

La manipulation des dates

Table des matières

I. Contexte	3
II. Le timestamp Unix	3
III. Exercice : Appliquez la notion	6
IV. Le timestamp en PHP	6
V. Exercice : Appliquez la notion	10
VI. Afficher une date depuis un timestamp Unix	10
VII. Exercice : Appliquez la notion	12
VIII. Essentiel	12
IX. Auto-évaluation	12
A. Exercice final	12
B. Exercice : Défi.....	14
Solutions des exercices	15

I. Contexte

Durée : 1 h

Environnement de travail : repl.it

Pré-requis : Bases de PHP, boucles, tableaux

Contexte

Il est courant que les fonctionnalités des applications nécessitent une gestion ou une manipulation de dates, pour un tri de données ou un affichage, par exemple. Cependant, une date peut avoir différents formats, prendre en compte un fuseau horaire spécifique ou suivre un système d'heure d'été ou d'hiver selon les pays.

Manipuler des dates est donc un mécanisme fastidieux qui nécessite de prendre en compte de nombreux facteurs, comme la gestion du nombre de jours dans un mois ou encore la gestion des années bissextiles.

Nous allons voir ici qu'il existe un format de données permettant d'unifier la gestion des dates, le **timestamp Unix**. Puis nous décrirons comment PHP propose différentes fonctions permettant d'en faire une gestion fine, en nous fournissant des dates directement dans ce format et en nous offrant la possibilité de formater ce mode de données en une valeur humainement compréhensible.

II. Le timestamp Unix

Objectif

- Découvrir le timestamp Unix

Mise en situation

Une date est une donnée numéraire suivant un format et une notation complexe, qui diffèrent en fonction des pays. Cette disparité de la gestion des dates pose problème lors du développement d'applications. Il existe cependant un format de date unifié qui est devenu la référence d'utilisation pour les développements informatiques, le **timestamp Unix**. Nous allons ici voir ce qu'il représente et les avantages qu'il apporte.

Le timestamp Unix

Le format **timestamp Unix** est une représentation numérique du nombre de secondes écoulées entre une date spécifique et le 1^{er} janvier 1970. Il permet par conséquent de s'affranchir des notions de fuseaux horaires ou de jours.

Ainsi, la date du 1^{er} janvier 2020 00:00:00 est représentée par la valeur 1577836800.

En se basant sur ce format pour représenter une date, nous pouvons facilement effectuer des opérations de comparaison ou de tri. Il ne s'agit ni plus ni moins que de les effectuer sur des entiers.

Exemple

Comparaison de dates au format timestamp Unix

```
1 <html>
2 <body>
3 <?php
4     $startDate = 1577836800; // 01/01/2020 00:00:00
5     $endDate = 1577901600; // 01/01/2020 18:00:00
6
```

```

7     if ($startDate < $endDate) {
8         echo 'La date de départ est avant celle de fin';
9     }
10    ?>
11    </body>
12 </html>

```

La date de départ est avant celle de fin

L'utilisation du format timestamp Unix permet de comparer des dates de manière très simple.

```

PHP 7.2.17-0ubuntu0.18.04.1 (cli) (built: Apr 18 2019 14:12:38) ( NTS )
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.2.0, Copyright (c) 1998-2018 Zend Technologies
> php -S 0.0.0.0:8000 -t
PHP 7.2.24-0ubuntu0.18.04.3 Development Server started at Thu Mar 26 11:05:50 2020
Listening on http://0.0.0.0:8000
Document root is /home/runner/NextAgileSpreadsheet
Press Ctrl-C to quit.
[Thu Mar 26 11:05:51 2020] 172.16.0.1:80230 [200]: /

```

Exemple Tri de dates

```

1 <html>
2 <body>
3 <?php
4     $startDate = 1577836800; // 01/01/2020 00:00:00
5     $endDate = 1577901600; // 01/01/2020 18:00:00
6     $randomDate = 1199145600; // 01/01/2008 00:00:00
7     $dates = [$startDate, $endDate, $randomDate];
8
9     echo 'tableau initial :<br>';
10    foreach ($dates as $date) {
11        echo $date.'<br>';
12    }
13
14    sort($dates);
15    echo 'tableau trié :<br>';
16    foreach ($dates as $date) {
17        echo $date.'<br>';
18    }
19    ?>
20 </body>
21 </html>

```

```
tableau initial :  
1577836800  
1577901600  
1199145600  
tableau trié :  
1199145600  
1577836800  
1577901600
```

L'utilisation du format timestamp Unix permet de trier des dates facilement.

```
PHP 7.2.17-Submodul-18-04-1 (cli) (built: Apr 18 2019 14:12:38) ( NTS )  
Copyright (c) 1997-2018 The PHP Group  
Zend Engine v3.2.0, Copyright (c) 1998-2018 Zend Technologies  
> php -S 0.0.0.0:8000 -t .  
PHP 7.2.17-Submodul-18-04-1 Development Server started at Thu Mar 26 11:11:04 2020  
Listening on http://0.0.0.0:8000  
Document root is /home/runner/HextAgileSpreadsheet  
Press Ctrl-C to quit.  
[Thu Mar 26 11:11:07 2020] [72.18.0.1:58096 [200]]: /
```

Attention Le bug de l'an 2038

Le format timestamp Unix étant une représentation entière du nombre de secondes écoulées depuis le 1^{er} janvier 1970, il est logique de considérer son évolution dans le temps.

En effet, selon les systèmes, la représentation des entiers n'est pas la même. Les systèmes 32 bits stockent majoritairement cette donnée sous la forme d'un entier signé sur 32 bits, dont la valeur maximale ne peut pas dépasser $2^{31} - 1$, soit 2 147 483 647. Cette valeur est donc la valeur maximale que peut atteindre un timestamp Unix au sein d'un système 32 bits, la date qui lui est associée est le 19/01/2038 à 03:14:07.

Concrètement, cela signifie que, le 19 janvier 2038 à 3 h 14 m 08 s, tous les systèmes 32 bits se basant encore sur des entiers signés sur 32 bits afficheront une date erronée, ce qui causera de nombreuses erreurs de fonctionnement.

Les systèmes 64 bits n'ont pas ce problème : ils peuvent gérer des dates jusqu'à l'année 292 277 026 596.

Syntaxe À retenir

- Afin d'éviter une gestion complexe des formats de dates liés aux spécificités locales, nous pouvons utiliser un format numérique unifié, le timestamp Unix.

Complément

Timestamp Unix¹

Bug de l'an 2038²

1 https://fr.wikipedia.org/wiki/Heure_Unix

2 https://fr.wikipedia.org/wiki/Bug_de_l'an_2038

III. Exercice : Appliquez la notion

Nous avons pu voir qu'il était possible de comparer des timestamps entre eux.

Vous disposez du tableau de timestamps suivant :

```
1 <?php
2
3 $timestamps = [1654863436, 1407673368, 1581337036, 1644495436, 1399724236, 1586521368,
4 1628598168];
```

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



Question 1

[solution n°1 p.17]

À l'aide de ce site², déterminez le timestamp de la date de votre choix, et assignez sa valeur à une variable que vous nommerez `$comparisonTimestamp`.

Question 2

[solution n°2 p.17]

En partant de la solution de la question précédente, effectuez un tri dans le tableau afin de ne conserver que les valeurs inférieures à `$comparisonTimestamp`.

Aidez-vous des fonctions internes de PHP concernant la gestion des tableaux.

Indice :

Vous pouvez vous aider de la fonction `array_filter` et créer une fonction de comparaison.

Indice :

Aidez-vous du mot-clé "use" afin d'utiliser un paramètre externe dans votre fonction de comparaison.

IV. Le timestamp en PHP

Objectif

- Apprendre à convertir une date en un timestamp Unix

Mise en situation

Afin de manipuler des dates au cours de nos développements, nous allons avoir besoin de les obtenir au format timestamp Unix. PHP propose nativement de nombreuses fonctions de récupération et de manipulation de dates, que nous allons voir ici.

1 <https://repl.it/>

2 <https://123timestamp.fr/>

Récupérer la date actuelle

PHP propose plusieurs fonctions permettant de récupérer une date au format timestamp Unix :

- La fonction `time()`, qui retourne la date courante du serveur au format timestamp Unix.
- La fonction `microtime()`, qui retourne la date courante du serveur au format timestamp Unix en précisant les microsecondes. Le résultat retourné est une chaîne de caractères dont la première partie représente le nombre de microsecondes écoulées exprimé en secondes, tandis que la seconde partie est le timestamp Unix exprimé lui aussi en secondes. Il est possible de passer la valeur `true` à la fonction `microtime` pour qu'elle retourne la date actuelle sous forme d'un nombre flottant exprimé en secondes depuis le 1^{er} janvier 1970.

Exemple

```
1 <html>
2   <body>
3     <?php
4       echo 'time : '.time().<br>';
5       echo 'microtime : '.microtime().<br>';
6       echo 'microtime flottant : '.microtime(true);
7     ?>
8   </body>
9 </html>
```



```
time : 1585230156
microtime : 0.23188700 1585230156
microtime flottant : 1585230156.2319
```

L'exemple présenté montre comment récupérer le timestamp Unix de la date actuelle.

Convertir une date au format timestamp Unix

Lorsque nous devons manipuler des dates autres que la date actuelle, PHP nous offre la possibilité de convertir une date pour nous retourner sa valeur au format timestamp Unix.

Le fonction `mktime()` permet de récupérer le timestamp d'une date voulue.

À l'origine elle permet de calculer le nombre de secondes en fonction du 1er janvier 1970 et la date donnée. Mais elle est plus couramment utilisée pour calculer le nombre de secondes entre deux dates données (il faut donc donner deux timestamps). Cette fonction prend donc plusieurs paramètres en compte afin de définir chaque composante de la date souhaitée. On la retrouve donc sous la forme: `mktime(heures, minutes, secondes, mois, jour, année)`.

Depuis le passage à PHP 8.0.0 le paramètre heure est obligatoire. Par conséquent si la fonction `mktime()` n'est pas remplie alors elle retournera false. Tous les autres paramètres laissés vide après la définition de l'heure prendront la valeur courante au moment de l'utilisation de la fonction. Un paramètre peut être positif ou négatif. La valeur 0 représente 00:00 l'équivalent de minuit.

`mktime(-2)` avec une valeur négative sur le paramètre heure cela nous indique que le timestamp se calcule à -2h du point de départ du jour actuel soit 22h de la journée d'hier (on peut considérer minuit comme le point 0 entre 2 journées)

`mktime(9)` correspond à 9 heure de ce jour, les paramètres “minutes, secondes, mois, jour, année” étant laissé vide ils prendront automatiquement la valeur minutes seconde mois jour et année au moment de l'appel de la fonction.

```
1 <code>
2 <?php
3
4 // On calcul le timeStamP et on le met au format souhaité avec date()
5 echo "Réalisé le 02/12/2021 à 14h15 : ".date('Y-m-d H:i:s', mktime(-2))."<br>";
6 // affiche Réalisé le 02/12/2021 à 14h15 : 2021-12-01 22:15:32
7
8 echo "Réalisé le 02/12/2021 à 14h15 : ".date('Y-m-d H:i:s', mktime(2));
9 // affiche Réalisé le 02/12/2021 : 2021-12-01 02:15:32
10
11 ?>
12
13 </code>
```

Exemple

```
1 <html>
2 <body>
3 <?php
4     echo 'time : '.time().'<br>';
5     echo '01/01/2020 00:00:00 : '.mktime(0,0,0,1,1,2020);
6 >
7 </body>
8 </html>
```

```
time 1585231090
mktime: 1585231090
01/01/2020 00:00:00: 1577356500
```

On peut constater que la fonction `mktime()` retourne bien le timestamp Unix courant par défaut et le timestamp Unix correspondant à la décomposition des composantes d'une date ciblée passées en paramètres.

```
PHP 7.2.17-0ubuntu0.18.04.1 (cli) (built: Apr 18 2019 14:12:38) (NTS)
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.2.0, Copyright (c) 1998-2018 Zend Technologies
> php -S 0.0.0.0:8000 -t .
PHP 7.2.24-0ubuntu0.18.04.3 Development Server started at Thu Mar 26 13:58:09 2020
Listening on http://0.0.0.0:8000
Document root is /home/runner/NextAgileSpreadsheet
Press Ctrl-C to quit.
[Thu Mar 26 13:58:10 2020] 172.16.0.1:49610 [200]: /
```

Remarque

La fonction `mktime()` présente l'avantage de corriger les erreurs qui peuvent survenir lors de l'utilisation de valeurs issues de calculs sur des composantes de la date. En effet, si une erreur est présente sur une des composantes, `mktime()` va automatiquement réaliser l'opération nécessaire pour obtenir une date valide à condition que la paramètre heure soit au minimum renseigné.

Exemple

```

1 <html>
2   <body>
3     <?php
4       echo '03/01/2020 00:00:00 : '.mktime(0, 0, 0, 12, 34, 2019).'

```

```

03/01/2020 00:00:00 : 1578009600
01/12/2018 00:00:00 : 1543622400
01/11/2018 00:00:00 : 1541030400
01/11/2018 00:00:00 : 1577836800
01/01/2020 00:00:00 : 1577836800

```

```

PHP 7.2.17-0ubuntu0.18.04.1 (cli) (built: Apr 18 2019 14:12:38) ( NTS )
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.2.0, Copyright (c) 1998-2018 Zend Technologies
[PHP 7.2.17-0ubuntu0.18.04.3 Development Server started at Thu Mar 26 14:21:50 2020]
Listening on http://0.0.0.0:8000
Document root is /home/runner/NextAgileSpreadsheet
Press Ctrl-C to quit.
[Thu Mar 26 14:21:50 2020] 172.18.0.1:54872 [200]: /

```

L'exemple montre que, si une valeur est supérieure à celle attendue pour la composante en cours (jour du mois supérieur à sa valeur maximale, par exemple), la fonction opérera une addition à partir de la composante suivante en réinitialisant la composante en erreur. On peut voir ainsi que, si le nombre de jours est supérieur au nombre de jours du mois ciblé, alors l'écart est reporté sur le mois suivant.

Inversement, si la valeur d'une composante est inférieure au minimum attendu, une soustraction sera effectuée sur celle-ci. Ainsi, lorsque l'on passe la valeur 0 à un mois, le mois de la date retournée correspondra au mois précédant le premier mois de l'année passée en paramètre.

Attention

La valeur 0 pour les composantes de jour ou de mois correspond à une soustraction de 1 par rapport à la composante de niveau supérieur.

Syntaxe **À retenir**

- Le langage PHP permet d'obtenir la date courante au format timestamp Unix grâce aux fonctions `time()` et `microtime()`.
- Il met aussi à notre disposition la fonction `mktime()`, qui convertit une date donnée au format timestamp Unix.

Complément

`time`¹

`microtime`²

`mktime`³

1 <https://www.php.net/manual/fr/function.time.php>

2 <https://www.php.net/manual/fr/function.microtime.php>

3 <https://www.php.net/manual/fr/function.mktime.php>

V. Exercice : Appliquez la notion

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



Question 1

[solution n°3 p.17]

Affichez les trois timestamps Unix suivants :

1. Le 21 décembre 2012 à 0 h 41 m00 s
2. Aujourd'hui à minuit
3. Aujourd'hui à midi

Question 2

[solution n°4 p.17]

En vous aidant des fonctions vues précédemment, calculez le temps d'exécution du code suivant :

```
1 $result = 1;
2
3 for ($i = 1; $i <= 10000000; $i++) {
4     $result = $i * $result;
5 }
```

VI. Afficher une date depuis un timestamp Unix

Objectif

- Apprendre à formater un timestamp Unix

Mise en situation

Les manipulations de dates se font communément en passant par le format timestamp Unix, mais celui-ci n'est pas compréhensible tel quel à la lecture par un humain. Nous allons voir que PHP nous permet de le transformer dans un format lisible grâce à la fonction `date()`.

Obtenir une valeur de date lisible

La fonction `date()` attend en paramètres une chaîne de caractères représentant le format d'affichage souhaité et, optionnellement, une date au format timestamp Unix.

La définition du format de retour souhaité se fait sous la forme d'un masque contenant un code, sous la forme d'une lettre, pour chaque composante de la date.

Ainsi, par exemple, un masque de la forme `'d/m/Y H:i:s'` affichera le jour sur deux chiffres suivi d'un '/', le mois sur deux chiffres suivi d'un '/', l'année sur quatre chiffres, puis un espace et l'heure au format 24 h sur deux chiffres suivis de ':', les minutes sur deux chiffres suivies de ':' et les secondes sur deux chiffres.

Remarque

La liste de tous les codes et leur correspondance est disponible ici : <https://www.php.net/manual/fr/function.date.php>.

1 <https://repl.it/>

Exemple

```

1 <html>
2   <body>
3     <?php
4       echo '03/01/2020 00:00:00 : '.date('d/m/Y H:i:s', mktime(0, 0, 0, 12, 34, 2019)).'<br>';
5       echo '01/12/2018 00:00:00 : '.date('d/m/y h:i:s', mktime(0, 0, 0, 0, 1, 2019)).'<br>';
6       echo '01/11/2018 00:00:00 : '.date('l jS \of F Y h:i:s A', mktime(0, 0, 0, -1, 1,
2019));
7     ?>
8   </body>
9 </html>

```

```

03/01/2020 00:00:00 : 03/01/2020 00:00:00
01/12/2018 00:00:00 : 01/12/18 12:00:00
01/11/2018 00:00:00 : Thursday 1st of November 2018 12:00:00 AM

```

Les dates au format timestamp Unix obtenues par la fonction `mktime()` sont bien converties dans un format humainement lisible.

```

exited, terminated
PHP 7.2.17-0ubuntu0.18.04.1 (cli) (built: Apr 18 2019 14:12:38) ( NTS )
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.2.0, Copyright (c) 1998-2018 Zend Technologies
> php -S 0.0.0.0:8000 -t .
PHP 7.2.24-0ubuntu0.18.04.3 Development Server started at Thu Mar 26 15:00:41 2020
Listening on http://0.0.0.0:8000
Document root is /home/runner/NextAgileSpreadsheet
Press Ctrl-C to quit.
[Thu Mar 26 15:00:41 2020] 172.18.0.1:39270 [200]: /

```

Remarque

Il est possible de combiner les fonction `date()` et `mktime()` pour réaliser des opérations sur les composantes d'une date.

Exemple

```

1 <html>
2   <body>
3     <?php
4       echo 'Demain : '.date('Y/m/d H:i:s', mktime(0, 0, 0, date('m'), date('d') + 1,
date('Y'))).'<br>';
5       echo 'Le mois dernier : '.date('Y/m/d H:i:s', mktime(0, 0, 0, date('m') - 1, date('d'),
date('Y')));
6     ?>
7   </body>
8 </html>

```

```

Demain : 2020/03/27 00:00:00
Le mois dernier : 2020/02/26 00:00:00
    
```

Dans cet exemple, chaque composante de la date est obtenue grâce à un appel à la fonction `date()` pour être passée à la fonction `mktime()` qui retourne alors un timestamp Unix, lui-même formaté par la fonction `date()` pour le rendre lisible.

Pour rappel, les paramètres attendus par la fonction `mktime()` sont des entiers. Il est donc possible d'effectuer des opérations de calcul sur chacun d'entre eux.

Cette approche permet de s'assurer que toutes les spécificités d'une date sont prises en compte lors d'un ajout ou d'une soustraction à une composante de la date.

Syntaxe À retenir

- Le langage PHP nous permet de transformer un timestamp Unix en un format humainement compréhensible grâce à la fonction `date()`.

Complément

`date`¹

VII. Exercice : Appliquez la notion

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



Question

[solution n°5 p.18]

En vous aidant de la fonction `date()` et de la documentation, affichez :

- Le numéro de la semaine en cours
- La date du jour à minuit au format ISO 8601
- Le jour textuel correspondant au 25 décembre de l'année prochaine

VIII. Essentiel

IX. Auto-évaluation

A. Exercice final

Exercice 1

[solution n°6 p.18]

Exercice

¹ <https://www.php.net/manual/fr/function.date.php>

² <https://repl.it/>

Qu'est-ce qu'un timestamp Unix ?

- ☐ Un entier représentant le nombre de millisecondes depuis le 1^{er} janvier 1970
- ☐ Un entier représentant le nombre de millisecondes depuis le 1^{er} janvier 1960
- ☐ Un entier représentant le nombre de secondes depuis le 1^{er} janvier 1970
- ☐ Un entier représentant le nombre de secondes depuis le 1^{er} janvier 1960

Exercice

Comment peut-on représenter le timestamp Unix d'une date des années 1940 ?

- ☐ On ne peut pas
- ☐ Avec un nombre négatif

Exercice

Quelle fonction PHP permet de récupérer uniquement le timestamp Unix courant ?

Exercice

Quelle fonction PHP permet de récupérer le timestamp Unix d'une date donnée ?

Exercice

Quelle fonction PHP permet d'afficher une date dans un format humainement lisible ?

Exercice

J'ai besoin de calculer le temps d'exécution de mon script en millisecondes : quelle fonction PHP dois-je utiliser pour récupérer la date de début ?

- ☐ time(true)
- ☐ time(false)
- ☐ microtime(true)
- ☐ microtime(false)

Exercice

Comment récupère-t-on le timestamp Unix du 31 mars 2020 à minuit ?

- ☐ mktime(2020, 3, 31, 0, 0, 0))
- ☐ mktime(0, 0, 0, 31, 3, 2020))
- ☐ mktime(0, 0, 0, 3, 31, 2020))
- ☐ mktime(0, 0, 0, 4, 0, 2020))

Exercice

Comment récupérer-t-on le timestamp Unix d'hier à 8 h 15 ?

- ☐ mktime(8, 15, 0, null, -1)
- ☐ mktime(8, 15, 0, date('n'), -1, date('Y'))
- ☐ mktime(8, 15, 0, date('m'), date('d') - 1, date('Y'))
- ☐ mktime(8, 15, 0, date('n'), date('j') - 1, date('Y'))

Exercice

Quel masque permet d'afficher une date au format `mercredi 1 avril 2020` ?

Exercice

Quel masque permet d'afficher une heure au format `08h05 EDT` ?

B. Exercice : Défi

Dans cet exercice, nous allons voir comment afficher le calendrier du mois en cours sous forme de tableau.

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



Question

[solution n°7 p.19]

Vous devez constituer un tableau contenant :

- en en-tête, le mois et l'année en cours
- suivis des jours de la semaine
- puis des jours du mois répartis à raison d'une semaine par ligne

1 <https://repl.it/>

Le résultat devrait être le suivant :

04/2020

L	M	M	J	V	S	D
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

Solutions des exercices

p. 6 Solution n°1

Dans notre cas, le timestamp sera 1586521036.

```

1 <?php
2
3 <?php
4
5 $timestamps = [1654863436, 1407673368, 1581337036, 1644495436, 1399724236, 1586521368,
6 1628598168];
7 $comparisonTimestamp = 1586521036;
```

p. 6 Solution n°2

Voici ce que vous avez dû obtenir :

```

1 <?php
2
3 $timestamps = [1654863436, 1407673368, 1581337036, 1644495436, 1399724236, 1586521368,
4 1628598168];
5 $comparisonTimestamp = 1586521036;
6
7 $comparisonFunction = function ($timestamp) use ($comparisonTimestamp) {
8     return $timestamp <= $comparisonTimestamp;
9 };
10
11 print_r(array_filter($timestamps, $comparisonFunction));
```

p. 10 Solution n°3

```

1 <?php
2
3 // 1
4 echo mktime(0, 41, 0, 12, 21, 2012) . PHP_EOL;
5
6 // 2
7 echo mktime(0, 0, 0) . PHP_EOL;
8
9 // 3
10 echo mktime(12, 0, 0);
```

p. 10 Solution n°4

```

1 <?php
2
3 $start = microtime(true);
4
5 $result = 1;
6
7 for ($i = 1; $i <= 100000000; $i++) {
8     $result = $i * $result;
9 }
```

```
10
11 echo 'Script terminé en ' . (microtime(true) - $start) . ' secondes';
```

p. 12 Solution n°5

```
1 <?php
2
3 // 1
4 echo 'Nous sommes en semaine ' . date('W') . PHP_EOL;;
5
6 // 2
7 echo 'La date d\'aujourd\'hui au format ISO 8601 est : ' . date('c', mktime(0, 0, 0)) .
    PHP_EOL;;
8
9 // 3
10 $nextXmas = mktime(0, 0, 0, 12, 25, date('Y') + 1);
11 echo 'Le 25 décembre ' . date('Y', $nextXmas) . ' nous serons un ' . date('l', $nextXmas) .
    PHP_EOL;;
```

Exercice p. 12 Solution n°6

Exercice

Qu'est-ce qu'un timestamp Unix ?

- ☐ Un entier représentant le nombre de millisecondes depuis le 1^{er} janvier 1970
- ☐ Un entier représentant le nombre de millisecondes depuis le 1^{er} janvier 1960
- ☒ Un entier représentant le nombre de secondes depuis le 1^{er} janvier 1970
- ☐ Un entier représentant le nombre de secondes depuis le 1^{er} janvier 1960

Exercice

Comment peut-on représenter le timestamp Unix d'une date des années 1940 ?

- ☐ On ne peut pas
- ☒ Avec un nombre négatif

Exercice

Quelle fonction PHP permet de récupérer uniquement le timestamp Unix courant ?

time

Exercice

Quelle fonction PHP permet de récupérer le timestamp Unix d'une date donnée ?

mktime

Exercice

Quelle fonction PHP permet d'afficher une date dans un format humainement lisible ?

date

Exercice

J'ai besoin de calculer le temps d'exécution de mon script en millisecondes : quelle fonction PHP dois-je utiliser pour récupérer la date de début ?

- ☐ time(true)
- ☐ time(false)
- ☒ microtime(true)
- ☐ microtime(false)

Exercice

Comment récupère-t-on le timestamp Unix du 31 mars 2020 à minuit ?

- ☐ mktime(2020, 3, 31, 0, 0, 0)
- ☐ mktime(0, 0, 0, 31, 3, 2020)
- ☒ mktime(0, 0, 0, 3, 31, 2020)
- ☒ mktime(0, 0, 0, 4, 0, 2020)

Exercice

Comment récupère-t-on le timestamp Unix d'hier à 8 h 15 ?

- ☐ mktime(8, 15, 0, null, -1)
- ☐ mktime(8, 15, 0, date('n'), -1, date('Y'))
- ☒ mktime(8, 15, 0, date('m'), date('d') - 1, date('Y'))
- ☒ mktime(8, 15, 0, date('n'), date('j') - 1, date('Y'))

Exercice

Quel masque permet d'afficher une date au format mercredi 1 avril 2020 ?

l j F Y

Exercice

Quel masque permet d'afficher une heure au format 08h05 EDT ?

H\hi T

p. 14 Solution n°7

```

1 <?php
2
3 $currentMonth = (int) date('m');
4 $currentYear = date('Y');
5 $numDaysInMonth = (int) date('t');
6 $firstDayOfMonth = (int) date('N', mktime(0, 0, 0, $currentMonth, 1, $currentYear));
7
8 ?>
9 <table>
10     <caption><?php echo date('m/Y') ?></caption>
11     <thead><tr><th>L</th><th>M</th><th>M</th><th>J</th><th>V</th><th>S</th><th>D</th></tr>
    </thead>

```

```

12     <tbody>
13     <tr>
14         <?php
15
16         if (1 !== $firstDayOfMonth) {
17             echo '<td colspan="' . ($firstDayOfMonth - 1) . '"></td>';
18         }
19
20         for ($i = 1; $i <= $numDaysInMonth; $i++) {
21             echo '<td>'. $i . '</td>';
22
23             if ((int) date('N', mktime(0, 0, 0, $currentMonth, $i, $currentYear)) === 7)
24         {
25             echo '</tr><tr>';
26         }
27
28         $daysLeft = ($numDaysInMonth + $firstDayOfMonth) % 7;
29         if (0 !== $daysLeft) {
30             echo '<td colspan="' . ((7 - $daysLeft) + 1) . '"></td>';
31         }
32
33         ?>
34     </tr>
35 </tbody>
36 </table>

```