

PHP

Arnaud Sallaberry

arnaud.sallaberry@univ-montp3.fr

Plan

- **Les principaux langages du Web**
 - **Côté client**
 - Côté serveur
 - Autres
 - Récapitulatif
- **PHP**
 - Introduction
 - Types de données
 - Opérateurs
 - Structures de contrôle
 - Commentaires
 - Fichiers CSV

HTML

Voir cours de la première séance

Feuilles de style

- Directives d'affichage (css)
 - Le source de la page reste le même (html)
 - L'affichage est décrit en CSS
 - La feuille de style est envoyée par le serveur
- XML, XSL, XSLT
 - Séparer le contenu de son affichage
 - Contenu décrit en XML (source)
 - Affichage défini en XSL et/ou XSLT (ou CSS)
- Prise en compte d'évènements

Javascript

- Exécution sur le client
 - Script transmis avec la page
 - Interprété par le navigateur
- DOM: Document Object Model
 - Modèle objet du document (Inspecteur DOM)
 - Permet de consulter et de modifier la page en cours d’affichage

Plan

- **Les principaux langages du Web**
 - Côté client
 - **Côté serveur**
 - Autres
 - Récapitulatif
- PHP
 - Introduction
 - Types de données
 - Opérateurs
 - Structures de contrôle
 - Commentaires
 - Fichiers CSV

PHP

- Exécution sur le serveur
 - Calcul de la page à afficher
 - Transmission au client
- Connexion avec une base de données
- Permet lecture et écriture de fichiers sur le serveur

Python

- Même principe que PHP
- Common Gateway Interface (CGI) : Interface permettant l'exécution de scripts Python sur le serveur
- Principe:
 - mettre un script Python dans un répertoire approprié configuré par l'administrateur de votre système.
 - Le script est ensuite appelé par l'URL
<http://localhost/cgi-bin/nomDuScript.py>

Plan

- **Les principaux langages du Web**
 - Côté client
 - Côté serveur
 - **Autres**
 - Récapitulatif
- PHP
 - Introduction
 - Types de données
 - Opérateurs
 - Structures de contrôle
 - Commentaires
 - Fichiers CSV

Java

- Langage de programmation à objets
- Gestion sécurisée de la mémoire
- Existe sur toute machine
- *Applet* : exécution par le client
- *Servlet* : exécution sur le serveur

Ajax

- « Asynchronous Javascript and XML »
- Combiner de façon asynchrone les processus clients et les requêtes au serveur
- Voir la page
[http://fr.wikipedia.org/wiki/Asynchronous JavaScript and XML](http://fr.wikipedia.org/wiki/Asynchronous_JavaScript_and_XML)
- Exemple combinant javascript et php

Plan

- **Les principaux langages du Web**
 - Côté client
 - Côté serveur
 - Autres
 - **Récapitulatif**
- PHP
 - Introduction
 - Types de données
 - Opérateurs
 - Structures de contrôle
 - Commentaires
 - Fichiers CSV

Les langages du Web

Exécution sur le Serveur	Exécution sur le Client
Python (CGI) PHP Servlet	HTML XLS, XLST, CSS Javascript Applet

Plan

- Les principaux langages du Web
 - Côté client
 - Côté serveur
 - Autres
 - Récapitulatif
- **PHP**
 - **Introduction**
 - Types de données
 - Opérateurs
 - Structures de contrôle
 - Commentaires
 - Fichiers CSV

Introduction

- Extension d'un fichier : .php (exemple : index.php)
- PHP est un langage serveur => nécessité d'utiliser un serveur en local pour développer votre site
- **Client** : votre navigateur web, par exemple Firefox
- **Serveur installé en local** : WAMP sur les ordinateurs salle 108
 - Placez votre site dans le répertoire de WAMP :
c:/wamp/www/
 - Appelez vos pages depuis votre navigateur :
<http://localhost/.../nomPage.php>

Introduction

- **Bonjour le monde en HTML**

```
<html>
  <head><title>Bonjour</title></head>
  <body>
    Bonjour le monde !
  </body>
</html>
```

- **Bonjour le monde en PHP**

```
<html>
  <head><title>Bonjour</title></head>
  <body>
    <?php
      echo "Bonjour le monde !";
    ?>
  </body>
</html>
```

Introduction

- Le code se place entre les balises `<?php` et `?>`
- Chaque instruction se termine par un `;`
- Le serveur exécute le script et envoie ensuite le code HTML résultant au client
- `echo` permet « d'écrire » une phrase (équivalent de `print` en Python)
- `<?php echo "Bonjour le monde !"; ?>` affiche `Bonjour le monde !`

Si mon code php est :

```
<html>
  <head><title>Bonjour</title></head>
  <body>
    <?php
      echo "Bonjour le monde !";
    ?>
  </body>
</html>
```

Introduction

- Le code se place entre les balises `<?php` et `?>`
- Chaque instruction se termine par un `;`
- Le serveur exécute le script et envoie ensuite le code HTML résultant au client
- `echo` permet « d'écrire » une phrase (équivalent de `print` en Python)
- `<?php echo "Bonjour le monde !"; ?>` affiche `Bonjour le monde !`

Le client va recevoir :

```
<html>
  <head><title>Bonjour</title></head>
  <body>
    Bonjour le monde !
  </body>
</html>
```

Plan

- Les principaux langages du Web
 - Côté client
 - Côté serveur
 - Autres
 - Récapitulatif
- **PHP**
 - Introduction
 - **Types de données**
 - Opérateurs
 - Structures de contrôle
 - Commentaires
 - Fichiers CSV

Types de données

- Pour définir une variable, précéder son nom de \$:

```
$maVar = 3000;
```

- Comme en Python, le type est défini par l'affectation :

```
$maVar = 3000;
```

type entier

```
$maVar = "Sallaberry";
```

type chaine de caractères

Types de données

- **Booléens** (boolean)
 - 2 valeurs possibles : TRUE ou FALSE
 - Exemples :
 - `$monBool = TRUE;`
 - `$monBool = FALSE;`
- **Nombres entiers** (integer)
 - Suite de chiffres
 - Peut être précédé de – si l'entier est négatif
 - Exemples :
 - `$monEnt = 1276;`
 - `$monEnt = -67;`

Types de données

- **Nombres flottants** (double, float)
 - Suite de chiffres avec virgule
 - Virgule codée par un `.` (notation anglaise)
 - Peut être précédé de `-` si le nombre est négatif
 - Exemples :
 - `$monFlo = 1276.89;`
 - `$monFlo = -67.28;`
- **Chaines de caractères** (string)
 - Suite de caractères compris entre `"` et `"`.
 - Exemples :
 - `$maChaine = "Bonjour";`
 - `$maChaine = "Toute une phrase";`

Types de données

- **Tableau (array) :**

- Liste ordonnée d'éléments

- Exemples :

- `$monTab = array("mot1", "mot2", "mot3");`

- `$monTab = array(1, "mot2", 3.9, TRUE);`

- `$monTab = array($maVar1, "ab", 3.9, $mavVar3);`

- Accès aux éléments :

- `$monTab = array(1, "mot2", 3.9, TRUE);`

- `echo $monTab[2];`

- écrit 3.9 dans le code HTML

Types de données

- Fonction `gettype` : permet de connaître le type d'une variable
- Exemples :

```
echo gettype(3);  
        écrit integer dans le code HTML
```

```
$maChaine = "Bonjour";  
echo gettype($maChaine);  
        écrit string dans le code HTML
```

Plan

- Les principaux langages du Web
 - Côté client
 - Côté serveur
 - Autres
 - Récapitulatif
- **PHP**
 - Introduction
 - Types de données
 - **Opérateurs**
 - Structures de contrôle
 - Commentaires
 - Fichiers CSV

Opérateurs

- **Affectation (=) :**

- Par copie pour tous les types prédéfinis, y compris les **chaines de caractères** et les **tableaux** !
- Par référence pour les objets que vous créez.

- Exemple 1 :

```
$v1 = 1;  
$v2 = $v1;  
$v1 = 2  
echo $v1 // Affiche 2  
echo $v2 // Affiche 1
```

- Exemple 2 :

```
$v1 = array(1,2,3);  
$v2 = $v1;  
$v1[0] = "test";  
echo $v1[0]; // Affiche "test"  
echo $v2[0]; // Affiche 1
```

Opérateurs

- **Arithmétiques :**

Exemple	Nom	Résultat
$-\$a$	Négation	Opposé de $\$a$.
$\$a + \b	Addition	Somme de $\$a$ et $\$b$.
$\$a - \b	Soustraction	Différence de $\$a$ et $\$b$.
$\$a * \b	Multiplication	Produit de $\$a$ et $\$b$.
$\$a / \b	Division	Quotient de $\$a$ et $\$b$.
$\$a \% \b	Modulo	Reste de $\$a$ divisé par $\$b$.
$\$a ** \b	Exponentielle	Résultat de l'élévation de $\$a$ à la puissance $\$b$.

Opérateurs

- **Combinés :**

- Tout opérateur arithmétique peut être combiné avec l'opérateur d'affectation

- Exemple 1 :

```
$v = 1;  
$v += 2;           // Equivalent à $v=$v+2;  
echo $v;          // Affiche 3
```

- Exemple 2 :

```
$v = 2;  
$v **= 3;         // Equivalent à $v=$v**3;  
echo $v;          // Affiche 8
```

Opérateurs

- **D'incrémentation :**

Exemple	Nom	Résultat
<code>++\$a</code>	Pre-incrémente	Incrémente $\$a$ de 1, puis retourne $\$a$.
<code>\$a++</code>	Post-incrémente	Retourne $\$a$, puis incrémente $\$a$ de 1.
<code>--\$a</code>	Pré-décrémente	Décrémente $\$a$ de 1, puis retourne $\$a$.
<code>\$a--</code>	Post-décrémente	Retourne $\$a$, puis décrémente $\$a$ de 1.

Opérateurs

- De comparaison :

Exemple	Nom	Résultat
$\$a == \b	Egal	TRUE si $\$a$ est égal à $\$b$ après le transtypage.
$\$a === \b	Identique	TRUE si $\$a$ est égal à $\$b$ et qu'ils sont de même type.
$\$a != \b	Différent	TRUE si $\$a$ est différent de $\$b$ après le transtypage.
$\$a <> \b	Différent	TRUE si $\$a$ est différent de $\$b$ après le transtypage.
$\$a !== \b	Différent	TRUE si $\$a$ est différent de $\$b$ ou bien s'ils ne sont pas du même type.
$\$a < \b	Plus petit que	TRUE si $\$a$ est strictement plus petit que $\$b$.
$\$a > \b	Plus grand	TRUE si $\$a$ est strictement plus grand que $\$b$.
$\$a <= \b	Inférieur ou égal	TRUE si $\$a$ est plus petit ou égal à $\$b$.
$\$a >= \b	Supérieur ou égal	TRUE si $\$a$ est plus grand ou égal à $\$b$.

Opérateurs

- Logiques :

Exemple	Nom	Résultat
$\$a$ and $\$b$	And (Et)	TRUE si $\$a$ ET $\$b$ valent TRUE .
$\$a$ or $\$b$	Or (Ou)	TRUE si $\$a$ OU $\$b$ valent TRUE .
$\$a$ xor $\$b$	XOR	TRUE si $\$a$ OU $\$b$ est TRUE , mais pas les deux en même temps.
! $\$a$	Not (Non)	TRUE si $\$a$ n'est pas TRUE .
$\$a$ && $\$b$	And (Et)	TRUE si $\$a$ ET $\$b$ sont TRUE .
$\$a$ $\$b$	Or (Ou)	TRUE si $\$a$ OU $\$b$ est TRUE .

Opérateurs

- **Chaines de caractères :**

- Concaténation : .

- Exemple :

```
$v1 = "Bonjour " ;  
$v2 = "Monsieur " ;  
$v3 = $v1.$v2 ;  
echo $v3 ;                // Affiche "Bonjour Monsieur"
```

- Concaténation et assignation : .=

- Exemple :

```
$v = "Bonjour " ;  
$v .= "Monsieur" ;      // Equivalent à $v=$v."Monsieur" ;  
echo $v ;                // Affiche "Bonjour Monsieur"
```

Opérateurs

- Tableaux :

Exemple	Nom	Résultat
$\$a + \b	Union	Union de $\$a$ et $\$b$.
$\$a == \b	Egalité	TRUE si $\$a$ et $\$b$ contiennent les mêmes paires clés/valeurs.
$\$a === \b	Identique	TRUE si $\$a$ et $\$b$ contiennent les mêmes paires clés/valeurs dans le même ordre et du même type.
$\$a != \b	Inégalité	TRUE si $\$a$ n'est pas égal à $\$b$.
$\$a <> \b	Inégalité	TRUE si $\$a$ n'est pas égal à $\$b$.
$\$a !== \b	Non-identique	TRUE si $\$a$ n'est pas identique à $\$b$.

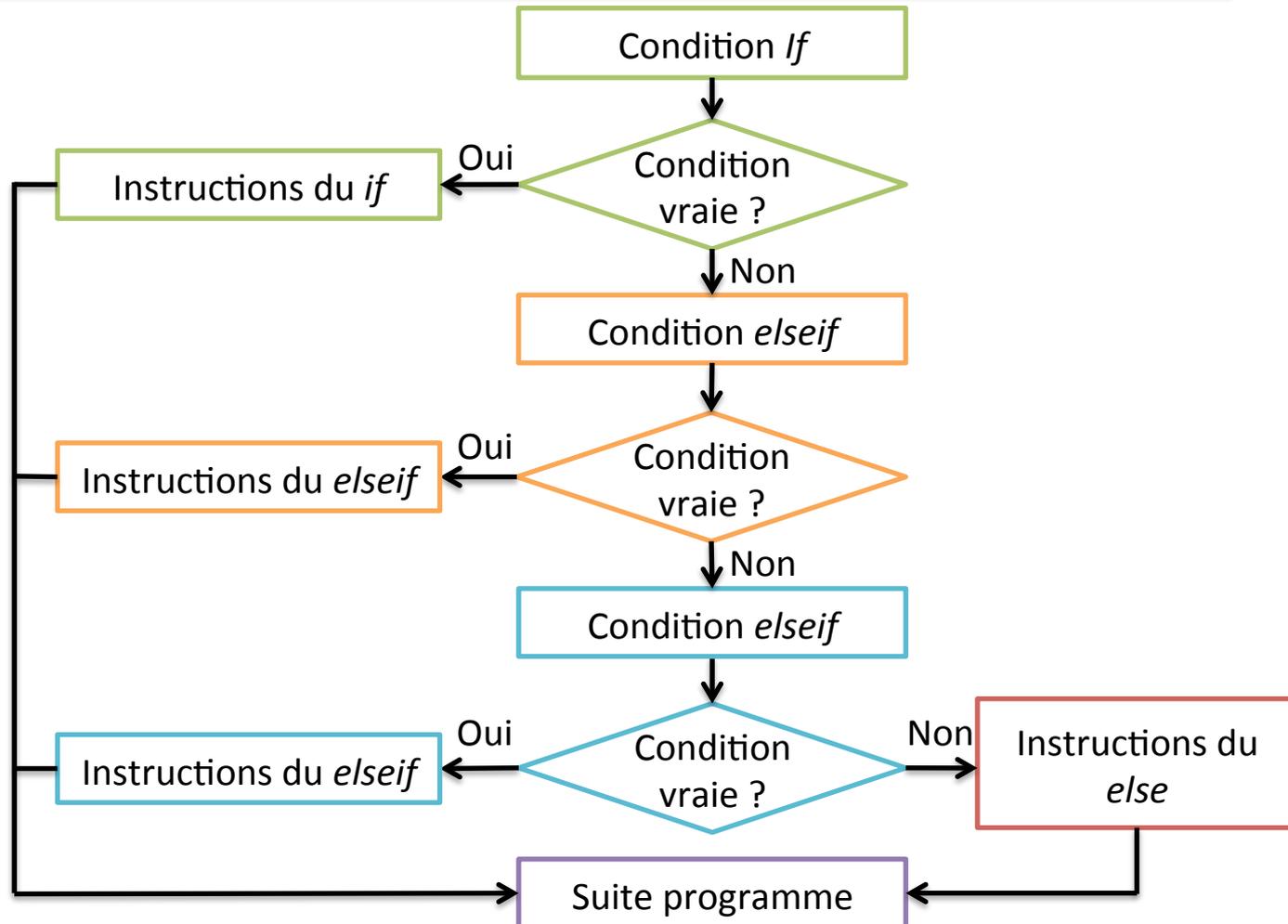
Plan

- Les principaux langages du Web
 - Côté client
 - Côté serveur
 - Autres
 - Récapitulatif
- **PHP**
 - Introduction
 - Types de données
 - Opérateurs
 - **Structures de contrôle**
 - Commentaires
 - Fichiers CSV

Structures de contrôle

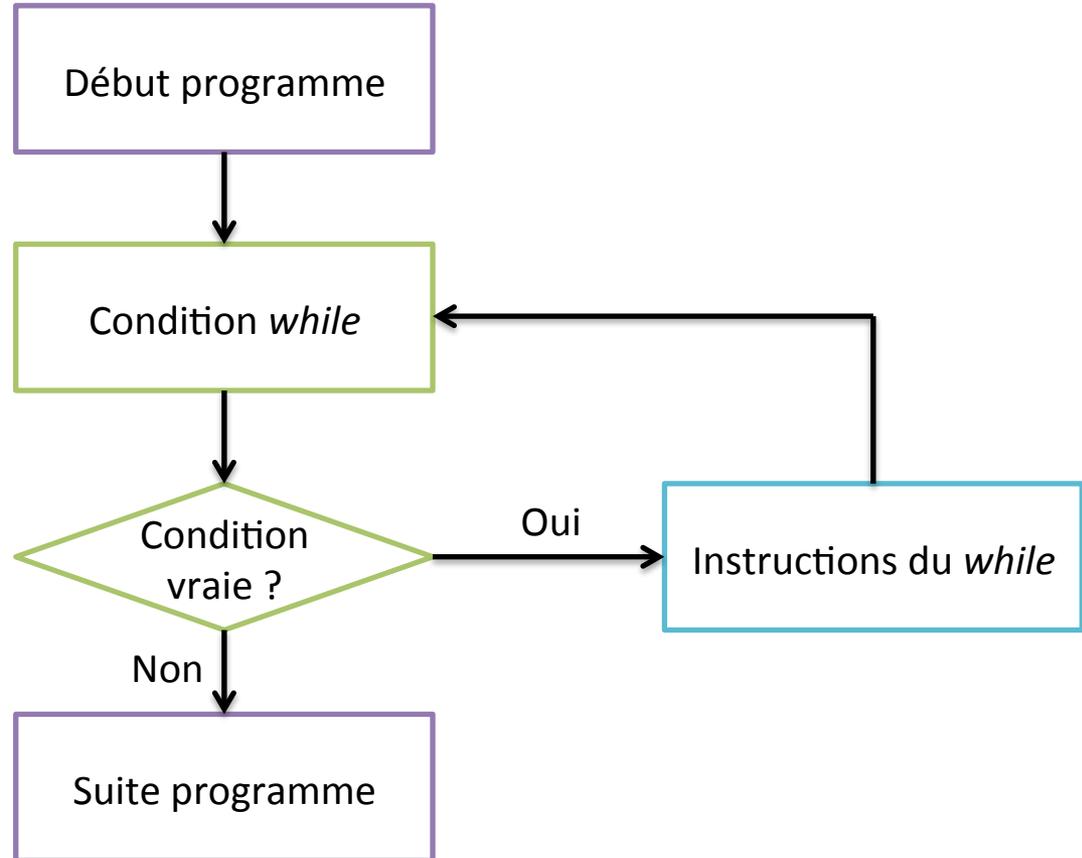
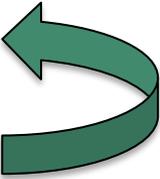
```

if ($v > 15){
    echo "Très bien";
}
elseif ($v > 12.5){
    echo "Bien";
}
elseif ($v > 10){
    echo "Passable";
}
else{
    echo "Insuffisant";
}
Suite du programme
...
    
```



Structures de contrôle

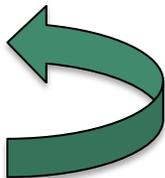
```
...  
$i = 0;  
while ($i < 100){  
    echo $i;  
    $i++;  
}  
echo "On est sorti de la boucle !";  
...
```



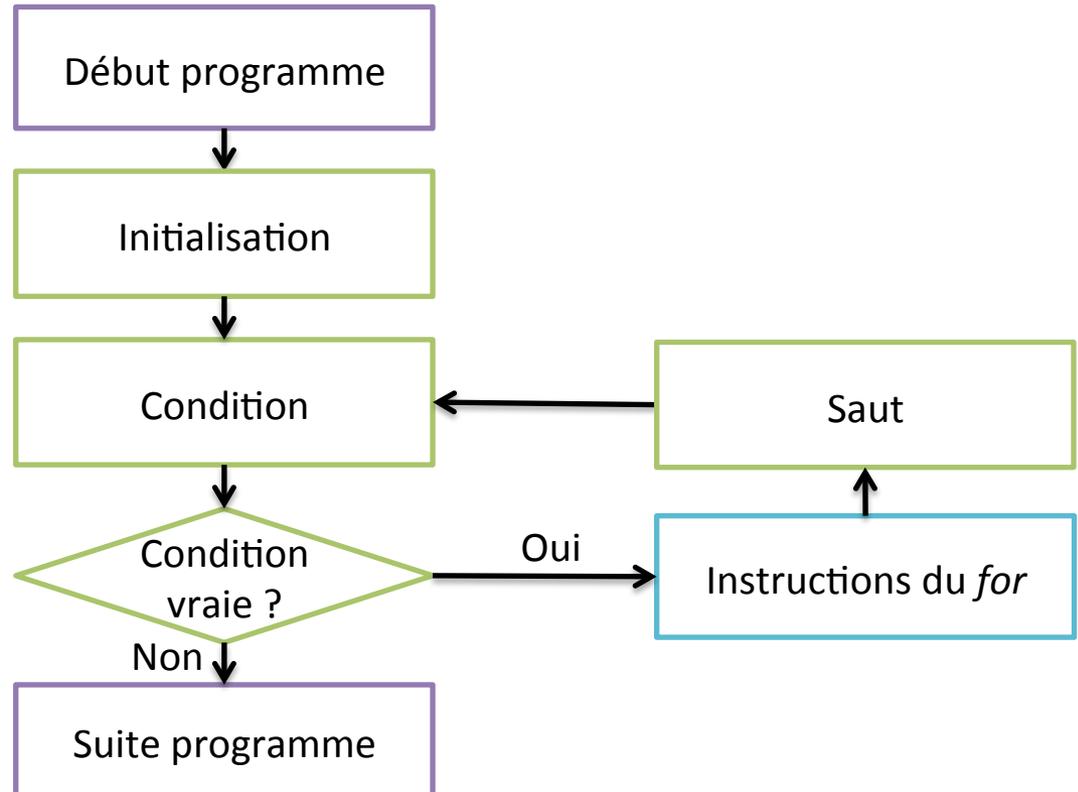
Structures de contrôle

```
...
$i = 0;
while ($i < 100){
    echo $i;
    $i++;
}
echo "On est sorti de la boucle !";
...
-----
...

```



```
...
for ($i=0 ; $i < 100 ; $i++){
    echo $i;
}
echo "On est sorti de la boucle !";
...
-----
for(Initialisation ; Condition ; Saut)
```

Plan

- Les principaux langages du Web
 - Côté client
 - Côté serveur
 - Autres
 - Récapitulatif
- **PHP**
 - Introduction
 - Types de données
 - Opérateurs
 - Structures de contrôle
 - **Commentaires**
 - Fichiers CSV

Commentaires

- Ligne commençant par **//**

```
// Cette phrase est un commentaire.  
// Cette phrase aussi.  
echo "Celle-là ne l'est pas";
```
- Ensemble de lignes encadrées par **/*** et ***/**

```
/*  
Cette phrase est un commentaire.  
Cette phrase aussi.  
*/  
echo "Celle-là ne l'est pas";
```

Plan

- Les principaux langages du Web
 - Côté client
 - Côté serveur
 - Autres
 - Récapitulatif
- **PHP**
 - Introduction
 - Types de données
 - Opérateurs
 - Structures de contrôle
 - Commentaires
 - **Fichiers CSV**

Fichiers CSV

- Lire un fichier CSV contenant 3 colonnes :

The screenshot shows the Microsoft Excel interface with a CSV file named 'articles.csv' open. The ribbon is set to 'Tableaux' (Tables). The data is displayed in a table with 7 rows and 7 columns (A-F). The first four rows contain data, and the last three rows are empty.

	A	B	C	D	E	F
1		1 Crime et chatiment	10			
2		2 Mme Bovary	2			
3		3 Les diaboliques	3			
4		4 Le rouge et le noir	12			
5						
6						
7						

Fichiers CSV

- Lire un fichier CSV contenant 3 colonnes :

```
ini_set('auto_detect_line_endings', TRUE);  
if (($file = fopen("monFichier.csv", "r")) != FALSE) {  
    while (($tmp= fgetcsv($file, 0, ";")) != FALSE) {  
        echo $tmp[0]." et ".$tmp[1]." et ".$tmp[2];  
    }  
    fclose($file);  
}
```

- `$file = fopen("monFichier.csv", "r")` ouvre le fichier et le place dans la variable `$file`
- `while (($tmp= fgetcsv($file, 0, ";")) != FALSE) {}` lit chaque ligne du fichier et place les valeurs séparées par des `;` dans un tableau nommé `tmp`.