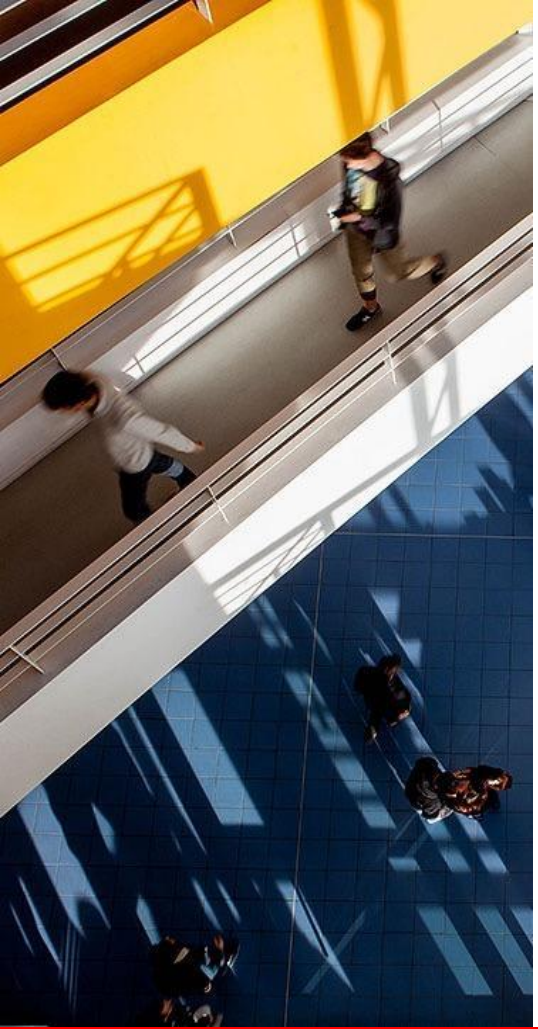

Développement web en PHP - Opérateurs, tableaux et structures de contrôle -

Groupe des étudiants : CIR2

Ayoub KARINE
E-Mail : ayoub.karine@isen-ouest.yncrea.fr
Numéro de bureau : A3-78

Opérateurs

Types d'opérateurs

- Les opérateurs arithmétiques :
  - **+** : addition ;
  - **-** : soustraction ;
  - ***** : multiplication ;
  - **/** : division ;
  - **%** : modulo ;
  - ****** : exponentiation.

Types d'opérateurs

- Les opérateurs arithmétiques :

- Les opérateurs d’affectation :
  - **=** : affectation ;
  - **+=** : addition combinée à une affectation ;
  - **-=** : soustraction combinée à une affectation ;
  - ***=** : multiplication combinée à une affectation ;
  - **/=** : division combinée à une affectation ;
  - **%**= : modulo combiné à une affectation.

Types d'opérateurs

- Les opérateurs arithmétiques :
- Les opérateurs d’affectation :
- Les opérateurs de comparaison :
  - **==** : égalité ;
  - **===** : identité (égalité et même type) ;
  - **!=** et **<>** : non-égalité ;
  - **!==** : non-identité ;
  - **>** et **<** : strictement supérieur et inférieur ;
  - **>=** et **<=** : supérieur ou égal et inférieur ou égal.

Types d'opérateurs

- Les opérateurs arithmétiques :
- Les opérateurs d’affectation :
- Les opérateurs de comparaison :
- Les opérateurs d’incrémentation / décrémentation :
  - **++$var** : pré incrémentation (avant évaluation) ;
  - **$var++** : post incrémentation (après évaluation) ;
  - **--$var** : pré décrémentation (avant évaluation) ;
  - **$var--** : post décrémentation (après évaluation).

Types d'opérateurs

- Les opérateurs arithmétiques :
- Les opérateurs d’affectation :
- Les opérateurs de comparaison :
- Les opérateurs d’incrémentation / décrémentation :
- Les opérateurs logiques :
  - **&&** et **and** : ET logique ;
  - **||** et **or** : OU logique ;
  - **xor** : OU exclusif ;
  - **!** : NON logique.

Types d'opérateurs

- Les opérateurs arithmétiques :
- Les opérateurs d’affectation :
- Les opérateurs de comparaison :
- Les opérateurs d’incrémentation / décrémentation :
- Les opérateurs logiques :
- Les opérateurs sur les chaînes de caractères

Tableaux

Tableaux indicés vs associatifs

- Le tableau PHP regroupe en un seul type, array, les deux notions suivantes :
  - **Les tableaux indicés** : indicé par des entiers positifs (tpe de la clé : entier) ;
    - Déclaration

```php
$tab = array('apple', 'juice');
print_r($tab); // Affiche : Array([0]=>apple [1]=>juice)
```

Tableaux indicés vs associatifs

- Le tableau PHP regroupe en un seul type, array, les deux notions suivantes :
  - **Les tableaux indicés** : indicé par des entiers positifs (tpe de la clé : entier) ;
    - Déclaration

```php
$tab = array('apple', 'juice');
print_r($tab); // Affiche : Array([0]=>apple [1]=>juice)
```

  - **Les tableaux associatifs** : indicé par des chaînes de caractères (type de la clé : chaîne)
    - Déclaration

```php
$tab =  array('Paul' => '01 23 45 67 89',
              'Virginie' => '06 05 04 03 02',
              'Pierre' => 'unknown');
print_r($tab); // Affiche :
               // Array([Paul]=>01 23 45 67 89
               //       [Virginie]=>06 05 04 03 02
               //       [Pierre]=>unknown)
```

Affectation

- Affectation à un indice donné :

```php
$tab[0] = 4;
$tab['Jacques'] = '01 23 45 67 89';
$tab[$i] = 'truc';
```

- Affectation sans indice (ajout en fin du tableau) :

```php
$tab[] = 'pear';
```

Accès aux éléments

- Avec un indice :

```php
echo $tab[1]; // OK.
```

- Avec une clé constante :

```php
echo $tab['my_key']; // OK.
echo "Value: $tab['my_key']"; // Faux
echo "Value: {$tab['my_key']}"; // OK.
echo "Value: ".$tab['my_key']; // OK.
```

- Avec une clé variable :

```php
$i = 'my_key';
echo $tab['$i']; // Faux
echo $tab[$i]; // OK.
```

Suppression

- Suppression d'un élément (utilisation de la fonction unset) :

```php
$tab = array('a', 'b', 'c');
unset($tab[1]); // Attention, le tableau n'est pas ré-indexé
print_r($tab); // Affiche : Array([0]=>a [2]=>c)
```

- Suppression de tous les éléments :

```php
$tab = array();
```

- Suppression d'un tableau (utilisation de la fonction unset):

```php
unset($tab);
```

Structures de contrôles

Structure conditionnelle : If Else ElseIf

```
if (condition ) { //condition vraie
  instructions;
} else {//condition fausse
  instructions;
}
```

```
if ( condition ) {
  instructions;
}
elseif  (condition) {
  instructions;
}
else {
  instructions;
}
```

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

Structure conditionnelle : Opérateur ternaire

(condition) ? Instruction si vrai : instruction si faux

Exemple :
$note > 15 ? $mention = ‘très bien’ : $mention = ‘bof’ ;

Structure conditionnelle : Switch-case

```
switch ( variable ) {
  case valeur1 :
    Instructions_1;
    break ;
  case valeur2 :
    Instructions_2;
    break ;
...
  default :
    Instructions_x;
    break ;
}
```

```
$nb = mt_rand(0,4) ;
switch ( $nb ) {
  case 4 :
    echo ''$nb sup à 3 <br>'' ;
  case 3 :
    echo ''$nb sup à 2 <br>'' ;
  case 2 :
    echo ''$nb sup à 1 <br>'' ;
  case 1 :
    echo ''$nb sup à 0 <br>'' ;
    break ;
  default :
     echo ''$nb égal à 0 <br>'' ;
    break ;
}
```

Boucles : While & Do While

- While

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

- Do While

```
do {
  Instructions;
} while ( condition )
```

Boucle For

- For :
  - Syntaxe :

```
for (init counter; test counter; increment counter) {
    // Code à exécuter.
}
```

  - Exemple :

```
for ($i = 0; $i < 10; $i++) {
  echo "The number is: $i";
}
```

- Foreach :
  - Syntaxe :

```
foreach ($array as $value) {
  // Code à exécuter.
}
```

  - Exemples :

```
foreach ($tab as $value) {
  echo "Value: $value";
}
```

```
foreach ($tab as $key => $value) {
  echo "Key: $key, value: $value";
}
```

Instruction particulière

- Les instructions **continue** et **break** permettent de réaliser un arrêt partiel ou total d'une boucle
  - **continue** : continuer l'itération d'une boucle sans réaliser la suite de traitement

```
<?php
for($i=1; $i<=10; $i++)
{
        if($i%3==0)
          {
            continue;
          }
echo $i.'<br>';
}
?>
```

```
1
2
4
5
7
8
10
```

Instruction particulière

- Les instructions **continue** et **break** permettent de réaliser un arrêt partiel ou total d'une boucle
  - **continue** : continuer l'itération d'une boucle sans réaliser la suite de traitement
  - **break** : sortir d'une structure conditionnelle

```
<?php
for($i=1; $i<=10; $i++)
{
        if($i%3==0){
                break;
        }
        echo $i.'<br>';
}
?>
```

1
2