codecademy

# Validation de formulaire PHP

## Jeux de caractères dans les expressions régulières

Les jeux de caractères d'expressions régulières indiqués par une paire de crochets `[]` correspondront à tous les caractères inclus entre crochets. Par exemple, l'expression régulière `con[sc]en[sc]us` correspondra à n'importe laquelle des orthographes `consensus`, `concensus`, `consencus` et `concencus`.

## Quantificateurs facultatifs dans les expressions régulières

Dans les expressions régulières, les quantificateurs facultatifs sont signalés par un point d'interrogation `?`. Il indique qu'un caractère peut apparaître 0 ou 1 fois. Par exemple, l'expression régulière `humou?r` correspondra au texte `humour` ainsi qu'au texte `humor`.

## Littéraux dans les expressions régulières

Dans l'expression régulière, les `literals` caractères sont les caractères les plus simples qui correspondront au texte exact des littéraux. Par exemple, la regex `monkey` correspondra complètement au texte `monkey` mais correspondra également `monkey` dans le texte `The monkeys like to eat bananas.`

## Quantificateurs fixes dans les expressions régulières

Dans les expressions régulières, les quantificateurs fixes sont indiqués par des accolades `{}`. Il contient soit la quantité exacte, soit la plage de quantités de caractères à faire correspondre. Par exemple, l'expression régulière `roa{3}r` correspondra au texte `roaaar`, tandis que l'expression régulière `roa{3,6}r` correspondra `roaaar` à, `roaaaar`, `roaaaaar` ou `roaaaaaar`.

## Alternance dans les expressions régulières

L'alternance indiquée par le symbole pipe `|`, permet la correspondance de l'une des deux sous-expressions. Par exemple, la regex `baboons|gorillas` correspondra au texte `baboons` ainsi qu'au texte `gorillas`.

## Ancres dans les expressions régulières

Les ancres (chapeau `^` et signe dollar `$` ) sont utilisées dans les expressions régulières pour faire correspondre le texte au début et à la fin d'une chaîne, respectivement. Par exemple, la regex `^Monkeys: my mortal enemy$` correspondra complètement au texte `Monkeys: my mortal enemy` mais pas à `Spider Monkeys: my mortal enemy` ou `Monkeys: my mortal enemy in the wild` . Le `^` garantit que le texte correspondant commence par `Monkeys` , et le `$` garantit que le texte correspondant se termine par `enemy` .

## Expressions régulières

Les expressions régulières sont des séquences de caractères définissant un modèle de texte qui doit être trouvé. Ils peuvent être utilisés pour analyser les fichiers texte pour un modèle spécifique, vérifier les résultats des tests et rechercher des mots-clés dans les e-mails ou les pages Web.

## Caractères génériques dans les expressions régulières

Dans l'expression régulière, les caractères génériques sont désignés par le point `.` et peuvent correspondre à n'importe quel caractère (lettre, chiffre, symbole ou espace) dans un morceau de texte. Par exemple, l'expression régulière `.........` correspondra au texte `orangutan , marsupial` ou à tout autre texte de 9 caractères.

## Plages d'expressions régulières

Les plages d'expressions régulières sont utilisées pour spécifier une plage de caractères pouvant être mis en correspondance. Les plages d'expressions régulières courantes incluent : [AZ]. : correspond à n'importe quelle lettre majuscule [az]. : correspond à n'importe quelle lettre minuscule [0-9]. : correspond à n'importe quel chiffre [A-Za-z] : correspond à n'importe quelle lettre majuscule ou minuscule.

## Classes de caractères abrégés dans les expressions régulières

Les classes de caractères abrégés simplifient l'écriture d'expressions régulières. Par exemple, `\w` représente la plage d'expressions régulières `[A-Za-z0-9_]` , `\d` représente [0-9], `\W` représente `[^A-Za-z0-9_]` la correspondance de tout caractère non inclus par `\w` , `\D` représente `[^0-9]` la correspondance de tout caractère non inclus par `\d` .

## Kleene Star & Kleene Plus dans les expressions régulières

Dans les expressions régulières, l'étoile de Kleene ( `*` ) indique que le caractère précédent peut apparaître 0 fois ou plus. Par exemple, `meo*w` correspondra `mew` à , `meow` , `meooow` et `meooooooooooow` . Le Kleene plus( `+` ) indique que le caractère précédent peut apparaître 1 ou plusieurs fois. Par exemple, `meo+w` correspondra `meow` à , `meooow` et `meooooooooooow` , mais pas à `mew` .

## Regroupement dans les expressions régulières

Dans les expressions régulières, le regroupement est effectué par `(` des parenthèses ouvrantes et fermantes `)` . Ainsi, l'expression régulière `I love (baboons|gorillas)` correspondra au texte `I love baboons` ainsi qu'à `I love gorillas` , car le regroupement limite la portée du `|` au texte entre parenthèses.

## $_POSTVariable superglobale

`$_POST` , la variable superglobale PHP, est un tableau qui contient les données de la requête POST du client. Ces données sont traduites à partir des en-têtes de requête HTTP

## Fonction PHP filter_var

En PHP, `filter_var` est une fonction qui filtre une variable avec un filtre spécifié.
Comme premier argument, `filter_var()` prend une variable. En second lieu, il prend un ID représentant le type de filtrage qui doit être effectué. Le troisième argument est facultatif et peut être utilisé pour spécifier des options pour le filtre.
Il existe plusieurs filtres pour nettoyer les types d'entrée courants, y compris `FILTER_SANITIZE_EMAIL` , qui est un filtre courant qui supprime les caractères illégaux pour une adresse e-mail.
Les filtres peuvent également être utilisés pour valider que le texte correspond à un modèle. Par exemple, `FILTER_VALIDATE_EMAIL` renvoie la variable si elle ne contient que des caractères d'e-mail valides. Sinon, ça revient `false` .

```
echo filter_var("
<p>u</p>pies@codecademy.com",
FILTER_SANITIZE_EMAIL);
# prints "puppies@codecademy.com"
echo filter_var("
<p>u</p>pies@codecademy.com",
FILTER_VALIDATE_EMAIL);
# returns false and prints nothing
```

## Fonction PHP preg_match

En PHP, `preg_match` effectue une correspondance d'expression régulière. Le premier argument est le modèle à rechercher et le second argument est la variable à vérifier. Il renvoie `1` si cela correspond, `0` si ce n'est pas le cas et `FALSE` s'il y a eu une erreur.

```
$pattern = '/^[(]*([0-9]{3})[- .)]*[0-9]
{3}[- .]*[0-9]{4}$/';

preg_match($pattern, "(999)-555-2222");
// Returns: 1

preg_match($pattern, "555-2222"); //
Returns: 0
```

## Fonction PHP Strlen

En PHP, `strlen` est une fonction qui renvoie la longueur de la chaîne qui lui est transmise.

```
echo strlen("Codecademy");
# prints "10"
```

## Fonction PHP preg_replace

En PHP, `preg_replace` est une fonction qui peut remplacer des parties d'une chaîne d'entrée basée sur une expression régulière.
Le premier argument est le motif à rechercher. Le deuxième argument est le nouveau texte avec lequel remplacer le motif. Le troisième argument est la chaîne sur laquelle opérer.

```
$one = "codeacademy";
$two = "CodeAcademy";
$three = "code academy";
$four = "Code Academy";

$pattern = "/[cC]ode\s*[aA]cademy/";
$codecademy = "Codecademy";

echo preg_replace($pattern, $codecademy,
$one);
// Prints: Codecademy

echo preg_replace($pattern, $codecademy,
$two);
// Prints: Codecademy

echo preg_replace($pattern, $codecademy,
$three);
// Prints: Codecademy

echo preg_replace($pattern, $codecademy,
$four);
// Prints: Codecademy
```

## Fonction d'en-tête PHP

La fonction PHP `header()` peut être utilisée pour effectuer des redirections.
Nous appelons la `header()` fonction sur une chaîne qui commence par `"Location: "` , suivie de l'URL vers laquelle nous voulons rediriger l'utilisateur. Par exemple : `"Location: https://www.codecademy.com/learn/learn-php/"` . Après avoir appelé la fonction header(), nous voudrons utiliser la construction du langage `exit` pour terminer le script en cours.

```
if (/* Is the submission data validated?
*/) {
  header("Location:
https://www.codecademy.com/learn/learn-
php/");
  exit;
}
```

## Fonction PHP htmlspecialchars

En PHP, `htmlspecialchars` est une fonction qui transforme les caractères spéciaux en entités HTML.

```
$text = "This text is <b>bold</b>.";

echo htmlspecialchars($text);
// prints This text is
&lt;b&gt;bold&lt;/b&gt;.
```

## Fonction de découpage PHP

La fonction PHP `trim()` intégrée est utilisée pour supprimer les espaces au début et à la fin d'une chaîne.

```
echo trim("         hello world    ");
# prints "hello world"
```