PHP

Les conditions

Dans ce module

- Les conditions simple
- Les conditions par défaut
- Les conditions multiples
- Alternative d’écriture des conditions
- Les conditions ternaires
- Imbrications
- Opérateurs de comparaisons
- Opérateurs logiques
- Pratique

Condition simple

- Il est possible d'applique des conditions à l'exécution d'un code ou bloc de code.
- Le code est dans des accolades {} qui suit la condition.
- Si la CONDITION est vraie le code dans des accolades {} est exécuté.
- Si la CONDITION est fausse le code dans des accolades {} est ignoré.
- L'instruction (mot clé) pour réalisé une condition est « if »

```
if (CONDITION) {
    // CODE if CONDITION true
}
```

Forme normalisée

```
if (CONDITION):
    // CODE if CONDITION true
endif
```

Forme alternative

Condition simple - Exemple

```php
$phoneIsColored = true;
$phoneColor = null;

if ($phoneIsColored) {
    echo 'Le smartphone est coloré.';
    $phoneColor = '#336BFF';
}
```

Condition simple par défaut

- Il est possible d'ajouter un code qui ne s'exécute que si la condition est fausse.
- Attention, un seul des codes sera exécuté suivant la CONDITION (VRAI -> IF ou FAUX -> ELSE)
- « else » est optionnel.

```
if (CONDITION) {
    // CODE if CONDITION true
} else {
    // CODE if CONDITION false
}
```

Forme normalisée

```
if (CONDITION):
    // CODE if CONDITION true
else :
    // CODE if CONDITION false
endif
```

Forme alternative

Condition simple par défaut - Exemple

```php
$phoneIsColored = true;
$phoneColor = null;

if ($phoneIsColored) {
    echo "Le smartphone est coloré.";
    $phoneColor = '#336BFF'; // bleu
} else {
    echo "Le smartphone n'est pas coloré.";
    $phoneColor = '#000000'; // black
}
```

Condition multiple

- Il est possible d'ajouter plusieurs conditions qui exécutent un code différent.
- Les conditions sont évalué/testé/examiné une par une de haut en bas. La première à être vraie lance l'exécution de son code et toutes les autres instructions (« elseif » et « else ») qui suivent sont ignorées.
- « elseif » est optionnel et peut être utilisé plusieurs fois dans un même bloc de conditions.

```
if (CONDITION_1) {
    // CODE if CONDITION_1 true
} elseif (CONDITION_2) {
    // CODE if CONDITION_2 true
} else {
    // CODE if CONDITION false
}
```

Forme normalisée

```
if (CONDITION_1):
    // CODE if CONDITION_1 true
elseif (CONDITION_2):
    // CODE if CONDITION_2 true
else :
    // CODE if CONDITION false
endif
```

Forme alternative

Condition multiple - Exemple

```php
$phoneIsColored = true;
$phoneIsSpecialEdition = false;
$phoneColor = null;

if ($phoneIsSpecialEdition) {
    echo "Le smartphone est coloré en édition spéciale.";
    $phoneColor = '#F5B01E'; // or
} elseif ($phoneIsColored) {
    echo "Le smartphone est coloré.";
    $phoneColor = '#336BFF'; // bleu
} else {
    echo "Le smartphone n'est pas coloré.";
    $phoneColor = '#000000'; // black
}
```

Alternative d’écriture des conditions

- Il est possible de retirer les accolades { } d’une condition qui ne possède qu’une ligne de code.
- Utilisation peu commune et moyenne apprécié.

```
if (CONDITION_1)
    // 1 line CODE if CONDITION_1 true
elseif (CONDITION_2)
    // 1 line CODE if CONDITION_2 true
else
    // 1 line CODE if CONDITION false
```

Alternative d’écriture des conditions - Exemple

```php
$phoneIsColored = true;
$phoneIsSpecialEdition = false;
$phoneColor = null;

if ($phoneIsSpecialEdition)
    $phoneColor = '#F5B01E'; // or
elseif ($phoneIsColored)
    $phoneColor = '#336BFF'; // bleu
else
    $phoneColor = '#000000'; // black
```

Condition ternaire

- Il est existe une simplifié d'une condition « IF … ELSE … ».
- Utilisé pour des conditions simples et courtes dans le cas d'une attribution ou modification de variables.
- A le grand avantage de pouvoir être utilisé un peu partout et dans des imbrications complexes de codes où une condition normale n'est pas plaçable.

```php
// Attribution d'une variable
$var = CONDITION ? VAR_IF_TRUE : VAR_IF_FLASE;

// Injection dans les paramètres d'une fonction
FUNCTION(CONDITION ? VALUE_IF_TRUE : VALUE_IF_FLASE);
```

Condition ternaire - Exemples

```php
$phoneIsColored = true;
$phoneColor = null;

$phoneColor = $phoneIsColored ? '#336BFF' : '#000000';
```

```php
$phoneIsColored = true;
$phoneColor = null;

echo $phoneIsColored ? '#336BFF' : '#000000';
var_dump($phoneIsColored ? '#336BFF' : '#000000');
```

```php
$phoneIsColored = true;
$phoneColor = null;

echo $phoneIsColored
    ? "Le smartphone est coloré."
    : "Le smartphone n'est pas coloré.";
```

```php
$phoneIsColored = true;
$phoneColor = null;

if ($phoneIsColored) {
    $phoneColor = '#336BFF';
} else {
    $phoneColor = '#000000';
}
```

Condition au format simple

Imbrications

```
if (CONDITION) {
    if (CONDITION) {
        if (CONDITION) {
            // ...
        } else {
            if (CONDITION) {
                // ...
            }
        }
    }
} elseif (CONDITION) {
    if (CONDITION) {
        // ...
    } else {
        // ...
    }
} else {
    // ...
}
```

Opérateurs de comparaison

Exemple	Nom	Résultat
$var1 == $var2	Egal	**true** si *$var1* est égal à *$var2* (inclus des conversions automatisées)
$var1 === $var2	Identique	**true** si *$var1* est égal à *$var2* et qu'ils sont de même type (pas de conversions automatisées)
$var1 != $var2	Différent	**true** si *$var1* est différent de *$var1* (inclus des conversions automatisées)
$var1 < $var2	Plus petit que	**true** si *$var1* est strictement plus petit que *$var2*
$var1 > $var2	Plus grand que	**true** si *$var1* est strictement plus grand que *$var2*
$var1 <= $var2	Plus petit ou égal	**true** si *$var1* est plus petit ou égal à *$var2*
$var1 >= $var2	Plus grand ou égal	**true** si *$var1* est plus grand ou égal à *$var2*
! $var1	Inversion	**true** si *$var1* est fausse. **false** si *$var2* est vrai

Opérateurs logiques

Exemple	Nom	Résultat
CONDITION_1 and CONDITION_2	ET	**TRUE** si CONDITION_1 ET CONDITION_2 sont tous les deux **TRUE**
CONDITION_1 && CONDITION_2	ET	idem
CONDITION_1 or CONDITION_2	OU	**TRUE** si au moins CONDITION_1 OU CONDITION_2 est **TRUE**
CONDITION_1 || CONDITION_2	OU	idem

Pratique

1. Total d'un achat : Créer 3 variables (un prix unitaire HT, un taux de TVA et un nombre d'articles) puis calculer le montant TTC de l'achat.
2. Conversion en date : Créer une variable contenant une durée en secondes, puis afficher la en heures, minutes, secondes. Par exemple « $time = 12334 » affichera « 3 heures, 25 minutes et 34 secondes ». Chercher l'utilité de la fonction « time() » en PHP et afficher le résultat en heures, minutes et secondes.
3. Condition de remise : Un commerçant accorde une remise de 5 % pour tout achat d'un montant compris entre 100 et 500 € et 15 % au-delà. Écrire un programme de calcul du montant de la remise sur un achat donné.
4. Message par heure : Le but est d'afficher un message différent selon l'heure de la journée. N'oublier pas de tester toutes vos conditions avec une variable. Pour récupérer l'heure actuelle utilisez « date('G') », pour les minutes « date('i') ». Condition à mettre en place :
   - *Heure sous 12h : Bonne matinée*
   - *Heure sous 13h30 : Bonne appétit*
   - *Heure sous 18h : Bonne après-midi*
   - *Heure sous 23h : Bonne soirée*
   - *Heure sous 5h : Bonne nuit*
5. Il existe encore une autre forme de condition. Chercher l'utilité des « switch » en PHP.
6. Avec PHP 8, il existe une dernière forme de condition. Chercher l'utilité des « match » en PHP.

PHP

Les conditions

Fin du module