

Les conditions

Table des matières

I. Contexte	3
II. Qu'est-ce qu'une condition ?	3
III. Exercice : Appliquez la notion	4
IV. if et else	5
V. Exercice : Appliquez la notion	8
VI. else if et écriture ternaire	9
VII. Exercice : Appliquez la notion	13
VIII. switch	14
IX. Exercice : Appliquez la notion	17
X. Auto-évaluation	17
A. Exercice final	17
B. Exercice : Défi	19
Solutions des exercices	19

I. Contexte

Durée : 1 h

Environnement de travail : Repl.it

Pré-requis : Bases de PHP

Contexte

Un programme devra adapter son comportement en fonction de paramètres, par exemple le type de l'utilisateur (simple utilisateur ou administrateur), la date ou une configuration serveur. Les conditions sont là pour aider à indiquer au programme le chemin à suivre et ainsi prévoir les cas d'exécution possibles.

II. Qu'est-ce qu'une condition ?

Objectifs

- Définir une condition
- Comprendre le mécanisme d'une condition

Mise en situation

Imaginons développer un site de vente en ligne de matériel informatique. Nous pourrions avoir besoin d'afficher les produits d'une certaine catégorie et pas les autres, ou bien d'afficher les produits en fonction d'un profil utilisateur. Les conditions vont nous permettre de répondre à ces besoins.

Définition

Une condition, ou structure conditionnelle, permet de réaliser des actions en fonction d'une expression booléenne. Il sera alors possible d'exécuter une portion de code selon l'évaluation de l'expression.

Syntaxe

Syntaxe générale

La logique est la même, quel que soit le langage de programmation pratiqué, à savoir **si - alors** ou **si - alors - sinon**.

```
1 si condition alors
2   code 1
3 sinon
4   code 2
```

Si la **condition** est **vraie**, l'instruction **code 1** sera exécutée ; si la **condition** est **fausse** (*sinon*), l'instruction **code 2** sera exécutée.

Exemple

En PHP

```
1 <?php
2 $age = 21;
3
4 if ($age >= 18) {
5     echo "Vous êtes majeur";
```

```
6 } else {
7   echo "Vous êtes mineur";
8 }
9 ?>
```

Le code ci-dessus affiche "*Vous êtes majeur*", car la valeur de `$age` est supérieure à 18. Si la variable `$age` avait été égale à 15, le contenu du `if` aurait été ignoré et seul le contenu du `else` aurait été exécuté, affichant ainsi "*Vous êtes mineur*".

Nous ne nous attardons pas sur la syntaxe pour le moment, c'est la compréhension du mécanisme de condition qui est importante.

Complément

Dans de nombreux cas, nous utiliserons les opérateurs de comparaison dans les conditions :

- `==` égale à
- `!=` différent de
- `>`, `<` supérieur à, inférieur à
- `>=`, `<=` supérieur ou égal, inférieur ou égal
- `===` égal à **et** de même type
- `!==` strictement différent de
- `<=>` retourne -1, 0 ou 1

Exemple En PHP

```
1 <code>
2 <?php
3 $a = 1;
4 $b = 2;
5 $c = 1.0;
6
7 if($a !== $c){
8   echo 'a est strictement différent de c';
9 }
10 echo ($a <=> $b); //retourne -1
11 echo ($a <=> $a); //retourne 0
12 echo ($b <=> $a); //retourne 1
13 <code>
```

Syntaxe À retenir

Le mécanisme d'une condition est sous la forme : **si** (condition) **alors** // code **sinon** // code ; la partie "sinon" étant facultative.

Exercice : Appliquez la notion

[solution n°1 p.21]

Exercice

À quoi servent les conditions ?

- ☐ À exécuter plusieurs fois un même bloc de code
- ☐ À centraliser un bloc de code pour qu'il puisse être appelé dans plusieurs endroits
- ☐ À exécuter un bloc de code seulement dans certains cas

Exercice

Qu'est-ce que ce code va afficher ?

```
1 <?php
2
3 $userType = 'admin';
4 $userName = 'John';
5
6 if ($userType === 'admin') {
7     echo 'Bonjour administrateur ' . $userName;
8 } else {
9     echo 'Bonjour ' . $userName;
10 }
```

IV. if et else

Objectifs

- Écrire une condition `if`
- Écrire une condition `else`

Mise en situation

La condition la plus utilisée en programmation, est le **si - alors - sinon** : si la condition est vraie, alors on exécute le script A, sinon on exécute le script B.

Syntaxe if

1. Indiquez le mot-clé `if`
2. Suivi de la condition entre parenthèses (`condition`)
3. Puis du code à exécuter, si la condition est vraie, entre accolades { }

```
1 <?php
2 if ($condition) {
3     // code;
4 }
```

Exemple if

```
1 <?php
2 $valeur = 1;
3
4 if ($valeur === 1) {
5     echo 'La variable $condition est égale à 1';
6 }
```

L'expression `$valeur === 1` est évaluée. Si `$valeur` est effectivement un entier égal à 1, la condition sera **vraie**, et l'instruction placée dans le bloc `if` sera exécutée.

Exemple Site de vente en ligne

Notre site de vente en ligne possède une section "compte utilisateur". Nous souhaitons afficher le nom d'utilisateur si, et seulement si, celui-ci est renseigné.

```
1 <?php
2 $nomUtilisateur = 'John Doe';
3
4 if ($nomUtilisateur !== '') {
5     echo 'Utilisateur : '.$nomUtilisateur;
6 }
```

Utilisateur : John Doe

Le nom d'utilisateur est bien différent de "vide", il est donc affiché.

Conseil

Si une variable est déjà de type booléen, on pourrait être tentés d'écrire : `if ($condition === true)`. Cette condition est valide et fonctionne, mais, pour alléger le code, il est possible de simplement écrire `if ($condition)`, `$condition` étant déjà un booléen.

Syntaxe else

`else` va toujours de pair avec `if`. Il permet d'exécuter les instructions pour le cas où la condition n'est pas vérifiée.

```
1 <?php
2 if ($condition) {
3     // code si $condition est vraie;
4 } else {
5     // code si $condition est fausse;
6 }
```

Exemple else

```
1 <?php
2 $valeur = 2;
3
4 if ($valeur === 1) {
5     echo 'La variable $condition est égale à 1';
6 } else {
7     echo 'La variable $condition n\'est pas égale à 1';
8 }
9 ?>
```

L'expression `$valeur === 1` est évaluée. Si `$valeur` n'est pas égale à 1, la condition sera **fausse**, c'est donc l'instruction placée dans le bloc `else` qui sera exécutée.

La variable \$condition n'est pas égale à 1

Exemple Site de vente en ligne

Pour afficher les comptes utilisateurs d'un site, nous affichons sur la page d'accueil "Bonjour Madame" ou "Bonjour Monsieur" en fonction de ce qu'a indiqué l'utilisateur.

```
1 <?php
2 $sexeUtilisateur = 'F';
3
4 if ($sexeUtilisateur === 'M') { // Si l'utilisateur est un homme
5     echo 'Bonjour Monsieur';
6 } else { // Sinon, c'est une femme.
7     echo 'Bonjour Madame';
8 }
9 ?>
```

Bonjour Madame

La condition du `if` étant fausse, la seule instruction exécutée sera celle du bloc `else`.

Attention Test d'égalité et affectation

Soyons vigilants avec l'utilisation des opérateurs `==` et `===` dans le `if`.

Par habitude, il peut arriver d'écrire simplement un seul égal "=", ce qui aura pour effet d'affecter une valeur à une variable, et non de comparer les valeurs. Or :

- si la condition d'un `if` n'est pas un booléen, l'interpréteur PHP **va essayer de convertir cette variable en booléen**,
- la règle qui va s'appliquer le plus souvent est que, si une variable n'est pas vide, alors PHP la traite comme "vraie".

```
1 <?php
2 $sexeUtilisateur = 'F';
3
4 if ($sexeUtilisateur = 'M') {
5     echo 'Bonjour Monsieur<br>';
6 }
7
8 echo "Sexe de l'utilisateur : $sexeUtilisateur";
9 ?>
```

Bonjour Monsieur
Sexe de l'utilisateur : M

Dans le code ci-dessus, nous avons oublié un égal à notre condition.

Dans le `if` :

1. la variable `$sexeUtilisateur` prend la valeur 'M',
2. puis PHP résout l'expression `if ('M') : 'M' n'est pas une valeur vide, donc l'expression appelle "vrai" ; donc on passe dans le if.`

Syntaxe À retenir

```
1 <?php
2 if (expression) {
3     // Code à exécuter si expression est vraie.
4 } else {
5     // Code à exécuter si expression est fausse.
6 }
7 ?>
```

Complément

if¹

else²

Les booléens et la conversion en booléen³

V. Exercice : Appliquez la notion

Rien de tel que la pratique pour comprendre les conditions ! Dans cet exercice, nous allons écrire un petit script de gestion de combat de boxe.

En boxe, afin d'éviter les trop gros écarts entre deux participants, les championnats sont divisés en catégories de poids, et deux boxeurs ne peuvent s'affronter que s'ils sont dans la même catégorie. Par exemple, un boxeur pesant 92 kg, considéré comme "Poids lourd", ne peut pas affronter un autre boxeur pesant 80 kg, considéré comme "Lourd-léger".

Pour réaliser cet exercice, vous aurez besoin de travailler sur l'environnement de travail :



Question

[solution n°2 p.21]

Écrivez un script permettant de déterminer, à partir du poids de deux boxeurs, s'ils peuvent combattre ensemble ou non. Pour simplifier, nous n'allons gérer que le cas des poids lourds, c'est-à-dire les boxeurs ayant un poids de 90,7 kg ou plus. Le script devra donc vérifier les deux poids et afficher "FIGHT !" si les deux boxeurs sont tous les deux des poids lourds,

ou "Impossible de combattre !" dans le cas contraire.

Les deux poids seront stockés dans des variables.

```
1 <?php
2
3 $boxer1Weight = 80.4;
4 $boxer2Weight = 92.2;
5
6 // Faire le traitement ici
```

1 <https://www.php.net/manual/fr/control-structures.if.php>

2 <https://www.php.net/manual/fr/control-structures.else.php>

3 <https://www.php.net/manual/fr/language.types.boolean.php>

4 <https://repl.it/>

Voici les différents cas de tests qui devront être gérés par votre script :

```
1 <?php
2
3 $boxer1Weight = 80.4;
4 $boxer2Weight = 92.2;
5
6 // Doit afficher "Impossible de combattre !" car boxer1 n'est pas poids lourd
7
8 $boxer1Weight = 95.6;
9 $boxer2Weight = 78.1;
10
11 // Doit afficher "Impossible de combattre !" car boxer2 n'est pas poids lourd
12
13 $boxer1Weight = 54.4;
14 $boxer2Weight = 70.2;
15
16 // Doit afficher "Impossible de combattre !" car les deux boxeurs ne sont pas des poids lourds
17
18 $boxer1Weight = 91.4;
19 $boxer2Weight = 98.3;
20
21 // Doit afficher "FIGHT !" car les deux boxeurs sont tous les deux des poids lourds
22
23 $boxer1Weight = 90.7;
24 $boxer2Weight = 98.3;
25
26 // Doit afficher "FIGHT !" car les deux boxeurs sont tous les deux des poids lourds, même
    celui qui pese exactement 90.7kg
```

Indice :

Il est possible de réaliser ce script avec un seul `if`, mais cela peut peut-être vous aider de faire une condition par boxeur, puis de comparer les résultats.

VI. else if et écriture ternaire

Objectifs

- Écrire une condition `else if`
- Utiliser l'écriture ternaire à bon escient

Mise en situation

Dans certains cas, une seule condition n'est pas suffisante pour couvrir l'ensemble des cas d'utilisation. Pour répondre à cette problématique, nous utiliserons la structure **si - sinon si - sinon**. Soit : si la condition est vraie, alors on exécute le script A, sinon si la condition suivante est vraie, nous exécuterons le script B, sinon nous exécuterons le script C.

Syntaxe else if

Le mot-clé `else if` (*sinon si*) devra être placé après l'accolade fermante du bloc `if "}"` et, contrairement au `else`, sera suivi d'une condition entre parenthèses. Le code à exécuter sera placé à la suite, entre les accolades.

L'ordre des conditions est le suivant :

- `if`
- `else if`
- `else if`
- `else`

Les `else if` peuvent être enchaînés autant de fois que besoin. Pour finir, n'oublions pas le `else` à la fin, dans le cas où toutes les conditions en amont seraient fausses.

```
1 <?php
2 $valeur = 3;
3
4 if ($valeur === 1) {
5     echo 'La variable $valeur est égale à 1';
6 } else if ($valeur === 2) {
7     echo 'La variable $valeur est égale à 2';
8 } else if ($valeur === 3) {
9     echo 'La variable $valeur est égale à 3';
10 } else {
11     echo 'La variable $valeur n\'est pas égale à 1, 2 ou 3';
12 }
13 ?>
```

Les expressions `$valeur ===` sont évaluées les unes après les autres, jusqu'à en trouver une vraie. Le code correspondant sera alors exécuté.

Remarque

`else if` peut aussi s'écrire sans espace : `elseif`.

Attention

Dès qu'une condition est évaluée comme vraie, les suivantes sont ignorées.

Exemple Site de vente en ligne

Considérons que notre site de vente en ligne soit internationalisé et que nous souhaitons dire "*Bonjour*" à l'utilisateur dans sa langue.

```
1 <?php
2 $pays = 'Espagne';
3
4 if ($pays === 'Allemagne') {
5     echo 'Hallo';
6 } else if ($pays === 'Angleterre') {
7     echo 'Hello';
8 } else if ($pays === 'Espagne') {
9     echo 'Buenos días';
10 } else if ($pays === 'France') {
11     echo 'Bonjour';
12 } else if ($pays === 'Italie') {
13     echo 'Buongiorno';
```

```
14 } else {  
15     echo 'Bonjour';  
16 }  
17 ?>
```

Buenos dias

L'écriture ternaire

Les ternaires vont nous permettre de condenser notre code, en écrivant une structure conditionnelle sur une seule ligne.

Syntaxe

Une écriture ternaire se décompose ainsi : **(condition) ? alors : sinon**

- **(condition)** est l'expression booléenne à évaluer,
- **?** est l'opérateur ternaire, équivalent du `if`,
- **alors** est exécuté si la condition est vraie,
- **:** est l'équivalent du `else`,
- **sinon** est exécuté si la condition est fausse.

Exemple

```
1 <?php  
2 $pays = 'france';  
3  
4 if ($pays === 'france') {  
5     $francais = true;  
6 } else {  
7     $francais = false;  
8 }  
9  
10 // if-else transformé en ternaire.  
11  
12 $francais = ($pays === 'france') ? true:false;  
13 ?>
```

Si `$pays` est égal à 'france', alors on affecte la valeur `true` à `$francais`; sinon on lui affecte la valeur `false`.

Attention

L'écriture ternaire est à utiliser avec parcimonie, car elle peut très vite devenir piègeuse, source d'erreurs et, paradoxalement, faire perdre en lisibilité. Elle est recommandée pour remplacer un **if - else**, mais à éviter dans le cas des **if - else if - else**.

Exemple Transformation d'un if - else if - else en ternaire

Reprenons l'exemple du "Bonjour" en différentes langues. Nous pouvons juger la structure un peu lourde et pourrions être tentés d'écrire ceci :

```
1 <?php
2 $pays = 'Espagne';
3
4 $bonjour =
5     ($pays === 'Allemagne') ? 'Hallo' :
6     (($pays === 'Angleterre') ? 'Hello' :
7     (($pays === 'Espagne') ? 'Buenos dias' :
8     (($pays === 'France') ? 'Bonjour' :
9     (($pays === 'Italie') ? 'Buongiorno' : 'Salut'))));
10
11 echo $bonjour;
12 ?>
```

Buenos dias

Dans ce cas, la structure a effectivement moins de lignes, mais n'est pas pour autant moins lourde. Elle est même moins lisible et a été plus longue à écrire.

Syntaxe À retenir

```
1 <?php
2 if (expression1) {
3     // Code à exécuter si expression1 est vraie.
4 } else if (expression2) {
5     // Code à exécuter si expression2 est vraie.
6 } else {
7     // Code à exécuter si expression1 et expression2 sont fausses.
8 }
9 ?>
```

Complément

else if¹

&& (et logique)

Imaginons que le nom d'utilisateur puisse être soit une chaîne vide (si l'utilisateur n'a pas rempli le champ du formulaire), soit la valeur nulle, qui symbolise l'absence de données (dans un retour de base de données, par exemple). Il faut alors vérifier ces deux cas au moment d'afficher le nom.

```
1 <?php
2 $nomUtilisateur = 'John Doe';
3
4 if ($nomUtilisateur !== '' && $nomUtilisateur !== null) {
5     echo "Utilisateur : $nomUtilisateur";
6 }
7 ?>
```

¹ <https://www.php.net/manual/fr/control-structures.elseif.php>

Dans cet exemple on veut s'assurer qu'un nom soit bien inscrit, pour cela on vérifie d'abord que `$nomUtilisateur` soit **strictement différent** de `''`. Mais quand on ne rentre rien il arrive parfois que la valeur retournée soit **nulle**, par précaution on rajoute une condition pour s'assurer cette fois que `$nomUtilisateur` soit strictement différent de **nulle**.

On ne pourra donc pas afficher "Utilisateur : John Doe" si les deux conditions ne sont pas remplies !

|| (ou logique)

Contrairement au `&&` le `||` (qui se prononce "double pipe") permet de vérifier plusieurs options, pour que ce soit plus clair, reprenons l'exemple vu plus tôt avec le "Bonjour" qui change de langue selon le pays.

```
1 <code>
2 <?php
3 if ($pays === 'France' || $pays === 'Belgique') {
4     echo "Bonjour";
5 }
6 <code>
```

Avec cet exemple il est plus aisé de comprendre le `||`, en effet il n'y a pas qu'en France que l'on parle français, cette condition permet de dire "Bonjour" si notre pays est la France OU la Belgique. Donc que `$pays` soit égal à 'France' ou à 'Belgique' le résultat sera identique.

VII. Exercice : Appliquez la notion

Pour réaliser cet exercice, vous aurez besoin de travailler sur l'environnement de travail :



Question

[solution n°3 p.21]

On considère des combats de boxe entre :

- des poids mi-lourds (entre 76,2 et 79,3 kg)
- des poids lourds-légers (entre 79,3 et 90,7 kg)
- des poids lourds (plus de 90,7 kg)

Si les deux boxeurs sont dans la même catégorie, il faudra afficher "*FIGHT*" suivi du nom de la catégorie (par exemple, "*FIGHT POIDS LOURDS*" ou "*FIGHT POIDS MI-LOURDS*"). Sinon, le message "*Impossible de combattre*" devra être affiché.

Si un participant a un poids à la limite de deux catégories, on considère qu'il peut combattre dans les deux. Ainsi, un combattant pesant 79,3 kg peut affronter un combattant lourd-léger ou un combattant mi-lourd.

Voici les différents cas de tests que votre script devra respecter :

```
1 <?php
2 $boxer1Weight = 54.4;
3 $boxer2Weight = 70.2;
4
5 // Doit afficher "Impossible de combattre !" car les deux boxeurs ne sont pas dans la même
   catégorie
6
7 $boxer1Weight = 77.1;
8 $boxer2Weight = 79.1;
9
```

1 <https://repl.it/>

```

10 // Doit afficher "FIGHT MI-LOURDS !"
11
12 $boxer1Weight = 86.2;
13 $boxer2Weight = 79.9;
14
15 // Doit afficher "FIGHT LOURDS-LEGRS !"
16
17 $boxer1Weight = 100.4;
18 $boxer2Weight = 90.7;
19
20 // Doit afficher "FIGHT POIDS LOURDS !" car le boxer2 pese pile 90.7 kg, il peut combattre un
    poids lourd
21
22 $boxer1Weight = 79.9;
23 $boxer2Weight = 90.7;
24
25 // Doit afficher "FIGHT LOURDS-LEGRS !" car le boxer2 pese pile 90.7 kg, il peut aussi
    combattre un poids lourd-leger
26
27 $boxer1Weight = 90.7;
28 $boxer2Weight = 90.7;
29
30 // Doit afficher soit "FIGHT POIDS LOURDS !", soit "FIGHT LOURDS-LEGRS !", mais pas les deux
    messages !

```

Indice :

Il n'est pas possible de tout faire en un seul `if`, sinon on perd la distinction entre les différentes catégories.

Indice :

Attention : si vous enchaînez simplement les `if`, vous risquez d'avoir des problèmes avec les valeurs qui se chevauchent. Si on a deux combattants qui pèsent 79,3 kg, on ne veut qu'un seul message. C'est là tout l'intérêt du `else if` : s'arrêter au premier cas valide.

VIII. switch

Objectifs

- Écrire un `switch`
- Savoir quand utiliser un `else if` ou un `switch`

Mise en situation

La structure **if - else if - else** peut devenir lourde et répétitive si de nombreux cas sont à traiter. L'écriture ternaire n'étant pas une alternative crédible, nous allons voir une nouvelle structure permettant de simplifier le code : le `switch`.

Syntaxe switch

Le principe du `switch` est de prendre une variable et de la comparer à plusieurs valeurs, chaque cas ayant un code correspondant :

```

1 switch ($variable) {
2     case 'valeur1':
3         // Code à exécuter.
4         break;
5     case 'valeur2':
6         // Code à exécuter.

```

```
7         break;
8     default:
9         // Code à exécuter.
10 }
```

La `$variable` paramètre du `switch` n'est pas une expression booléenne, mais une valeur qui sera comparée ensuite avec chaque `case`. Une fois le `case` trouvé, le code correspondant est exécuté et l'exécution du `switch` est interrompue quand le mot-clé `break` est rencontré.

Un `switch` se décompose ainsi :

- `switch` : le mot-clé pour indiquer qu'on va utiliser le `switch`.
- `$variable`) : la valeur à comparer dans les `case`.
- `case 'valeur'` : les différentes valeurs de comparaison.
- `break` : indique de sortir du `switch` quand on a rencontré un `case`.
- `default` : si valeur n'a pas été trouvée dans les cases. Équivalent du `else` classique.

Attention

Si nous ne mettons pas le mot-clé `break` à chaque fin de bloc `case`, le programme continuera à comparer les `case` les uns après les autres. Utiliser un `break` est l'équivalent du `else if`, là où ne pas le mettre équivaut à enchaîner les `if`.

Exemple

```
1 <?php
2 $pays = 'Angleterre';
3
4 switch ($pays) {
5     case 'Allemagne':
6         echo "Hallo";
7         break;
8     case 'Angleterre':
9         echo "Hello";
10        break;
11    case 'France':
12        echo "Bonjour";
13        break;
14    case 'Espagne':
15        echo "Buenos dias";
16        break;
17    case 'Italie':
18        echo "Buongiorno";
19        break;
20    default:
21        echo "De quel pays êtes-vous ?";
22 }
23 ?>
```

Hello

Attention

Le `switch` ne traite que l'égalité `==`, et non la supériorité ou l'infériorité.

Remarque

Cas particulier du `switch`

Si pour plusieurs `case` nous devons exécuter le même code, il est possible d'enchaîner les `case` sans répéter le bloc d'instructions. Nous écrirons alors seulement `case 'valeur'`.

```
1 <?php
2 $pays = 'Norvège';
3
4 switch ($pays) {
5     case 'Norvège':
6     case 'Allemagne':
7         echo "Hallo";
8         break;
9     case 'Angleterre':
10        echo "Hello";
11        break;
12    default:
13        echo "De quel pays êtes-vous ?";
14 }
15 ?>
```

Ici, "*Bonjour*" se dit "*Hallo*" en allemand et en norvégien. Nous avons donc écrit une seule fois l'affichage de "*Hallo*".

Hallo

Conseil

Le `switch` étant plus lisible, mieux structuré (moins d'accolades, une indentation plus adaptée), il est préféré à un enchaînement trop important de `if` ou de `else if`.

Syntaxe

À retenir

```
1 switch (valeur) {
2     case 'valeur0':
3     case 'valeur1':
4         // Code à exécuter.
5         break;
6     case 'valeur2':
7         // Code à exécuter.
8         break;
9     default:
10        // Code à exécuter.
11 }
```

Complément

`switch`¹

¹ <https://www.php.net/manual/fr/control-structures.switch.php>

IX. Exercice : Appliquez la notion

Pour réaliser cet exercice, vous aurez besoin de travailler sur l'environnement de travail :



Question

[solution n°4 p.22]

Il existe de nombreux types de boxe, qui permettent l'utilisation de plus ou moins de parties du corps :

- La boxe anglaise permet de n'utiliser que les poings,
- La savate BF et le kick-boxing américain permettent d'utiliser les poings et les pieds,
- Le kick-boxing japonais permet d'utiliser les poings, pieds et genoux,
- La boxe birmane, la boxe khmère, la boxe thaï et le shoot-boxing permettent d'utiliser tout le haut du corps.

Écrivez un script permettant d'afficher les règles d'un combat de boxe selon le type voulu. Si un type n'est pas dans la liste, on considère que l'on peut utiliser tout le haut du corps. Voici les cas de tests :

```
1 <?php
2
3 $boxeType = 'savate BF';
4 // Doit afficher "Pieds et poings"
5
6 $boxeType = 'boxe khmère';
7 // Doit afficher "Tout le haut du corps"
8
9 $boxeType = 'kick-boxing américain';
10 // Doit afficher "Pieds et poings"
11
12 $boxeType = 'MMA';
13 // Doit afficher "Tout le haut du corps", car MMA n'est pas dans la liste
14
15 $boxeType = 'kick-boxing japonais';
16 // Doit afficher "Poings, pieds et genoux"
```

Indice :

Si un cas précis a le même résultat qu'un cas par défaut, il n'est pas utile de traiter ce cas : il passera dans le cas par défaut.

X. Auto-évaluation

A. Exercice final

Exercice 1

[solution n°5 p.22]

Exercice

Que va afficher ce code ?

```
1 <?php
2
3 $budget = 350000;
4
5 $house1Price = 500000;
6 $house2Price = 300000;
7 $house3Price = 250000;
```

1 <https://repl.it/>

```

8
9 if ($budget > $house1Price) {
10     echo 'Maison1';
11 }
12
13 if ($budget > $house2Price) {
14     echo 'Maison2';
15 }
16
17 if ($budget > $house3Price) {
18     echo 'Maison3';
19 }

```

Exercice

Que va afficher ce code ?

```

1 <?php
2
3 $animal = 'chat';
4
5 switch($animal) {
6     case 'chien':
7         $aliment = 'croquettes';
8         break;
9     case 'chat':
10        $aliment = 'pâtée';
11        break;
12    default:
13        $aliment = 'Inconnu';
14 }
15
16 echo $aliment;

```

Exercice

Que va afficher ce code ?

```

1 <?php
2
3 $animal = 'chien';
4
5 switch($animal) {
6     case 'chien':
7         $aliment = 'croquettes';
8     case 'chat':
9         $aliment = 'pâtée';
10    default:
11        $aliment = 'Inconnu';
12 }
13
14 echo $aliment;

```

B. Exercice : Défi

Pour réaliser cet exercice, vous aurez besoin de travailler sur l'environnement de travail :



Question

[solution n°6 p.23]

Un garage répare des voitures, des quads, des scooters, des motos et des trikes. Pour chaque réparation, il a besoin d'acheter des pneus neufs : il préfère remplacer toutes les roues du véhicule, par précaution. Chaque type de véhicule possède un nombre variable de roues : les voitures et les quads en ont 4, les trikes en ont 3, et les scooters et motos en ont 2. Pour tous les autres véhicules (remorques, monocycles, et autres), il achète quand même un pneu par sécurité.

Il a le choix entre deux fabricants de pneus, qui ont différents tarifs :

- "Pneufs" vend ses pneus 60 € pièce, mais si on lui en achète 4, alors il propose une remise de 15 % sur la facture finale.
- "Roues cool" vend ses pneus 75 € pièce, mais, à partir du troisième, il propose une remise de 25 % sur la facture finale.

À chaque réparation, c'est donc un casse-tête pour savoir chez quel fournisseur il est le plus intéressant d'acheter ses pneus.

Écrivez un script qui sera appelé à chaque réparation de véhicule. Ainsi, à partir d'une variable contenant le type de véhicule, votre script va afficher le nom du fournisseur le plus intéressant.

```
1 <?php
2
3 $type = 'voiture'; // À changer à chaque réparation
4
5 // Affectation du nombre de pneus à la variable $nbPneus
6
7 // Calcul des deux prix $pneufPrice et $rouesCoolPrice
8
9 // Affichage du nom du fournisseur le plus intéressant
```

Indice :

Le meilleur moyen de résoudre un problème complexe est de le diviser en plusieurs étapes. Ici, vous pouvez diviser votre script en 4 grandes parties : le calcul du nombre de roues, le calcul du prix chez Pneu, le calcul du prix chez Roues Cool, et l'affichage du résultat.


Faire 15 % de remise sur un montant équivaut à le multiplier par 0,85. D'une manière générale, faire X % de remise, c'est multiplier par $(100 - X)/100$.

Solutions des exercices

1 <https://repl.it/>

Exercice p. 4 Solution n°1**Exercice**

À quoi servent les conditions ?

- ☐ À exécuter plusieurs fois un même bloc de code
- ☐ À centraliser un bloc de code pour qu'il puisse être appelé dans plusieurs endroits
- ☒ À exécuter un bloc de code seulement dans certains cas
-  Les conditions permettent d'exécuter une partie du code seulement quand certaines conditions sont requises.

Exercice

Qu'est-ce que ce code va afficher ?

```

1 <?php
2
3 $userType = 'admin';
4 $userName = 'John';
5
6 if ($userType === 'admin') {
7     echo 'Bonjour administrateur '.$userName;
8 } else {
9     echo 'Bonjour '.$userName;
10 }
```

Bonjour administrateur John

p. 8 Solution n°2

```

1 <?php
2
3 $boxer1Weight = 80.4;
4 $boxer2Weight = 92.2;
5
6 if ($boxer1Weight >= 90.7 && $boxer2Weight >= 90.7) {
7     echo "FIGHT !";
8 } else {
9     echo "Impossible de combattre !";
10 }
```

p. 13 Solution n°3

```

1 <?php
2
3 if ($boxer1Weight >= 90.7 && $boxer2Weight >= 90.7) {
4     echo "FIGHT POIDS LOURDS !";
5 } elseif ($boxer1Weight <= 90.7 && $boxer1Weight >= 79.3 && $boxer2Weight <= 90.7 &&
6     $boxer2Weight >= 79.3) {
7     echo "FIGHT LOURDS-LEGERS !";
8 } elseif ($boxer1Weight <= 79.3 && $boxer1Weight >= 76.2 && $boxer2Weight <= 79.3 &&
9     $boxer2Weight >= 76.2) {
10     echo "FIGHT MI-LOURDS !";
11 } else {
```

```
10     echo "Impossible de combattre !";
11 }
```

p. 17 Solution n°4

```
1 <?php
2
3 switch ($boxeType) {
4     case 'boxe anglaise':
5         echo 'Poings';
6         break;
7     case 'savate BF':
8     case 'kick-boxing américain':
9         echo 'Pieds et poings';
10        break;
11    case 'kick-boxing japonais':
12        echo 'Poings, pieds et genoux';
13        break;
14    default:
15        echo 'Tout le haut du corps';
16 }
```


Exercice p. 17 Solution n°5

Exercice

Que va afficher ce code ?

```
1 <?php
2
3 $budget = 350000;
4
5 $house1Price = 500000;
6 $house2Price = 300000;
7 $house3Price = 250000;
8
9 if ($budget > $house1Price) {
10     echo 'Maison1';
11 }
12
13 if ($budget > $house2Price) {
14     echo 'Maison2';
15 }
16
17 if ($budget > $house3Price) {
18     echo 'Maison3';
19 }
```

Maison2Maison3

 Les `if` sont enchaînés : ce ne sont pas des `else if`, donc on passe dans chacun des `if`. Les deuxième et troisième `if` ont une condition vraie, donc ils affichent "*Maison2*" et "*Maison3*" chacun leur tour.

Exercice

Que va afficher ce code ?

```

1 <?php
2
3 $animal = 'chat';
4
5 switch($animal) {
6     case 'chien':
7         $aliment = 'croquettes';
8         break;
9     case 'chat':
10        $aliment = 'pâtée';
11        break;
12    default:
13        $aliment = 'Inconnu';
14 }
15
16 echo $aliment;

```

pâtée

Q La variable `animal` contient *"chat"*, donc lorsque le `switch` compare cette valeur à tous ses `case`, il affecte la variable `$aliment` à *"pâtée"* et sort du `switch` (grâce au `break`).

Exercice

Que va afficher ce code ?

```

1 <?php
2
3 $animal = 'chien';
4
5 switch($animal) {
6     case 'chien':
7         $aliment = 'croquettes';
8     case 'chat':
9         $aliment = 'pâtée';
10    default:
11        $aliment = 'Inconnu';
12 }
13
14 echo $aliment;

```

Inconnu

Q La variable `animal` contient *"chien"*, donc lorsque le `switch` compare cette valeur à tous ses `case`, il affecte la variable `$aliment` à *"croquettes"*. Mais, cette fois, nous avons oublié les `break`, donc plutôt que de sortir du `switch`, nous allons continuer à comparer cette valeur avec les autres `case`. Ce n'est pas un *"chat"*, donc rien n'est fait ; cependant, le `default` est exécuté à chaque fois qu'il est atteint, donc l'aliment *"croquettes"* est écrasé par *"Inconnu"*, qui est ensuite affiché. C'est pourquoi il est très important de ne pas oublier les `break` dans un `switch` !

p. 19 Solution n°6

```

1 <?php
2
3 $type = 'voiture'; // À changer à chaque réparation
4
5 switch ($type) {
6     case 'voiture':
7     case 'quad':
8         $nbPneus = 4;

```

```

9         break;
10     case 'trike':
11         $nbPneus = 3;
12         break;
13     case 'moto':
14     case 'scooter':
15         $nbPneus = 2;
16         break;
17     default:
18         $nbPneus = 1;
19 }
20
21 $pneufPrice = $nbPneus*60;
22 if ($nbPneus === 4) {
23     $pneufPrice *= 0.85;
24 }
25
26 $rouesCoolPrice = $nbPneus*75;
27 if ($nbPneus >= 3) {
28     $rouesCoolPrice *= 0.75;
29 }
30
31 if ($pneufPrice < $rouesCoolPrice) {
32     echo 'pneuf';
33 } else {
34     echo 'rouesCool';
35 }
36

```