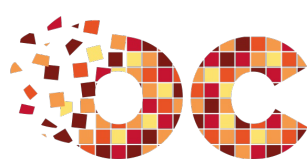


Un compte à rebours en PHP

Par SpriteWare



OPENCLASSROOMS

www.openclassrooms.com

*Licence Creative Commons 6 2.0
Dernière mise à jour le 8/02/2012*

Sommaire


Sommaire	2
Un compte à rebours en PHP	3
Bases du script	3
Les conversions	4
Finalisation & améliorations	6
Améliorations	7
Annexe : allons plus loin avec JavaScript !	7
Partager	9



Un compte à rebours en PHP



Mise à jour : 08/02/2012

Difficulté : Facile 



Salut les Zéros !

Je vais ici vous apprendre comment faire un compte à rebours en PHP, avec une petite touche de **JavaScript** pour ceux qui en veulent plus. Vous verrez, il n'y a rien de plus simple, et quelquefois on va chercher dans le bien compliqué inutilement (j'ai été dans ce cas 😊).

Je vous propose de faire, tout au long de ce tutoriel, un **TP**. Nous apprendrons à faire un compte à rebours avant Noël. (*C'est vrai quoi, qui n'est pas pressé d'être à Noël ?* 🤖)

⇒ Ce que vous saurez faire dans pas très très longtemps...

⇒ Et en JavaScript...

Compétences requises :

- *Maîtrise du PHP (évidemment) ;*
- *Fonction date() ;*
- *Timestamp.*

Sommaire du tutoriel :



- Bases du script
- Les conversions
- Finalisation & améliorations
- Annexe : allons plus loin avec JavaScript !

Bases du script

Comme je vous l'ai dit, notre petit script fera en sorte de calculer le temps qu'il reste jusqu'à Noël. Pour ce faire, nous allons utiliser le timestamp. Puis, nous calculerons la différence entre le timestamp actuel et le timestamp de Noël prochain.

~~Pour commencer, nous allons créer le timestamp du prochain Noël :~~

Changement de programme ! Est-ce plus palpitant d'attendre le 24 ou l'ouverture des cadeaux ? 😊 Personnellement, chez moi on les ouvre à 8h, car ma petite sœur croit toujours au Père Noël. À vous d'adapter en fonction de votre situation.



Dans la suite du tutoriel, je continuerai de citer le mot « Noël ». J'entends par là, bien sûr, l'ouverture des cadeaux, comme je l'ai dit plus haut.



Il ne faut pas oublier que nous cherchons à calculer le Noël prochain avec `date('Y')`.

Code : PHP

```
<?php
$annee = date('Y');
$noel = mktime(8, 0, 0, 12, 25, $annee); // Le timestamp du 25
décembre de l'année actuelle à 8h00 (ouverture des cadeaux).
?>
```

Voilà qui est fait ! Toutefois, nous avons négligé un détail : imaginez que nous sommes le **26 décembre**. Le compteur va donc être réglé sur le Noël qui vient de passer vu que l'année, elle, sera toujours la même. 😊 Le compteur sera donc en négatif, sauf si nous appliquons une condition simple à notre code :

Code : PHP

```
<?php
$annee = date('Y');
$noel = mktime(8, 0, 0, 12, 25, $annee);

if ($noel < time())
    $noel = mktime(8, 0, 0, 12, 25, ++$annee);

?>
```



Pourquoi as-tu placé le signe de l'incréméntation avant \$annee ? 😊

Il s'agit de la pré-incrémentation (au contraire de la post-incrémentation pour \$annee++). La valeur de la variable est d'abord incrémentée d'une unité puis renvoyée, alors que dans l'autre cas, c'est l'inverse. D'où l'utilité de faire un ++\$annee ici. 😊
De cette manière, nous avançons d'une année, et cela, (au risque de me répéter) tout simplement car nous cherchons à calculer le Noël **PROCHAIN** ! 😊

Bon ensuite, il faut que l'on récupère la différence entre le timestamp actuel et celui de Noël. Cette différence sera donc le temps qu'il nous reste (en secondes) avant d'ouvrir les cadeaux !

Récapitulatif :

Code : PHP

```
<?php
$annee = date('Y');
$noel = mktime(8, 0, 0, 12, 25, $annee);

if ($noel < time())
    $noel = mktime(8, 0, 0, 12, 25, ++$annee);

$tps_restant = $noel - time(); // Différence entre Noël et
maintenant.
?>
```

Avez-vous compris le principe ? Jusque là, c'était assez facile.
Maintenant devinez ce qu'il nous reste à faire pour afficher des jours, des minutes, etc. ?



Terminer le tuto ? 😊

Aussi, mais je pensais plutôt à convertir toutes ces secondes en durées plus courantes ; car un compte à rebours qui indique « 13459054423 secondes restantes » (par exemple), c'est pas pratique. 😊

Les conversions

Nous arrivons dans la deuxième partie du tuto, et la plus coriace. Jusqu'ici, nous avons construit un script qui nous renvoie le temps restant en secondes. Et ce temps là, nous allons le transformer en quelque chose de plus lisible.

Vous savez quoi ? Je vais vous laisser chercher un peu ! 😊



- utilisez le **modulo %** pour prendre un reste d'une division ;
- utilisez **floor()** pour arrondir ;
- pensez à la division par **60** !

Pom... pom... pom... Fini ? 🤖

Allez, ne vous torturez pas plus, je vais vous expliquer ma solution : j'ai divisé le **temps restant par 60** pour obtenir les minutes, puis les **minutes par 60** pour obtenir les heures, et pour finir, **les heures par 24** pour obtenir les jours. Ce qui me donne ça :

Code : PHP

```
<?php
$i_restantes = $tps_restant / 60; // Minutes restantes
$H_restantes = $i_restantes / 60; // Heures restantes
$d_restants = $H_restantes / 24; // Jours restants
?>
```

À ce stade, j'espère que vous avez déjà écarquillé les yeux sur ma façon de faire, qui a l'air fausse. **Sinon, relisez !** Car en effet elle l'est. Pourquoi ? Car en fin de compte, c'est comme si notre compteur nous disait :

Citation

*Il reste **tant d'heures**, ou si vous préférez **tant de minutes**, ou...*

Et ainsi de suite. Cette méthode revient à traduire le timestamp restant en plusieurs unités de mesure temporelle. Vous suivez ?



Nous ce qu'on veut c'est que SI il reste 32 heures, le compte à rebours indique « 1 jour et 8 heures restantes ». 😊

Pour ça, on va prendre le reste des divisions grâce au **modulo (%)**. C'est-à-dire que s'il y a un peu plus d'un jour, on convertira le reste en heures.

Vous avez compris ? Application du raisonnement :

Code : PHP

```
<?php
$H_restantes = $H_restantes % 24;
?>
```

C'est au reste de la division de \$H_restantes par 24 qu'est maintenant égal \$H_restantes. 😊

Pour finir, appliquons ces changements avec un petit mixage « **floor() + %** » pour arrondir les restes :

Code : PHP

```
<?php
$s_restantes = floor($tps_restant % 60); // Secondes restantes
$i_restantes = floor($i_restantes % 60); // Minutes restantes
$H_restantes = floor($H_restantes % 24); // Heures restantes
$d_restants = floor($d_restants); // Jours restants
?>
```

Si vous avez mal compris cette partie, relisez-la, c'est important d'avoir bien compris mon raisonnement. Je sais que c'est dur de comprendre un code que vous n'avez pas fait, mais c'est important de ne pas recopier bêtement. Je vous parle en connaissance de cause, c'est d'ailleurs ce qui m'a amené à faire ce tutoriel. Je ne comprenais rien à ce que faisaient les autres, alors j'ai essayé de réfléchir par moi-même. 😊

Finalisation & améliorations

Nous voici à la dernière étape ! Comme nous avons le script de fonctionnement, rien de plus simple que d'afficher les résultats !

Code : PHP

```
<?php
echo 'Il reste exactement <strong>'. $d_restants .' jours</strong>,
<strong>'. $H_restantes .' heures</strong>,'
. ' <strong>'. $i_restantes .' minutes</strong> et <strong>'.
$s_restantes .'s</strong> avant d\'ouvrir les cadeaux <:o>.';
?>
```

Pour finir, que diriez-vous d'afficher la date dans la page ? Pour cela, nous allons utiliser les fonctions `setlocale()` et `strftime()` pour afficher la date en français :

Code : PHP

```
<?php
setlocale(LC_ALL, 'fr_FR');

echo 'Nous sommes le '. strftime('<strong>%d %B %Y</strong>, et il
est <strong>%Hh%M</strong>') .'.<br />'

?>
```

Ça y est, nous avons fini ! 🎉👑

Pour les retardataires, voici le script entier :

Code : PHP

```
<?php
$annee = date('Y');
$noel = mktime(8, 0, 0, 12, 25, $annee);

if ($noel < time())
    $noel = mktime(8, 0, 0, 12, 25, ++$annee);

$tps_restant = $noel - time(); // $noel sera toujours plus grand
que le timestamp actuel, vu que c'est dans le futur. ;)

//===== CONVERSIONS

$i_restantes = $tps_restant / 60;
$H_restantes = $i_restantes / 60;
$d_restantes = $H_restantes / 24;

$s_restantes = floor($tps_restant % 60); // Secondes restantes
$i_restantes = floor($i_restantes % 60); // Minutes restantes
$H_restantes = floor($H_restantes % 24); // Heures restantes
$d_restantes = floor($d_restantes); // Jours restants

//=====
```

```
setlocale(LC_ALL, 'fr_FR');

echo 'Nous sommes le '. strftime('<strong>%d %B %Y</strong>, et il
est <strong>%H%M</strong>') .'.<br />'

    . 'Il reste exactement <strong>'. $d_restants . ' jours</strong>,
<strong>'. $H_restantes . ' heures</strong>,'
    . ' <strong>'. $i_restantes . ' minutes</strong> et <strong>'.
$s_restantes . 's</strong> avant d\'ouvrir les cadeaux <:o>.';

?>
```

Améliorations

Maintenant, il ne vous reste plus qu'à faire un joli design de Noël. 🎅

Si vous souhaitez en savoir plus, et améliorer votre compte à rebours en voyant le temps restant défiler, rendez-vous dans l'annexe qui suit. 🤖

Vous pouvez aussi rajouter un système de mois : c'est exactement la même démarche à suivre que celle que nous avons vue ensemble. *(Et pourquoi pas des années ? Soyons fous !)*

Annexe : allons plus loin avec JavaScript !

Tiens, vous revoilà ? 🤖

Je savais que vous seriez curieux. Je vais vous apprendre ici comment faire ce compte à rebours avec le langage **JavaScript**. Ce langage s'exécute du côté utilisateur, ce qui va nous permettre de voir le temps défiler. Néanmoins, je déconseille à tous ceux ne s'y connaissant pas de s'aventurer par ici. Car le copier / coller sans rien comprendre, ça n'avance à rien !

Compétences requises :

- *connaissance minimale du JavaScript ;*
- *fonction Date().*

Bon, eh bien allons-y ! 🤖

Tout d'abord, nous allons préparer le « terrain » avec cette portion de code (x)HTML, qu'il faudra placer en haut du script :

Code : HTML

```
<body>
<p id="affichage"></p>
</body>
```

Fait ? Poursuivons !

Nous allons commencer par déclarer notre fonction CompteAREbours().

Code : JavaScript

```
function CompteAREbours ()
{
    // Notre script ici
}
```



Une fonction ? Mais on n'en avait pas besoin en PHP ! 🤖



Effectivement, mais nous avons besoin de placer notre script dans une fonction pour ensuite faire appel à une autre fonction nommée « setInterval() », qui appellera la fonction toutes les x millisecondes, et donc qui actualisera le temps restant.

Ensuite, vous verrez, rien de compliqué. En fait, nous allons poursuivre les mêmes étapes qu'en PHP : c'est-à-dire que nous allons d'abord déclarer les deux « timestamps » (pour parler PHP 😊), et ensuite la condition qui permet le compte à rebours jusqu'à Noël prochain :

Code : JavaScript

```
function CompteAREbours()
{
    var date_actuelle = new Date(); // On déclare la date
    d'aujourd'hui.
    var annee = date_actuelle.getFullYear();

    var noel = new Date(annee, 11, 25, 8, 0, 0); // On déclare la
    date de Noël.

    if (noel.getTime() < date_actuelle.getTime()) // Si Noël est
    dépassé, on passe au Noël suivant !
        noel = new Date(++annee, 11, 25, 8, 0, 0); // On re-déclare Noël
    pour qu'il ne soit pas passé.

    // Reste du script.
}
```



Remarquez que dans l'argument du mois, j'ai tapé « 11 ». Il faut tout simplement tenir compte du fait que les mois commencent à partir de 0 (tiens, comme vous, bizarre hein ? 😊), donc janvier sera égal à 0, et décembre à 11. C'est à l'origine de pas mal d'erreurs, et de casse-tête.

Maintenant, rien de plus simple, nous allons recopier bêtement la soustraction des deux timestamps ainsi que les conversions. Pour arrondir, nous allons utiliser Math.floor(). Bien sur, sans oublier qu'en JavaScript les timestamps sont exprimés en millisecondes, une division pour obtenir des secondes s'impose donc :

Code : JavaScript

```
var tps_restant = noel.getTime() - date_actuelle.getTime(); // Temps
restant en millisecondes

//===== CONVERSIONS

var s_restantes = tps_restant / 1000; // On convertit en secondes
var i_restantes = s_restantes / 60;
var H_restantes = i_restantes / 60;
var d_restants = H_restantes / 24;

s_restantes = Math.floor(s_restantes % 60); // Secondes
restantes
i_restantes = Math.floor(i_restantes % 60); // Minutes restantes
H_restantes = Math.floor(H_restantes % 24); // Heures restantes
d_restants = Math.floor(d_restants); // Jours restants
//=====
```

Nous voici presque à la fin : nous allons afficher les résultats. 😊 Pour afficher le mois en français, nous allons utiliser la bonne vieille méthode du tableau contenant les mois. Ensuite, rien de plus facile que d'afficher le texte avec `document.getElementById("affichage").innerHTML` :

Code : JavaScript

```
var mois_fr = new Array('Janvier', 'Février', 'Mars', 'Avril',  
    'Mai', 'Juin', 'Juillet', 'Août', 'Septembre', 'Octobre',  
    'Novembre', 'Décembre');  
  
var texte = "Nous sommes le <strong>" + date_actuelle.getDate() + " "  
+ mois_fr[date_actuelle.getMonth()] + " "  
+ date_actuelle.getFullYear() + "</strong>," +  
    " et il est <strong>" + date_actuelle.getHours() + "h"  
+ date_actuelle.getMinutes() + "</strong>.<br />";  
  
    texte += "Il reste exactement <strong>" + d_restants + "  
jours</strong>, <strong>" + H_restantes + " heures</strong>," +  
    " <strong>" + i_restantes + " minutes</strong> et <strong>"  
+ s_restantes + "s</strong> avant d'ouvrir les cadeaux <:o>.";  
  
document.getElementById("affichage").innerHTML = texte;
```

Pour finir, nous allons faire en sorte de rappeler la fonction toutes les secondes, pour qu'elle se mette à jour, avec **setInterval(fonction, millisecondes)** (Sans oublier de fermer la fonction avant !):

Code : JavaScript

```
}  
setInterval(CompteAREbours, 1000); // Rappel de la fonction toutes  
les 1000 millisecondes (toutes les secondes quoi !).
```

Et voilà le travail ! Vous devriez obtenir [ce résultat](#). Encore une fois, il est possible d'améliorer ce script pour qu'il rende mieux. Moi, par exemple, j'ai affiché les dixièmes de secondes après la virgule, ça rend sympa en JavaScript ! En plus, vous connaissez déjà le principe, c'est pas bien dur. 🤔



Notez que si vous souhaitez afficher les dixièmes de secondes, n'oubliez pas de diminuer le délai de la fonction **setInterval** à 100.



Attention, le script en JavaScript calcule le temps restant à partir de l'heure du visiteur, contrairement au script PHP, qui affiche en fonction du serveur !

Voilà tout ! Si vous avez des questions, n'hésitez pas à me contacter par [MP](#), et me faire des remarques par le biais des commentaires.

Bonne continuation à vous ! 😊

Partager

