

# PHP

## Serveur



## Client



# PHP

## Serveur



## Client



# PHP Hypertext Preprocessor

Langage compilé à la volée



- Fabrication de contenu (code HTML, CSS, ...) sur le serveur
- Envoie du code généré au client

Un *indispensable* du développement web

# Exemple de code PHP

```
<h1>Série d'attaques des noobs :</h1>
<?php
    echo '<ul>';
    for ($i = 0 ; $i <= 4 ; $i++) {
        echo '<li>Hadoken ! ('.$i.')</li>';
    }
    echo '</ul>';
```

Série d'attaques des noobs :

- Hadoken (0)
- Hadoken (1)
- Hadoken (2)
- Hadoken (3)
- Hadoken (4)

Une page HTML peut être intégralement générée par du PHP !



# Les types

## Les entiers

-12    5    42    0

```
$var = 3;  
$var = $var - 5;  
$var += 1;
```

## Les Réels

-12.2    5.0    0.5  
42.42

```
$var = 3.1;  
$var = $var - 5.0;  
$var += 1.5;
```

## Les Booléens

Faux	Vrai
false 0 ... ...	true 1 "toto" ...

```
$var = !0;  
$var = false && 1;  
$var = 1 or 0;
```

Recommandation :  
Utiliser true et false

## Les chaînes de caractères

Codé par des guillemets

```
$var = "hello";  
echo "hello ".$var."  
echo 'hello '$var."  
echo 'long : 'strlen($var).'  
';
```

## Les tableaux

### Usuels



### Associatifs



```
<?php  
$x = 3; // entier 3  
$y = $x - 3; /* entier (0) */  
$x = 'toto'; # string ('toto')  
$y = '3' - 1; // entier (2)  
?>
```

Typage dynamique !

comme python  
pas comme java

# Les entiers

-12      5      42      0

```
$var = 3;  
$var = $var - 5;  
$var += 1;
```

# Les Réels

-12.2

5.0

0.5

42.42

```
$var = 3.1;  
$var = $var - 5.0;  
$var += 1.5;
```

# Les Booléens

Faux

false

0

""

...

Vrai

true

1

"toto"

...

```
$var = !0;  
$var = false && 1;  
$var = 1 or 0;
```

Recommandation :  
Utiliser true et false

# Les chaînes de caractères

Codé par des guillemets

```
$var = "hello";
echo "hello ".$var."<br/>";
echo 'hello '.$var.'<br/>';

echo 'long : '.strlen($var).<br/>';
```

# Les tableaux

## Usuels

Indexé par des entiers

```
$tab = array();
$tab = array(1, 'toto', 0.5);
echo $tab[1]; // affiche toto
$tab[0] = 'titi';
$tab[12] = 'RTFM !';
```

Extensible mais pas forcément continu

```
$tab = array(1, 'toto', 0.5);
echo count($tab); // affiche 3
$tab[] = 42; // ajoute 42 à la fin
unset($tab[0]) // supprime 1
```

## Associatifs

Indexé par des chaînes de caractères

```
$tab = array('Matt' => 'Smith', 'David' => 'Tennant');

echo $tab['Matt'] // affiche Smith

$tab['Peter'] = 'Capaldi';
```

Même chose que pour le tableau usuel ... mais avec des chaînes de caractères

# Indexé par des entiers

```
$tab = array();  
$tab = array(1, 'toto', 0.5);  
echo $tab[1]; // affiche toto  
$tab[0] = 'titi';  
$tab[12] = 'RTFM !';
```

Extensible mais pas forcément continu

```
$tab = array(1, 'toto', 0.5);  
echo count($tab); // affiche 3  
$tab[] = 42; // ajoute 42 à la fin  
unset($tab[0]) // supprime 1
```

# Indexé par des chaînes de caractères

```
$tab = array('Matt' => 'Smith', 'David' => 'Tennant');
```

```
echo $tab['Matt'] // affiche Smith
```

```
$tab['Peter'] = 'Capaldi';
```

Même chose que pour le tableau usuel ... mais  
avec des chaînes de caractères



```
<?php  
    $x = 3;           // entier 3  
    $y = $x - 3;     /* entier (0) */  
    $x = 'toto';      # string ('toto')  
    $y = '3' - 1;     // entier (2)  
?>
```

Typage dynamique !

comme python  
pas comme java

# Les boucles

## Boucle while

```
while (condition){  
    instructions;  
}
```

```
$tab = array();  
$i = 0;  
while ($i < 5){  
    $tab[] = $i*2;  
    $i++;  
}
```

Contenu de \$tab :  
0, 2, 4, 6, 8

## Boucle For

```
for ($i = 0 ; $i < count($t); $i ++){  
    instructions;  
}
```

```
$tab = array();  
for ($i = 0 ; $i < count($t); $i ++){  
    $tab[] = $i*2;  
}
```

Contenu de \$tab :  
0, 2, 4, 6, 8

## Boucle Foreach

```
foreach (tableau as élément){  
    instructions;  
}
```

```
$tab = array(0,2,3,4);  
foreach ($tab as $val){  
    echo $val<br/>;  
}
```

0  
1  
2  
3  
4

## Définitions entre les boucles

```
$tab = array();  
$tab[0] = 1;  
$tab[1] = 2;  
$tab[3] = 3;  
$compteur = 0;
```

```
for ($i = 0 ; $i < count($tab) ; i++){  
    $compteur += $tab[$i];  
}  
echo $compteur;
```

```
$i = 0;  
while ($i < count($tab)){  
    $compteur += $tab[$i];  
    $i++;  
}  
echo $compteur;
```

```
foreach ($tab as $val){  
    $compteur += $val;  
}  
echo $compteur;
```



# Boucle while

```
while (condition){  
    instructions;  
}
```

```
$tab = array();  
$i = 0;  
while ($i < 5){  
    $tab[] = $i*2;  
    $i++;  
}
```

Contenu de \$tab :  
0, 2, 4, 6, 8



# Boucle For

```
for ($i = 0 ; $i < count($t); $i ++){  
    instructions;  
}
```

```
$tab = array();  
for ($i = 0 ; $i < count($t); $i ++){  
    $tab[] = $i*2;  
}
```

Contenu de \$tab :  
0, 2, 4, 6, 8



# Boucle Foreach

```
foreach (tableau as élément){  
    instructions;  
}
```

```
$tab = array(0,2,3,4);  
foreach ($tab as $val){  
    echo $val.'<br/>';  
}
```

0  
1  
2  
3  
4

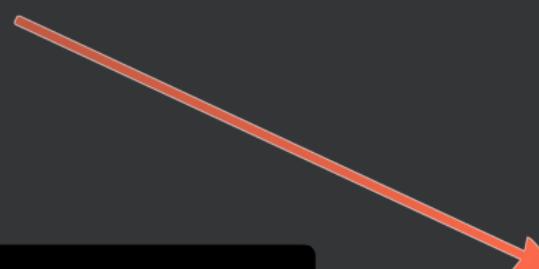


# Différences entre les boucles

```
$tab = array();  
$tab[0] = 1;  
$tab[1] = 2;  
$tab[3] = 3;  
$compteur = 0;
```



```
for ($i = 0 ; $i < count($tab) ; i++){  
    $compteur += $tab[$i];  
}  
echo $compteur;
```



```
$i = 0;  
while ($i < count($tab)){  
    $compteur += $tab[$i];  
    $i++;  
}  
echo $compteur;
```

```
foreach ($tab as $val){  
    $compteur += $val;  
}  
echo $compteur;
```

# Génération de code

```
$abonnements = ('Dirty Biology' => 421834,
                 'Sissy MUA' => 662515,
                 'Taupe10' => 1167541,
                 'Dr Nozman' => 1030732);
echo '<p>Youtubeurs présents au Play Azur Festival</p>';
echo '<ul>';
foreach($abonnement => $youtubeur as $nbAbonnes){
    echo '<li>' . $youtubeur . ' (' . $nbAbonnes . ' abonnés)</li>';
}
echo '</ul>';
```



# Les fonctions / procédures

Grouper une série d'instructions permettant d'effectuer une tâche

```
function nomFonction(arguments) {  
    instructions;  
    return data;  
}
```

ex : une fonction qui génère une liste html à partir d'un tableau



# Bonnes pratiques en PHP

## répertoire du TP

```
└── index.php  
└── img/  
└── css/  
└── js/  
└── php/  
    └── data/  
        └── html/
```

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="utf-8" />  
    <link href="css/style.css" />  
  </head>  
  <body>  
    <?php  
      include_once('php/html/file.php');  
    ?>  
    <title>Titre</title>  
  </body>  
</html>
```

- Contient les fichiers permettant de
- générer des données
  - lire des données externes
  - faire des calculs
  - modifier des données

Les fonctions contenu dans ce répertoire ne rendent que des données (pas de code HTML)

- Contient les fichiers permettant de
- d'afficher les données
  - générer des bouts de code HTML

Les fonctions contenu dans ce répertoire ne font qu'afficher des données en générant du code HTML

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <link src="css/style.css" />
    <?php
      include_once("php/html/file.php");
    ?>
    <title>Titre</title>
  </head>
  <body>
    <?php
      $liste = (1,2,3,4,5);
      genereList($liste);
    ?>
    </body>
  </html>
```



# data/

Contient les fichiers permettant de

- générer des données
- lire des données externes
- faire des calculs
- modifier des données

Les fonctions contenu dans ce répertoire ne rendent que des données (pas de code HTML)

Les fonctions contenu dans ce répertoire ne rendent que des données (pas de code HTML)

# html/

Contient les fichiers permettant de

- d'afficher les données
- générer des bouts de code HTML

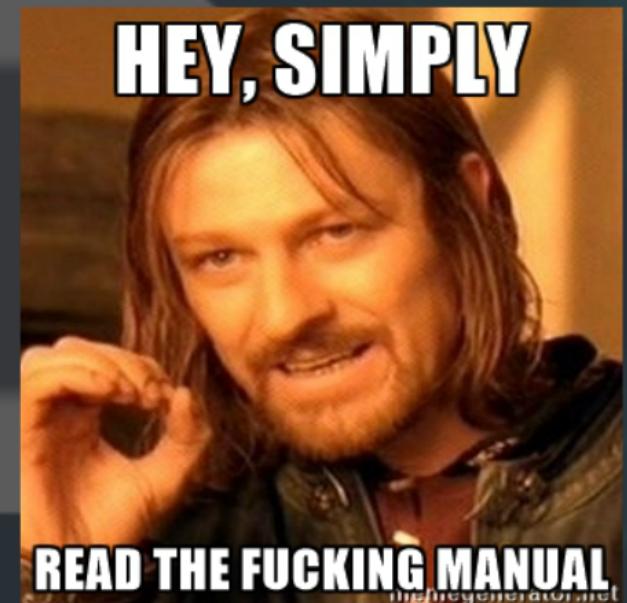
Les fonctions contenu dans ce répertoire ne font qu'afficher des données en générant du code HTML

# En plus !

## Utiliser des fonctions

Quand vous avez un bug :

- 1 - Localiser le problème
- 2 - Regarder la doc  
([php.net/manual/fr/](http://php.net/manual/fr/))
- 3 - Appelez l'enseignant



Si  $n-1$  n'est pas fait, ne pas faire  $n$