

Instructions PHP If ...Else

Dans ce chapitre, vous apprendrez à écrire du code décisionnel à l'aide des instructions if...else...elseif en PHP.

Instructions conditionnelles PHP

Comme la plupart des langages de programmation, PHP vous permet également d'écrire du code qui exécute différentes actions en fonction des résultats d'un test logique ou comparatif au moment de l'exécution. Cela signifie que vous pouvez créer des conditions de test sous la forme d'expressions évaluées à **true** ou **false** et sur la base de ces résultats, vous pouvez effectuer certaines actions.

Il existe plusieurs instructions en PHP que vous pouvez utiliser pour prendre des décisions :

- L' instruction **if**
- L' instruction **if...else**
- L' instruction **if...elseif...else**
- L' instruction **switch...case**

Nous allons explorer chacune de ces déclarations dans les prochaines sections.

La déclaration if

L'instruction **if** est utilisée pour exécuter un bloc de code uniquement si la condition spécifiée est vraie. Ce sont les instructions conditionnelles les plus simples de PHP et peuvent être écrites comme suit :

```
if (condition){  
    // Code à exécuter  
}
```

L'exemple suivant affichera "Have a nice weekend!" si le jour en cours est vendredi :

```
<?php  
$d = date("D");  
if($d == "Fri"){  
    echo "Have a nice weekend!";  
}  
?>
```

La déclaration if...else

Vous pouvez améliorer le processus de prise de décision en proposant un choix alternatif en ajoutant une instruction **else** à l'instruction **if**. L'instruction **if...else** vous permet d'exécuter un bloc de code si la condition spécifiée est évaluée comme vraie et un autre bloc de code si elle est évaluée comme fausse. Il peut s'écrire, comme ceci :

```
if (condition){  
    // Code à exécuter si la condition est vraie  
} else {  
    // Code à exécuter si la condition est fausse  
}
```

L'exemple suivant affichera "Have a nice weekend!" si le jour actuel est vendredi, sinon il affichera "Have a nice day!"

```
<?php  
$d = date("D");  
if($d == "Fri"){  
    echo "Have a nice weekend!";  
} else{  
    echo "Have a nice day!";  
}  
?>
```

La déclaration if...elseif...else

Le **if...elseif...else** est une instruction spéciale utilisée pour combiner plusieurs instructions **if...else** .

```
if (condition1){  
    // Code à exécuter si la condition1 est vraie  
} elseif (condition2){  
    // Code à exécuter si la condition1 est fausse et la condition2 est vraie  
} else {  
    // Code à exécuter si condition1 et condition2 sont faux  
}
```

L'exemple suivant affichera "Have a nice weekend!" si le jour actuel est vendredi, et "Bon dimanche !" si le jour actuel est dimanche, sinon il affichera "Have a nice day!"

```
<?php  
$d = date("D");  
if($d == "Fri"){  
    echo "Have a nice weekend!";  
} elseif($d == "Sun"){  
    echo "Have a nice Sunday!";  
} else{  
    echo "Have a nice day!";  
}?>
```

L'opérateur ternaire

L'opérateur **ternaire** fournit une manière abrégée d'écrire les instructions **if...else**. L'opérateur ternaire est représenté par le symbole point d'interrogation (?) et il prend trois opérandes : une **condition** à **vérifier**, un **résultat** pour **true** et un **résultat** pour **false**.

Pour comprendre le fonctionnement de cet opérateur, considérons les exemples suivants :

```
<?php
if($age < 18)
{ echo 'Enfant';
} else{
echo 'Adulte';
}
?>
```

En utilisant l'opérateur ternaire, le même code pourrait être écrit de manière plus compacte :

```
<?php
$age = 15;
echo ($age < 18) ? 'Enfant' : 'Adulte';
?>
```

L'opérateur ternaire dans l'exemple ci-dessus sélectionne la valeur à gauche des deux-points (c'est-à-dire 'Enfant') si la condition est vraie (c'est-à-dire si \$age est inférieur à 18), et sélectionne la valeur à droite des deux-points (c'est-à-dire 'Adulte') si la condition est fausse.

Conseil : Le code écrit à l'aide de l'opérateur ternaire peut être difficile à lire. Cependant, il fournit un excellent moyen d'écrire des instructions if-else compactes.

Switch ... Cas

L'instruction **switch-case** est une alternative à l'instruction **if-elseif-else**, qui fait presque la même chose. L'instruction switch-case teste une variable par rapport à une série de valeurs jusqu'à ce qu'elle trouve une correspondance, puis exécute le bloc de code correspondant à cette correspondance.

```
switch (n){  
    case label1:  
        // Code à exécuter si n=label1  
        break;  
    case label2:  
        // Code à exécuter si n=label2  
        break;  
    ...  
    default :  
        // Code à exécuter si n est différent de tous les labels  
}
```


Prenons l'exemple suivant, qui affiche un message différent pour chaque jour.

```
<?php
$today = date("D");
switch($today){
    case "Mon":
        echo "Aujourd'hui, c'est lundi. Nettoyez votre maison.";
        break;
    case "Tue":
        echo "On est mardi. Achetez de la nourriture.";
        break;
    case "Wed":
        echo "Nous sommes mercredi. Visitez un médecin.";
        break;
    case "Thu":
        echo "Nous sommes jeudi. Réparez votre voiture.";
        break;
    case "Fri":
        echo "Nous sommes vendredi. Faites la fête ce soir.";
        break;
    case "Sat":
        echo "Nous sommes samedi. C'est l'heure du cinéma..";
        break;
    case "Sun":
        echo "Aujourd'hui, c'est dimanche. Reposez-vous..";
        break;
    default:
        echo "Aucune information disponible pour ce jour.";
        break;
}
?>
```

La déclaration switch-case diffère de la déclaration if-elseif-else d'une manière importante. L'instruction switch s'exécute ligne par ligne (c'est-à-dire instruction par instruction) et une fois que PHP trouve une instruction **case** évaluée à **true**, il exécute non seulement le code correspondant à cette instruction **case**, mais exécute également toutes les instructions **case** suivantes jusqu'à la fin du bloc **switch** automatiquement.

Pour éviter cela, ajoutez une instruction **break** à la fin de chaque bloc case. L'instruction **break** indique à PHP de sortir du bloc switch-case d'instructions une fois qu'il a exécuté le code associé au premier vrai cas.