Les structures conditionnelles en PHP

Contenu

- **Les booléens**
- **Les opérateurs de comparaison**
- **Les conditions (*if – else if – else*)**
- **Les opérateurs logiques**

Les booléens

Types de variables

- **Quels types de variables avons-nous vus au dernier cours?**

Le type booléen

- **Autre type de variable: booléen (*bool*)**
  - Deux valeurs possibles: true (vrai) ou false (faux)

```
$jaiFaim = true;

$ilPleut = false;
```

Les opérateurs de comparaison

Comparaison

- **Chaque énoncé ci-dessous est soit vrai, soit faux:**
  - **37** est plus petit que **52**
  - **$a** est égal à **$b**
  - **$a** est plus grand que **$b**
  - **$a** est plus grand ou égal à **$b**
  - **$a** est plus petit ou égal à **$b**
  - **$a** est différent de **$b**

Opérateurs de comparaison

==	est égal à
<	est plus petit que
>	est plus grand que
<=	est plus petit ou égal à
>=	est plus grand ou égal à
!=	est différent de

Comparaison et booléens

- **La comparaison de deux variables produit (retourne) un booléen**
  - Il s'agit donc d'une expression booléenne

```
$a = 27;
$b = 31;

$estEgal = $a == $b;
$estPlusPetit = $a < $b;
$estPlusGrand = $a > $b;
$estPlusPetitOuEgal = $a <= $b;
$estPlusGrandOuEgal = $a >= $b;
$estDifferent = $a != $b;
```

Comparer des chaînes de caractères

```
"Bonjour" == "Bonjour"; // true
"Bonjour" == "Au revoir"; // false
"Bonjour" < "Au revoir"; // false
"Bonjour" > "Au revoir"; // true
```

Opérateurs === et !==

- **Il existe aussi un opérateur « === »**
  - Est égal et de même type

```
17 == "17"; // true
17 === "17"; // false
```

- **Opérateur correspondant « !== »**

```
17 != "17"; // false
17 !== "17"; // true
```

Les conditions

Les conditions (*if*)

- **En programmation, les conditions permettent d'exécuter des instructions seulement si une expression boléenne est vrai**
  - Si *ceci* est vrai, fais *cela*

- **Syntaxe:**

```
if (expression booléenne) {
    Instructions à exécuter si l'expression est vraie
}
```

Les conditions (*if*)

```php
$limiteVitesse = 50;
$vitesseConducteur = 60;

if ($vitesseConducteur > $limiteVitesse) {
    echo 'Vous allez trop vite!';
    $donnerTicket = true;
}
```

Les conditions (*else*)

- **On peut utiliser *else* (sinon) pour exécuter des instructions différentes si la condition est fausse**
  - Si *ceci*, fais *cela*. Sinon, fais *plutôt cela*.

- **Syntaxe:**

```
if (expression booléenne) {
    Instructions à exécuter si l'expression est vraie
} else {
    Instructions à exécuter si l'expression est fausse
}
```

Les conditions (*else*)

```
if ($vitesseConducteur > $limiteVitesse) {
    echo 'Vous allez trop vite!';
    $donnerTicket = true;
} else {
    $donnerTicket = false;
}
```

Les conditions (*else if*)

- **On peut utiliser *else if* (sinon si) pour exécuter des instructions différentes si la condition est fausse, MAIS qu'une autre condition est vraie**

- **Syntaxe:**

```
if (expression booléenne #1) {
    Instructions à exécuter si l'expression #1 est vraie
} else if (expression booléenne #2)  {
    Instructions à exécuter si l'expression #1 est fausse
    MAIS que l'expression #2 est vraie
}
```

Les conditions (*else if*)

```
if ($vitesseConducteur > 100) {
    echo 'Vous allez trop vite!';
    $donnerTicket = true;
} else if ($vitesseConducteur < 60) {
    echo 'Vous allez trop lentement!';
    $donnerTicket = true;
}
```

Les conditions (*if - else if - else*)

```
if ($vitesseConducteur > 100) {
    echo 'Vous allez trop vite!';
    $donnerTicket = true;
} else if ($vitesseConducteur < 60) {
    echo 'Vous allez trop lentement!';
    $donnerTicket = true;
} else {
    $donnerTicket = false;
}
```

Conditions imbriquées

- **On peut imbriquer des structures conditionnelles**

```
if ($jaiFaim) {
    if ($jaiUnLunch) {
        echo 'Je mange mon lunch.';
    } else {
        echo 'Je vais au restaurant.';
    }
}
```

Les opérateurs logiques

Les opérateurs logiques

- **Les opérateurs logiques en programmation permettent de créer des expressions booléennes plus complexes**
- **Il en existe quatre en PHP**
  - ET (**&&**)
  - OU (**||**)
  - OU EXCLUSIF (**xor**)
  - PAS (**!**)

ET

- **L'opérateur ET (&&) retourne vrai seulement si les deux expressions sont vraies**

```
if ($jaiFaim && $jaiUnLunch) {
    echo "J'ai faim et j'ai un lunch, donc je mange mon lunch.";
} else {
    echo "Soit je n'ai pas faim, soit je n'ai pas de lunch, soit les deux.";
}
```

OU

- **L'opérateur OU (||) retourne vrai si *au moins une* des deux expressions est vraie**

```
if ($jaiFaim || $ilEstMidi) {
    echo "Je mange, soit parce que j'ai faim, soit parce qu'il est midi,
          soit les deux.";
} else {
    echo "Je ne mange pas car je n'ai pas faim et il n'est pas midi.";
}
```

Table de vérité (ET / OU)

$a	$b	$a && $b	$a || $b
false	false	false	false
false	true	false	true
true	false	false	true
true	true	true	true

and / or

- Les opérateurs « && » et « || » peuvent aussi s’écrire « and » et « or » en PHP
- « && » et « || » sont utilisés dans beaucoup d’autres langages (dont JavaScript)

OU EXCLUSIF

- **L'opérateur OU EXCLUSIF (xor) retourne vrai si une seule des deux expressions est vraie**
- **C'est un opérateur qu'on voit très rarement et qui n'existe pas dans la plupart des langages de programmation courants (mais il existe en PHP)**

$a	$b	$a xor $b
false	false	false
false	true	true
true	false	true
true	true	false

Opérateur !

- **L'opérateur PAS (!) permet d'inverser une expression booléenne**

```
$a = true;
$b = !$a; // $b est false

$c = !(10 < 20); // $c est false

$d = !($a || $b); // $d est false
```

Fin de la présentation

Des questions?

Photo par Jules Bss sur Unsplash