

# Programmation WEB

## Un survol

[serge.tahe@istia.univ-angers.fr](mailto:serge.tahe@istia.univ-angers.fr)

<b>1. Généralités.....</b>	<b>6</b>
<b><u>1.1</u> LES COMPOSANTES D'UNE APPLICATION WEB.....</b>	<b>6</b>
<b><u>1.2</u> LES ÉCHANGES DANS UNE APPLICATION WEB AVEC FORMULAIRE DE SAISIE.....</b>	<b>7</b>
<b><u>1.3</u> QUELQUES RESSOURCES.....</b>	<b>8</b>
<b>2. Serveurs Web, Navigateurs, Langages de scripts.....</b>	<b>9</b>
<b>3. Les navigateurs.....</b>	<b>9</b>
<b><u>3.1</u> Netscape.....</b>	<b>9</b>
<b><u>3.2</u> Internet Explorer (IE).....</b>	<b>9</b>
<b>4. Les langages du WEB.....</b>	<b>9</b>
<b><u>4.1</u> PHP.....</b>	<b>9</b>
<b><u>4.1.1</u> Installation.....</b>	<b>9</b>
<b><u>4.1.2</u> Un premier script php.....</b>	<b>10</b>
<b><u>4.2</u> PERL.....</b>	<b>10</b>
<b><u>4.2.1</u> Installation.....</b>	<b>10</b>
<b><u>4.2.2</u> Tests.....</b>	<b>10</b>
<b><u>4.3</u> JAVA.....</b>	<b>11</b>
<b><u>4.3.1</u> Installation.....</b>	<b>11</b>
<b><u>4.3.2</u> Tests.....</b>	<b>11</b>
<b>5. Serveurs WEB.....</b>	<b>12</b>
<b><u>5.1</u> Serveur Apache.....</b>	<b>12</b>
<b><u>5.1.1</u> Installation.....</b>	<b>12</b>
<b><u>5.1.2</u> Configuration.....</b>	<b>12</b>
<b><u>5.1.3</u> Lien PHP - Apache.....</b>	<b>13</b>
<b><u>5.1.4</u> Lien PERL-APACHE.....</b>	<b>13</b>
<b><u>5.2</u> Le serveur PWS.....</b>	<b>15</b>
<b><u>5.2.1</u> Installation.....</b>	<b>15</b>
<b><u>5.2.2</u> Configuration.....</b>	<b>15</b>
<b><u>5.2.3</u> Lien PHP - PWS.....</b>	<b>15</b>
<b><u>5.3</u> Tomcat : servlets Java et pages JSP (Java Server Pages).....</b>	<b>15</b>
<b><u>5.3.1</u> Installation.....</b>	<b>15</b>
<b><u>5.3.2</u> Lancement du serveur Web Tomcat.....</b>	<b>16</b>
<b><u>5.3.3</u> Premiers tests.....</b>	<b>17</b>
<b><u>5.3.4</u> Lien Apache-Tomcat.....</b>	<b>17</b>
<b><u>5.3.5</u> Tests Tomcat-Apache.....</b>	<b>18</b>
<b>6. Page statique HTML.....</b>	<b>19</b>
<b><u>6.1</u> Le script <code>essai1.html</code>.....</b>	<b>19</b>
<b><u>6.2</u> Les tests.....</b>	<b>20</b>
<b>7. Une page ASP (Active Server Pages).....</b>	<b>20</b>
<b><u>7.1</u> Le script <code>essai2.asp</code>.....</b>	<b>20</b>
<b><u>7.2</u> Le test.....</b>	<b>20</b>
<b>8. Un script Perl.....</b>	<b>20</b>
<b><u>8.1</u> Le script : <code>essai3.pl</code>.....</b>	<b>20</b>
<b><u>8.2</u> Le test.....</b>	<b>21</b>
<b>9. Un script PHP.....</b>	<b>21</b>

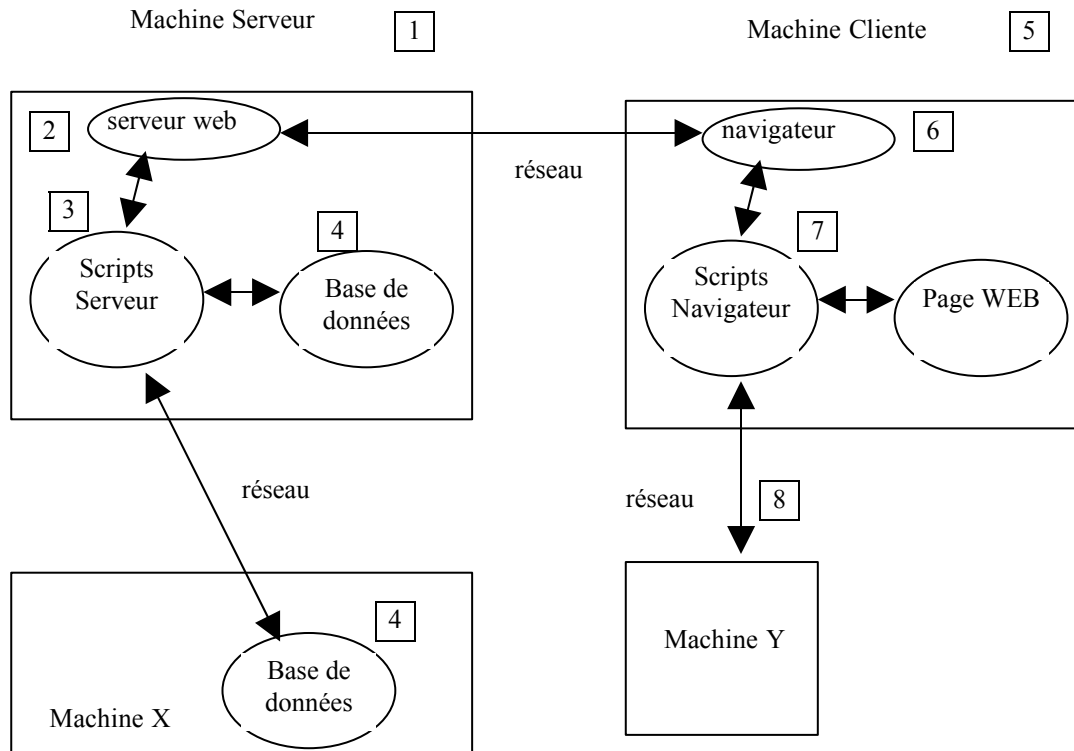
9.1	Le script : essai4.php.....	21
9.2	Les tests.....	22
10.	<i>Un script JSP.....</i>	22
10.1	Le script : heure.jsp.....	22
10.2	Les tests.....	23
11.	<i>Une page Web avec un script Vbscript, côté navigateur.....</i>	24
11.1	La page : vbs1.html.....	24
11.2	Les tests.....	25
12.	<i>Une page Web avec un script Javascript, côté navigateur.....</i>	25
12.1	La page : js1.html.....	25
12.2	Le test.....	25
13.	<i>Une page Web avec un script Perl, côté navigateur.....</i>	26
13.1	La page : ps1.html.....	26
13.2	Le test.....	26
14.	<i>Une page Web avec un script VB, côté serveur.....</i>	27
14.1	La page : vbs2.asp.....	27
14.2	Le test.....	28
15.	<i>Formulaire Web, javascript côté navigateur.....</i>	29
15.1	La page : form1.html.....	29
15.2	Les tests.....	30
16.	<i>Formulaire Web, vbscript côté navigateur.....</i>	30
16.1	La page : form2.html.....	30
16.2	Le test.....	31
17.	<i>Formulaire Web, Perlscript côté navigateur.....</i>	31
17.1	La page : form3.html.....	31
17.2	Le test.....	32
18.	<i>Vérification du formulaire avant l'envoi, par un script côté navigateur.....</i>	32
18.1	Le texte HTML.....	32
18.2	Test.....	33
19.	<i>Le dialogue client/serveur.....</i>	34
19.1	Le formulaire : form4.asp.....	34
19.2	Le texte HTML du formulaire.....	34
19.3	Demande initiale du formulaire.....	35
19.4	Envoi du formulaire : méthode GET.....	35
19.5	Envoi du formulaire : méthode POST.....	36
20.	<i>Traitement du formulaire Web avec un script ASP côté serveur.....</i>	38
20.1	La page ASP pour une méthode d'envoi GET.....	38
20.2	Le test.....	39
20.3	La page ASP pour une méthode d'envoi POST.....	39

<b>20.4</b> Le test.....	40
<b>21. Traitement d'un formulaire Web avec un script Perl côté serveur.....</b>	<b>41</b>
<b>21.1</b> Le script Perl.....	41
<b>21.2</b> Le test.....	43
<b>22. Technologie ASP.....</b>	<b>44</b>
<b>22.1</b> Le formulaire à construire.....	44
<b>22.2</b> Les balises HTML de base.....	44
<b>22.3</b> Une page HTML simplifiée.....	46
<b>22.4</b> Passage des valeurs du formulaire par la méthode GET.....	47
<b>22.5</b> Passage des valeurs du formulaire par la méthode POST.....	48
<b>22.6</b> Récupération des variables d'environnement du serveur Web.....	49
<b>23. Technologie Perl.....</b>	<b>50</b>
<b>23.1</b> Le formulaire à construire.....	50
<b>23.2</b> Le script Perl d'affichage et de traitement du formulaire.....	50
<b>23.3</b> Le test.....	52
<b>23.4</b> Récupération des variables d'environnement du serveur.....	52
<b>24. Amélioration du script Perl.....</b>	<b>54</b>
<b>24.1</b> Le script.....	54
<b>24.2</b> Les tests.....	56
<b>25. Technologie JSP.....</b>	<b>57</b>
<b>25.1</b> Le script JSP.....	57
<b>25.2</b> Le test.....	59
<b>26. Technologie PHP.....</b>	<b>60</b>
<b>26.1</b> Le script PHP.....	60
<b>26.2</b> Les tests.....	62
<b>27. L'application.....</b>	<b>64</b>
<b>27.1</b> Pilote ODBC/Nom DSN.....	65
<b>27.2</b> Technologie ASP.....	67
<b>27.2.1</b> Le script.....	67
<b>27.2.2</b> Les tests.....	70
<b>27.3</b> Technologie JSP.....	72
<b>27.3.1</b> Le script.....	72
<b>27.3.2</b> Les tests.....	75
<b>27.4</b> Technologie PHP.....	76
<b>27.4.1</b> Le script.....	76
<b>27.4.2</b> Les tests.....	77
<b>27.5</b> Technologie PERL.....	79
<b>27.5.1</b> Le script.....	79
<b>27.5.2</b> Les tests.....	81
<b>28. Les sessions WEB.....</b>	<b>82</b>
<b>28.1</b> Technologie Perl.....	82
<b>28.1.1</b> Les champs cachés.....	82

<a href="#">28.1.2</a> Le cookie initialisé côté serveur.....	83
<a href="#">28.1.3</a> Le cookie initialisé côté navigateur.....	84
<b><a href="#">28.2</a> Technologie ASP.....</b>	<b>85</b>
<a href="#">28.2.1</a> Les champs cachés.....	85
<a href="#">28.2.2</a> Le cookie initialisé côté serveur.....	85
<a href="#">28.2.3</a> Le cookie initialisé côté navigateur.....	86
<b><a href="#">28.3</a> Technologie PHP.....</b>	<b>86</b>
<a href="#">28.3.1</a> Les champs cachés.....	86
<a href="#">28.3.2</a> Le cookie initialisé côté serveur.....	87
<a href="#">28.3.3</a> Le cookie initialisé côté navigateur.....	87
<b><a href="#">28.4</a> Technologie JSP.....</b>	<b>88</b>
<a href="#">28.4.1</a> Les champs cachés.....	88
<a href="#">28.4.2</a> Le cookie initialisé côté serveur.....	89
<a href="#">28.4.3</a> Le cookie initialisé côté navigateur.....	89
<b><a href="#">29</a> JAVASCRIPT.....</b>	<b>91</b>
<b><a href="#">29.1</a> Récupérer les informations d'un formulaire.....</b>	<b>91</b>
<a href="#">29.1.1</a> Le formulaire.....	91
<a href="#">29.1.2</a> Le code.....	91
<b><a href="#">29.2</a> Les expressions régulières en Javascript.....</b>	<b>93</b>
<a href="#">29.2.1</a> La page de test.....	93
<a href="#">29.2.2</a> Le code de la page.....	94
<b><a href="#">29.3</a> Gestion des listes en JavaScript.....</b>	<b>96</b>
<a href="#">29.3.1</a> Le formulaire.....	96
<a href="#">29.3.2</a> Le code.....	96

# 1.Généralités

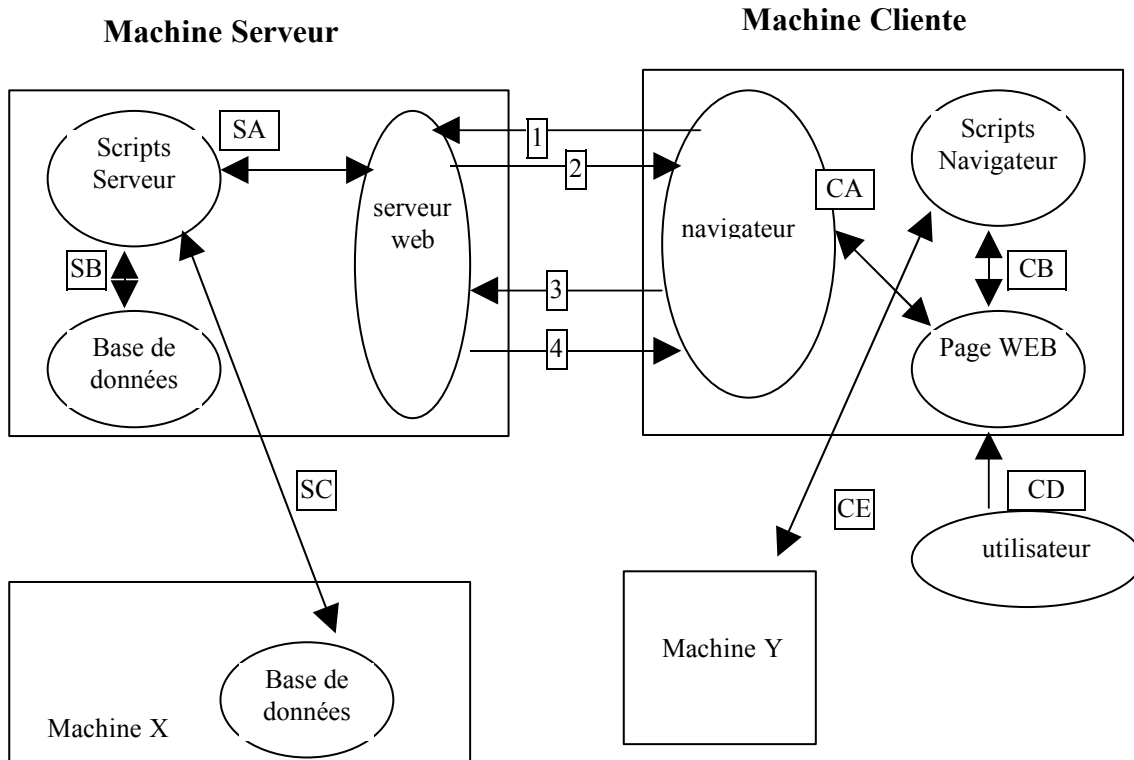
## 1.1 LES COMPOSANTES D'UNE APPLICATION WEB



Numéro	Rôle	Exemples courants
1	OS Serveur	Linux, Windows
2	Serveur Web	Apache (Linux, Windows) IIS (NT), PWS(Win9x)
3	Scripts exécutés côté serveur. Ils peuvent l'être par des modules du serveur ou par des programmes externes au serveur (CGI).	PERL (Apache, IIS, PWS) VBSCRIPT (IIS,PWS) JAVASCRIPT (IIS,PWS) PHP (Apache, IIS, PWS) JAVA (Apache, IIS, PWS)
4	Base de données - Celle-ci peut être sur la même machine que le programme qui l'exploite ou sur une autre via Internet.	Oracle (Linux, Windows) MySQL (Linux, Windows) Access (Windows) SQL Server (Windows)
5	OS Client	Linux, Windows
6	Navigateur Web	Netscape, Internet Explorer
7	Scripts exécutés côté client au sein du navigateur. <b>Ces scripts n'ont aucun accès aux disques du poste client.</b>	VBscript (IE) Javascript (IE, Netscape) Perlscript (IE) Applets JAVA
8	Côté navigateur, les scripts peuvent atteindre d'autres machines via Internet si le site du client n'est pas protégé	

par une machine pare-feu (firewall).

## 1.2 LES ÉCHANGES DANS UNE APPLICATION WEB AVEC FORMULAIRE DE SAISIE



Numéro	Rôle
1	Le navigateur demande une URL pour la 1ère fois ( <i>http://machine/url</i> ). Aucun paramètre n'est passé.
2	Le serveur Web lui envoie la page Web de cette URL. Elle peut être statique ou bien dynamique, générée par un script serveur (SA) qui a pu utiliser le contenu de bases de données (SB, SC). Ici, le programme détectera que l'URL a été demandée sans passage de paramètres et générera la page WEB initiale. Le navigateur reçoit la page. Il l'affiche (CA). Des scripts côté navigateur (CB) ont pu modifier la page initiale envoyée par le serveur. Ensuite par des interactions entre l'utilisateur (CD) et les scripts (CB) et éventuellement d'autres machines (CE) la page Web va être modifiée. Les formulaires vont notamment être remplis.
3	Le navigateur redemande la même URL mais cette fois-ci avec des paramètres : méthode <b>GET</b> <i>url?param1=val1&amp;param2=val2&amp;...</i> ou méthode <b>POST</b> <i>url</i> et les paramètres sont envoyés en fin de dialogue navigateur/serveur sous la forme <b>param1=val1&amp;param2=val2&amp;...</b> Le serveur déclenche le programme (SA) associé à l'URL qui va détecter qu'il y a maintenant des paramètres et va donc les traiter
4	Le serveur délivre la page WEB construite par programme (SA, SB, SC). Cette étape est identique à l'étape 2 précédente. Les échanges se font désormais selon les étapes 2 et 3.

### 1.3 QUELQUES RESSOURCES

<b>Serveur Apache</b>	<a href="http://www.apache.org">http://www.apache.org</a> - Apache, Installation et Mise en œuvre, O'Reilly
<b>Serveur IIS, PWS</b>	<a href="http://www.microsoft.com">http://www.microsoft.com</a>
<b>PERL</b>	<a href="http://www.activestate.com">http://www.activestate.com</a> - Programmation en Perl, Larry Wall, O'Reilly - Applications CGI en Perl, Neuss et Vromans, O'Reilly - la documentation HTML livrée avec Active Perl
<b>PHP</b>	<a href="http://www.php.net">http://www.php.net</a> - Prog. Web avec PHP, Lacroix, Eyrolles - Manuel d'utilisation de PHP récupérable sur le site de PHP
<b>VBSCRIPT, ASP</b>	<a href="http://msdn.microsoft.com/scripting/vbscript/download/vbsd.exe">http://msdn.microsoft.com/scripting/vbscript/download/vbsd.exe</a> <a href="http://msdn.microsoft.com/scripting/default.htm?scripting/vbscript/">http://msdn.microsoft.com/scripting/default.htm?scripting/vbscript/</a> - Interface entre WEB et Base de données sous WinNT, Alex Homer, Eyrolles
<b>JAVASCRIPT</b>	<a href="http://msdn.microsoft.com/scripting/jscript/download/jsdoc.exe">http://msdn.microsoft.com/scripting/jscript/download/jsdoc.exe</a> <a href="http://developer.netscape.com/docs/manuals/index.html">http://developer.netscape.com/docs/manuals/index.html</a>
<b>HTML</b>	<a href="http://developer.netscape.com/docs/manuals/index.html">http://developer.netscape.com/docs/manuals/index.html</a>
<b>JAVA</b>	<a href="http://www.sun.com">http://www.sun.com</a> - JAVA Servlets, Jason Hunter, O'Reilly - Programmation réseau avec Java, Elliotte Rusty Harold, O'Reilly - JDBC et Java, George Reese, O'Reilly
<b>Base de données</b>	<a href="http://www.mysql.com">http://www.mysql.com</a> <a href="http://www.oracle.com">http://www.oracle.com</a> - Le manuel de MySQL est disponible sur le site de MySQL - Oracle 8i sous Linux, Gilles Briard, Eyrolles - Oracle 8i sous NT, Gilles Briard, Eyrolles



# Les outils

## 2. Serveurs Web, Navigateurs, Langages de scripts

### 1. Serveurs Web principaux

- Apache (Linux, Windows)
- Internet Information Server IIS (NT), Personal Web Server PWS (Windows 9x)

### 2. Navigateurs principaux

- Internet Explorer (Windows)
- Netscape (Linux, Windows)

### 3. Langages de scripts côté serveur

- VBScript (IIS, PWS)
- JavaScript (IIS, PWS)
- Perl (Apache, IIS, PWS)
- PHP (Apache, IIS, PWS)
- Java (Apache, Tomcat)

### 4. Langages de scripts côté navigateur

- VBScript (IE)
- Javascript (IE, Netscape)
- Perlscript (IE)
- Java (IE, Netscape)

## 3. Les navigateurs

### 3.1 Netscape

Ce navigateur est disponible à l'URL <http://home.netscape.com/browsers>.

### 3.2 Internet Explorer (IE)

Ce navigateur est disponible à l'URL <http://www.microsoft.com/windows/ie/download>.

## 4. Les langages du WEB

### 4.1 PHP

#### 4.1.1 Installation

Site de PHP : <http://www.php.net>

PHP est disponible à l'URL : <http://www.php.net>. On récupère une archive qu'on décompresse dans un répertoire que nous appellerons <php>. Il contient l'arborescence suivante :

```
MSVCRT  DLL          266 293 11/02/99  3:33 MSVCRT.DLL
PHP      EXE           16 384 10/05/00 15:29 php.exe
PHP~1   INI            20 540 10/05/00  1:55 php.ini-dist
PHP~2   INI            20 327 20/04/00 16:25 php.ini-optimized
PHP4IS~1 DLL           20 480 10/05/00 15:31 php4isapi.dll
PHP4TS  DLL           839 680 10/05/00 15:29 php4ts.dll
README  TXT              2 448 06/05/00 22:34 README.txt
```

L'exécutable est *php.exe*.

## 4.1.2 Un premier script php

PHP n'est pas utilisable que dans le cadre du Web. On peut l'utiliser comme langage de scripts sous Windows. Créez le script suivant et sauvegardez-le sous le nom *date.php* :

```
<?
// script php affichant l'heure
$maintenant=date("j/m/y, H:i:s",time());
echo "Nous sommes le $maintenant";
?>
```

Dans une fenêtre DOS, placez-vous dans le répertoire **<php>** de *php.exe* et exécutez le script précédent :

```
C:\php4>php d:\data\php\essais\date.php
X-Powered-By: PHP/4.0RC2
Content-type: text/html

Nous sommes le 16/11/00, 08:59:50
```

## 4.2 PERL

### 4.2.1 Installation

Il est préférable que Internet Explorer soit déjà installé. S'il est présent, Active Perl va le configurer afin qu'il accepte des scripts PERL dans les pages HTML, scripts qui seront exécutés par IE lui-même côté client.

Site de Active Perl : <http://www.activestate.com>

Le fichier d'installation d'Active perl est disponible à l'URL :

<http://www.activestate.com/download/ActivePerl/Windows>

A l'installation, PERL sera installé dans un répertoire que nous appellerons **<perl>**. Il contient l'arborescence suivante :

```
DEISL1  ISU      32 403 23/06/00 17:16 DeIsL1.isu
BIN      <REP>      23/06/00 17:15 bin
LIB      <REP>      23/06/00 17:15 lib
HTML     <REP>      23/06/00 17:15 html
EG       <REP>      23/06/00 17:15 eg
SITE     <REP>      23/06/00 17:15 site
HTMLHELP <REP>      28/06/00 18:37 htmlhelp
```

L'exécutable *perl.exe* est dans **<perl>\bin**.

### 4.2.2 Tests

Perl est un langage de scripts fonctionnant sous Windows et Unix. Il est de plus utilisé dans la programmation WEB. Écrivons un premier script :

```
# script PERL affichant l'heure

# modules
use strict;
```

```
# programme
my ($secondes,$minutes,$heure)=localtime(time);
print "Il est $heure:$minutes:$secondes\n";
```

Sauvegardez ce script dans un fichier *heure.pl*. Ouvrez une fenêtre DOS, placez-vous dans le répertoire **<perl>\bin** contenant l'exécutable *perl.exe* et lancez l'exécution du script :

```
C:\Perl\bin>perl d:\data\perl\essais\heure.pl
Il est 9:16:44
```

## 4.3 JAVA

### 4.3.1 Installation

Java est disponible à l'URL : <http://www.sun.com/download>.

Java s'installe dans une arborescence qu'on appellera **<java>**. Elle contient les éléments suivants :

```
UNINST  ISU      112 984  23/06/00  17:31 Uninst.isu
BIN      <REP>      23/06/00  17:31 bin
README  TXT        5 063  28/04/00  10:23 README.txt
LICENSE  <REP>      10 849  28/04/00  10:23 LICENSE
COPYRI~1 <REP>      945  28/04/00  10:23 COPYRIGHT
README~1 HMT      18 374  28/04/00  10:23 readme.html
JRE      <REP>      23/06/00  17:31 jre
LIB      <REP>      23/06/00  17:31 lib
INCLUDE <REP>      23/06/00  17:31 include
INCLUD~1 <REP>      23/06/00  17:31 include-old
DEMO     <REP>      23/06/00  17:31 demo
SRC      JAR      19 565 690  28/04/00  10:24 src.jar
```

Dans bin, on trouvera **javac.exe**, le compilateur Java et **java.exe** la machine virtuelle Java.

### 4.3.2 Tests

1. Écrire le script suivant :

```
//programme Java affichant l'heure

import java.io.*;
import java.util.*;

public class heure{
    public static void main(String arg[]){
        // on récupère date & heure
        Date maintenant=new Date();
        // on affiche
        System.out.println("Il est "+maintenant.getHours()+
            ":"+maintenant.getMinutes()+":"+maintenant.getSeconds());
    }
}
```

2. Sauvegarder ce programme sous le nom *heure.java*. Ouvrir une fenêtre DOS. Se mettre dans le répertoire du fichier *heure.java* et le compiler :

```
D:\data\java\essais>c:\jdk1.3\bin\javac heure.java
Note: heure.java uses or overrides a deprecated API.
Note: Recompile with -deprecation for details.
```

Dans la commande ci-dessus `c:\jdk1.3\bin\javac` doit être remplacé par le chemin exact du compilateur *javac.exe*. Vous devez obtenir dans le même répertoire que *heure.java* un fichier *heure.class* qui est le

programme qui va maintenant être exécuté par la machine virtuelle *java.exe*.

### 3. Exécuter le programme :

```
D:\data\java\essais>c:\jdk1.3\bin\java heure
Il est 10:44:2
```

## 5. Serveurs WEB

### 5.1 Serveur Apache

#### 5.1.1 Installation

Site d'Apache : <http://www.apache.org>

Le fichier d'installation est disponible à l'URL :

<http://httpd.apache.org/dist/binaries/win32>

L'installation crée une arborescence où on trouve tous les fichiers nécessaires au serveur. Appelons *<apache>* ce répertoire. Il contient l'arborescence suivante :

```
UNINST  ISU      118 805 23/06/00 17:09 Uninst.isu
HTDOCS  <REP>        23/06/00 17:09 htdocs
APACHE~1 DLL    299 008 25/02/00 21:11 ApacheCore.dll
ANNOUN~1 3 000 23/02/00 16:51 Announcement
ABOUT_~1 13 197 31/03/99 18:42 ABOUT_APACHE
APACHE  EXE     20 480 25/02/00 21:04 Apache.exe
KEYS     36 437 20/08/99 11:57 KEYS
LICENSE  2 907 01/01/99 13:04 LICENSE
MAKEFI~1 TMP    27 370 11/01/00 13:47 Makefile.tpl
README  2 109 01/04/98  6:59 README
README  NT      3 223 19/03/99  9:55 README.NT
WARNIN~1 TXT    339 21/09/98 13:09 WARNING-NT.TXT
BIN      <REP>        23/06/00 17:09 bin
MODULES  <REP>        23/06/00 17:09 modules
ICONS    <REP>        23/06/00 17:09 icons
LOGS     <REP>        23/06/00 17:09 logs
CONF     <REP>        23/06/00 17:09 conf
CGI-BIN  <REP>        23/06/00 17:09 cgi-bin
PROXY    <REP>        23/06/00 17:09 proxy
INSTALL LOG    3 779 23/06/00 17:09 install.log
```

**conf** dossier des fichiers de configuration d'Apache  
**logs** dossier des fichiers de logs (suivi) d'Apache  
**bin** les exécutables d'Apache

#### 5.1.2 Configuration

Dans le dossier *<Apache>conf*, on trouve les fichiers suivants : *httpd.conf*, *srm.conf*, *access.conf*. Dans les dernières versions d'Apache, les trois fichiers ont été réunis dans *httpd.conf*. La lecture de ces fichiers de configuration n'est pas simple au départ. Voici les points importants :

ligne(s)	rôle
<b>ServerRoot "D:/Program Files/Apache Group/Apache"</b>	indique le dossier où se trouve l'arborescence de Apache
<b>Port 80</b>	indique sur quel port va travailler le serveur Web. Classiquement c'est 80. En changeant cette ligne, on peut faire travailler le serveur Web sur un autre port


<b>ServerAdmin</b> root@istia.univ-angers.fr	l'adresse email de l'administrateur du serveur Apache
<b>ServerName</b> stahe.istia.uang	le nom de la machine sur laquelle "tourne" le serveur Apache
<b>DocumentRoot</b> "D:/Apache/htdocs"	le dossier racine de l'arborescence des pages Web délivrées par le serveur. Ici, l'url <i>http://machine/rep1/fic1.html</i> correspondra au fichier <i>d:\apache\htdocs\rep1\fic1.html</i>
<b>&lt;Directory "D:/Apache/htdocs"&gt;</b>	fixe les propriétés du dossier précédent
<b>ErrorLog</b> d:/apache/logs/error.log	dossier des logs
<b>ScriptAlias</b> /cgi-bin/ "D:/Apache/cgi-bin/"	<i>d:\apache\cgi-bin</i> sera la racine de l'arborescence où l'on pourra mettre des scripts CGI. Ainsi l'URL <i>http://machine/cgi-bin/rep1/script1.pl</i> sera l'url du script CGI <i>d:\apache\cgi-bin\rep1\script1.pl</i> .
<b>&lt;Directory "D:/Data/Apache/cgi-bin"&gt;</b>	fixe les propriétés du dossier ci-dessus
<b>ScriptAlias</b> /php4/ "d:/php4/" <b>AddType</b> application/x-httpd-php4 .php4 <b>Action</b> application/x-httpd-php4 /php4/php.exe	ces lignes permettent au serveur Apache de délivrer des pages générées par des scripts php4. Ceux-ci sont repérés par leur suffixe .php4. L'exécutable php.exe est dans d:\php4. <b>L'erreur la plus fréquente est d'oublier le dernier / dans le nom du répertoire "d:/php4".</b>

### 5.1.3 Lien PHP - Apache

- installer php si ce n'est déjà fait
- modifier le fichier *httpd.conf* de Apache comme indiqué dans le paragraphe ci-dessus
- Pour tester, créer le fichier **intro.php4** avec la seule ligne suivante :

<? phpinfo() ?>

et le mettre à la racine des pages du serveur Apache (*DocumentRoot* ci-dessus). Demander l'URL **http://localhost/intro.php4**. On doit voir une liste d'informations php.

<b>PHP Version 4.0RC2</b> 	
<b>System</b>	Windows 95/98 4.10
<b>Build Date</b>	May 10 2000
<b>Server API</b>	CGI
<b>Virtual Directory Support</b>	disabled
<b>Configuration File (php.ini) Path</b>	php.ini
<b>ZEND_DEBUG</b>	disabled
<b>Thread Safety</b>	enabled

### 5.1.4 Lien PERL-APACHE

Il est fait grâce à une ligne de la forme :

```
ScriptAlias /cgi-bin/ "D:/Apache/cgi-bin/"
```

du fichier `<apache>\conf\httpd.conf`. Sa syntaxe est

```
ScriptAlias /cgi-bin/ "<cgi-bin>"
```

où `<cgi-bin>` est le dossier où on pourra placer des scripts CGI. CGI (Common Gateway Interface) est une norme de dialogue serveur WEB <--> Applications. Un client demande au serveur Web une page dynamique, c.a.d. une page générée par un programme. Le serveur WEB doit donc demander à un programme de générer la page. CGI définit le dialogue entre le serveur et le programme, notamment le mode de transmission des informations entre ces deux entités.

Si besoin est, modifiez cette ligne et relancez le serveur Apache. Faites ensuite le test suivant :

1. Écrire le script suivant :

```
#!c:\perl\bin\perl.exe

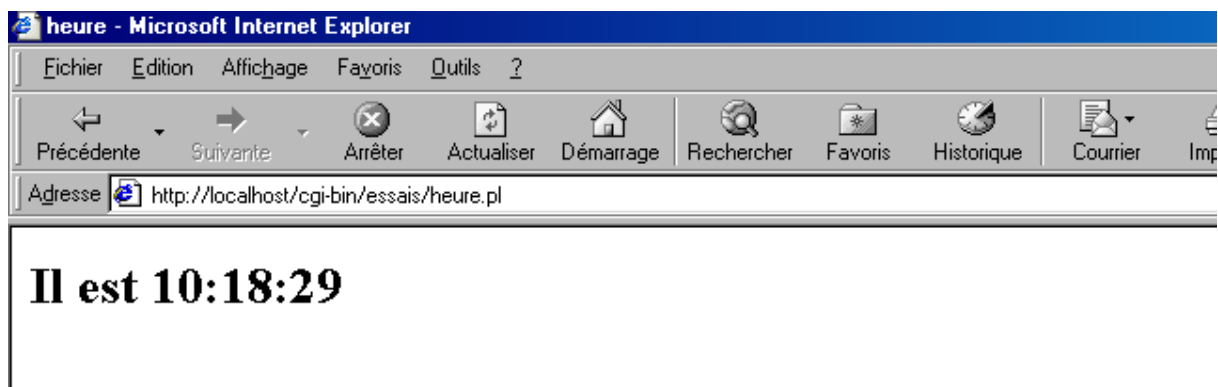
# script PERL affichant l'heure

# modules
use strict;

# programme
my ($secondes,$minutes,$heure)=localtime(time);
print <<FINHTML
Content-Type: text/html

<html>
<head>
<title>heure</title>
</head>
<body>
<h1>Il est $heure:$minutes:$secondes</h1>
</body>
FINHTML
;
```

2. Mettre ce script dans `<cgi-bin>\heure.pl` où `<cgi-bin>` est le dossier pouvant recevoir des scripts CGI (cf `httpd.conf`). La première ligne `#!c:\perl\bin\perl.exe` désigne le chemin de l'exécutable `perl.exe`. Le modifier si besoin est.
3. Lancer Apache si ce n'est fait
4. Demander avec un navigateur l'URL `http://localhost/cgi-bin/heure.pl`. On obtient la page suivante :



## 5.2 Le serveur PWS

### 5.2.1 Installation

Avec Internet Explorer. Prendre une installation personnalisée.

### 5.2.2 Configuration

La racine des pages Web du serveur PWS est `d:\inetpub\wwwroot`. Ainsi l'url `http://machine/rep1/page1.html` correspondra au fichier `d:\inetpub\wwwroot\rep1\page1.html`. Le serveur PWS interprète tout fichier de suffixe `.asp` (Active Server pages) comme étant un script qu'il doit exécuter pour produire une page HTML.

### 5.2.3 Lien PHP - PWS

1. Ci-dessous un fichier `.reg` destiné à modifier la base de registres. Double-cliquer sur ce fichier pour modifier la base. Ici la `dll` nécessaire se trouve dans `d:\php4` avec l'exécutable de php. Modifier si besoin est. Les \ doivent être doublés dans le chemin de la dll.

```
REGEDIT4
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\w3svc\parameters\Script Map]  
".php"="d:\php4\php4isapi.dll"
```

2. Relancer la machine pour que la modification de la base de registres soit prise en compte.
3. Créer un dossier **php** dans `d:\inetpub\wwwroot` qui est la racine du serveur PWS. Ceci fait, activez PWS et prendre l'onglet « Avancé ». Sélectionner le bouton « Ajouter » pour créer un dossier virtuel :

```
Répertoire/Parcourir : d:\inetpub\wwwroot\php  
Alias : php  
Cocher la case exécuter.
```

4. Valider le tout et relancer PWS. Mettre dans `d:\inetpub\wwwroot\php` le fichier **intro.php** ayant la seule ligne suivante :

```
<? phpinfo() ?>
```

5. Demander au serveur PWS l'URL `http://localhost/php/intro.php`. On doit voir une liste d'informations php4.

## 5.3 Tomcat : servlets Java et pages JSP (Java Server Pages)

Tomcat est un serveur Web permettant de générer des pages HTML grâce à des servlets (programmes Java exécutés par le serveur web) où des pages JSP (Java Server Pages), pages mélangeant code Java et code HTML. C'est l'équivalent des pages ASP (Active Server Pages) du serveur IIS/PWS de Microsoft où là on mélange code VBScript ou Javascript avec du code HTML.

### 5.3.1 Installation

Tomcat est disponible à l'URL : <http://jakarta.apache.org/downloads/binindex.html>

On récupère un fichier compressé. L'installation consiste simplement à décompresser cette archive dans un répertoire. Prenez un répertoire ne contenant dans son chemin que des noms sans espace (pas par exemple

"Program Files"), ceci parce qu'il y a un bogue dans le processus d'installation de Tomcat. Prenez par exemple C:\tomcat ou D:\tomcat. Appelons ce répertoire <tomcat>. On y trouvera dedans un dossier appelé **jakarta-tomcat** et dans celui-ci l'arborescence suivante :

```
LOGS          <REP>          15/11/00    9:04 logs
LICENSE      2 876    18/04/00    15:56 LICENSE
CONF         <REP>          15/11/00    8:53 conf
DOC          <REP>          15/11/00    8:53 doc
LIB          <REP>          15/11/00    8:53 lib
SRC          <REP>          15/11/00    8:53 src
WEBAPPS     <REP>          15/11/00    8:53 webapps
BIN          <REP>          15/11/00    8:53 bin
WORK        <REP>          15/11/00    9:04 work
```

### 5.3.2 Lancement du serveur Web Tomcat

Tomcat est un serveur Web comme l'est Apache ou PWS. Pour le lancer, procéder ainsi :

- ouvrir une fenêtre DOS
- définir une variable TOMCAT\_HOME ayant pour valeur le chemin du répertoire contenant l'arborescence précédente :  
 SET TOMCAT\_HOME=<tomcat>\jakarta-tomcat  
 par exemple **set TOMCAT\_HOME=d:\tomcat\jakarta-tomcat**, si vous avez décompressé l'archive tomcat dans *d:\tomcat*.
- définir une variable JAVA\_HOME ayant pour valeur le chemin du répertoire contenant une version récente d'un JDK Java, par exemple  
**set JAVA\_HOME=d:\jdk1.3** si vous avez un JDK en *d:\jdk1.3*
- mettre le répertoire contenant l'exécutable **java.exe** du JDK dans le PATH de votre machine. En suivant l'exemple précédent, cela se ferait par **set PATH=d:\jdk1.3\bin;%PATH%** qui rajoute **d:\jdk1.3\bin** à votre path actuel (%PATH%)
- ceci fait, le serveur Tomcat peut être lancé par la commande **%TOMCAT\_HOME%\bin\startup** qui lance un fichier batch *startup.bat* se trouvant dans <tomcat>\jakarta-tomcat\bin.

On obtient alors une fenêtre DOS ressemblant à ceci :

```
MS-DOS JAVA
Auto
Trop de paramètres
Starting tomcat in new window
Using classpath: c:\tomcat\jakarta-tomcat\classes;c:\tomcat\jakarta-tomcat\lib\we
ebserver.jar;c:\tomcat\jakarta-tomcat\lib\jasper.jar;c:\tomcat\jakarta-tomcat\li
b\xml.jar;c:\tomcat\jakarta-tomcat\lib\servlet.jar;c:\jdk1.3\lib\tools.jar;C:\Pr
ogram Files\Jamba\Java;
Context log: path="/examples" Adding context path="/examples" docBase="webapps/
examples"
Context log: path="" Adding context path="" docBase="webapps/ROOT"
Context log: path="/test" Adding context path="/test" docBase="webapps/test"
Starting tomcat. Check logs/tomcat.log for error messages
Starting tomcat install="c:\tomcat\jakarta-tomcat" home="C:\tomcat\jakarta-tomca
t" classPath="c:\tomcat\jakarta-tomcat\classes;c:\tomcat\jakarta-tomcat\lib\webs
erver.jar;c:\tomcat\jakarta-tomcat\lib\jasper.jar;c:\tomcat\jakarta-tomcat\lib\x
ml.jar;c:\tomcat\jakarta-tomcat\lib\servlet.jar;c:\jdk1.3\lib\tools.jar;C:\Progr
am Files\Jamba\Java;"
-
```

L'ensemble peut être automatisé dans un fichier batch :



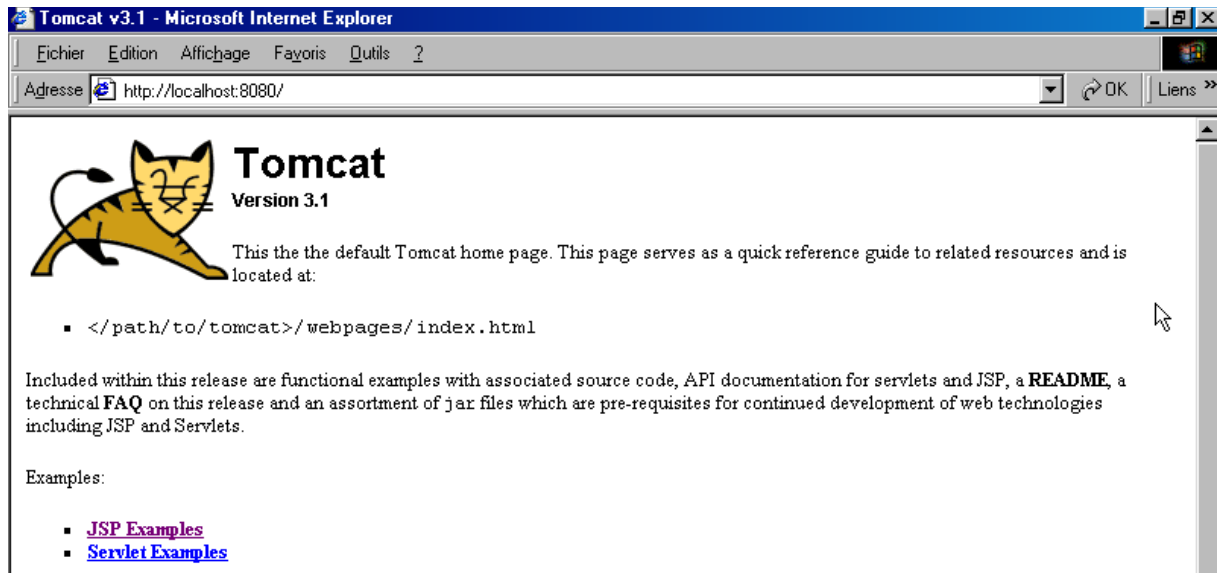
```

rem lance le serveur web Tomcat
set TOMCAT_HOME=d:\tomcat\jakarta-tomcat
set JAVA_HOME=d:\jdk1.3
set PATH=d:\jdk1.3\bin;%PATH%
%TOMCAT_HOME%\bin\startup

```

### 5.3.3 Premiers tests

Le serveur Web Tomcat travaille sur le port 8080. Une fois Tomcat lancé, prenez un navigateur Web et demandez l'URL <http://localhost:8080>. Vous devez obtenir la page suivante :



Suivez le lien [Servlet Examples](#) :

Hello World	 <a href="#">Execute</a>	 <a href="#">Source</a>
Request Info	 <a href="#">Execute</a>	 <a href="#">Source</a>
Request Headers	 <a href="#">Execute</a>	 <a href="#">Source</a>
Request Parameters	 <a href="#">Execute</a>	 <a href="#">Source</a>
Cookies	 <a href="#">Execute</a>	 <a href="#">Source</a>

Cliquez sur le lien *Execute* de *RequestParameters* puis sur celui de *Source*. Vous aurez un premier aperçu de ce qu'est un servlet Java.

### 5.3.4 Lien Apache-Tomcat

Peut-on visualiser ces servlets et pages JSP avec un serveur autre que Tomcat ? La réponse est oui pour Apache. Il faut pour cela modifier son fichier de configuration et récupérer une dll : **ApacheModuleJServ.dll**. Celle-ci, au moment de l'écriture de ce document était disponible à l'URL :

<http://jakarta.apache.org/builds/tomcat/release/v3.1/bin/win32/i386/ApacheModuleJServ.dll>

Elle est disponible à l'endroit où vous avez récupéré Tomcat.

Appelons **<apache>**, le répertoire d'installation du serveur Apache (par exemple *d:\Program files\Apache*

Group\Apache). *ApacheModuleJServ.dll* doit être placé dans le dossier <apache>\modules.

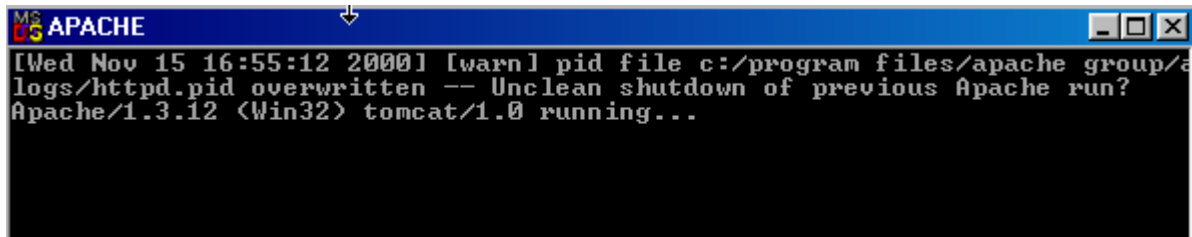
Dans <apache>\conf on trouvera le fichier de configuration **httpd.conf**. Ajouter en fin de fichier les lignes suivantes :

```
# tomcat
Include "D:\tomcat\jakarta-tomcat\conf\tomcat-apache.conf"
```

si vous avez installé Tomcat dans *d:\tomcat*. Auparavant, vérifiez que le fichier *D:\tomcat\jakarta-tomcat\conf\tomcat-apache.conf* existe bien.

### 5.3.5 Tests Tomcat-Apache

- lancer Tomcat
- lancer Apache. La fenêtre DOS d'Apache doit indiquer qu'il travaille avec Tomcat :



```
MS-DOS APACHE
[Wed Nov 15 16:55:12 2000] [warn] pid file c:/program files/apache group/a
logs/httpd.pid overwritten -- Unclean shutdown of previous Apache run?
Apache/1.3.12 (Win32) tomcat/1.0 running...
```

- demander l'URL **http://localhost/examples/servlets/** (ne pas oublier le dernier /). Vous obtiendrez la même page que lorsque vous avez fait les tests des servlets avec le serveur Tomcat. Si vous demandez l'URL **http://localhost/examples/jsp/**, vous pourrez tester des pages JSP délivrées par Apache.

# Pages statiques, Pages dynamiques

Une page statique est représentée par un fichier HTML. Une page dynamique est générée "à la volée" par un programme.

Dans la suite, nous adopterons les notations suivantes :

notation	signification	habituellement à l'ISTIA
<apache>	racine de l'arborescence du serveur apache	d:\program files\apache group\apache
<apache-DocumentRoot>	racine des pages Web délivrées par Apache. C'est sous cette racine que doivent se trouver les pages Web. Ainsi l'URL <i>http://localhost/page1.htm</i> correspond au fichier <apache-DocumentRoot>\page1.htm	d:\apache\htdocs
<apache-cgi-bin>	racine de l'arborescence où l'on peut placer des scripts CGI pour Apache. Ainsi l'URL <i>http://localhost/cgi-bin/test1.pl</i> correspond au fichier <apache-cgi-bin>\test1.pl	d:\apache\cgi-bin
<pws-DocumentRoot>	racine des pages Web délivrées par PWS. C'est sous cette racine que doivent se trouver les pages Web. Ainsi l'URL <i>http://localhost/page1.htm</i> correspond au fichier <pws-DocumentRoot>\page1.htm	d:\inetpub\wwwroot
<perl>	racine de l'arborescence de perl. L'exécutable perl.exe se trouve en général dans <perl>\bin	d:\perl
<php>	racine de l'arborescence de php. L'exécutable php.exe se trouve en général dans <php>	d:\php4
<java>	racine de l'arborescence de java. Les exécutables liés à java se trouvent dans <java>\bin	d:\jdk1.3 j:\jdk12
<tomcat>	racine du serveur Tomcat. On trouve des exemples de servlets dans <tomcat>\webapps\examples\servlets et des exemples de pages JSP dans <tomcat>\webapps\examples\jsp	d:\tomcat\jakarta-tomcat

## 6. Page statique HTML

### 6.1 Le script *essai1.html*

```
<html>
  <head>
    <title>essai 1 : une page statique</title>
  </head>
  <body>
    <center>
      <h1>Une page statique...</h1>
    </center>
  </body>
</html>
```



**Une page statique...**

## 6.2 Les tests

- lancer serveur Apache
  - mettre le script *essai1.html* dans *<apache-DocumentRoot>*
  - visualiser l'URL *http://localhost/essai1.html* avec un navigateur
  - arrêter le serveur Apache
- 
- lancer serveur PWS
  - mettre le script *essai1.html* dans *<pws-DocumentRoot>*
  - visualiser l'URL *http://localhost/essai1.html* avec un navigateur

## 7. Une page ASP (Active Server Pages)

### 7.1 Le script *essai2.asp*

```
<html>
  <head>
    <title>essai 1 : une page asp</title>
  </head>
  <body>
    <center>
      <h1>Une page asp générée dynamiquement par le serveur PWS</h1>
      <h2>Il est <% =time %></h2>
      <br>
      A chaque fois que vous rafraîchissez la page, l'heure change.
    </body>
</html>
```

## Une page asp générée dynamiquement par le serveur PWS

**Il est 11:41:40**



A chaque fois que vous rafraîchissez la page, l'heure change.

### 7.2 Le test

- lancer le serveur PWS
- mettre le script *essai2.asp* dans *<pws-DocumentRoot>*
- demander l'URL *http://localhost/essai2.asp* avec un navigateur

## 8. Un script Perl

### 8.1 Le script : *essai3.pl*

```
#!/d:\perl\bin\perl.exe

($secondes,$minutes,$heure)=localtime(time);

print <<HTML
Content-type: text/html

<html>
  <head>
    <title>essai 1 : un script Perl</title>
```

```

</head>
<body>
  <center>
    <h1>Une page générée dynamiquement par un script Perl</h1>
    <h2>Il est $heure:$minutes:$secondes</h2>
    <br>
    A chaque fois que vous rafraîchissez la page, l'heure change.
  </body>
</html>

```

```

HTML
;

```

La première ligne est le chemin de l'exécutable *perl.exe*. L'adapter si besoin est.

## Une page générée dynamiquement par un script Perl



**Il est 11:48:11**

A chaque fois que vous rafraîchissez la page, l'heure change.

### 8.2 Le test

- serveur Web : Apache
- pour information, visualiser le fichier de configuration *srm.conf* et rechercher la ligne parlant de *cgi-bin*
- mettre le script *essai3.pl* dans *<apache-cgi-bin>*
- demander l'url *http://localhost/cgi-bin/essai3.pl*

A noter qu'il faut davantage de temps pour avoir la page *perl* que la page *asp*. Ceci parce que le script Perl est généré par un interpréteur Perl qu'il faut charger avant qu'il puisse exécuter le script. Il ne reste pas en permanence en mémoire.

## 9. Un script PHP

### 9.1 Le script : *essai4.php*

```

<html>
<head>
  <title>essai 4 : une page php</title>
</head>
<body>
  <center>
    <h1>Une page PHP générée dynamiquement par le serveur Apache</h1>
  <h2>
    <?
    $maintenant=time();
    echo date("j/m/y, h:i:s",$maintenant);
    ?>
  </h2>
  <br>
  A chaque fois que vous rafraîchissez la page, l'heure change.
  </body>
</html>

```

# Une page PHP générée dynamiquement



12/09/00, 11:54:19

A chaque fois que vous rafraîchissez la page, l'heure change.

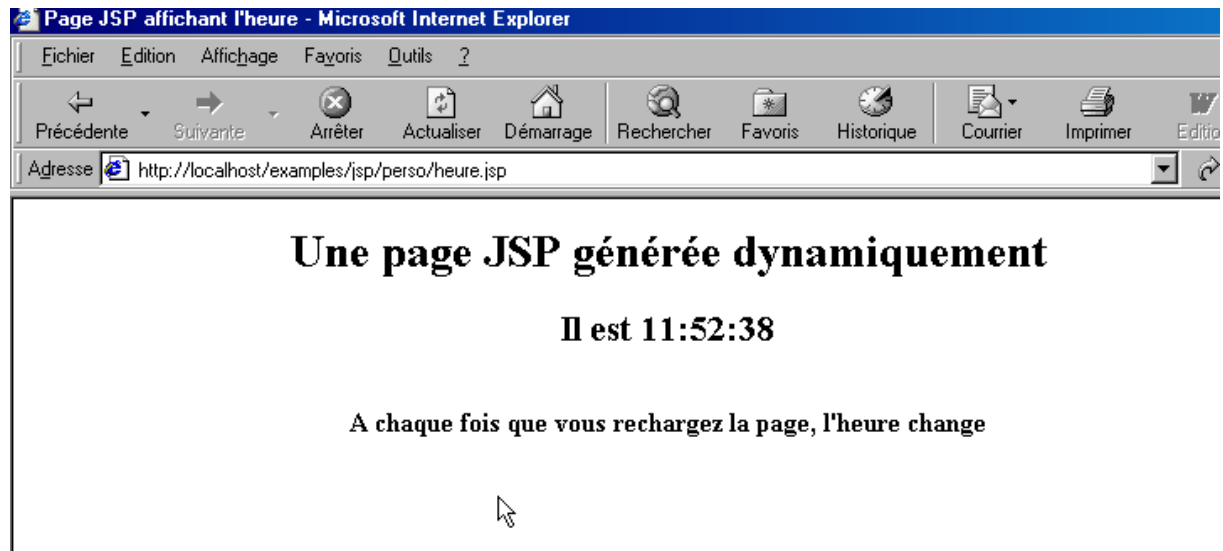
## 9.2 Les tests

- consulter le fichier de configuration *srm.conf* ou *httpd.conf* d'Apache dans `<Apache>\conf`
- pour information, vérifier les lignes de configuration de *php*
- lancer le serveur Apache
- mettre *essai4.php4* dans `<apache-DocumentRoot>`
- demander l'URL `http://localhost/essai4.php4`
  
- lancer le serveur PWS
- pour information, vérifier la configuration de PWS à propos de *php*
- mettre *essai4.php* dans `<pws-DocumentRoot>\php`
- demander l'URL `http://localhost/essai4.php`

## 10. Un script JSP

### 10.1 Le script : *heure.jsp*

```
<%  
    //programme Java affichant l'heure  
%>  
<%// classes importées %>  
<%@ page import="java.util.*" %>  
  
<%// code JAVA pour calculer l'heure  
    Date maintenant=new Date();  
    String heures="" +maintenant.getHours();  
    String minutes="" +maintenant.getMinutes();  
    String secondes="" +maintenant.getSeconds();  
    // heures, minutes, secondes sont des variables globales  
    // qui pourront être utilisées dans le code HTML  
%>  
  
<%// code HTML %>  
<html>  
    <head>  
        <title>Page JSP affichant l'heure</title>  
    </head>  
    <body>  
        <center>  
            <h1>Une page JSP générée dynamiquement</h1>  
            <h2>Il est <%=heures%>: <%=minutes%>: <%=secondes%></h2>  
            <br>  
            <h3>A chaque fois que vous rechargez la page, l'heure change</h3>  
        </body>  
</html>
```



## 10.2 Les tests

- mettre le script *heure.jsp* dans `<tomcat>\jakarta-tomcat\webapps\examples\jsp`
- lancer le serveur Tomcat
- lancer le serveur Apache
- demander l'URL `http://localhost/examples/jsp/heure.jsp`

# Scripts côté navigateur

Une page Web peut contenir des scripts qui seront exécutés par le navigateur. Les langages de script disponibles sont nombreux. En voici quelques-uns :

Langage	Navigateurs utilisables
Vbscript	IE
Javascript	IE, Netscape
PerlScript	IE
Java	IE, Netscape

## 11. Une page Web avec un script Vbscript, côté navigateur

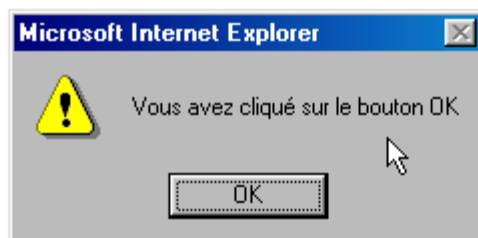
### 11.1 La page : vbs1.html

```
<html>
  <head>
    <title>essai : une page web avec un script vb</title>
    <script language="vbscript">
      function reagir
        alert "Vous avez cliqué sur le bouton OK"
      end function
    </script>
  </head>

  <body>
    <center>
      <h1>Une page Web avec un script VB</h1>
      <table>
        <tr>
          <td>Cliquez sur le bouton</td>
          <td><input type="button" value="OK" name="cmdOK" onclick="reagir"></td>
        </tr>
      </table>
    </body>
  </html>
```

## Une page Web avec un script VB

Cliquez sur le bouton





## 11.2 Les tests

- serveur Apache
- script *vbs1.html* dans *<apache-DocumentRoot>*
- demander l'URL *http://localhost/vbs1.html* avec le navigateur IE

- serveur PWS
- script *vbs1.html* dans *<pws-DocumentRoot>*
- demander l'URL *http://localhost/vbs1.html* avec le navigateur IE

## 12. Une page Web avec un script Javascript, côté navigateur

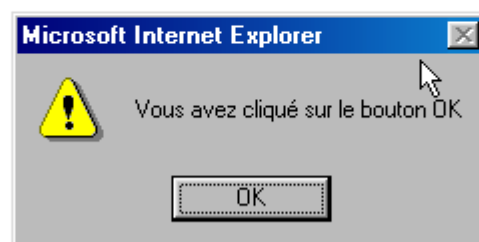
### 12.1 La page : *js1.html*

```
<html>
  <head>
    <title>essai 4 : une page web avec un script Javascript</title>
    <script language="javascript">
      function reagir(){
        alert ("Vous avez cliqué sur le bouton OK");
      }
    </script>
  </head>

  <body>
    <center>
      <h1>Une page Web avec un script Javascript</h1>
      <table>
        <tr>
          <td>Cliquez sur le bouton</td>
          <td><input type="button" value="OK" name="cmdOK" onclick="reagir()"></td>
        </tr>
      </table>
    </body>
</html>
```

## Une page Web avec un script Javascript

Cliquez sur le bouton



### 12.2 Le test

- serveur Apache
- script *js1.html* dans *<apache-DocumentRoot>*
- demander l'URL *http://localhost/js1.html* avec le navigateur IE

- serveur PWS
- script *js1.html* dans *<pws-DocumentRoot>*
- demander l'url *http://localhost/js1.html* avec le navigateur IE

## 13. Une page Web avec un script Perl, côté navigateur

### 13.1 La page : *ps1.html*

```
<html>
  <head>
    <title>Une page web avec un script Perl, côté navigateur</title>
    <script language="perlscript">
      sub reagir{
        $window->alert ("Vous avez cliqué sur le bouton OK");
      }
    </script>
  </head>

  <body>
    <center>
      <h1>Une page Web avec un script Perl, côté navigateur</h1>
      <table>
        <tr>
          <td>Cliquez sur le bouton</td>
          <td><input type="button" value="OK" name="cmdOK" onclick="reagir()"></td>
        </tr>
      </table>
    </body>
</html>
```

## Une page Web avec un script Perl, côté navigateur

Cliquez sur le bouton



### 13.2 Le test

- serveur Apache
- script *ps1.html* dans *<apache-DocumentRoot>*
- demander l'url *http://localhost/ps1.html* avec le navigateur IE
- serveur PWS
- script *ps1.html* dans *<pws-DocumentRoot>*
- demander l'url *http://localhost/ps1.html* avec le navigateur IE

# Scripts côté serveur

## 14. Une page Web avec un script VB, côté serveur

### 14.1 La page : vbs2.asp

```
<html>
<head>
  <title>Une page web avec un script Javascript</title>
  <script language="javascript">
    function reagir(){
      alert ("Vous avez cliqué sur le bouton OK");
    }
  </script>
  <script language="vbscript" runat="server">
    function infos
      infos=time
    end function
  </script>
</head>

<body>
  <center>
    <h1>Une page Web avec </h1>
    <h1>-un script Javascript, côté navigateur</h1>
    <h1>-un script VB, côté serveur</h1>
    <hr>
    <% heure=infos %>
    <h2>Il est <% =heure %></h2>
    <br>
    <table>
      <tr>
        <td>Cliquez sur le bouton</td>
        <td><input type="button" value="OK" name="cmdOK" onclick="reagir()"></td>
      </tr>
    </table>
  </body>
</html>
```

**Une page Web avec**  
**-un script Javascript, côté navigateur**  
**-un script VB, côté serveur**

---

Il est 12:06:10

Cliquez sur le bouton

## **14.2Le test**

- serveur PWS
- page *vbs2.asp* dans *<pws-DocumentRoot>*
- demander url *http://localhost/vbs2.asp* avec un navigateur

# Formulaires Web

## 15. Formulaire Web, javascript côté navigateur

### 15.1 La page : form1.html

```
<html>
<head>
  <title>Formulaire web</title>
  <script language="javascript">
    function effacer() {
      document.frmPersonne.txtNom.value="";
      document.frmPersonne.txtAge.value="";
    }
  </script>
</head>

<body>
  <center>
    <h1>Un formulaire Web</h1>
    <h2>Récupération des valeurs des champs par un script Javascript côté
navigateur</h2>
    <hr>
    <form name="frmPersonne" method="post">
      <br>
      <table>
        <tr>
          <td>Nom</td>
          <td><input type="text" value="" name="txtNom"></td>
        </tr>
        <tr>
          <td>Age</td>
          <td><input type="text" value="" name="txtAge"></td>
        </tr>
        <tr></tr>
        <tr>
          <td><input type="button" name="cmdEffacer" value="Effacer"
onclick="effacer()"></td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

## Un formulaire Web

### Récupération des valeurs des champs par un script Javascript côté navigateur

---

Nom	<input type="text"/>
Age	<input type="text"/>
	<input type="button" value="Effacer"/>

## 15.2 Les tests

- serveur PWS
- mettre le script *form1.html* dans *<pws-DocumentRoot>*
- demander l'URL *http://localhost/form1.html* avec un navigateur
  
- serveur Apache
- mettre le script *form1.html* dans *<apache-DocumentRoot>*
- demander l'URL *http://localhost/form1.html* avec un navigateur

## 16. Formulaire Web, vbscript côté navigateur

### 16.1 La page : *form2.html*

```
<html>
<head>
  <title>Formulaire web</title>
  <script language="vbscript">
    function effacer
      frmPersonne.txtNom.value=""
      frmPersonne.txtAge.value=""
    end function
  </script>
</head>

<body>
  <center>
    <h1>Un formulaire Web</h1>
    <h2>Récupération des valeurs des champs par un script VBscript côté
navigateur</h2>
    <hr>
    <form name="frmPersonne" method="post">
    <br>
    <table>
      <tr>
        <td>Nom</td>
        <td><input type="text" value="" name="txtNom"></td>
      </tr>
      <tr>
        <td>Age</td>
        <td><input type="text" value="" name="txtAge"></td>
      </tr>
      <tr></tr>
      <tr>
        <td><input type="button" name="cmdEffacer" value="Effacer"
onclick="effacer"></td>
      </tr>
    </table>
  </form>
</body>
</html>
```

# Un formulaire Web

## Récupération des valeurs des champs par un script VBscript côté navigateur

---

Nom	<input type="text"/>
Age	<input type="text"/>
<input type="button" value="Effacer"/>	

### 16.2Le test

- serveur PWS
- mettre le script *form2.html* dans *<pbs-DocumentRoot>*
- demander l'URL *http://localhost/form2.html* avec navigateur IE
  
- serveur Apache
- mettre le script *form2.html* dans *<apache-DocumentRoot>*
- demander l'URL *http://localhost/form2.html* avec navigateur IE

## 17.Formulaire Web, Perlscript côté navigateur

### 17.1La page : *form3.html*

```
<html>
<head>
  <title>Formulaire web</title>
  <script language="Perlscript">
    sub effacer{
      $window->document->frmPersonne->txtNom->{value}="";
      $window->document->frmPersonne->txtAge->{value}="";
    }
  </script>
</head>

<body>
  <center>
    <h1>Un formulaire Web</h1>
    <h2>Récupération des valeurs des champs par un script Perlscript côté
navigateur</h2>
    <hr>
    <form name="frmPersonne" method="post">
      <br>
      <table>
        <tr>
          <td>Nom</td>
          <td><input type="text" value="" name="txtNom"></td>
        </tr>
        <tr>
          <td>Age</td>
          <td><input type="text" value="" name="txtAge"></td>
        </tr>
        <tr></tr>
        <tr>
          <td><input type="button" name="cmdEffacer" value="Effacer"
onclick="effacer()"></td>
        </tr>
      </table>
    </form>
  </center>
</body>
</html>
```

```
</form>
</body>
</html>
```

## Un formulaire Web

### Récupération des valeurs des champs par un script Perlscript côté navigateur

---

Nom

Age

#### 17.2 Le test

- serveur PWS
- mettre le script *form3.html* dans *<pws-DocumentRoot>*
- demander l'URL *http://localhost/form3.html* avec le navigateur Internet Explorer
- serveur Apache
- mettre le script *form3.html* dans *<apache-DocumentRoot>*
- demander l'URL *http://localhost/form3.html* avec le navigateur Internet Explorer

## 18. Vérification du formulaire avant l'envoi, par un script côté navigateur

### 18.1 Le texte HTML

```
<html>
<head>
  <title>Formulaire web</title>
  <script language="javascript">
    function effacer(){
      document.frmPersonne.txtNom.value="";
      document.frmPersonne.txtAge.value="";
    }
    function envoyer(){
      // vérification des données
      with (document.frmPersonne) {
        champs=/^\s*(\d+)\s*$/.exec(txtAge.value);
        if(champs == null){
          alert("Age incorrect");
          txtAge.focus();
          return;
        }
        // if
        // l'âge est correct
        txtAge.value=champs[1];
        submit();
      }
    }
  </script>
</head>

<body>
  <center>
    <h1>Un formulaire Web</h1>
    <h2>Test de validité avant envoi</h2>
```



```

<hr>
<form name="frmPersonne" method="post">
<br>
<table>
  <tr>
    <td>Nom</td>
    <td><input type="text" value="" name="txtNom"></td>
  </tr>
  <tr>
    <td>Age</td>
    <td><input type="text" value="" name="txtAge"></td>
  </tr>
  <tr></tr>
  <tr>
    <td><input type="button" name="cmdEffacer" value="Effacer"
onclick="effacer()"></td>
    <td><input type="button" name="cmdEnvoyer" value="Envoyer"
onclick="envoyer()"></td>
  </tr>
</table>
</body>
</html>

```

Nom

Age



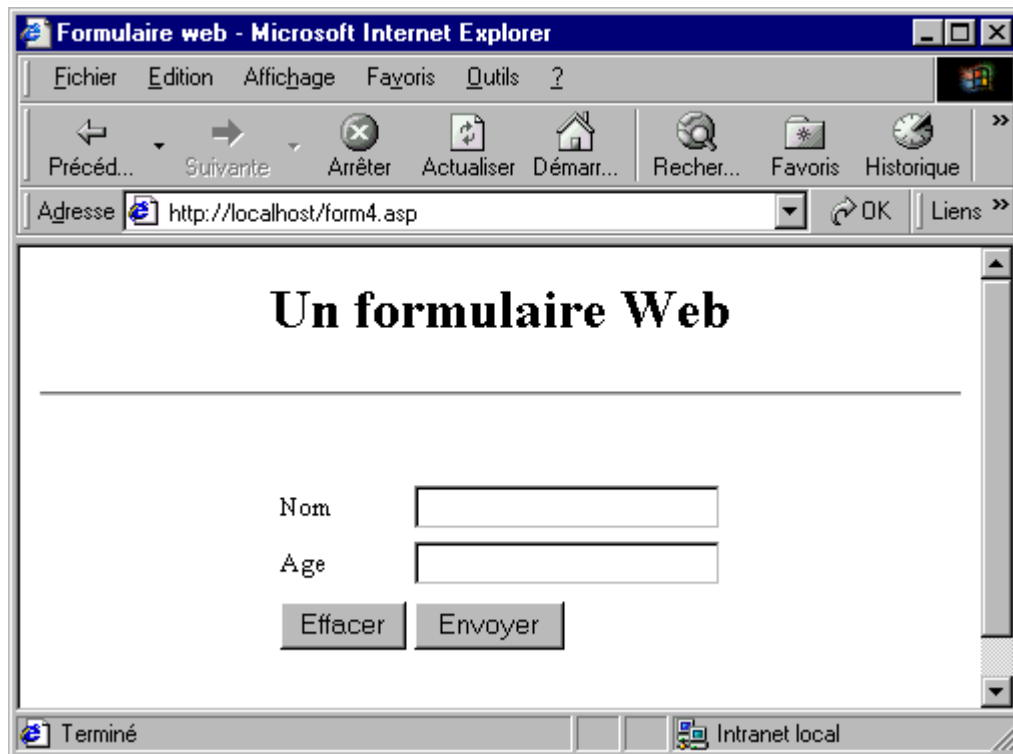
## 18.2 Test

- serveur Apache ou PWS
- navigateur supportant javascript

# Dialogue Client-serveur

## 19. Le dialogue client/serveur

### 19.1 Le formulaire : form4.asp



### 19.2 Le texte HTML du formulaire

```
<html>
  <head>
    <title>Formulaire web</title>
    <script language="javascript">
      function effacer(){
        document.frmPersonne.txtNom.value="";
        document.frmPersonne.txtAge.value="";
      }
    </script>
  </head>

  <body>
    <center>
      <h1>Un formulaire Web</h1>
    </center>
    <hr>
    <form name="frmPersonne" method="get">
      <br>
      <table>
        <tr>
          <td>Nom</td>
          <td><input type="text" value="" name="txtNom"></td>
        </tr>
        <tr>
          <td>Age</td>
          <td><input type="text" value="" name="txtAge"></td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

```

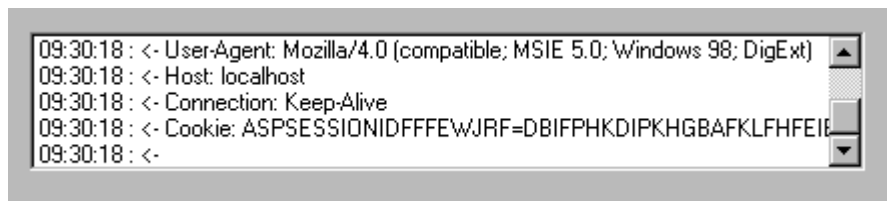
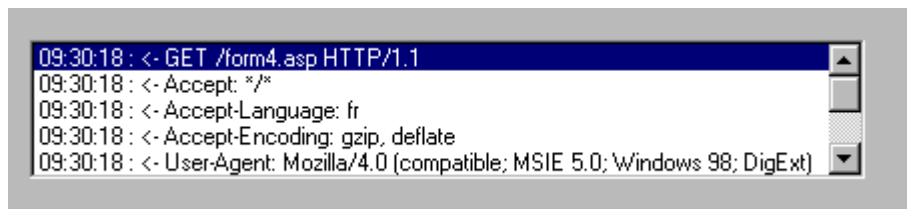
        <td><input type="button" name="cmdEffacer" value="Effacer"
onclick="effacer()"></td>
        <td><input type="submit" name="cmdEnvoyer" value="Envoyer"></td>
    </tr>
</table>
</form>
<hr>
</body>
</html>

```

### 19.3 Demande initiale du formulaire

Pour savoir quelles information le client navigateur envoie au serveur Web lors de la demande initiale du formulaire :

- on ferme tout serveur web fonctionnant sur le port 80
- on lance un serveur tcp générique sur le port 80. Ce serveur fait un suivi de tout ce que les clients lui envoient
- on demande l'URL `http://localhost/form4.asp` avec un navigateur quelconque
- on obtient le résultat suivant avec IE4



### 19.4 Envoi du formulaire : méthode GET

On remplit le formulaire et on appuie sur le bouton Envoyer. Le navigateur va alors transmettre les valeurs du formulaire au serveur de la façon suivante :

## Un formulaire Web

---

Nom	<input type="text" value="Valentin"/>
Age	<input type="text" value="45"/>
	<input type="button" value="Effacer"/> <input type="button" value="Envoyer"/>

- le serveur tcp générique est actif sur le port 80
- on appuie sur le bouton Envoyer
- le navigateur envoie alors les commandes suivantes

```
09:44:27 : <- GET /form4.asp?txtNom=Valentin&txtAge=45&cmdEnvoyer=Envoyer H
09:44:27 : <- Accept: application/vnd.ms-excel, application/msword, application/vn
09:44:27 : <- Referer: http://localhost/form4.asp
09:44:27 : <- Accept-Language: fr
09:44:27 : <- Accept-Encoding: gzip, deflate
```

```
09:44:27 : <- Accept-Encoding: gzip, deflate
09:44:27 : <- User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)
09:44:27 : <- Host: localhost
09:44:27 : <- Connection: Keep-Alive
09:44:27 : <- Cookie: ASPSESSIONIDFFFEWJRF=DBIFPHKDIPKHGBAFKLFHFEIE
```

```
09:44:27 : <- User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)
09:44:27 : <- Host: localhost
09:44:27 : <- Connection: Keep-Alive
09:44:27 : <- Cookie: ASPSESSIONIDFFFEWJRF=DBIFPHKDIPKHGBAFKLFHFEIE
09:44:27 : <-
```

On voit que les paramètres ont été envoyés par le navigateur au serveur par la commande :

```
09:44:27 : <- GET /form4.asp?txtNom=Valentin&txtAge=45&cmdEnvoyer=Envoyer H
```

c.a.d. derrière le nom de l'URL qui va les traiter.

### **19.5 Envoi du formulaire : méthode POST**

On modifie dans le formulaire la méthode d'envoi des valeurs du formulaire au serveur par le navigateur :

```
<form name="frmPersonne" method="POST">
```

On remplit le formulaire et on appuie sur le bouton Envoyer. Le navigateur transmet les valeurs du formulaire au serveur de la façon suivante :

```
10:04:58 : <- POST /form4.asp HTTP/1.1
10:04:58 : <- Accept: application/vnd.ms-excel, application/msword, application/vn
10:04:58 : <- Referer: http://localhost/form4.asp
10:04:58 : <- Accept-Language: fr
10:04:58 : <- Content-Type: application/x-www-form-urlencoded
```

```
10:04:58 : <- Content-Type: application/x-www-form-urlencoded
10:04:58 : <- Accept-Encoding: gzip, deflate
10:04:58 : <- User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)
10:04:58 : <- Host: localhost
10:04:58 : <- Content-Length: 44
```

```
10:04:58 : <- Content-Length: 44
10:04:58 : <- Connection: Keep-Alive
10:04:58 : <- Cookie: ASPSESSIONIDFFFEWJRF=DBIFPHKDIPKHGBAFKLFHEIE
10:04:58 : <-
10:04:58 : <- txtNom=Valentin&txtAge=45&cmdEnvoyer=Envoyer
```

Les commandes ci-dessus nous montrent comment ont été envoyés les paramètres :

- la commande *Content-Length: 44* précise que le navigateur va envoyer 44 caractères
- ces 44 caractères sont envoyés après une ligne vide

```
10:04:58 : <-
10:04:58 : <- txtNom=Valentin&txtAge=45&cmdEnvoyer=Envoyer
```

# Traitement des formulaires côté serveur

## 20. Traitement du formulaire Web avec un script ASP côté serveur

### 20.1 La page ASP pour une méthode d'envoi GET

```
<%
' on récupère les éventuels paramètres - méthode GET
nom=Request.QueryString("txtNom")
age=Request.QueryString("txtAge")
%>

<html>
  <head>
    <title>Formulaire web</title>
    <script language="javascript">
      function effacer(){
        document.frmPersonne.txtNom.value="";
        document.frmPersonne.txtAge.value="";
      }
      function envoyer(){
        // vérification des données
        with (document.frmPersonne) {
          champs=/^\s*(\d+)\s*$/ .exec(txtAge.value);
          if(champs == null){
            alert("Age incorrect");
            txtAge.focus();
            return;
          } // if
          // l'âge est correct
          txtAge.value=champs[1];
          submit();
        } // with
      } // envoyer
    </script>
  </head>

  <body>
    <center>
      <h1>Un formulaire Web</h1>
      <h2>Test de validité avant envoi</h2>
      <hr>
      <form name="frmPersonne" method="get">
        <br>
        <table>
          <tr>
            <td>Nom</td>
            <td><input type="text" name="txtNom" value="<%=nom%>"></td>
          </tr>
          <tr>
            <td>Age</td>
            <td><input type="text" name="txtAge" value="<%=age%>"></td>
          </tr>
          <tr></tr>
          <tr>
            <td><input type="button" name="cmdEffacer" value="Effacer"
            onclick="effacer()"></td>
            <td><input type="button" name="cmdEnvoyer" value="Envoyer"
            onclick="envoyer()"></td>
          </tr>
        </table>
      </form>
      <hr>
      <h2>Valeurs des différents champs du formulaire reçues par le serveur</h2>
      <br>
      <table border=1>
        <%
          response.write("<tr><td>Nom</td><td>" & nom & "</td></tr>")
        %>
      </table>
    </center>
  </body>
</html>
```

```

        response.write("<tr><td>Age</td><td>" & age & "</td></tr>")
    %>
</table>
</body>
</html>

```

## 20.2 Le test

- serveur PWS
- navigateur supportant javascript

# Un formulaire Web

## Test de validité avant envoi

---

Nom	<input type="text" value="mélanie"/>
Age	<input type="text" value="11"/>
	<input type="button" value="Effacer"/> <input type="button" value="Envoyer"/>

---

## Valeurs des différents champs du formulaire reçues par le serveur

Nom	mélanie
Age	11

## 20.3 La page ASP pour une méthode d'envoi POST

```

<%
' on récupère les éventuels paramètres - méthode POST
nom=Request.Form("txtNom")
age=Request.Form("txtAge")
%>

```

```

<html>
<head>
<title>Formulaire web</title>
<script language="javascript">
function effacer(){
document.frmPersonne.txtNom.value="";
document.frmPersonne.txtAge.value="";
}
function envoyer(){
// vérification des données
with (document.frmPersonne) {
champs=/^\s*(\d+)\s*$/.exec(txtAge.value);
if(champs == null){
alert("Age incorrect");
txtAge.focus();
return;
}
}
// l'âge est correct
txtAge.value=champs[1];
submit();

```

```

    }// with
    }// envoyer
</script>
</head>

<body>
  <center>
    <h1>Un formulaire Web</h1>
    <h2>Test de validité avant envoi</h2>
    <hr>
    <form name="frmPersonne" method="post">
      <br>
      <table>
        <tr>
          <td>Nom</td>
          <td><input type="text" name="txtNom" value="<%=nom%>"></td>
        </tr>
        <tr>
          <td>Age</td>
          <td><input type="text" name="txtAge" value="<%=age%>"></td>
        </tr>
        <tr></tr>
        <tr>
          <td><input type="button" name="cmdEffacer" value="Effacer"
onclick="effacer()"></td>
          <td><input type="button" name="cmdEnvoyer" value="Envoyer"
onclick="envoyer()"></td>
        </tr>
      </table>
    </form>
    <hr>
    <h2>Valeurs des différents champs du formulaire reçues par le serveur</h2>
    <br>
    <table border=1>
      <%
        response.write("<tr><td>Nom</td><td>" & nom & "</td></tr>")
        response.write("<tr><td>Age</td><td>" & age & "</td></tr>")
      %>
    </table>
  </body>
</html>

```

## 20.4Le test

- serveur PWS
- navigateur supportant javascript



# Un formulaire Web

## Test de validité avant envoi

---

Nom	<input type="text" value="joachim"/>
Age	<input type="text" value="13"/>
	<input type="button" value="Effacer"/> <input type="button" value="Envoyer"/>

---

### Valeurs des différents champs du formulaire reçues par le serveur

Nom	joachim
Age	13

## 21. Traitement d'un formulaire Web avec un script Perl côté serveur

### 21.1 Le script Perl

```
#!d:\perl\bin\perl.exe
```

```
use CGI qw/:all/;  
use strict;
```

```
# on récupère les éventuels paramètres
```

```
my $nom=param("txtNom");  
my $age=param("txtAge");
```

```
# on écrit la page Web  
print <<FINHTML  
Content-Type : text/html
```

```
<html>  
<head>  
  <title>Formulaire web</title>  
  <script language="javascript">  
    function effacer(){  
      document.frmPersonne.txtNom.value="";  
      document.frmPersonne.txtAge.value="";  
    }  
    function envoyer(){  
      // vérification des données  
      with (document.frmPersonne) {  
        champs= /^\s*(\d+)\s*$/ .exec(txtAge.value);  
        if(champs == null){
```

```

        alert("Age incorrect");
        txtAge.focus();
        return;
    }// if
    // l'âge est correct
    txtAge.value=champs[1];
    submit();
    }// with
} // envoyer
</script>
</head>

<body>
<center>
<h1>Un formulaire Web</h1>
<h2>Récupération des valeurs des champs d'un formulaire</h2>
<h2>par un script Perl côté serveur</h2>
<hr>
<form name="frmPersonne" method="get">
<br>
<table>
<tr>
<td>Nom</td>
<td><input type="text" value="$nom" name="txtNom"></td>
</tr>
<tr>
<td>Age</td>
<td><input type="text" value="$age" name="txtAge"></td>
</tr>
<tr></tr>
<tr>
<td><input type="button" name="cmdEffacer" value="Effacer" onclick="effacer()"></td>
<td><input type="button" name="cmdEnvoyer" value="Envoyer" onclick="envoyer()"></td>
</tr>
</table>
</form>
<hr>
<h2>Valeurs des différents champs du formulaire reçus par le serveur</h2>
<br>
<table border=1>
<tr>
<td>Nom</td><td>$nom</td>
</tr>
<tr>
<td>Age</td><td>$age</td>
</tr>
</table>
</body>
</html>
FINHTML
;

```

Noter que l'une des lignes de la fonction javascript a été modifiée afin de prendre en compte le fait que certains caractères ont une signification en Perl.

```
champs=/^\s*(\d+)\s*$/ .exec (txtAge.value) ;
```

C'est le cas du caractère \ : \c veut dire que le caractère c doit être pris pour ce qu'il est et non être interprété.

Ainsi le caractère \$ a une signification en Perl. Pour éviter l'interprétation on écrit \\$. Le caractère \ a une signification en Perl. Pour éviter son interprétation, on écrira \\.

## 21.2 Le test

- serveur Apache
- mettre le script *form5.pl* dans *<apache-cgi-bin>*
- demander l'URL *http://localhost/cgi-bin/form5.pl* avec un navigateur
- modifier le texte HTML pour que la méthode d'envoi des valeurs du formulaire par le navigateur soit successivement GET puis POST. Constaté que le script Perl ci-dessus est valable dans les deux cas
- noter l'URL de la page ramenée par le GET :  
*http://localhost:8080/cgi-bin/form5.pl?txtNom=valentin&txtAge=45*

## Un formulaire Web

### Récupération des valeurs des champs d'un formulaire par un script Perl côté serveur

---

Nom	<input type="text" value="joachim"/>
Age	<input type="text" value="13"/>
	<input type="button" value="Effacer"/> <input type="button" value="Envoyer"/>

---

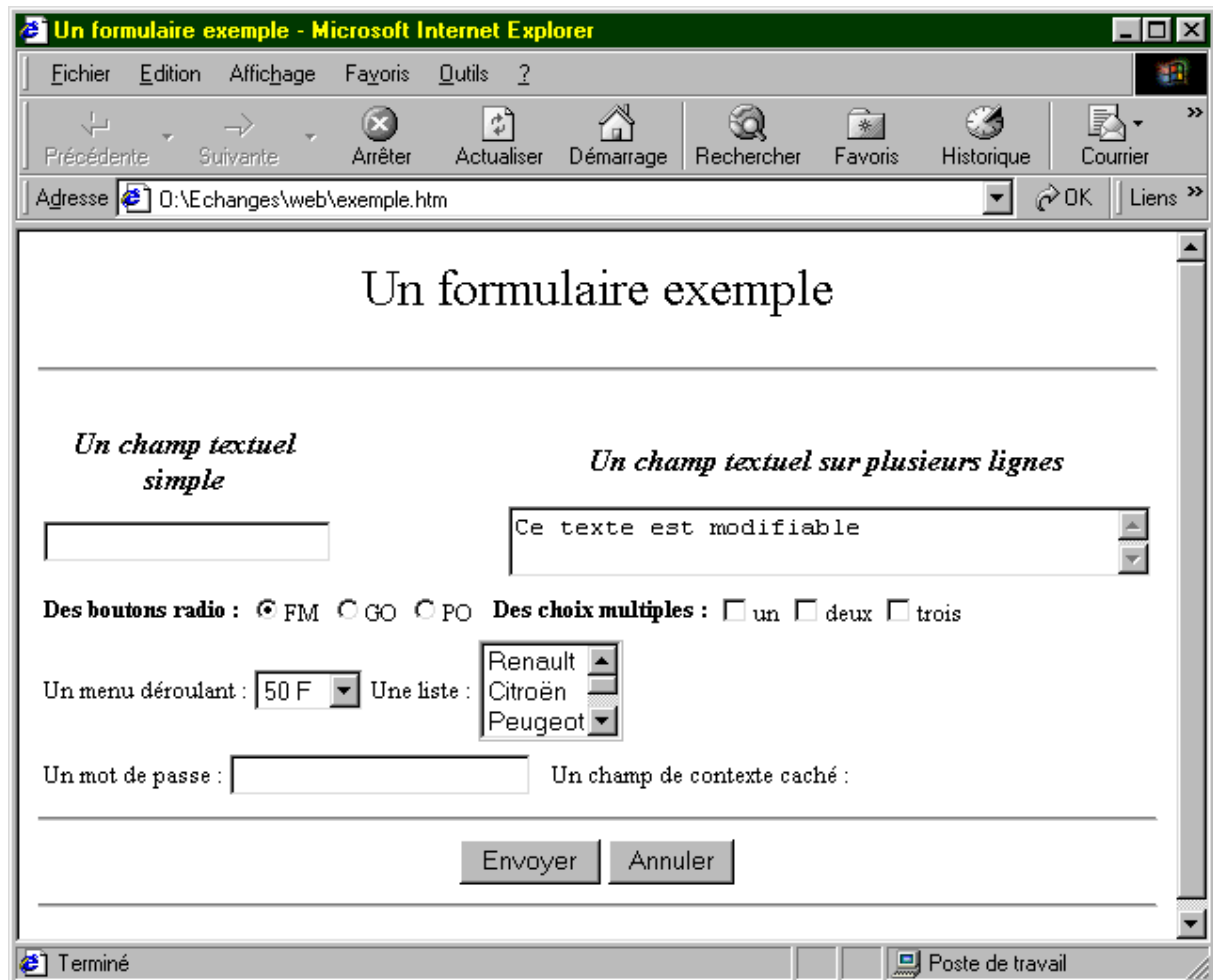
### Valeurs des différents champs du formulaire reçues par le serveur

Nom	joachim
Age	13

# Un exemple général de formulaire Web

## 22. Technologie ASP

### 22.1 Le formulaire à construire



Ce formulaire a été construit avec Microsoft FrontPage Express. Il présente les principaux contrôles qu'on peut trouver dans un formulaire Web. En construisant un formulaire avec un outil de génération de pages HTML et en consultant ensuite le code HTML généré on peut découvrir facilement les balises dont on a besoin.

### 22.2 Les balises HTML de base

```
<html>
<head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">
<meta name="GENERATOR" content="Microsoft FrontPage Express 2.0">
<title>Un formulaire exemple</title>
</head>
<body bgcolor="#FFFFFF">
```

```

<p align="center"><font size="6">Un formulaire exemple</font></p>

<hr>

<form method="GET" name="frmExemple">
  <input type="hidden" name="cache" value="secret"><div
    align="center"><center><table border="0">
      <tr>
        <td align="center"><font size="4"><em><strong>Un
          champ textuel simple</strong></em></font></td>
        <td align="center" width="100">&nbsp;</td>
        <td align="center"><font size="4"><em><strong>Un
          champ textuel sur plusieurs lignes</strong></em></font></td>
      </tr>
      <tr>
        <td align="center"><input type="text" size="20"
          name="simple"></td>
        <td align="center" width="100">&nbsp;</td>
        <td align="center"><textarea name="lignes" rows="2"
          cols="40">Ce texte est modifiable</textarea></td>
      </tr>
    </table>
  </center></div><table border="0">
    <tr>
      <td><strong>Des boutons radio :</strong></td>
      <td><input type="radio" checked name="radio"
        value="FM">FM</td>
      <td><input type="radio" name="radio" value="GO">GO</td>
      <td><input type="radio" name="radio" value="PO">PO</td>
      <td>&nbsp;</td>
      <td><strong>Des choix multiples :</strong></td>
      <td><input type="checkbox" name="qcm" value="un">un</td>
      <td><input type="checkbox" name="qcm" value="deux">deux</td>
      <td><input type="checkbox" name="qcm" value="trois">trois</td>
    </tr>
  </table>
  <table border="0">
    <tr>
      <td>Un menu déroulant : </td>
      <td><select name="menu" size="1">
        <option>50 F</option>
        <option>60 F</option>
        <option>70 F</option>
        <option>100 F</option>
      </select></td>
      <td>Une liste :</td>
      <td><select name="lstVoitures" multiple size="3">
        <option>Renault</option>
        <option>Citroën</option>
        <option>Peugeot</option>
        <option>Fiat</option>
        <option>Audi</option>
      </select></td>
    </tr>
  </table>
  <table border="0">
    <tr>
      <td>Un mot de passe : </td>
      <td><input type="password" size="21" name="passwd"></td>
      <td>&nbsp;</td>
      <td>Un champ de contexte caché : </td>
    </tr>
  </table>
  <hr>
  <div align="center"><center><table border="0">
    <tr>
      <td align="center"><input type="submit"
        name="cmdEnvoyer" value="Envoyer"></td>
      <td align="center"><input type="reset"
        name="cmdAnnuler" value="Annuler"></td>
    </tr>
  </table>
  </center></div><hr>
</form>
</body>
</html>

```

## 22.3 Une page HTML simplifiée

Avec un peu d'expérience, on voit que la page générée par FrontPage est alourdie de balises sans réelle utilité. Une page simplifiée serait la suivante :

```
<html>

<head>
  <title>Un formulaire exemple</title>
</head>

<body bgcolor="#FFFFFF">
  <center>
    <h1>Un formulaire exemple</h1>
    <hr>
    <form method="GET" name="frmExemple">
      <input type="hidden" name="cache" value="secret">
      <table border="0">
        <tr>
          <td align="center">Un champ textuel simple</td>
          <td align="center" width="100">&nbsp;</td>
          <td align="center">Un champ textuel sur plusieurs lignes</td>
        </tr>
        <tr>
          <td align="center"><input type="text" size="20" name="simple"></td>
          <td align="center" width="100">&nbsp;</td>
          <td align="center">
            <textarea name="lignes" rows="2" cols="40">
              Ce texte est modifiable
            </textarea>
          </td>
        </tr>
      </table>
      <table border="0">
        <tr>
          <td><strong>Des boutons radio :</strong></td>
          <td>
            <input type="radio" checked name="radio" value="FM">FM
          </td>
          <td>
            <input type="radio" name="radio" value="GO">GO
          </td>
          <td>
            <input type="radio" name="radio" value="PO">PO
          </td>
          <td>&nbsp;</td>
          <td><strong>Des choix multiples :</strong></td>
          <td>
            <input type="checkbox" name="qcm" value="un">un
          </td>
          <td>
            <input type="checkbox" name="qcm" value="deux">deux
          </td>
          <td>
            <input type="checkbox" name="qcm" value="trois">trois
          </td>
        </tr>
      </table>
      <table border="0">
        <tr>
          <td>Un menu déroulant : </td>
          <td>
            <select name="menu" size="1">
              <option>50 F</option>
              <option>60 F</option>
              <option>70 F</option>
              <option>100 F</option>
            </select>
          </td>
          <td>Une liste :</td>
          <td>
            <select name="lstVoitures" multiple size="3">
              <option>Renault</option>
              <option>Citroën</option>
              <option>Peugeot</option>
              <option>Fiat</option>
              <option>Audi</option>
            </select>
          </td>
        </tr>
      </table>
    </form>
  </center>
</body>
</html>
```

```

        </td>
    </tr>
</table>
<table border="0">
    <tr>
        <td>Un mot de passe : </td>
        <td><input type="password" size="21" name="passwd"></td>
        <td>&nbsp;</td>
        <td>Un champ de contexte caché : </td>
    </tr>
</table>
<hr>
<table border="0">
    <tr>
        <td align="center">
            <input type="submit" name="cmdEnvoyer" value="Envoyer">
        </td>
        <td align="center">
            <input type="reset" name="cmdAnnuler" value="Annuler">
        </td>
    </tr>
</table>
<hr>
</form>
</body>
</html>

```

## 22.4 Passage des valeurs du formulaire par la méthode GET

objectif : récupérer les valeurs d'un formulaire transmises par un navigateur via la méthode GET

```

<hr>
<form>
<center><h3>Collection QueryString (méthode GET)</h3></center>
<table border=1>
    <% for each item in Request.QueryString
        for i=1 to Request.QueryString(item).count
            response.write("<tr><td " & color1 & ">" & item & "(" & i & ")" & "</td><td " &
color2 & "> " & request.querystring(item)(i) & "</td></tr>")
        next
    next
    %>
</table>
</form>
</body>
</html>

```

### Tests

Ajouter le texte précédent derrière la balise </form> de la page html précédente. Sauvegarder le tout dans un fichier .asp. Tester avec PWS.

On fait la saisie suivante :

# Un formulaire exemple

---

Un champ textuel simple	Un champ textuel sur plusieurs lignes
<input type="text" value="qqs mots"/>	<input style="width: 150px;" type="text" value="ligne1"/> <input style="width: 150px;" type="text" value="ligne2"/>
Des boutons radio : <input type="radio"/> FM <input type="radio"/> GO <input checked="" type="radio"/> PO	
Des choix multiples : <input checked="" type="checkbox"/> un <input checked="" type="checkbox"/> deux <input type="checkbox"/> trois	
Un menu déroulant : <input type="text" value="100 F"/>	Une liste : <input type="text" value="Renault"/> <input type="text" value="Citroën"/> <input type="text" value="Peugeot"/>
Un mot de passe : <input style="width: 100px;" type="password" value="azerty"/>	Un champ de contexte caché :

---

<input type="submit" value="Envoyer"/>	<input type="submit" value="Annuler"/>
--	--

---

Le résultat obtenu est le suivant :

## Collection QueryString (méthode GET)

cache(1)	secret
menu(1)	100 F
simple(1)	qqs mots
lignes(1)	ligne1 ligne2
qcm(1)	un
qcm(2)	deux
radio(1)	PO
lstVoitures(1)	Renault
lstVoitures(2)	Peugeot
passwd(1)	azerty
cmdEnvoyer(1)	Envoyer

## 22.5 Passage des valeurs du formulaire par la méthode POST

objectif : récupérer les valeurs d'un formulaire transmises par un navigateur via la méthode POST

```
<hr>
<form>
<center><h3>Collection Form (méthode POST)</h3></center>
<table border=1>
  <% for each item in Request.Form
    for i=1 to Request.Form(item).count
      response.write("<tr><td " & color1 & ">" & item & "(" & i & ")" & "</td><td " &
color2 & ">" & request.form(item)(i) & "</td></tr>")
    next
  next
  %>
</table>
</form>
</body>
```



```
<html>
```

## Tests

Ajouter le texte précédent entre les balises `</form>` et `</body>` de la page html initiale du formulaire. Sauvegarder le tout dans un fichier .asp. Tester avec PWS.

## 22.6 Récupération des variables d'environnement du serveur Web

Un script côté serveur a accès à un certain nombre de variables dites d'environnement, qui permettent de connaître diverses informations :

- Les entêtes envoyés par le navigateur
- L'identité du navigateur

Voyons comment un script ASP peut récupérer ces informations :

```
<html>
<head>
  <title>Environnement ASP</title>
</head>
<body>
  <h2>Variables d'environnement du serveur PWS</h2>
  <form>
    <table border=1>
      <%
        for each item in Request.ServerVariables
          for i=1 to Request.ServerVariables(item).count
            response.write("<tr><td>" & item & "(" & i & ")" & "</td><td> " &
Request.ServerVariables(item)(i) & "</td></tr>")
          next
        next
      <%>
    </table>
  </form>
</body>
</html>
```

On obtient un résultat dont voici le début :

### Variables d'environnement du serveur PWS

ALL_HTTP(1)	HTTP_ACCEPT:image/gif, image/x-bitmap, image/jpeg, image/pjpeg, application/vnd.ms-excel, application/msword, application/vnd.ms-powerpoint, /* HTTP_ACCEPT_LANGUAGE:fr HTTP_CONNECTION:Keep-Alive HTTP_HOST:localhost HTTP_USER_AGENT:Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt) HTTP_COOKIE:ASPSESSIONIDFFERZBIV=BBMBILOBACLDNMBFJKICBGLF HTTP_ACCEPT_ENCODING:gzip, deflate
ALL_RAW(1)	Accept: image/gif, image/x-bitmap, image/jpeg, image/pjpeg, application/vnd.ms-excel, application/msword, application/vnd.ms-powerpoint, /* Accept-Language: fr Connection: Keep-Alive Host: localhost User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt) Cookie: ASPSESSIONIDFFERZBIV=BBMBILOBACLDNMBFJKICBGLF Accept-Encoding: gzip, deflate
APPL_MD_PATH(1)	/LM/W3SVC/1/ROOT
APPL_PHYSICAL_PATH(1)	C:\inetpub\wwwroot\
AUTH_PASSWORD(1)	
AUTH_TYPE(1)	
AUTH_USER(1)	
CERT_COOKIE(1)	
CERT_FLAGS(1)	

## 23. Technologie Perl

On reprend le problème traité précédemment avec la technologie ASP.

### 23.1 Le formulaire à construire

#### Un formulaire exemple

---

*Un champ textuel simple*

une ligne

*Un champ textuel sur plusieurs lignes*

ligne1  
ligne2

Des boutons radio :  FM  GO  PO Des choix multiples :  un  deux  trois

Un menu déroulant : 70 F

Une liste :

Renault  
Citroën  
Peugeot

Un mot de passe :

secret

Un champ de contexte caché :

Envoyer

Annuler

### 23.2 Le script Perl d'affichage et de traitement du formulaire

Le formulaire précédent est construit et traité avec le script Perl suivant :

```
#!d:\perl\bin\perl.exe

# modules
use CGI qw/:all/;
use strict;

# on envoie la 1ère partie de la page

print <<HTML1
Content-type: text/html

<html>

<head>
  <title>Un formulaire exemple</title>
</head>

<body bgcolor="#FFFFFF">
  <center>
    <h1>Un formulaire exemple</h1>
    <hr>
    <form method="POST" name="frmExemple">
      <input type="hidden" name="cache" value="secret">
      <table border="0">
        <tr>
          <td align="center">Un champ textuel simple</td>
          <td align="center" width="100">&nbsp;</td>
          <td align="center">Un champ textuel sur plusieurs lignes</td>
```

```

</tr>
<tr>
  <td align="center"><input type="text" size="20" name="simple"></td>
  <td align="center" width="100">&nbsp;</td>
  <td align="center">
    <textarea name="lignes" rows="2" cols="40">
      Ce texte est modifiable
    </textarea>
  </td>
</tr>
</table>
<table border="0">
  <tr>
    <td><strong>Des boutons radio :</strong></td>
    <td>
      <input type="radio" checked name="radio" value="FM">FM
    </td>
    <td>
      <input type="radio" name="radio" value="GO">GO
    </td>
    <td>
      <input type="radio" name="radio" value="PO">PO
    </td>
    <td>&nbsp;</td>
  <td><strong>Des choix multiples :</strong></td>
  <td>
    <input type="checkbox" name="qcm" value="un">un
  </td>
  <td>
    <input type="checkbox" name="qcm" value="deux">deux
  </td>
  <td>
    <input type="checkbox" name="qcm" value="trois">trois
  </td>
</tr>
</table>
<table border="0">
  <tr>
    <td>Un menu déroulant : </td>
    <td>
      <select name="menu" size="1">
        <option>50 F</option>
        <option>60 F</option>
        <option>70 F</option>
        <option>100 F</option>
      </select>
    </td>
  <td>Une liste :</td>
  <td>
    <select name="lstVoitures" multiple size="3">
      <option>Renault</option>
      <option>Citroën</option>
      <option>Peugeot</option>
      <option>Fiat</option>
      <option>Audi</option>
    </select>
  </td>
</tr>
</table>
<table border="0">
  <tr>
    <td>Un mot de passe : </td>
    <td><input type="password" size="21" name="passwd"></td>
    <td>&nbsp;</td>
    <td>Un champ de contexte caché : </td>
  </tr>
</table>
<hr>
<table border="0">
  <tr>
    <td align="center">
      <input type="submit" name="cmdEnvoyer" value="Envoyer">
    </td>
    <td align="center">

```

```

        <input type="reset" name="cmdAnnuler" value="Annuler">
    </td>
</tr>
</table>
<hr>
</form>
<hr>
HTML1
;

# on envoie un complément s'il y avait des paramètres
if(param()){
    # traitement du formulaire
    print "<table>\n";
    my @parametres=param();
    foreach my $param (@parametres){
        my @valeurs=param($param);
        print "<tr><td>$param</td><td>".join(", ",@valeurs)."</td></tr>\n";
    }#foreach
    print "</table>\n";
}#if

# on envoie la fin de la page
print <<HTML2
</body>
</html>
HTML2
;

```

### 23.3 Le test

Lorsque ce script est exécuté avec le formulaire présenté précédemment, on obtient les résultats suivants :

### Valeurs du formulaire reçues par le serveur

cache	secret
simple	une ligne
lignes	ligne1 ligne2
radio	PO
qcm	un,trois
menu	70 F
lstVoitures	Renault,Peugeot
passwd	mdp
cmdEnvoyer	Envoyer

### 23.4 Récupération des variables d'environnement du serveur

Voici un script Perl qui récupère les variables d'environnement du serveur Apache :

```

#!D:\perl\bin\perl.exe

# modules
use CGI qw/:all/;
use strict;

```

```

# on envoie la lère partie de la page

print <<HTML1
Content-type: text/html

<html>
  <head>
    <title>Environnement PERL/APACHE</title>
  </head>
  <body>
    <h2>Variables d'environnement du serveur APACHE</h2>
    <form>
      <table border=1>
HTML1
;

# affichage des variables d'environnement
foreach my $param (keys %ENV){
  print "<tr><td>$param</td><td>".$ENV{$param}."</td></tr>";
}

# fin de page
print <<HTML3
  </table>
</body>
</html>
HTML3
;

```

Voici un début de la page obtenue :

## Variables d'environnement du serveur APACHE

SERVER_SOFTWARE	Apache/1.3.12 (Win32) tomcat/1.0
GATEWAY_INTERFACE	CGI/1.1
DOCUMENT_ROOT	d:/data/apache/htdocs
REMOTE_ADDR	127.0.0.1
SERVER_PROTOCOL	HTTP/1.1
SERVER_SIGNATURE	Apache/1.3.12 Server at stake.istia.uang Port 80
REQUEST_METHOD	GET
COMSPEC	C:\COMMAND.COM
QUERY_STRING	
HTTP_USER_AGENT	Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)
WINDIR	C:\WINDOWS
PATH	C:\Program Files\Apache Group\Apache;C:\jdk1.3\bin;C:\ORACLE\ORA81\BIN;C:\PROGRAM FILES\ORACI\BIN;C:\NOVELL\CLIENT32;C:\WINDOWS;C:\WINDOWS\COMMAND;C:\PROGRAM FILES\MTS;C\ADTL\PF\BORLAND\CBUILD~1\BIN;C\PROGRA~1\BORLAND\CBUILD~1\PROJECTS\BPL;C\PROGRA~1\BORLAND\BIN;C\PROGRA~1\BORLAND\DELPHI\PROJECTS\BPL;Z:\WINSHARE;Y:\PUBLIC;X:\BIN;V:\ACCESS
HTTP_ACCEPT	image/gif, image/x-bitmap, image/jpeg, image/pjpeg, application/vnd.ms-excel, application/msword, application/

## 24. Amélioration du script Perl

Dans l'exemple précédent, lorsque l'utilisateur appuie sur le bouton *Envoyer*, la page lui revient vide avec en plus les résultats demandés. On voudrait que la page revienne avec les valeurs qu'elle avait lorsqu'elle a été envoyée.

### 24.1 Le script

```
#!c:\perl\bin\perl.exe

use strict;
use CGI qw/:all/;

# initialisations
my $lignes="Ce texte est modifiable";
my ($simple, $radio, @qcm, $qcm, $menu, @lstVoitures, $lstVoitures, $passwd, $cache);

# on récupère les paramètres
if (param()){
    $simple=param("simple");
    $lignes=param("lignes");
    $radio=param("radio");
    @qcm=param("qcm");
    $qcm=join(' ',@qcm);
    $menu=param("menu");
    @lstVoitures=param("lstVoitures");
    $lstVoitures=join(' ',@lstVoitures);
    $passwd=param("passwd");
    $cache=param("cache");
}

# qqs traitements
my $FM="checked" if $radio eq "FM";
my $GO="checked" if $radio eq "GO";
my $PO="checked" if $radio eq "PO";
my $QCM1="checked" if $qcm =~ /1/;
my $QCM2="checked" if $qcm =~ /2/;
my $QCM3="checked" if $qcm =~ /3/;
my $menu50="selected" if $menu =~ /50/;
my $menu60="selected" if $menu =~ /60/;
my $menu70="selected" if $menu =~ /70/;
my $menu100="selected" if $menu =~ /100/;
my $renault="selected" if $lstVoitures =~ /Renault/;
my $citroen="selected" if $lstVoitures =~ /Citroën/;
```

```

my $peugeot="selected" if $lstVoitures=~Peugeot/;
my $fiat="selected" if $lstVoitures=~Fiat/;
my $audi="selected" if $lstVoitures=~Audi/;

# on affiche le formulaire de base
&afficheDebutFormulaire;
&afficheValeursFormulaires if param();
&afficheFinFormulaire;

sub afficheDebutFormulaire{
print <<HTML1
Content-type: text/html

<html>
  <head>
    <title>Script CGI/PERL</title>
  </head>
  <body bgcolor="#C0C0C0">
    <center>
      <h1>Un formulaire exemple </h1>
      <hr>
    </center>
    <form name="frmExemple" method="post">
      <table>
        <tr>
          <td><strong>Un champ textuel simple</strong></td><td><input name="simple" size=30
value="$simple"></td></tr>
        </tr>
        <tr>
          <td><strong>Un champ textuel sur plusieurs lignes</strong></td>
          <td><textarea name="lignes" rows=5 cols=40>$lignes
          </textarea></td>
        </td>
        </tr>
        <tr>
          <td><strong>Des boutons type radio</strong></td>
          <td>
            <input type="radio" name="radio" value="FM" $FM>Modulation de fréquences
            <input type="radio" name="radio" value="GO" $GO>Grandes Ondes
            <input type="radio" name="radio" value="PO" $PO>Petites Ondes
          </td>
        </tr>
        <tr>
          <td><strong> Des choix multiples</strong></td>
          <td>
            <input type="checkbox" name="qcm" value=1 $QCM1> un
            <input type="checkbox" name="qcm" value=2 $QCM2> deux
            <input type="checkbox" name="qcm" value=3 $QCM3> trois
          </td>
        </tr>
        <tr>
          <td><strong>Un menu déroulant</strong></td>
          <td>
            <select name="menu">
              <option $menu50> 50F </option>
              <option $menu60> 60F </option>
              <option $menu70> 70F </option>
              <option $menu100> 100F </option>
            </select>
          </td>
        </tr>
        <tr>
          <td><strong>Une liste à choix multiple</strong></td>
          <td>
            <select name="lstVoitures" multiple>
              <option $renault> Renault </option>
              <option $citroen> Citroën </option>
              <option $peugeot> Peugeot </option>
              <option $fiat> Fiat </option>
              <option $audi> Audi </option>
            </select>
          </td>
        </tr>
        <tr>
          <td><strong>Un mot de passe</strong></td>
          <td>
            <input type="password" name="passwd" value="passwd" size=40 value="$passwd">
          </td>
        </tr>
      </table>
    </form>
  </body>
</html>

```

```

        <td><strong>Un champ de contexte caché</strong></td>
        <td>
            <input type="hidden" name="cache" value="secret">
        </td>
    </tr>
</table>
<hr>
<center>
<table>
    <tr>
        <td><input type="reset" value="Annuler"></td>
        <td><input type="submit" value="Envoyer"></td>
    </tr>
</table>
</form>
HTML1
;
}#afficheDebutFormulaire

sub afficheFinFormulaire{
print <<HTML3
    </body></html>
HTML3
;
}#afficheFinFormulaire

sub afficheValeursFormulaires{
print <<HTML2
    <hr>
    <center>
    <h2>Paramètres saisis</h2>
    <table>
        <tr><td><strong>Simple</strong></td><td>$simple</td></tr>
        <tr><td><strong>Lignes</strong></td><td>$lignes</td></tr>
        <tr><td><strong>Radio</strong></td><td>$radio</td></tr>
        <tr><td><strong>Qcm</strong></td><td>$qcm</td></tr>
        <tr><td><strong>Menu</strong></td><td>$menu</td></tr>
        <tr><td><strong>lstVoitures</strong></td><td>@lstVoitures</td>
        <tr><td><strong>Passwd</strong></td><td>$passwd</td></tr>
        <tr><td><strong>Cache</strong></td><td>$cache</td></tr>
    </table>
    </center>
HTML2
;
}#afficheValeursFormulaires

```

## 24.2 Les tests

**Un formulaire exemple**

---

<b>Un champ textuel simple</b>	<input type="text" value="qqq mots"/>
<b>Un champ textuel sur plusieurs lignes</b>	<div style="border: 1px solid #ccc; padding: 2px;">ligne1 ligne2 ligne3</div>
<b>Des boutons type radio</b>	<input type="radio"/> Modulation de fréquences <input type="radio"/> Grandes Ondes <input checked="" type="radio"/> Petites Ondes
<b>Des choix multiples</b>	<input type="checkbox"/> un <input checked="" type="checkbox"/> deux <input checked="" type="checkbox"/> trois
<b>Un menu déroulant</b>	<div style="border: 1px solid #ccc; padding: 2px;">70F</div>
<b>Une liste à choix multiple</b>	<div style="border: 1px solid #ccc; padding: 2px;">Renault Citroën Peugeot Fiat</div>
<b>Un mot de passe</b>	<input type="password" value="aaaa"/>
<b>Un champ de contexte caché</b>	



## Paramètres saisis

<b>Simple</b>	qqs mots
<b>Lignes</b>	ligne1 ligne2 ligne3
<b>Radio</b>	PO
<b>Qcm</b>	2 3
<b>Menu</b>	70F
<b>lstVoitures</b>	Renault Peugeot
<b>Passwd</b>	tahe
<b>Cache</b>	secret

## 25. Technologie JSP

Nous reprenons l'étude du formulaire traité précédemment avec les technologies ASP et PERL.

### 25.1 Le script JSP

```
<%@ page import="java.util.*" %>

<html>
<head>
  <title>Essai page jsp</title>
</head>
<body bgcolor="#C0C0C0">
  <center>
    <h1>Formulaire traité par JSP(Java Server Pages)</h1>
    <hr>
    <form method="post">
      <input type="hidden" name="cache" value="secret">
      <table border="0">
        <tr>
          <td align="center">Un champ textuel simple</td>
          <td align="center" width="100">&nbsp;</td>
          <td align="center">Un champ textuel sur plusieurs lignes</td>
        </tr>
        <tr>
          <td align="center"><input type="text" size="20" name="simple"></td>
          <td align="center" width="100">&nbsp;</td>
          <td align="center">
            <textarea name="lignes" rows="2" cols="40">Ce texte est modifiable
            </textarea>
          </td>
        </tr>
      </table>
      <table border="0">
        <tr>
          <td><strong>Des boutons radio :</strong></td>
          <td>
            <input type="radio" checked name="radio" value="FM">FM
          </td>
          <td>
            <input type="radio" name="radio" value="GO">GO
          </td>
          <td>
            <input type="radio" name="radio" value="PO">PO
          </td>
          <td>&nbsp;</td>
        </tr>
        <tr>
          <td><strong>Des choix multiples :</strong></td>
          <td>
            <input type="checkbox" name="qcm" value="un">un
          </td>
          <td>
            <input type="checkbox" name="qcm" value="deux">deux
          </td>
        </tr>
      </table>
    </form>
  </center>
</body>
</html>
```

```

        </td>
        <td>
            <input type="checkbox" name="qcm" value="trois">trois
        </td>
    </tr>
</table>
<table border="0">
    <tr>
        <td>Un menu déroulant : </td>
        <td>
            <select name="menu" size="1">
                <option>50 F</option>
                <option>60 F</option>
                <option>70 F</option>
                <option>100 F</option>
            </select>
        </td>
        <td>Une liste :</td>
        <td>
            <select name="lstVoitures" multiple size="3">
                <option>Renault</option>
                <option>Citroën</option>
                <option>Peugeot</option>
                <option>Fiat</option>
                <option>Audi</option>
            </select>
        </td>
    </tr>
</table>
<table border="0">
    <tr>
        <td>Un mot de passe : </td>
        <td><input type="password" size="21" name="passwd"></td>
        <td>&nbsp;</td>
        <td>Un champ de contexte caché : </td>
    </tr>
</table>
<hr>
<table border="0">
    <tr>
        <td align="center">
            <input type="submit" name="cmdEnvoyer" value="Envoyer">
        </td>
        <td align="center">
            <input type="reset" name="cmdAnnuler" value="Annuler">
        </td>
    </tr>
</table>
<hr>
</form>
<hr>
<h1>Paramètres du formulaire</h1>
<table border=1>
<%
    // on recupere la liste des parametres du formulaire
    Enumeration liste;
    liste=request.getParameterNames();
    // on parcourt cette liste
    while(liste.hasMoreElements()){
        // nom et valeurs du parametre courant
        String nomParam=(String) liste.nextElement();
        String[] valeursParam=request.getParameterValues(nomParam);
        // parcours & affichage des valeurs
        if(valeursParam!=null)
            for(int i=0; i<valeursParam.length;i++){
                String valeur=valeursParam[i];
                if(valeur.equals("")) valeur="&nbsp;";
                out.println("<tr><td>" +nomParam+" (" +i+")</td><td>" +valeur+"</td></tr>");
            }
        else out.println("<tr><td>" +nomParam+"</td><td>&nbsp;</td></tr>");
    }
%>
</table>
<h1>Entêtes du client</h1>
<table border=1>
<%
    // on recupere la liste des entetes
    Enumeration listeHeaders;
    listeHeaders=request.getHeaderNames();

```

```

// parcours de la liste des entetes
while(listeHeaders.hasMoreElements()){
// nom et valeur de l'entete courant
String nomHeader=(String) listeHeaders.nextElement();
String valHeader=(String) request.getHeader(nomHeader);
// affichage
if(valHeader==null || valHeader.equals("")) valHeader="&nbsp;";
out.println("<tr><td>"+nomHeader+"</td><td>"+valHeader+"</td></tr>");
}
%>
</table>
</body>
</html>

```

## 25.2 Le test

- Sauvegardez le script précédent sous le nom *parametres.jsp* dans `<tomcat>\jakarta-tomcat\webapps\examples\jsp`
- lancez les serveurs Tomcat et Apache
- demandez l'URL `http://localhost/examples/jsp/parametres.jsp`

Si les saisies sont les suivantes :

le résultat renvoyé est le suivant pour les paramètres :

Paramètres du formulaire	
cache(0)	secret
cmdEnvoyer(0)	Envoyer
simple(0)	qqs mots
lstVoitures(0)	Fiat
lstVoitures(1)	Audi
menu(0)	60 F
radio(0)	GO
passwd(0)	wxcvbn
qcm(0)	un
qcm(1)	deux
lignes(0)	ligne1 ligne2

et le suivant pour les entêtes :

Entêtes du client	
cookie	JSESSIONID=To1010mC3643558357907659At; ASPSESSIONIDFFERZBIV=BBMBILOBACLDNMBFJKICBGLF
connection	Keep-Alive
accept-encoding	gzip, deflate
referer	http://localhost/exemples/jsp/perso/parametres.jsp
accept	image/gif, image/x-bitmap, image/jpeg, image/png, application/vnd.ms-excel, application/msword, application/vnd.ms-powerpoint, */*
content-length	172
accept-language	fr
user-agent	Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)
content-type	application/x-www-form-urlencoded
host	localhost

## 26. Technologie PHP

Nous reprenons l'étude du formulaire traité précédemment avec les technologies ASP, PERL et JSP.

### 26.1 Le script PHP

```
<?
// pour avoir le dictionnaire des paramètres du formulaire,
// des variables du serveur,...
php_track_vars
?>

<? // on affiche la page vide ?>
<html>
<head>
<title>Essai page php</title>
</head>
<body bgcolor="#C0C0C0">
<center>
<h2>Formulaire traité par PHP(Personal Home Page)</h2>
<hr>
<form method="post">
<input type="hidden" name="cache" value="secret">
<table border="0">
<tr>
<td align="center">Un champ textuel simple</td>
<td align="center" width="100">&nbsp;</td>
<td align="center">Un champ textuel sur plusieurs lignes</td>
</tr>
<tr>
<td align="center"><input type="text" size="20" name="simple"></td>
<td align="center" width="100">&nbsp;</td>
<td align="center">
<textarea name="lignes" rows="2" cols="40">Ce texte est modifiable
</textarea>
</td>
</tr>
</table>
<table border="0">
<tr>
<td><strong>Des boutons radio :</strong></td>
<td>
<input type="radio" checked name="radio" value="FM">FM
</td>
<td>
<input type="radio" name="radio" value="GO">GO
</td>
<td>

```

```

        <input type="radio" name="radio" value="PO">PO
    </td>
    <td>&nbsp;&nbsp;&nbsp;</td>
    <td><strong>Des choix multiples :</strong></td>
    <td>
        <input type="checkbox" name="qcm[]" value="un">un
    </td>
    <td>
        <input type="checkbox" name="qcm[]" value="deux">deux
    </td>
    <td>
        <input type="checkbox" name="qcm[]" value="trois">trois
    </td>
</tr>
</table>
<table border="0">
    <tr>
        <td>Un menu déroulant : </td>
        <td>
            <select name="menu" size="1">
                <option>50 F</option>
                <option>60 F</option>
                <option>70 F</option>
                <option>100 F</option>
            </select>
        </td>
        <td>Une liste :</td>
        <td>
            <select name="lstVoitures[]" multiple size="3">
                <option>Renault</option>
                <option>Citroën</option>
                <option>Peugeot</option>
                <option>Fiat</option>
                <option>Audi</option>
            </select>
        </td>
    </tr>
</table>
<table border="0">
    <tr>
        <td>Un mot de passe : </td>
        <td><input type="password" size="21" name="passwd"></td>
        <td>&nbsp;&nbsp;&nbsp;</td>
        <td>Un champ de contexte caché : </td>
    </tr>
</table>
<hr>
<table border="0">
    <tr>
        <td align="center">
            <input type="submit" name="cmdEnvoyer" value="Envoyer">
        </td>
        <td align="center">
            <input type="reset" name="cmdAnnuler" value="Annuler">
        </td>
    </tr>
</table>
</form>
<hr>
<h1>Paramètres du formulaire</h1>
<table border=1>
<h3>Environnement global</h3>
<table border=1>
<?
    reset($GLOBALS);next($GLOBALS);
    for (; $item=key($GLOBALS);next($GLOBALS)) {
        print("<tr><td>$item</td><td>".pos($GLOBALS). "</td></tr>");
    }
?>
</table>
<hr>
<h3>Environnement POST</h3>
<table border=1>
<?
    if (is_array($_HTTP_POST_VARS)) {
        reset($_HTTP_POST_VARS);next($_HTTP_POST_VARS);
        for (; $item=key($_HTTP_POST_VARS);next($_HTTP_POST_VARS)) {
            $valeur=pos($_HTTP_POST_VARS);
            if (is_array($valeur))
                for ($i=0;$i<count($valeur);$i++)

```

```

        print("<tr><td>$item"." ($i)</td><td>".$valeur[$i]."</td></tr>\n");
    else print("<tr><td>$item</td><td>$valeur</td></tr>\n");
    }//for
} //if
?>
</table>
</body>
</html>

```

## 26.2 Les tests

### Test 1

- Sauvegardez le script précédent sous le nom *parametres.php4* dans *<apache-DocumentRoot>*
- lancez le serveur Apache
- demandez l'URL *http://localhost/parametres.php4*

### Test 2

- Sauvegardez le script précédent sous le nom *parametres.php* dans *<PWS-DocumentRoot>\php*
- lancez le serveur PWS
- demandez l'URL *http://localhost/parametres.php*

Si les saisies sont les suivantes :

**Formulaire traité par PHP(Personal Home Page)**

---

Un champ textuel simple

Un champ textuel sur plusieurs lignes

Des boutons radio :  FM  GO  PO    Des choix multiples :  un  deux  trois

Un menu déroulant :

Une liste :

Un mot de passe :

Un champ de contexte caché :

les résultats obtenus sont alors les suivants :

## Environnement POST

simple	qqs mots
lignes	ligne1 ligne2
radio	GO
qcm(0)	un
qcm(1)	trois
menu	70 F
lstVoitures(0)	Peugeot
lstVoitures(1)	Audi
passwd	monmotdepasse
cmdEnvoyer	Envoyer

## Paramètres du formulaire

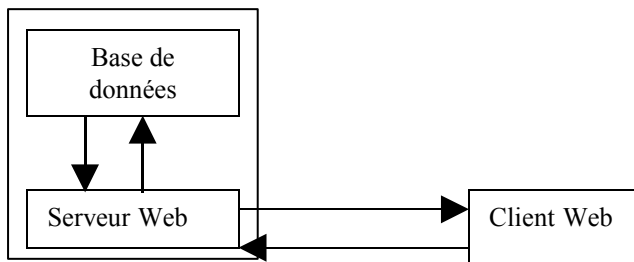
### Environnement global

CONTENT_LENGTH	214
CONTENT_TYPE	application/x-www-form-urlencoded
DOCUMENT_ROOT	d:/data/apache/htdocs
HTTP_ACCEPT	image/gif, image/x-bitmap, image/jpeg, image/pjpeg, application/vnd.ms-excel, application/msword, ap
HTTP_ACCEPT_ENCODING	gzip, deflate
HTTP_ACCEPT_LANGUAGE	fr
HTTP_CONNECTION	Keep-Alive
HTTP_HOST	localhost
HTTP_REFERER	http://localhost/php/parametres.php4
HTTP_USER_AGENT	Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)
	C:\Program Files\Apache Group\Apache;C:\WDK1.3\BIN;C:\ORACLE\ORA81\BIN;C:\PROGRAM

# WEB et Base de données

## 27.L'application

On se propose d'écrire une application Web permettant d'accéder à une base de données selon le schéma suivant :



La base de données sur un PC Windows. Le serveur Web se trouve sur ce même PC et est interrogé par des postes clients distants.

La base aura un identificateur DSN (Data Source Name) sur le PC qui l'héberge. Rappelons qu'un identificateur DSN permet aux clients de la base de se référer à ce seul nom et d'ignorer ainsi le chemin physique de la base. Nous supposons que la base utilisée (ACCESS, MySQL, ORACLE,...) a un pilote ODBC, le lien entre le programme d'interrogation et la base se faisant via ce pilote.

## Requêtes sur une base DSN/Windows

### Requêtes SQL sur une base DSN

Base de données	<input style="width: 80%;" type="text" value="articles"/>
Requête SQL	<input style="width: 80%;" type="text" value="select * from articles"/>



## Résultats de la requête

code	nom	prix	stock_actu	stock_mini
a300	vélo	1203	100	8
d600	arc	5001	100	8
d800	canoé	1503	100	7
x123	fusil	3001	100	8
s345	skis nautiques	1801	100	8
f450	essai3	4	100	9
z400	léopard	500001	100	7
g457	panthère	800001	150	7
x300	sac sport	146	7	5

## Requêtes SQL sur une base DSN

Base de données

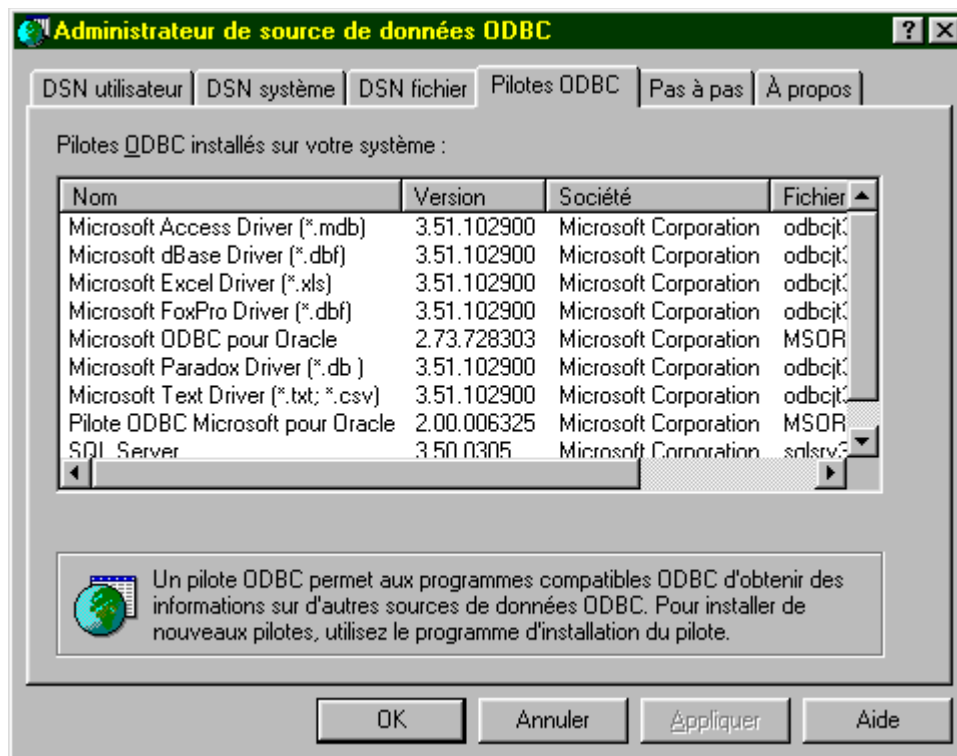
Requête SQL

## Résultats de la requête

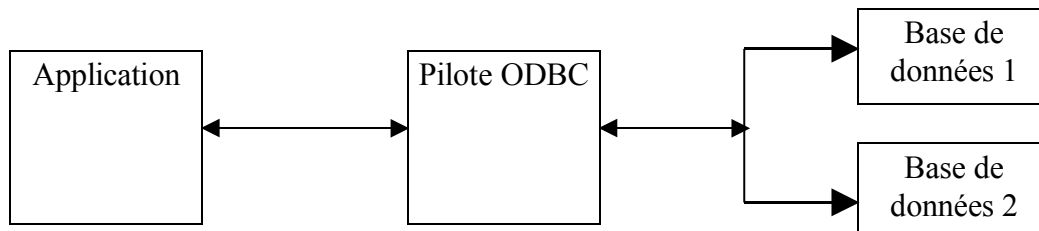
9 lignes ont été affectées par la requête

### 27.1 Pilote ODBC/Nom DSN

Il existe de nombreuses bases de données sur le marché. Afin d'uniformiser les accès aux bases de données sous MS Windows, Microsoft a développé une interface appelée ODBC (Open DataBase Connectivity). Cette couche cache les particularités de chaque base de données sous une interface standard. Il existe sous MS Windows de nombreux pilotes ODBC facilitant l'accès aux bases de données. Voici par exemple, une liste de pilotes ODBC installés sur une machine Win9x (*Menu Démarrer/Panneau de configuration/Source de données ODBC 32 bits*).



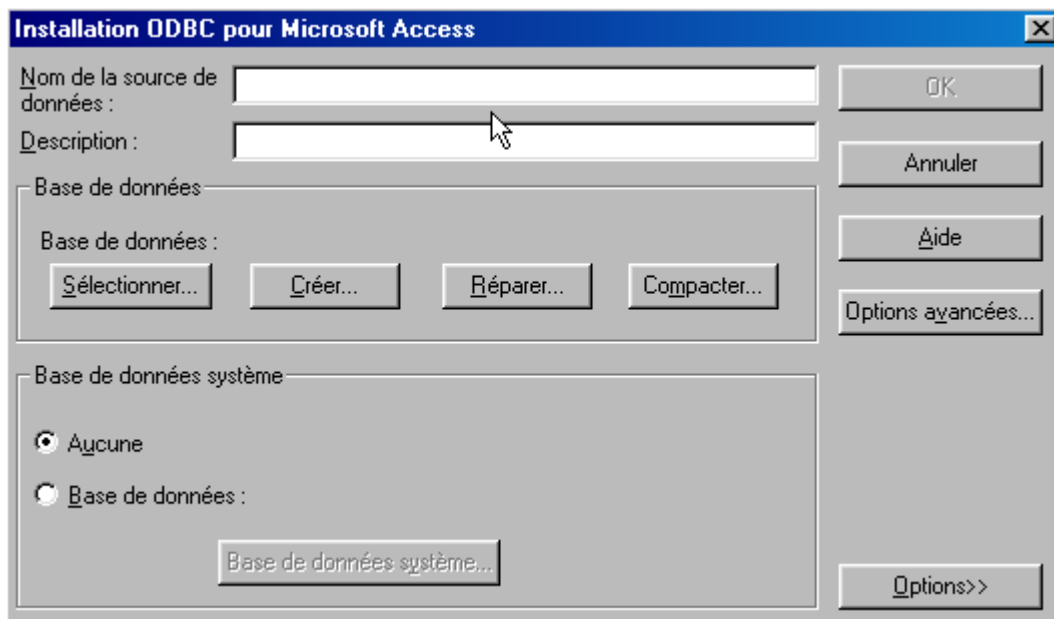
Une application s'appuyant sur un pilote ODBC peut utiliser n'importe quelle base de données ayant un tel pilote sans ré-écriture.



Pour faire d'une base de données, une base publique connue sous un certain nom, procéder de la façon suivante :

- *Menu Démarrer/Panneau de configuration/Source de données ODBC 32 bit/DSN utilisateur*
- *Ajouter*
- choisir le type de la base/*Terminer*
- ensuite, cela peut varier selon le type de la base de données. Néanmoins deux informations seront à donner :
  1. le choix de la base
  2. le nom DSN qu'on veut lui donner

Voici un exemple pour ACCESS :



Le nom DSN est à indiquer dans le champ "Nom de la source de données". Le choix de la base ACCESS à associer à ce nom se fait avec le bouton *Sélectionner*. Ceci fait, cette base ACCESS pourra être référencée via son nom DSN par les programmes d'interrogation de la base.

Nous traitons maintenant le problème posé avec les différentes technologies ASP, PERL, JSP et PHP.

## 27.2 Technologie ASP

### 27.2.1 Le script

```
<%
' formulaire d'accès à une base de données

' variables globales
dim couleurFond,couleurErreur,couleurResultats,couleurChamps,couleurCelluleChamp
dim client      ' identité client HTTP
dim vbDbName    ' nom DSN de la base de données à exploiter
dim vbsqlCmd    ' texte de la commande SQL

' on récupère qqs données
client=request.serverVariables("remote_host")
vbDbName=request.form("dbName")
vbsqlCmd=request.form("sqlCmd")
%>

<html>
<head>
<title>Gestion d'une base de données via le Web</title>

<!-- les scripts côté serveur -->
<script language="vbscript" runat="server">

    function initForm
    ' initialise des variables globales
    couleurFond="#C0C0C0"
    couleurErreur="#FF0000"
    couleurResultats="#0000FF"
    couleurChamps="#FF0080"
    couleurCelluleChamp="#FF8000"
    end function

    ' -----
    function erreur (codeErreur,msgErreur)
    ' fonction de gestion des erreurs
    ' codeErreur : entier code de l'erreur
```

```

' msgErreur : message d'erreur du système

' si codeErreur=0, il n'y a pas eu d'erreur
if codeErreur=0 then
    erreur=false
    exit function
end if

' il y a eu une erreur - on le dit
with response
    .write("<h3>L'erreur suivante s'est produite</h3>")
    .write("<h4><table border=1><tr><td cellpadding=5><font color='" & couleurErreur &
"><strong> " & msgErreur & "</strong></font></td></tr></table></h4>")
    .write("</body></html>")
end with
erreur=true
end function ' erreur

' -----
function executeSQL(dbName,sqlCmd)

' gestion des erreurs
on error resume next

' création d'un objet base de données
set db=server.createObject("ADODB.Connection")
if erreur(err.number,err.description) then
    db.close
    set db=nothing
    on error goto 0
    exit function
end if

' association de l'objet db à la base physique
db.open(dbName)
if erreur(err.number,err.description) then
    db.close
    set db=nothing
    on error goto 0
    exit function
end if

' exécution de la requête SQL
set resultats=db.execute(sqlCmd,nbLignes)
if erreur(err.number,err.description) then
    db.close
    set db=nothing
    resultats.close
    set resultats=nothing
    on error goto 0
    exit function
end if

' on rend la gestion des erreurs à vb
on error goto 0

' traitement du résultat de la requête
with response
    .write("<h2>Résultats de la requête<br> (émetteur : " & client & ")</h2>")

' select ou requête de mise à jour ?
nbColonnes=0
nbColonnes=resultats.fields.count
if(err.number<>0 or nbColonnes=0) then
    ' c'était une cmde SQL de mise à jour
    .write("<h3>" & nbLignes & " ligne(s) modifiée(s)</h3>")
    on error goto 0
    exit function
end if
on error goto 0

' c'était une commande SQL de recherche - on a récupéré une table
.write("<table border=1 cellpadding=10>")
' affichage des libellés des champs de la table des résultats
.write("<tr>")
for each champ in resultats.fields
    .write("<td align='center' bgcolor='" & couleurCelluleChamp & "'><font size=4
color='" & couleurChamps & "'><strong> " & champ.name & "</strong></font></td>")

```

```

next
.write("</tr>")
' parcours séquentiel de la table des résultats
do while not resultats.eof
' parcours des champs de la ligne courante de la table des résultats
.write("<tr>")
for each champ in resultats.fields
.write("<td<font color=' " & couleurResultats & "'><strong> " &
resultats.fields(champ.name) & "</strong></font></td>")
next
.write("</tr>")
' passage à la ligne suivante
resultats.moveNext
loop

.write("</table>")
' fermeture de la table des résultats
resultats.close
set resultats=nothing

' fermeture de la base
db.close
set db=nothing
end with
end function
</script>

```

```
<!-- scripts côté navigateur -->
```

```

<script language="javascript">
function verifier(){
// on vérifie que les deux champs ont été remplis
with (document.formSQL){
// nom de la base DSN
champ=/^\s*$/.exec(dbName.value);
if(champ!=null){
alert("Vous n'avez pas indiqué la base DSN");
dbName.focus();
return;
}
// requête SQL
champ=/^\s*$/.exec(sqlCmd.value);
if(champ!=null){
alert("Vous n'avez pas indiqué la requête SQL");
sqlCmd.focus();
return;
}
// on envoie le formulaire
submit();
} // with
} // function
</script>
</head>

```

```
<!-- fin de head, début de body -->
```

```
<!-- on initialise la page -->
```

```

<% ' init page
initForm
%>

```

```
<!-- on affiche la page -->
```

```
<body bgcolor="<% =couleurFond %>">
```

```

<center>
<h1>Requêtes SQL sur une base de données DSN</h1>
<hr>

```

```
<form name="formSQL" method="post">
```

```
<table>
```

```
<tr>
```

```
<td>Base de données DSN</td>
```

```
<td><input type="text" size=60 name="dbName" value="<%=vbdbName%>"></td>
```

```
</tr>
```

```
<tr>
```

```
<td>Requête SQL</td>
```

```
<td><input type="text" name="sqlCmd" value="<%=vbsqlCmd%>" size=60></td>
```

```
</tr>
```

```
</table>
```

```
<input type="button" value="Exécuter" name="cmdExecuter" onclick="verifier()">
```

```
</form>
```

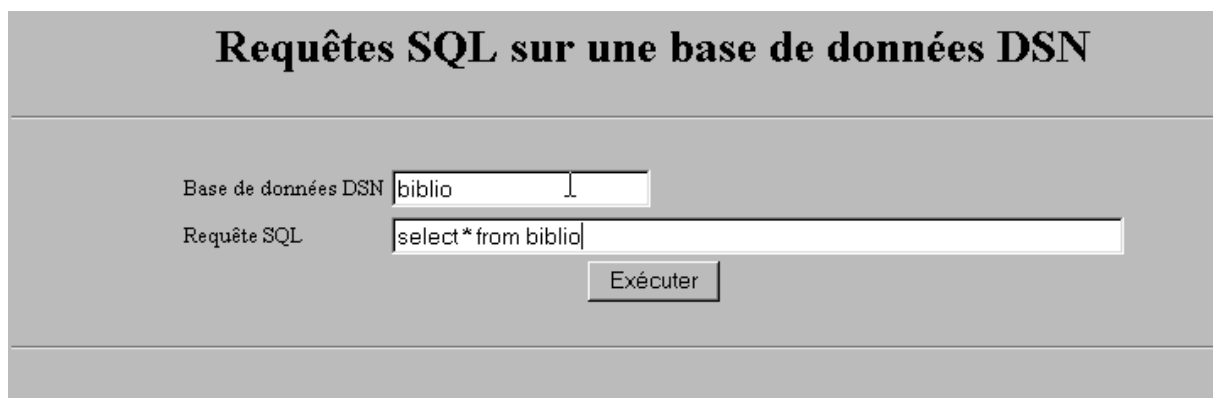
```
<hr>
```

```
<% ' y-a-t-il des arguments à traiter ?
  nbParametres=request.form.count
  if nbParametres<>0 then
    executeSQL request.form("dbName"),request.form("sqlCmd")
  end if
%>
</body>
</html>
```

## 27.2.2 Les tests

- mettre le script précédent dans `<PWS-DocumentRoot>\sql1.asp`
- lancer le serveur PWS
- assurez-vous d'avoir les noms DSN que vous voulez utiliser
- demander l'URL `http://localhost/sql1.asp`

Le formulaire de base, le nom DSN **biblio** a été associé à une base de données MySQL

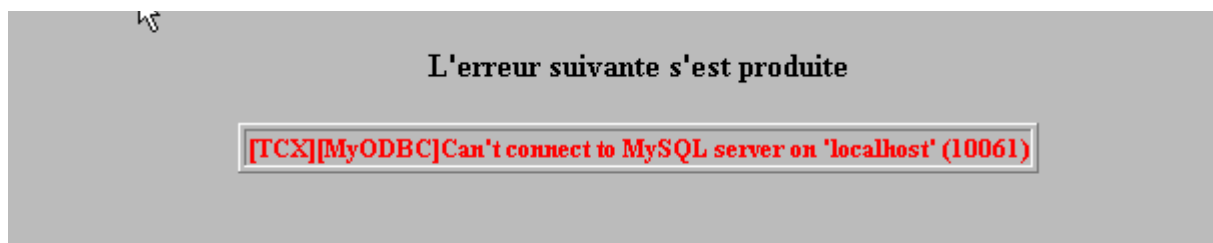


**Requêtes SQL sur une base de données DSN**

Base de données DSN

Requête SQL

Ici le serveur MySQL n'était pas lancé :



**L'erreur suivante s'est produite**

**[TCX][MyODBC]Can't connect to MySQL server on 'localhost' (10061)**

Maintenant on lance le serveur MySQL et on obtient le résultat suivant :

## Résultats de la requête (émetteur : 127.0.0.1)

titre	auteur	genre	prix	achat	disponible
introduction à python	lutz	développement	591,6	15/06/1900	o
programmmin perl	wall	développement	359,04	21/08/1900	o
prog. Internet en c:c++	jansa	développement	53	24/05/1900	n
trouver l'info sur internet	andrieu	web	273,36	31/07/1900	o
asp 2.0	francis	web	273,36	25/03/1901	o
interface web/Bd	homer	web	318,24	01/09/1900	n
javascript	McFarlane	web	359,04	06/09/1900	o
visual j++	winters	java	66	30/08/1900	o

Nouvelle requête :

## Requêtes SQL sur une base de données DSN

Base de données DSN

Requête SQL

Exécuter

Nouveaux résultats :

## Résultats de la requête (émetteur : 127.0.0.1)

titre	auteur	genre	prix	achat	disponible
introduction à python	lutz	développement	591,6	15/06/1900	o
programmmin perl	wall	développement	359,04	21/08/1900	o
trouver l'info sur internet	andrieu	web	273,36	31/07/1900	o
asp 2.0	francis	web	273,36	25/03/1901	o
javascript	McFarlane	web	359,04	06/09/1900	o
visual j++	winters	java	66	30/08/1900	o
java servlets	hunter	java	65	13/05/1900	o
nous n'irons plus au bois	higgins clark	policier	82	23/02/1900	o

Nouvelle requête :

### Requêtes SQL sur une base de données DSN

Base de données DSN

Requête SQL

---

**Résultats de la requête**  
(émetteur : 127.0.0.1)

1 ligne(s) modifiée(s)

Une requête sur une base Oracle :

### Requêtes SQL sur une base de données DSN

Base de données DSN

Requête SQL

---

**Résultats de la requête**  
(émetteur : 127.0.0.1)

NOCMD	ISBN	QTE
1	0-07-881551-7	2
1	0-07-881524-X	1
2	0-07-881537-1	4
3	0-07-881520-7	1

## 27.3 Technologie JSP

### 27.3.1 Le script

```
<!-- Gestion d'une base de données avec une page JSP -->
```

```
<%@ page import="java.sql.*" %>  
<%@ page import="java.util.*" %>
```



```

<html>
<head>
  <title>Gestion d'une base de donnees via le Web</title>

  <!-- scripts cote navigateur -->

  <script language="javascript">
    function verifier(){
      // on verifie que les deux champs ont ete remplis
      with (document.formSQL){
        // nom de la base DSN
        champ=/^\s*$/ .exec(dbName.value);
        if(champ!=null){
          alert("Vous n'avez pas indique la base DSN");
          dbName.focus();
          return;
        }
        // requete SQL
        champ=/^\s*$/ .exec(sqlCmd.value);
        if(champ!=null){
          alert("Vous n'avez pas indique la requete SQL");
          sqlCmd.focus();
          return;
        }
        // on envoie le formulaire
        submit();
      } // with
    } // function
  </script>
</head>

<!-- fin de head, debut de body -->

<!-- on initialise la page -->
<%
  initPage (request) ;
%>

<!-- on affiche la page -->
<body bgcolor='<%=couleurFond%>'>
  <center>
    <h1>Requêtes SQL sur une base de données DSN</h1>
    <hr>
    <form name="formSQL" method="post">
      <table>
        <tr>
          <td>Base de données DSN</td>
          <td><input type="text" size=60 name="dbName" value="<%=dbName%>"></td>
        </tr>
        <tr>
          <td>Requête SQL</td>
          <td><input type="text" name="sqlCmd" value="<%=sqlCmd%>" size=60</td>
        </tr>
      </table>
      <input type="button" value="Executer" name="cmdExecuter" onclick="verifier()">
    </form>
    <hr>
    <!-- Traitement de la requete -->
    <% //on traite la requete si besoin est
      if (!dbName.equals("")) executeSQL(out,dbName,sqlCmd) ;
    %>
  </body>
</html>

<!-- Methodes de la page JSP -->
<%!

  // variables globales
  String couleurFond,couleurErreur,couleurResultats,couleurChamps,couleurCelluleChamp;
  String client; // identite client HTTP
  String dbName; // nom DSN de la base de donnees a exploiter
  String sqlCmd; // texte de la commande SQL

  // init Formulaire
  private void initPage(HttpServletRequest request){

    // on recupere qqs donnees
    client=request.getHeader("remote_host");
  }

```

```

        dbName=request.getParameter("dbName");
        if(dbName==null) dbName="";
        sqlCmd=request.getParameter("sqlCmd");
        if(sqlCmd==null) sqlCmd="";

        // on fixe qqs couleurs
        couleurFond="#C0C0C0";
        couleurErreur="#FF0000";
        couleurResultats="#0000FF";
        couleurChamps="#FF0080";
        couleurCelluleChamp="#FF8000";
    }

    // execution de la requete SQL
    private void executeSQL(JspWriter out, String dbName, String sqlCmd)
        throws IOException{
        // Affichage titre
        out.print("<h1>Résultats de la requête</h1>");
        Connection connect=null;
        Statement S=null;
        ResultSet RS=null;
        try{
            // connexion a la base
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            // on récupère les composants dsn,user,uid dans dbName
            StringTokenizer st=new StringTokenizer(dbName,"");
            int nbComposants=st.countTokens();
            String dsn=st.nextToken();
            String user="";
            String passwd="";
            if (nbComposants>=1) user=st.nextToken();
            if (nbComposants>=2) passwd=st.nextToken();
            connect=DriverManager.getConnection("jdbc:odbc:"+dsn,user,passwd);
            // creation d'un objet Statement
            S=connect.createStatement();
            // execution de la requete SQL
            if(S.execute(sqlCmd)){
                // on a obtenu un ResultSet - on l'exploite
                RS=S.getResultSet();
                // nombre de colonnes
                int nbColonnes=RS.getMetaData().getColumnCount();
                // exploitation de la table des resultats
                out.print("<table border=1 cellspacing=10>");
                // affichage des titres de colonnes
                ResultSetMetaData RSMD=RS.getMetaData();
                out.print("<tr>");
                for(int i=1;i<=nbColonnes;i++)
                    out.print("<td align='center' bgcolor='\" + couleurCelluleChamp + \"'><font size=4
color='\" + couleurChamps + \"'><strong>"+RSMD.getColumnLabel(i)+"</strong></font></td>");
                out.print("</tr>");
                // exploitation des lignes de la table
                while(RS.next()){ // tant qu'il y a une ligne a exploiter
                    // on l'affiche a l'ecran
                    out.print("<tr>");
                    for(int i=1;i<=nbColonnes;i++)
                        out.print("<td><font color='\" + couleurResultats + \"'><strong>\" + RS.getString(i)+
\"</strong></font></td>");
                    out.print("</tr>");
                } // while
                out.print("</table>");
            } else {
                // c'etait une requete de mise a jour
                int nbLignes=S.getUpdateCount();
                // suivi
                out.print("<h4>"+nbLignes + " ligne(s) ont été mises à jour</h4>");
            } //if
        } catch (Exception e){
            out.print("<h3>L'erreur suivante s'est produite</h3><table
border=1><tr><td><h4>"+e+"</h4></td></tr></table>");
        }
        // fermeture de la base
        try{
            // on libere les ressources liees a la base
            RS.close();
            S.close();
            connect.close();
        } catch (Exception e){}
    } // executeSQL
} //

```

### 27.3.2 Les tests

- mettre le script précédent dans `<tomcat>\jakarta-tomcat\webapps\examples\jsp\sql1.jsp`
- arrêtez le serveur PWS si besoin est
- lancer le serveur *Tomcat*, puis le serveur *Apache*
- assurez-vous d'avoir les noms DSN que vous voulez utiliser
- demander l'URL `http://localhost/examples/jsp/sql1.jsp`

On obtient les mêmes résultats qu'avec le script ASP. La gestion des erreurs est légèrement différente. Voici l'erreur affichée lorsqu'on utilise un DSN inexistant :

**Requêtes SQL sur une base de données DSN**

Base de données DSN

Requête SQL

**Résultats de la requête**

L'erreur suivante s'est produite

```
java.sql.SQLException: [Microsoft][Gestionnaire de pilotes ODBC] Source de données non trouvée et nom de pilote non spécifié
```

La façon de préciser la séquence DSN, USER, PASSWORD également. Voici le cas de la base Oracle déjà utilisée précédemment :

**Requêtes SQL sur une base de données DSN**

Base de données DSN

Requête SQL

**Résultats de la requête**

NOCMD	ISBN	QTE
1	0-07-881551-7	2

## 27.4 Technologie PHP

### 27.4.1 Le script

```
<?php_track_vars?>

<!-- données PHP globales -->

<?php
    $couleurFond="#C0C0C0";
    $sqlCmd2=stripslashes($sqlCmd);
    $dbName2=stripslashes($dbName);

    // fonctions
    fonction executeSQL($dbName,$sqlCmd){

        // exécution de la requête $sqlCmd sur la base $dbName

        // on récupère les composantes DSN, USER, PASSWORD dans $dbName
        $champs=explode(",",$dbName);
        $dsn=$champs[0];
        $user=$champs[1];
        $passwd=$champs[2];

        // on se connecte à la base DSN
        if (!$odbc_link=odbc_connect($dsn,$user,$passwd)){
            return "$php_errormsg";
        }

        // interroger la base
        if (!$odbc_result=odbc_do($odbc_link,$sqlCmd))
            return "$php_errormsg";

        // a-t-on récupéré une table ?
        if (! odbc_field_name($odbc_result,1)){
            // on n'a pas une table
            $nbLignes=odbc_num_rows($odbc_result);
            print("$nbLignes lignes ont été affectées par la requête");
            return "";
        }

        // récupérer les champs de la table des résultats
        print("<table border=1><tr>\n");
        $nbChamps=odbc_num_fields($odbc_result);
        for($i=1;$i<=$nbChamps;$i++){
            print("<td>".odbc_field_name($odbc_result,$i)."</td>\n");
        }
        print("</tr>");

        // exploitation de la table
        while(odbc_fetch_into($odbc_result,&$row)){
            print("<tr>\n");
            for($i=0;$i<$nbChamps;$i++){
                print("<td>".$row[$i]."</td>");
            }
            print("</tr>\n");
        }

        // fin de la table
        print("</table>");
        // on ferme la connexion à la base
        odbc_close($odbc_link);
        // retour sans erreur
        return "";
    }
}

<?php

<html>
<head>
<title>Exécuter une commande SQL sur une base DSN</title>

<!-- scripts côté navigateur -->

<script language="javascript">
    fonction verifier(){
        // on vérifie que les deux champs ont été remplis
        with (document.formsSQL){
            // nom de la base DSN
            champ=/^\\s*$/ .exec(dbName.value);
```

```

        if(champ!=null){
            alert("Vous n'avez pas indiqué la base DSN");
            dbName.focus();
            return;
        }
        // requête SQL
        champ=/^\s*$/ .exec(sqlCmd.value);
        if(champ!=null){
            alert("Vous n'avez pas indiqué la requête SQL");
            sqlCmd.focus();
            return;
        }
        // on envoie le formulaire
        submit();
    } // with
} // fonction

function effacer(){
    with (document.formSQL){
        dbName.value="";
        sqlCmd.value="";
    } //with
} //effacer
</script>
</head>

    <!-- fin de head, début de body -->

</head>
<body bgcolor="<?php echo $couleurFond; ?>"><center>
    <h1>Requêtes SQL sur une base DSN</h1>
    <hr>
    <form name="formSQL" method="post">
        <table>
            <tr>
                <td>Base de données</td>
                <td><input type="text" size=30 name="dbName" value="<? echo $dbName2 ?>"></td>
            </tr>
            <tr>
                <td>Requête SQL</td>
                <td><input type="text" size=60 name="sqlCmd" value="<? echo $sqlCmd2 ?>"></td>
            </tr>
        </table>
        <br><br>
        <table>
            <tr>
                <td><input type="button" name="cmdExecuter" value="Exécuter"
                onclick="verifier()"></td>
                <td><input type="button" name="cmdEffacer" value="Effacer" onclick="effacer()"></td>
            </tr>
        </table>
    </form>
    <hr>
    <! -- traitement du formulaire -->
    <?
        // exploration de l'environnement
        if(is_array($_POST) && $dbName!=" && $sqlCmd!="){

            // on a récupéré des données
            print("<h1>Résultats de la requête</h1>");

            // on tente d'exécuter la requête SQL
            if ($erreur=executeSQL($dbName,$sqlCmd2)){
                print("<h3>L'erreur suivante s'est produite</h3>\n");
                print("<table border=1><tr><td>".$erreur."</td></tr></table>\n");
            }
        } //if
    ?>
</body>
</html>

```

## 27.4.2 Les tests

### Test 1

- mettre le script précédent dans `<Apache-DocumentRoot>\sql.php4`
- lancer le serveur Apache
- assurez-vous d'avoir les noms DSN que vous voulez utiliser
- demander l'URL `http://localhost/sql.php4`

## Test 2

- mettre le script précédent dans `<PWS-DocumentRoot>\php\sql.php`
- lancer le serveur PWS
- assurez-vous d'avoir les noms DSN que vous voulez utiliser
- demander l'URL `http://localhost/php/sql.php`

On obtient les mêmes résultats que précédemment. On remarque simplement qu'en cas d'erreur le script n'arrive pas à récupérer un message d'erreur significatif.

## Requêtes SQL sur une base DSN

Base de données

Requête SQL

---

## Résultats de la requête

**L'erreur suivante s'est produite**

SQL error, SQL state 00000 in SQLConnect

On a utilisé un nom DSN inexistant et le message d'erreur obtenu n'est pas compréhensible. Ici on avait positionné dans `<windows>\php.ini` les directives suivantes :

*display\_errors = Off ; Print out errors (as a part of the HTML script)*  
*track\_errors = On ; Store the last error/warning message in \$php\_errormsg (boolean)*

Si maintenant on configure `php` de la façon suivante :

*display\_errors = On ; Print out errors (as a part of the HTML script)*  
*track\_errors = On ; Store the last error/warning message in \$php\_errormsg (boolean)*

on obtient le résultat suivant :

## Résultats de la requête

**Warning:** SQL error: [Microsoft][Gestionnaire de pilotes ODBC] Source de données non trouvée et nom de pilote non spécifié, SQL state IM002 in SQLConnect in d:\data\apache\htdocs\php\sql2.php4 on line 22

**Warning:** SQL error: , SQL state 00000 in SQLConnect in d:\data\apache\htdocs\php\sql2.php4 on line 22

L'erreur suivante s'est produite

SQL error: , SQL state 00000 in SQLConnect

Ici, on voit clairement l'erreur qui s'est produite mais d'une façon pas acceptable dans une page renvoyée à un utilisateur. On voit par ailleurs, que le message d'erreur récupéré (*\$php\_errormsg*) correspond au second warning. Il aurait fallu avoir accès au premier.

## 27.5 Technologie PERL

### 27.5.1 Le script

```
#!/C:\Perl\bin\perl.exe"

# modules
use CGI qw/:all/;
use strict;
use DBI;
use English;

# données globales
my $couleurFond="#C0C0C0";

# paramètres
my $dbName=param("dbName");
my $sqlCmd=param("sqlCmd");

# affichage page principale
print <<FINHTML1
Content-type: text/html

<html>
  <head>
    <title>Exécuter une commande SQL sur une base DSN</title>

    <!-- scripts côté navigateur -->

    <script language="javascript">
      function verifier(){
        // on vérifie que les deux champs ont été remplis
        with (document.formSQL){
          // nom de la base DSN
          champ=/^\s*\$/ .exec(dbName.value);
          if(champ!=null){
            alert("Vous n'avez pas indiqué la base DSN");
            dbName.focus();
            return;
          }
          // requête SQL
          champ=/^\s*\$/ .exec(sqlCmd.value);
          if(champ!=null){
            alert("Vous n'avez pas indiqué la requête SQL");
            sqlCmd.focus();
            return;
          }
          // on envoie le formulaire
          submit();
        } // with
      } // function

      function effacer(){
```

```

        with (document.formsSQL){
            dbName.value="";
            sqlCmd.value="";
        }//with
    }//effacer
</script>
</head>

    <!-- fin de head, début de body -->

<body bgcolor="$couleurFond">
    <center>
    <h1>Requêtes SQL sur une base DSN</h1>
    <hr>
    <form name="formSQL" method="post">
        <table>
            <tr>
                <td>Base de données</td>
                <td><input type="text" size=60 name="dbName" value="$dbName"></td>
            </tr>
            <tr>
                <td>Requête SQL</td>
                <td><input type="text" size=60 name="sqlCmd" value="$sqlCmd"></td>
            </tr>
        </table>
        <br><br>
        <table>
            <tr>
                <td><input type="button" name="cmdExecuter" value="Exécuter"
                onclick="verifier()"></td>
                <td><input type="button" name="cmdEffacer" value="Effacer"
                onclick="effacer()"></td>
            </tr>
        </table>
    </form>
    <hr>
FINHTML1
;

# traitement du formulaire
if (param()){
    # on a récupéré des données
    print("<h1>Résultats de la requête</h1>");
    # on tente d'exécuter la requête SQL
    my $erreur;
    if ($erreur=executesSQL($dbName,$sqlCmd)){
        print("<h3>L'erreur suivante s'est produite</h3>\n");
        print("<table border=1><tr><td>.$erreur.</td></tr></table>\n");
    }# if erreur
}# if param

# fin du formulaire
print "</body></html>\n";
exit 0;

# procedure executesSQL
sub executesSQL{
    # on récupère les paramètres
    my ($dbName,$sqlCmd)=@_;

    # exécution de la requête $sqlCmd sur la base $dbName

    # on récupère les composantes DSN, USER, PASSWORD dans $dbName
    my ($dsn,$user,$passwd)=split(",",$dbName);

    # on se connecte à la base DSN
    my $dbh;
    return "Connexion à la base $dsn impossible"
        if ! ($dbh=DBI->connect("DBI:ODBC:$dsn",$user,$passwd));

    # préparer la requête
    my $statement;
    return $dbh->errstr if !($statement=$dbh->prepare($sqlCmd));

    # exécuter la requête
    my $resultats;
    return $dbh->errstr if !($resultats=$statement->execute());
}

```



```

# a-t-on récupéré une table ?
my $colonnes=$statement->{NAME_lc};
my $nbColonnes=$#{ $colonnes }+1;
if ($nbColonnes==0) {
    # on n'a pas une table
    print "$resultats lignes ont été affectées par la requête\n";
    return "";
}# if

# afficher les noms de colonnes
print "<table border=1><tr>\n";
my $colonne;
foreach $colonne (@{ $colonnes }) {
    print "<td>$colonne</td>\n";
}# foreach
print "</tr>";

# afficher la table

my @ligne;
print "<tr>";
while(@ligne=$statement->fetchrow_array) {
    for(my $i=0;$i<$nbColonnes;$i++){
        print("<td>$ligne[$i]</td>\n");
    }#for
    print "</tr>";
}#while

# fin de la table HTML
print("</table>");

# on libère les ressources
$statement->finish;
$dbh->disconnect;

# retour sans erreur
return "";
}#procedure executeSQL

```

## 27.5.2 Les tests

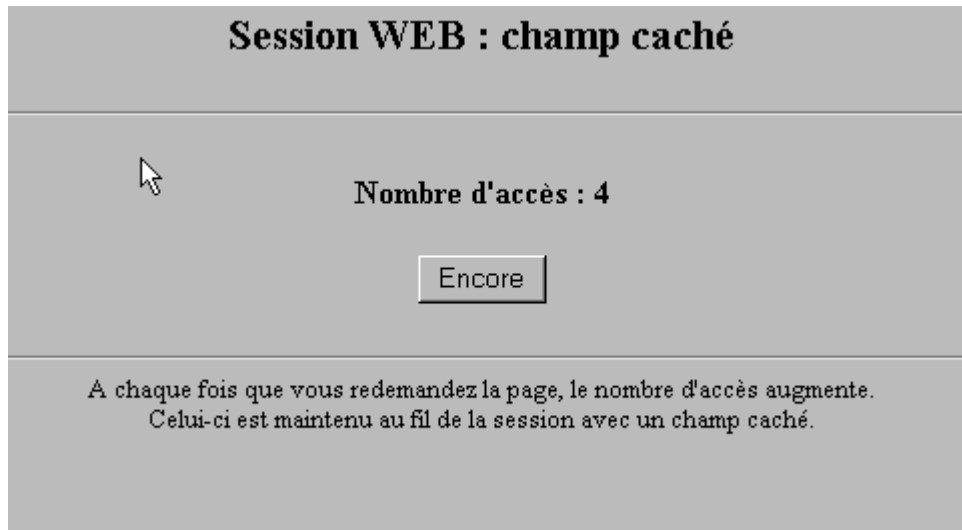
- mettre le script précédent dans `<Apache-DocumentRoot>\cgi-bin/sql1.pl`
- lancer le serveur Apache
- demander l'URL `http://localhost/cgi-bin/sql1.pl`
- assurez-vous d'avoir les noms DSN que vous voulez utiliser

Ils donnent les mêmes résultats qu'avec les technologies précédentes. Au contraire de PHP, le message d'erreur récupéré est ici significatif. Le script PERL a été obtenu par un copier/coller du script PHP tant les langages sont proches l'un de l'autre. Le principal travail a été de transposer à PERL les commandes PHP de gestion de base de données.

## 28. Les sessions WEB

### 28.1 Technologie Perl

#### 28.1.1 Les champs cachés



```
#!c:\perl\bin\perl.exe

# modules
use strict;
use CGI qw/:all/;

# on récupère le champ caché
my $hdAcces=param("hdAcces");
# on l'incrémente
$hdAcces++;

print <<HTML1
Content-type: text/html

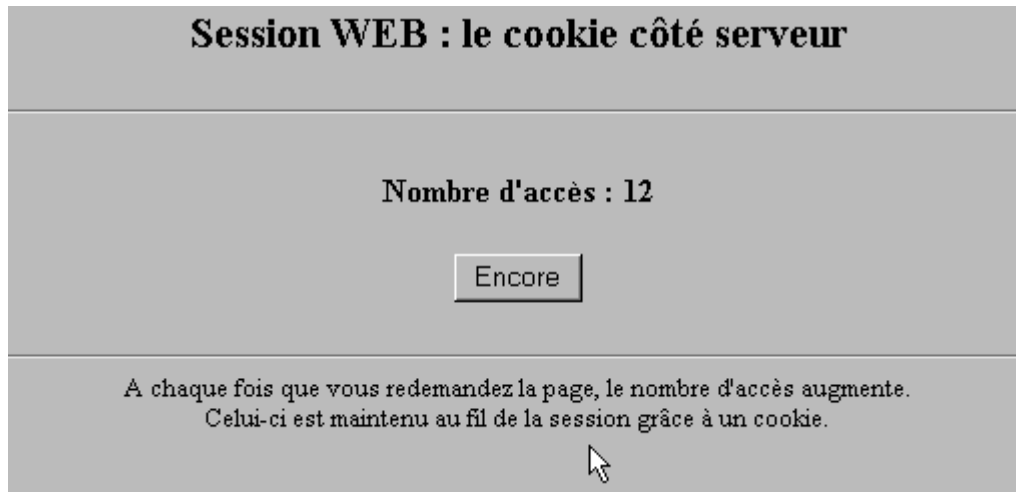
<html>

  <head>
    <title>Les sessions : champ caché</title>
    <script language="javascript">
    </script>
  </head>

  <body bgcolor="#C0C0C0">
    <center>
      <h2>Session WEB : champ caché</h2>
      <hr>
      <center>
        <form name="frmSessions" method="POST">
          <input type="hidden" name="hdAcces" value="$hdAcces">
          <table>
            <tr>
              <td><h3>Nombre d\'accès : </h3></td>
              <td><h3>$hdAcces</h3></td>
            </tr>
          </table>
          <input type="submit" value="Encore">
        </form>
      <hr>
      A chaque fois que vous redemandez la page, le nombre d'accès augmente.<br>
      Celui-ci est maintenu au fil de la session avec un champ caché.<br>
    </body>
  </html>
```

```
HTML1
;
```

## 28.1.2 Le cookie initialisé côté serveur



```
#!c:\perl\bin\perl.exe

# modules
use CGI qw/:all/;
use strict;
use English;

# on récupère le cookie
my $query=new CGI;
my $hdAcces=$query->cookie(-name=>"hdAcces");
# on incrémente le nbre d'accès
$hdAcces++;
# on renvoie le cookie
my $cookie=$query->cookie(-name=>"hdAcces",-value=>$hdAcces);
print $query->header(-cookie=>$cookie);

# on écrit la page
print <<HTML1
<html>

  <head>
    <title>Les sessions : cookie</title>
  </head>

  <body bgcolor="#C0C0C0">
    <center>
      <h2>Session WEB : le cookie côté serveur</h2>
      <hr>
      <center>
        <form name="frmSessions" method="POST">
          <input type="hidden" name="hdAcces" value="$hdAcces">
          <table>
            <tr>
              <td><h3>Nombre d'accès : </h3></td>
              <td><h3>$hdAcces</h3></td>
            </tr>
          </table>
          <input type="submit" value="Encore">
        </form>
      <hr>
      A chaque fois que vous redemandez la page, le nombre d'accès augmente.<br>
      Celui-ci est maintenu au fil de la session grâce à un cookie.<br>
    </body>
  </html>
HTML1
;
```

### 28.1.3 Le cookie initialisé côté navigateur



```
#!/c:\perl\bin\perl.exe

# modules
use CGI qw/:all/;
use strict;
use English;

# initialisations
my @liste=("élément0", "élément1", "élément2", "élément3");

# on récupère le cookie
my $query=new CGI;
my $index=$query->cookie(-name=>"index");
$index+=0;

# on écrit la page
print <<HTML1
Content-type: text/html

<html>

  <head>
    <title>Les sessions : cookie</title>
    <script language="javascript">
      function envoyer(){
        with(document.frmSessions){
          document.cookie="index="+lstElements.selectedIndex;
          submit();
        }//with
      }//envoyer()
    </script>
  </head>

  <body bgcolor="#C0C0C0">
    <center>
      <h2>Session WEB : le cookie côté navigateur</h2>
      <hr>
      <center>
        <form name="frmSessions" method="POST">
          <select name="lstElements" size="1">
HTML1
;
      for(my $i=0;$i<=#liste;$i++){
        my $selected=($i==$index) ? "selected" : "";
        print "<option $selected> $liste[$i] </option>";
      } #for

print <<HTML2
      <input type="button" value="Envoyer" onclick="envoyer()">
    </form>
    <hr>
    L'élément $index ($liste[$index]) a été sélectionné
  </body>
</html>
HTML2
;
```

## 28.2 Technologie ASP

On reprend les trois exemples développés en Perl.

### 28.2.1 Les champs cachés

```
<%
' on récupère le champ caché
hdAcces=request.form("hdAcces")
' on l'incrémente
hdAcces=hdAcces+1
' on écrit la page
%>

<html>
<head>
<title>Les sessions : champ caché</title>
</head>
<body bgcolor="#C0C0C0">
<center>
<h2>Session WEB : champ caché</h2>
<hr>
<form name="frmSessions" method="POST">
<input type="hidden" name="hdAcces" value="<% =hdAcces %>">
<table>
<tr>
<td><h3>Nombre d'accès : </h3></td>
<td><h3><% =hdAcces %></h3></td>
</tr>
</table>
<input type="submit" value="Encore">
</form>
<hr>
A chaque fois que vous redemandez la page, le nombre d'accès augmente.<br>
Celui-ci est maintenu au fil de la session avec un champ caché.<br>
</body>
</html>
```

### 28.2.2 Le cookie initialisé côté serveur

```
<%
' on récupère le cookie
hdAcces=request.cookies("hdAcces")
' on incrémente le nbre d'accès
if hdAcces="" then
hdAcces=0
else
hdAcces=hdAcces+1
end if
' on renvoie le cookie
response.cookies("hdAcces")=hdAcces
' on affiche la page
%>

<html>
<head>
<title>Les sessions : le cookie</title>
</head>
<body bgcolor="#C0C0C0">
<center>
<h2>Session WEB : le cookie</h2>
<hr>
<form name="frmSessions" method="POST">
<table>
<tr>
<td><h3>Nombre d'accès : </h3></td>
<td><h3><% =hdAcces %></h3></td>
</tr>
</table>
<input type="submit" value="Encore">
</form>
<hr>
A chaque fois que vous redemandez la page, le nombre d'accès augmente.<br>
```

```

    Celui-ci est maintenu au fil de la session avec un cookie.<br>
  </body>
</html>

```

### 28.2.3 Le cookie initialisé côté navigateur

```

<%
' initialisations
liste=array("élément0", "élément1", "élément2", "élément3")

' on récupère le cookie - il faut forcer index à être un nombre
' on lui ajoute 0 pour cela
index=request.cookies("index")+0

' on écrit la page
%>

<html>
<head>
<title>Les sessions : cookie</title>
<script language="javascript">
  function envoyer(){
    with(document.frmSessions){
      // on crée un cookie pour index
      document.cookie="index="+lstElements.selectedIndex;
      // on envoie le formulaire
      submit();
    }//with
  }//envoyer()
</script>
</head>

<body bgcolor="#C0C0C0">
  <center>
  <h2>Session WEB : le cookie côté navigateur</h2>
  <hr>
  <center>
  <form name="frmSessions" method="POST">
    <select name="lstElements" size="1">
<%
for i=0 to ubound(liste)
  selected=""
  ' attention ici il faut que i et index soient 2 nombres
  if i=index then
    selected="selected"
  end if
  response.write("<option " & selected & "> " & liste(i) & " </option>")
next
%>
  </select>
  <input type="button" value="Envoyer" onclick="envoyer()">
</form>
<hr>
  L'élément <% =index %> (<% =liste(index) %>) a été sélectionné
</body>
</html>

```

## 28.3 Technologie PHP

On reprend les trois exemples déjà traités avec Perl et ASP.

### 28.3.1 Les champs cachés

```

<?
// on incrémente le champ caché hdAcces
$hdAcces=$hdAcces+1;
// on écrit la page
?>

<html>
<head>

```

```

<title>Les sessions : champ caché</title>
</head>
<body bgcolor="#C0C0C0">
<center>
<h2>Session WEB : champ caché</h2>
<hr>
<form name="frmSessions" method="POST">
  <input type="hidden" name="hdAcces" value="<? echo $hdAcces ?>">
  <table>
    <tr>
      <td><h3>Nombre d'accès : </h3></td>
      <td><h3><? echo $hdAcces ?></h3></td>
    </tr>
  </table>
  <input type="submit" value="Encore">
</form>
<hr>
A chaque fois que vous redemandez la page, le nombre d'accès augmente.<br>
Celui-ci est maintenu au fil de la session avec un champ caché.<br>
</body>
</html>

```

### 28.3.2 Le cookie initialisé côté serveur

```

<?
// on incrémente le cookie hdAcces
$hdAcces=$hdAcces+1;
// on renvoie le cookie
setcookie("hdAcces", $hdAcces);
// on affiche la page
?>

<html>
<head>
  <title>Les sessions : le cookie</title>
</head>
<body bgcolor="#C0C0C0">
<center>
<h2>Session WEB : le cookie</h2>
<hr>
<form name="frmSessions" method="POST">
  <table>
    <tr>
      <td><h3>Nombre d'accès : </h3></td>
      <td><h3><? echo $hdAcces ?></h3></td>
    </tr>
  </table>
  <input type="submit" value="Encore">
</form>
<hr>
A chaque fois que vous redemandez la page, le nombre d'accès augmente.<br>
Celui-ci est maintenu au fil de la session avec un cookie.<br>
</body>
</html>

```

### 28.3.3 Le cookie initialisé côté navigateur

```

<?
// initialisations
$liste=array("élément0", "élément1", "élément2", "élément3");

// le cookie index a été récupéré dans $index
$index+=0; //pour la lère fois lorsque $index=""
// on écrit la page
?>

<html>
<head>
  <title>Les sessions : cookie</title>
  <script language="javascript">
    function envoyer(){
      with(document.frmSessions){

```

```

        // on crée un cookie pour index
        document.cookie="index="+lstElements.selectedIndex;
        // on envoie le formulaire
        submit();
    }//with
    }//envoyer()
</script>
</head>

<body bgcolor="#C0C0C0">
    <center>
    <h2>Session WEB : le cookie côté navigateur</h2>
    <hr>
    <center>
    <form name="frmSessions" method="POST">
        <select name="lstElements" size="1">
<?
    for ($i=0;$i<sizeof($liste);$i++){
        $selected= ($i==$index) ? "selected" : "";
        print ("<option $selected>$liste[$i]</option>\n");
    }//for
?>
    </select>
    <input type="button" value="Envoyer" onclick="envoyer()">
    </form>
    <hr>
    L'élément <? echo $index ?> (<? echo $liste[$index] ?>) a été sélectionné
</body>
</html>

```

## 28.4 Technologie JSP

### 28.4.1 Les champs cachés

```

<%
// on incrémente le champ caché hdAcces
String S_hdAcces=request.getParameter("hdAcces");
int hdAcces;
try{
    hdAcces=Integer.parseInt(S_hdAcces);
}catch(Exception e){
    hdAcces=0;
}
hdAcces++;

// on écrit la page
%>

<html>
<head>
    <title>Les sessions : champ caché</title>
</head>
<body bgcolor="#C0C0C0">
<center>
    <h2>Session WEB : champ caché</h2>
    <hr>
    <form name="frmSessions" method="POST">
        <input type="hidden" name="hdAcces" value="<%=hdAcces%>">
        <table>
            <tr>
                <td><h3>Nombre d'accès : </h3></td>
                <td><h3><%=hdAcces%></h3></td>
            </tr>
        </table>
        <input type="submit" value="Encore">
    </form>
    <hr>
    A chaque fois que vous redemandez la page, le nombre d'accès augmente.<br>
    Celui-ci est maintenu au fil de la session avec un champ caché.<br>
</body>
</html>

```



## 28.4.2 Le cookie initialisé côté serveur

```
<%
// on récupère tous les cookies
Cookie[] cookies=request.getCookies();
// on cherche le cookie "hdAcces"
int i;
for(i=0;i<cookies.length;i++)
    if (cookies[i].getName().equals("hdAcces")) break;
// on prépare le nouveau cookie
Cookie cookie=null;
if(i<cookies.length) cookie=cookies[i];
else cookie=new Cookie("hdAcces","0");
// on incrémente le nombre d'accès
int hdAcces=0;
try{
    hdAcces=Integer.parseInt(cookie.getValue());
} catch (Exception ignored){}
// incrémentation et mémorisation
hdAcces++;
cookie.setValue(""+hdAcces);
// on renvoie le nouveau cookie
response.addCookie(cookie);
// on affiche la page
%>

<html>
<head>
<title>Les sessions : le cookie</title>
</head>
<body bgcolor="#C0C0C0">
<center>
<h2>Session WEB : le cookie</h2>
<hr>
<form name="frmSessions" method="POST">
<table>
<tr>
<td><h3>Nombre d'accès : </h3></td>
<td><h3><%=hdAcces%></h3></td>
</tr>
</table>
<input type="submit" value="Encore">
</form>
<hr>
A chaque fois que vous redemandez la page, le nombre d'accès augmente.<br>
Celui-ci est maintenu au fil de la session avec un cookie.<br>
</body>
</html>
```

## 28.4.3 Le cookie initialisé côté navigateur

```
<%
// initialisations
String[] liste={"élément0", "élément1", "élément2", "élément3"};
// on récupère tous les cookies
Cookie[] cookies=request.getCookies();
// on cherche le cookie "index"
int i;
for(i=0;i<cookies.length;i++)
    if (cookies[i].getName().equals("index")) break;
// on récupère l'index s'il existe, sinon index=0
int index=0;
if(i<cookies.length)
    try{
        index=Integer.parseInt(cookies[i].getValue());
    } catch (Exception ignored){}

// on écrit la page
%>

<html>
<head>
<title>Les sessions : cookie</title>
<script language="javascript">
```

```

        function envoyer(){
            with(document.frmSessions){
                // on crée un cookie pour index
                document.cookie="index="+lstElements.selectedIndex;
                // on envoie le formulaire
                submit();
            }//with
        }//envoyer()
    </script>
</head>

<body bgcolor="#C0C0C0">
    <center>
        <h2>Session WEB : le cookie côté navigateur</h2>
        <hr>
        <center>
            <form name="frmSessions" method="POST">
                <select name="lstElements" size="1">
<%
                    for (i=0;i<liste.length;i++){
                        String selected= (i==index) ? "selected" : "";
                        out.println("<option " + selected + ">" + liste[i] + "</option>");
                    }//for
%>
                </select>
                <input type="button" value="Envoyer" onclick="envoyer()">
            </form>
            <hr>
            L'élément <%=index%> (<%=liste[index]%>) a été sélectionné
        </body>
</html>

```

## 29.JAVASCRIPT

Nous montrons dans cette partie quelques exemples d'utilisation de Javascript dans les pages WEB. Nous nous centrons sur la gestion des formulaires mais Javascript peut faire bien davantage.

### 29.1 Récupérer les informations d'un formulaire

L'exemple ci-dessous montre comment récupérer au sein du navigateur les données entrées par l'utilisateur au sein d'un formulaire. Cela permet en général de faire des pré-traitements avant de les envoyer au serveur.

#### 29.1.1 Le formulaire

On a un formulaire rassemblant les composants les plus courants et d'un bouton **Afficher** permettant d'afficher les saisies faites par l'utilisateur.

**Un formulaire traité par Javascript**

Un champ textuel simple

Un champ textuel sur plusieurs lignes

Des boutons radio :  FM  GO  PO

Des choix multiples :  un  deux  trois

Un menu déroulant : 50 F

Une liste : Renault, Citroën, Peugeot

Un mot de passe : [masqué]

Un champ de contexte caché :

**Informations du formulaire**

Effacer

champ caché=secret  
champ textuel simple=  
champ textuel multiple=Ce texte est modifiable

Afficher

#### 29.1.2 Le code

```
<html>

<head>
  <title>Un formulaire traité par Javascript</title>
  <script language="javascript">
    function afficher(){
      // affiche dans une liste les infos du formulaire

      // on efface d'abord
      effacerInfos();

      // on affiche la valeur des # champs
      with(document.frmExemple){
        // champ caché
        ecrire("champ caché="+cache.value);
      }
    }
  </script>
</head>

<body>
  <form>
    <input type="text" value="Un champ textuel simple" />
    <input type="text" value="Un champ textuel sur plusieurs lignes" />
    <input checked="" type="radio" value="FM" /> FM
    <input type="radio" value="GO" /> GO
    <input type="radio" value="PO" /> PO
    <input type="checkbox" value="un" /> un
    <input type="checkbox" value="deux" /> deux
    <input type="checkbox" value="trois" /> trois
    <input type="text" value="50 F" />
    <select>
      <option value="Renault">Renault
      <option value="Citroën">Citroën
      <option value="Peugeot">Peugeot
    </select>
    <input type="password" value="Un mot de passe" />
    <input type="hidden" value="Un champ de contexte caché" />
    <input type="button" value="Afficher" />
  </form>
  <div>
    <input type="button" value="Effacer" />
    <div>
      champ caché=secret
      champ textuel simple=
      champ textuel multiple=Ce texte est modifiable
    </div>
    <input type="button" value="Afficher" />
  </div>
</body>
</html>
```

```

// champ textuel simple
ecrire("champ textuel simple="+simple.value);
// champ textuel multiple
ecrire("champ textuel multiple="+lignes.value);
// boutons radio
for(i=0;i<radio.length;i++){
  texte="radio["+i+"]="+radio[i].value;
  if(radio[i].checked) texte+=", coché";
  écrire(texte);
}
//for
// cases à cocher
for(i=0;i<qcm.length;i++){
  texte="qcm["+i+"]="+qcm[i].value;
  if(qcm[i].checked) texte+=", coché";
  écrire(texte);
}
//for
//liste déroulante
ecrire("index sélectionné dans le menu="+menu.selectedIndex);
for(i=0;i<menu.length;i++){
  texte="menu["+i+"]="+menu.options[i].text;
  if(menu.options[i].selected) texte+=", sélectionné";
  écrire(texte);
}
//for
//liste à choix multiple
for(i=0;i<lstVoitures.length;i++){
  texte="lstVoitures["+i+"]="+lstVoitures.options[i].text;
  if(lstVoitures.options[i].selected) texte+=", sélectionné";
  écrire(texte);
}
//for
//mot de passe
ecrire("mot de passe="+passwd.value);
}
//with
}
//function

function écrire(texte){
  // écrit texte dans la liste des infos
  frmInfos.lstInfos.options[frmInfos.lstInfos.length]=new Option(texte);
}
//écrire

function effacerInfos(){
  frmInfos.lstInfos.length=0;
}
//effacerInfos
</script>
</head>

<body bgcolor="#C0C0C0" onload="afficher()">
  <center>
    <h2>Un formulaire traité par Javascript</h2>
    <hr>
    <form method="POST" name="frmExemple">
      <input type="hidden" name="cache" value="secret">
      <table border="0">
        <tr>
          <td align="center">Un champ textuel simple</td>
          <td align="center" width="100">&nbsp;  </td>
          <td align="center">Un champ textuel sur plusieurs lignes</td>
        </tr>
        <tr>
          <td align="center"><input type="text" size="20" name="simple"></td>
          <td align="center" width="100">&nbsp;  </td>
          <td align="center">
            <textarea name="lignes" rows="2" cols="40">Ce texte est modifiable</textarea>
          </td>
        </tr>
      </table>
      <table border="0">
        <tr>
          <td><strong>Des boutons radio :</strong></td>
          <td>
            <input type="radio" checked name="radio" value="FM">FM
          </td>
          <td>
            <input type="radio" name="radio" value="GO">GO
          </td>
          <td>
            <input type="radio" name="radio" value="PO">PO
          </td>
          <td>&nbsp;  </td>
          <td><strong>Des choix multiples :</strong></td>
          <td></td>
        </tr>
      </table>
    </form>
  </center>

```

```

        <input type="checkbox" name="qcm" value="un">un
    </td>
    <td>
        <input type="checkbox" name="qcm" value="deux">deux
    </td>
    <td>
        <input type="checkbox" name="qcm" value="trois">trois
    </td>
</tr>
</table>
<table border="0">
    <tr>
        <td>Un menu déroulant : </td>
        <td>
            <select name="menu" size="1">
                <option>50 F</option>
                <option>60 F</option>
                <option>70 F</option>
                <option>100 F</option>
            </select>
        </td>
        <td>Une liste :</td>
        <td>
            <select name="lstVoitures" multiple size="3">
                <option>Renault</option>
                <option>Citroën</option>
                <option>Peugeot</option>
                <option>Fiat</option>
                <option>Audi</option>
            </select>
        </td>
    </tr>
</table>
<table border="0">
    <tr>
        <td>Un mot de passe : </td>
        <td><input type="password" size="21" name="passwd"></td>
        <td>&nbsp;</td>
        <td>Un champ de contexte caché : </td>
    </tr>
</table>
</form>
<hr>
<h2>Informations du formulaire</h2>
<form name="frmInfos">
    <table>
        <tr>
            <td><input type="button" value="Effacer" onclick="effacerInfos()"></td>
            <td>
                <select name="lstInfos" multiple size="3">
                </select>
            </td>
            <td>
                <input type="button" name="cmdAfficher" value="Afficher" onclick="afficher()">
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

## 29.2 Les expressions régulières en Javascript

Côté navigateur, Javascript peut être utilisé pour vérifier la validité des données entrées par l'utilisateur avant de les envoyer au serveur. Voici un programme de test de ces expressions régulières.

### 29.2.1 La page de test

## Les expressions régulières en Javascript

Expression régulière

Jouer le test

Chaîne de test

Ajouter aux exemples

### Résultats de l'instruction champs=expression régulière.exec(chaine)

```
Il y a correspondance  
champs[0]=[45abcd67]  
champs[1]=[45]
```

### Exemples

Modèles

Chaînes

Jouer l'exemple

### 29.2.2 Le code de la page

```
<html>  
  
<head>  
<title>Les expressions régulières en Javascript</title>  
<script language="javascript">  
  function afficherInfos(){  
    with(document.frmRegExp){  
      // qq chose à faire ?  
      if (! verifier()) return;  
      // c'est bon - on efface les résultats précédents  
      effacerInfos();  
      // vérification du modèle  
      modele=new RegExp(txtModele.value);  
      champs=modele.exec(txtChaine.value);  
      if(champs==null)  
        // pas de correspondance entre modèle et chaîne  
        ecrireInfos("pas de correspondance");  
      else{  
        // correspondance - on affiche les résultats obtenus  
        ecrireInfos("Il y a correspondance");  
        for(i=0;i<champs.length;i++)  
          ecrireInfos("champs["+i+"]="+champs[i]+"");  
      }//else  
    }//with  
  }//function  
  
  function ecrireInfos(texte){  
    // écrit texte dans la liste des infos  
    document.frmRegExp.lstInfos.options[document.frmRegExp.lstInfos.length]=new  
Option(texte);  
  }//écrire  
  
  function effacerInfos(){  
    frmRegExp.lstInfos.length=0;  
  }//effacerInfos  
  
  function jouer(){  
    // teste le modèle contre la chaîne dans l'exemple choisi
```

```

        with(document.frmRegExp){
            txtModele.value=lstModeles.options[lstModeles.selectedIndex].text
            txtChaine.value=lstChaines.options[lstChaines.selectedIndex].text
            afficherInfos();
        }//with
    }//jouer

    function ajouter(){
        //ajoute le test courant aux exemples
        with(document.frmRegExp){
            // qq chose à faire ?
            if (! verifier()) return;
            // ajout
            lstModeles.options[lstModeles.length]=new Option(txtModele.value);
            lstChaines.options[lstChaines.length]=new Option(txtChaine.value);
            // raz saisies
            txtModele.value="";
            txtChaine.value="";
        }//with
    }//ajouter

    function verifier(){
        // vérifie que les champs de saisie sont non vides
        with(document.frmRegExp){
            champs=/^\s*$/ .exec(txtModele.value);
            if(champs!=null){
                alert("Vous n'avez pas indiqué de modèle");
                txtModele.focus();
                return false;
            }//if
            champs=/^\s*$/ .exec(txtChaine.value);
            if(champs!=null){
                alert("Vous n'avez pas indiqué de chaîne de test");
                txtChaine.focus();
                return false;
            }//if
            // c'est bon
            return true;
        }//with
    }//verifier
</script>
</head>

<body bgcolor="#C0C0C0">
<center>
    <h2>Les expressions régulières en Javascript</h2>
<hr>
    <form name="frmRegExp">
        <table>
            <tr>
                <td>Expression régulière</td>
                <td>Chaîne de test</td>
            </tr>
            <tr>
                <td><input type="text" name="txtModele" size="20"></td>
                <td><input type="text" name="txtChaine" size="20"></td>
            </tr>
            <tr>
                <td>
                    <input type="button" name="cmdAfficher" value="Jouer le test"
                    onclick="afficherInfos()" />
                </td>
                <td>
                    <input type="button" name="cmdAjouter" value="Ajouter aux exemples"
                    onclick="ajouter()" />
                </td>
            </tr>
        </table>
<hr>
    <h2>Résultats de l'instruction champs=expression régulière.exec(chaine)</h2>
    <table>
        <tr>
            <td>
                <select name="lstInfos" size="3">
                </select>
            </td>
        </tr>
    </table>
<hr>
    <h2>Exemples</h2>

```

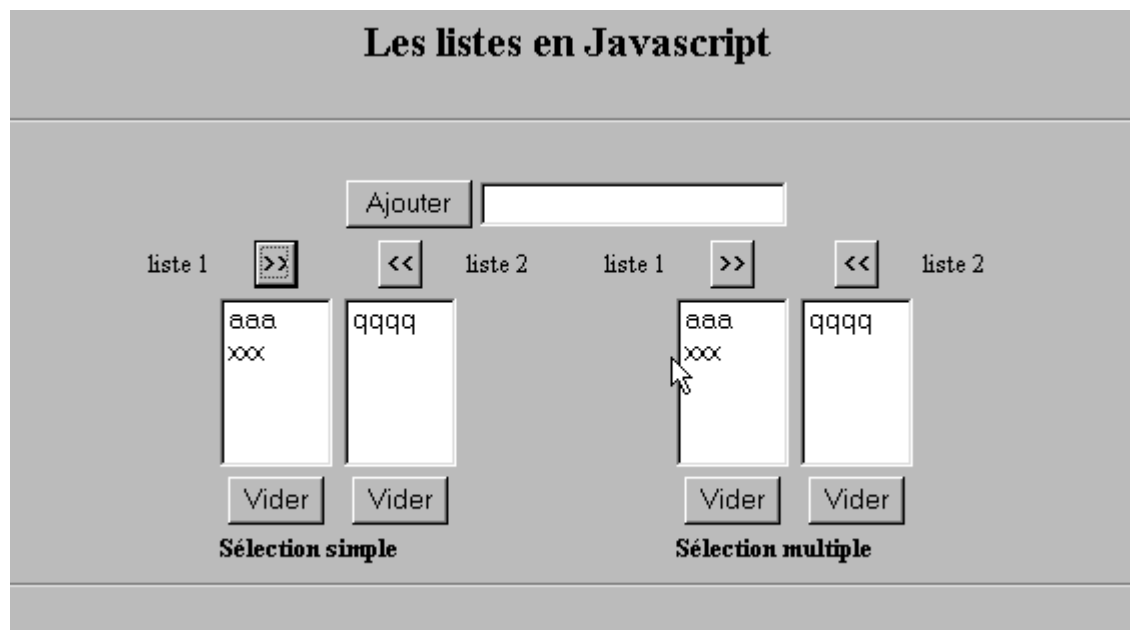
```

<table>
  <tr>
    <td align="center">Modèles</td>
    <td align="center">Chaînes</td>
  </tr>
  <tr>
    <td>
      <select name="lstModeles" size="1">
        <option>^\d+$</option>
        <option>^\d+ (\d+)$</option>
        <option>^\d+ (.*) (\d+)$</option>
        <option>^\d+ (\s+) (\d+)$</option>
      </select>
    </td>
    <td>
      <select name="lstChaines" size="1">
        <option>67</option>
        <option>56 84</option>
        <option>45abcd67</option>
        <option>45 67</option>
      </select>
    </td>
  </tr>
  <tr>
    <td colspan="2" style="text-align: center;">
      <input type="button" name="cmdJouer" value="Jouer l'exemple" onclick="jouer()">
    </td>
  </tr>
</table>
</form>
</body>
</html>

```

## 29.3 Gestion des listes en JavaScript

### 29.3.1 Le formulaire



### 29.3.2 Le code

```

<html>
  <head>
    <title>Les listes en Javascript</title>
  </head>
  <script language="javascript">
    // ajouter

```



```

function ajouter(L1,L2,T){
  // ajoute la valeur du champ T aux listes L1,L2
  // qq chose à faire ?
  champs=/^\s*$/ .exec(T.value);
  if(champs!=null){
    // le champ est vide
    alert("Vous n'avez pas indiqué la valeur à ajouter");
    txtElement.focus();
    return;
  }//if
  // on ajoute l'élément
  L1.options[L1.length]=new Option(T.value);
  L2.options[L2.length]=new Option(T.value);
  T.value="";
}//ajouter

//vider
function vider(L){
  // vide la liste L
  L.length=0;
}//vider

//transfert
function transfert(L1,L2, simple){
  //transfère dans L2 les éléments sélectionnés dans la liste L1

  // qq chose à faire ?
  // index de l'élément sélectionné dans L1
  index1=L1.selectedIndex;
  if(index1==-1){
    alert("Vous n'avez pas sélectionné d'élément");
    return;
  }//if
  // quel est le mode de sélection des éléments des listes
  if(simple){ // sélection simple
    element1=L1.options[index1].text;
    //ajout dans L2
    L2.options[L2.length]=new Option(element1);
    //suppression dans L1
    L1.options[index1]=null;
  }//simple
  if(! simple){ //sélection multiple
    //on parcourt la liste 1 en sens inverse
    for(i=L1.length-1;i>=0;i--){
      //élément sélectionné ?
      if(L1.options[i].selected){
        //on l'ajoute à L2
        L2.options[L2.length]=new Option(L1.options[i].text);
        //on le supprime de L1
        L1.options[i]=null;
      }//if
    }//for i
  }//if ! simple
}//transfert
</script>
</head>

<body bgcolor="#C0C0C0">
  <center>
    <h2>Les listes en Javascript</h2>
    <hr>
    <form name="frmListes">
      <table>
        <tr>
          <td>
            <input type="button" name="cmdAjouter" value="Ajouter"
onclick="ajouter(lst1A,lst1B,txtElement) ">
          </td>
          <td>
            <input type="text" name="txtElement">
          </td>
        </tr>
      </table>
      <table>
        <tr>
          <td align="center">liste 1</td>
          <td align="center"><input type="button" value=">>"
onclick="transfert(lst1A,lst2A,true) "</td>
          <td align="center"><input type="button" value="<<"
onclick="transfert(lst2A,lst1A,true) "</td>
        </tr>
      </table>
    </form>
  </center>
</body>

```

```

        <td align="center">liste 2</td>
        <td width="30"></td>
        <td align="center">liste 1</td>
        <td align="center"><input type="button" value="">
onclick="transfert(lst1B,lst2B,false) "</td>
        <td align="center"><input type="button" value="">
onclick="transfert(lst2B,lst1B,false) "</td>
        <td align="center">liste 2</td>
    </tr>
    <tr>
        <td></td>
        <td align="center">
            <select name="lst1A" size="5">
            </select>
        </td>
        <td align="center">
            <select name="lst2A" size="5">
            </select>
        </td>
        <td></td>
        <td></td>
        <td></td>
        <td align="center">
            <select name="lst1B" size="5" multiple >
            </select>
        </td>
        <td align="center">
            <select name="lst2B" size="5" multiple>
            </select>
        </td>
    </tr>
    <tr>
        <td></td>
        <td align="center"><input type="button" value="Vider" onclick="vider(lst1A) "</td>
        <td align="center"><input type="button" value="Vider" onclick="vider(lst2A) "</td>
        <td></td>
        <td></td>
        <td></td>
        <td align="center"><input type="button" value="Vider" onclick="vider(lst1B) "</td>
        <td align="center"><input type="button" value="Vider" onclick="vider(lst2B) "</td>
        <td></td>
    </tr>
    <tr>
        <td></td>
        <td colspan="2"><strong>Sélection simple</strong></td>
        <td></td>
        <td></td>
        <td colspan="2"><strong>Sélection multiple</strong></td>
        <td></td>
    </tr>
</table>
<hr>
</form>
</body>
</html>

```

<b><u>1. Généralités</u></b> .....	<b>6</b>
<b><u>1.1 LES COMPOSANTES D'UNE APPLICATION WEB</u></b> .....	<b>6</b>
<b><u>1.2 LES ÉCHANGES DANS UNE APPLICATION WEB AVEC FORMULAIRE DE SAISIE</u></b> .....	<b>7</b>
<b><u>1.3 QUELQUES RESSOURCES</u></b> .....	<b>8</b>
<b><u>2. Serveurs Web, Navigateurs, Langages de scripts</u></b> .....	<b>9</b>
<b><u>3. Les navigateurs</u></b> .....	<b>9</b>
<b><u>3.1 Netscape</u></b> .....	<b>9</b>
<b><u>3.2 Internet Explorer (IE)</u></b> .....	<b>9</b>
<b><u>4. Les langages du WEB</u></b> .....	<b>9</b>
<b><u>4.1 PHP</u></b> .....	<b>9</b>
<b><u>4.1.1 Installation</u></b> .....	<b>9</b>
<b><u>4.1.2 Un premier script php</u></b> .....	<b>10</b>
<b><u>4.2 PERL</u></b> .....	<b>10</b>
<b><u>4.2.1 Installation</u></b> .....	<b>10</b>
<b><u>4.2.2 Tests</u></b> .....	<b>10</b>
<b><u>4.3 JAVA</u></b> .....	<b>11</b>
<b><u>4.3.1 Installation</u></b> .....	<b>11</b>
<b><u>4.3.2 Tests</u></b> .....	<b>11</b>
<b><u>5. Serveurs WEB</u></b> .....	<b>12</b>
<b><u>5.1 Serveur Apache</u></b> .....	<b>12</b>
<b><u>5.1.1 Installation</u></b> .....	<b>12</b>
<b><u>5.1.2 Configuration</u></b> .....	<b>12</b>
<b><u>5.1.3 Lien PHP - Apache</u></b> .....	<b>13</b>
<b><u>5.1.4 Lien PERL-APACHE</u></b> .....	<b>13</b>
<b><u>5.2 Le serveur PWS</u></b> .....	<b>15</b>
<b><u>5.2.1 Installation</u></b> .....	<b>15</b>
<b><u>5.2.2 Configuration</u></b> .....	<b>15</b>
<b><u>5.2.3 Lien PHP - PWS</u></b> .....	<b>15</b>
<b><u>5.3 Tomcat : servlets Java et pages JSP (Java Server Pages)</u></b> .....	<b>15</b>
<b><u>5.3.1 Installation</u></b> .....	<b>15</b>
<b><u>5.3.2 Lancement du serveur Web Tomcat</u></b> .....	<b>16</b>
<b><u>5.3.3 Premiers tests</u></b> .....	<b>17</b>
<b><u>5.3.4 Lien Apache-Tomcat</u></b> .....	<b>17</b>
<b><u>5.3.5 Tests Tomcat-Apache</u></b> .....	<b>18</b>
<b><u>6. Page statique HTML</u></b> .....	<b>19</b>
<b><u>6.1 Le script essai1.html</u></b> .....	<b>19</b>
<b><u>6.2 Les tests</u></b> .....	<b>20</b>
<b><u>7. Une page ASP (Active Server Pages)</u></b> .....	<b>20</b>
<b><u>7.1 Le script essai2.asp</u></b> .....	<b>20</b>
<b><u>7.2 Le test</u></b> .....	<b>20</b>
<b><u>8. Un script Perl</u></b> .....	<b>20</b>
<b><u>8.1 Le script : essai3.pl</u></b> .....	<b>20</b>
<b><u>8.2 Le test</u></b> .....	<b>21</b>
<b><u>9. Un script PHP</u></b> .....	<b>21</b>

9.1	Le script : essai4.php.....	21
9.2	Les tests.....	22
10.	<i>Un script JSP.....</i>	22
10.1	Le script : heure.jsp.....	22
10.2	Les tests.....	23
11.	<i>Une page Web avec un script Vbscript, côté navigateur.....</i>	24
11.1	La page : vbs1.html.....	24
11.2	Les tests.....	25
12.	<i>Une page Web avec un script Javascript, côté navigateur.....</i>	25
12.1	La page : js1.html.....	25
12.2	Le test.....	25
13.	<i>Une page Web avec un script Perl, côté navigateur.....</i>	26
13.1	La page : ps1.html.....	26
13.2	Le test.....	26
14.	<i>Une page Web avec un script VB, côté serveur.....</i>	27
14.1	La page : vbs2.asp.....	27
14.2	Le test.....	28
15.	<i>Formulaire Web, javascript côté navigateur.....</i>	29
15.1	La page : form1.html.....	29
15.2	Les tests.....	30
16.	<i>Formulaire Web, vbscript côté navigateur.....</i>	30
16.1	La page : form2.html.....	30
16.2	Le test.....	31
17.	<i>Formulaire Web, Perlscript côté navigateur.....</i>	31
17.1	La page : form3.html.....	31
17.2	Le test.....	32
18.	<i>Vérification du formulaire avant l'envoi, par un script côté navigateur.....</i>	32
18.1	Le texte HTML.....	32
18.2	Test.....	33
19.	<i>Le dialogue client/serveur.....</i>	34
19.1	Le formulaire : form4.asp.....	34
19.2	Le texte HTML du formulaire.....	34
19.3	Demande initiale du formulaire.....	35
19.4	Envoi du formulaire : méthode GET.....	35
19.5	Envoi du formulaire : méthode POST.....	36
20.	<i>Traitement du formulaire Web avec un script ASP côté serveur.....</i>	38
20.1	La page ASP pour une méthode d'envoi GET.....	38
20.2	Le test.....	39
20.3	La page ASP pour une méthode d'envoi POST.....	39

20.4	Le test.....	40
21.	Traitement d'un formulaire Web avec un script Perl côté serveur.....	41
21.1	Le script Perl.....	41
21.2	Le test.....	43
22.	Technologie ASP.....	44
22.1	Le formulaire à construire.....	44
22.2	Les balises HTML de base.....	44
22.3	Une page HTML simplifiée.....	46
22.4	Passage des valeurs du formulaire par la méthode GET.....	47
22.5	Passage des valeurs du formulaire par la méthode POST.....	48
22.6	Récupération des variables d'environnement du serveur Web.....	49
23.	Technologie Perl.....	50
23.1	Le formulaire à construire.....	50
23.2	Le script Perl d'affichage et de traitement du formulaire.....	50
23.3	Le test.....	52
23.4	Récupération des variables d'environnement du serveur.....	52
24.	Amélioration du script Perl.....	54
24.1	Le script.....	54
24.2	Les tests.....	56
25.	Technologie JSP.....	57
25.1	Le script JSP.....	57
25.2	Le test.....	59
26.	Technologie PHP.....	60
26.1	Le script PHP.....	60
26.2	Les tests.....	62
27.	L'application.....	64
27.1	Pilote ODBC/Nom DSN.....	65
27.2	Technologie ASP.....	67
27.2.1	Le script.....	67
27.2.2	Les tests.....	70
27.3	Technologie JSP.....	72
27.3.1	Le script.....	72
27.3.2	Les tests.....	75
27.4	Technologie PHP.....	76
27.4.1	Le script.....	76
27.4.2	Les tests.....	77
27.5	Technologie PERL.....	79
27.5.1	Le script.....	79
27.5.2	Les tests.....	81
28.	Les sessions WEB.....	82
28.1	Technologie Perl.....	82
28.1.1	Les champs cachés.....	82

<a href="#">28.1.2</a> Le cookie initialisé côté serveur.....	83
<a href="#">28.1.3</a> Le cookie initialisé côté navigateur.....	84
<b><a href="#">28.2</a> Technologie ASP.....</b>	<b>85</b>
<a href="#">28.2.1</a> Les champs cachés.....	85
<a href="#">28.2.2</a> Le cookie initialisé côté serveur.....	85
<a href="#">28.2.3</a> Le cookie initialisé côté navigateur.....	86
<b><a href="#">28.3</a> Technologie PHP.....</b>	<b>86</b>
<a href="#">28.3.1</a> Les champs cachés.....	86
<a href="#">28.3.2</a> Le cookie initialisé côté serveur.....	87
<a href="#">28.3.3</a> Le cookie initialisé côté navigateur.....	87
<b><a href="#">28.4</a> Technologie JSP.....</b>	<b>88</b>
<a href="#">28.4.1</a> Les champs cachés.....	88
<a href="#">28.4.2</a> Le cookie initialisé côté serveur.....	89
<a href="#">28.4.3</a> Le cookie initialisé côté navigateur.....	89
<b><a href="#">29</a>.JAVASCRIPT.....</b>	<b>91</b>
<b><a href="#">29.1</a> Récupérer les informations d'un formulaire.....</b>	<b>91</b>
<a href="#">29.1.1</a> Le formulaire.....	91
<a href="#">29.1.2</a> Le code.....	91
<b><a href="#">29.2</a> Les expressions régulières en Javascript.....</b>	<b>93</b>
<a href="#">29.2.1</a> La page de test.....	93
<a href="#">29.2.2</a> Le code de la page.....	94
<b><a href="#">29.3</a> Gestion des listes en JavaScript.....</b>	<b>96</b>
<a href="#">29.3.1</a> Le formulaire.....	96
<a href="#">29.3.2</a> Le code.....	96