

## LABORATOIRE 2

### PREMIER CONTACT AVEC LE LANGAGE *PHP*

---

#### 1. Présentation

L'objectif de ce laboratoire est de prendre un premier (vrai) contact avec le langage *PHP*.

#### 2. Pour commencer

1. Créez le dossier **L:\420-4W5\laboratoire02**, puis démarrez l'application **NetBeans IDE**.

3. Création du projet *NetBeans IDE*;  
Modification de la configuration pour exécution sur le serveur;  
Création d'un fichier *index.php*

##### a. Création du projet

1. Cliquez sur **File/New Project**, assurez-vous que c'est la **catégorie PHP** qui est sélectionnée, cliquez sur **PHP Application with Existing Sources** du volet *Projects*, puis cliquez sur **NEXT>**.
2. En utilisant le bouton **BROWSE**, sélectionnez le dossier **L:\420-4W5\laboratoire02**, assurez-vous que le codage est bien **UTF-8**, puis cliquez sur **NEXT>**.

##### b. Modification des paramètres de configuration (1<sup>re</sup> partie)

1. Modifiez le contenu de la rubrique *Run As* pour **Remote Web Site (FTP, SFTP)**. De nouvelles rubriques s'affichent.
2. Sélectionnez la connexion **424w.cgodin.qc.ca** de la liste *Remote Connection*. Si la connexion n'apparaît pas, effectuez les étapes de la section *Définition des paramètres de connexion* ci-dessous; autrement poursuivez à l'étape *Modification des paramètres de configuration* (2<sup>e</sup> partie).

##### c. Définition des paramètres de connexion

1. Cliquez sur le bouton **MANAGE**.
2. Cliquez sur le bouton **ADD** (il est possible que cette manœuvre ne soit pas nécessaire), entrez **424w.cgodin.qc.ca** dans la zone *Connection Name*, puis confirmez le changement.
3. Complétez les rubriques à partir des informations du tableau ci-dessous...

Rubriques	Valeur
<i>Host Name</i>	<b>424w.cgodin.qc.ca</b>
<i>User Name</i>	<b>N.omUtilisateur</b> <span style="float: right;"><i>Référez-vous au laboratoire 1</i></span>

<i>Password</i>	<b>MotDePasse</b>
<i>Initial directory</i>	<b>/NomUtilisateur</b> <i>Référez-vous au laboratoire 1</i>
<i>Passive Mode</i>	<input checked="" type="checkbox"/>

4. Cliquez sur le bouton **TEST CONNECTION** pour vous assurer que les paramètres ont été définis correctement.
5. Confirmez les changements apportés.

#### d. Modification des paramètres de configuration (2e partie)

1. Complétez les rubriques à partir des informations du tableau ci-dessous...

<b>Rubriques</b>	<b>Valeur</b>
<i>Project URL</i>	<b>http://424w.cgodin.qc.ca/nomutilisateur/laboratoire02</b>
<i>Index File</i>	<b>index.php</b>
<i>Remote connection</i>	<b>424w.cgodin.qc.ca</b>
<i>Upload directory</i>	<b>/laboratoire02</b>
<i>Upload Files</i>	<b>On Run</b>

2. Cliquez sur le bouton **FINISH**.


#### e. Création d'un fichier *index.php*

Puisque nous avons spécifié que le fichier exécuté par défaut sera **index.php** et qu'il n'existe pas encore, nous devons le créer.

1. Cliquez droit sur le dossier **Source Files** du projet *laboratoire02* pour afficher son menu contextuel, puis sur **New/PHP File...**
2. Remplacez le nom actuel (*newEmptyPHP*) pour « **index** ». Ne saisissez pas le suffixe **.php**, car il sera inséré automatiquement (voir la zone *Created File*).
3. Cliquez sur **FINISH**.
4. Remplacez le contenu actuel du fichier pour...

```
<?php
    echo "Laboratoire 2";
?>
```

#### 4. Téléversement des fichiers du projet sur le serveur; Exécution de l'application *index.php*

1. Cliquez sur le bouton **RUN PROJECT**  de la barre d'outils *Run* pour déclencher le processus de téléversement. La boîte de dialogue *File Upload* s'affiche.
2. Assurez-vous que la case **index.php** est cochée, puis cliquez sur le bouton **UPLOAD**. Le téléversement s'effectue, puis après quelques secondes l'application s'exécute. Le message *Laboratoire 2* s'affiche dans la fenêtre du navigateur.

Notez que l'URL qui s'affiche est :

<http://424w.cgodin.qc.ca/nomutilisateur/laboratoire02/index.php>

3. Fermez la fenêtre du navigateur.

Nous pouvons commencer !

## 5. Présentation du problème : Affichage de la date courante

On vous demande de concevoir une application permettant d'afficher la date courante de façon formatée.

Deux exemples de sortie à produire :

Nous sommes le lundi 1<sup>er</sup> janvier 2018.  
Nous sommes le samedi 15 décembre 2018.

Réf. : <http://424w.cgodin.qc.ca/lmbrousseau/laboratoire02/index.php>

### a. Récupération de la date courante en utilisant la fonction *date()*

Comme tout bon langage de programmation qui se respecte, *PHP* permet de récupérer la date courante.

Réf. : <http://php.net/manual/fr/function.date.php>

1. Saisissez le code source ci-dessous en remplacement du code actuel...

```
<?php
    $strDate = date("d-m-Y");
    echo "Nous sommes le $strDate.<br /><br />";
?>
```

2. Exécutez l'application sur le serveur après vous être assuré d'avoir téléversé le fichier.

Nous sommes le 26-01-2018.

Exemple de sortie !

### b. Extraction du jour, du mois et de l'année en utilisant la fonction *date()* Utilisation de la fonction *var\_dump()* pour afficher le type d'une variable

Réf. : <http://php.net/manual/fr/function.var-dump.php>  
<http://php.net/manual/fr/function.intval.php>

1. Saisissez le code source ci-dessous...

```
<?php
    $strDate = date("d-m-Y");
    echo "Nous sommes le $strDate.<br /><br />";

    /* 1re technique : Récupération du jour, mois et année à partir de la fonction date(). */
    $strJour = date("d");var_dump($strJour);
    $strMois = date("m");var_dump($strMois);
```

```

$strAnnee = date("Y");var_dump($strAnnee);
/* Conversion de chaque variable chaîne en entier */
$intJour = intval($strJour); var_dump($intJour);
$intMois = intval($strMois); var_dump($intMois);
$intAnnee = intval($strAnnee); var_dump($intAnnee);

/* Affichage de la date */
echo "Nous sommes le $strJour-$strMois-$strAnnee.<br />";
echo "Nous sommes le $intJour-$intMois-$intAnnee.<br /><br />";
?>

```

2. Exécutez l'application sur le serveur après vous être assuré d'avoir téléversé le fichier.

```

Nous sommes le 26-01-2018.
C:\wamp64\www\nomutilisateur\laboratoire02\index.php:6:string '26' (length=2)
C:\wamp64\www\nomutilisateur\laboratoire02\index.php:7:string '01' (length=2)
C:\wamp64\www\nomutilisateur\laboratoire02\index.php:8:string '2018' (length=4)
C:\wamp64\www\nomutilisateur\laboratoire02\index.php:10:int 26
C:\wamp64\www\nomutilisateur\laboratoire02\index.php:11:int 1
C:\wamp64\www\nomutilisateur\laboratoire02\index.php:12:int 2018
Nous sommes le 26-01-2018.
Nous sommes le 26-1-2018.

```

c. Extraction du jour, du mois et de l'année à partir de la variable \$strDate

Réf. : <http://php.net/manual/fr/function.substr.php>

1. Saisissez le code source ci-dessous à la suite du code déjà saisi...

```

/* 2e technique : Extraction du jour, mois et année à partir de $strDate */
$intJour2 = intval(substr($strDate, 0, 2));
$intMois2 = intval(substr($strDate, 3, 2));
$intAnnee2 = intval(substr($strDate, 6, 4));

/* Affichage de la date */
echo "Nous sommes le $intJour2-$intMois2-$intAnnee2.<br /><br />";
?>

```

2. Exécutez l'application sur le serveur après vous être assuré d'avoir téléversé le fichier.

d. Création de la fonction *convertitSousChaineEnEntier*

Créons maintenant une fonction qui nous permettra de convertir n'importe quelle portion de chaîne en entier.

1. Saisissez le code source ci-dessous à la suite du code déjà saisi...

```

/*
|-----|
| convertitSousChaineEnEntier (2018-01-jj) |
|-----|
*/
function convertitSousChaineEnEntier($strChaine, $intDepart, $intLongueur) {
    $intEntier = intval(substr($strChaine, $intDepart, $intLongueur));
    return $intEntier;
}

$intJour3 = convertitSousChaineEnEntier($strDate, 0, 2);
$intMois3 = convertitSousChaineEnEntier($strDate, 3, 2);
$intAnnee3 = convertitSousChaineEnEntier($strDate, 6, 4);

```

```

/* Affichage de la date */
echo "Nous sommes le $intJour3-$intMois3-$intAnnee3.<br /><br />";
?>

```

2. Exécutez l'application sur le serveur.

### e. Création de la fonction *extraitJJMMAAAA*

Réf. : <http://php.net/manual/fr/function.func-num-args.php>  
<http://php.net/manual/fr/function.func-get-arg.php>

Créons maintenant une fonction qui nous permettra d'extraire, sous forme d'entiers, le jour, le mois et l'année de l'année courante ou de la date spécifiée.

1. Saisissez le code source ci-dessous à la suite du code déjà saisi...

```

/*
-----|
extraitJJMMAAAA (2018-01-jj)
Scénarios : extraitJJMMAAAA($intJour, $intMois, $intAnnee)           <= date()
              extraitJJMMAAAA($intJour, $intMois, $intAnnee, $strDate) <= $strDate
-----|
*/
function extraitJJMMAAAA(&$intJour, &$intMois, &$intAnnee) {
    /* Par défaut, l'extraction s'effectue à partir de la date courante;
     * autrement elle s'effectue à partir du 4e argument spécifié à l'appel */
    if (func_num_args() == 3) {
        /* Récupération de la date courante */
        $strDate = date("d-m-Y");
    }
    else {
        /* Récupération du 4e argument */
        $strDate = func_get_arg(3);
    }
    $intJour = intval(substr($strDate, 0, 2));
    $intMois = intval(substr($strDate, 3, 2));
    $intAnnee = intval(substr($strDate, 6, 4));
}

$intJour4;
$intMois4;
$intAnnee4;
/* Extraction à partir de la date courante */
extraitJJMMAAAA($intJour4, $intMois4, $intAnnee4);
echo "Nous sommes le $intJour4-$intMois4-$intAnnee4.<br /><br />";

$intJour5;
$intMois5;
$intAnnee5;
$strDate5 = "31-12-2018";
/* Extraction à partir de la date spécifiée */
extraitJJMMAAAA($intJour5, $intMois5, $intAnnee5, $strDate5);
echo "Nous sommes le $intJour5-$intMois5-$intAnnee5.<br /><br />";
?>

```

2. Exécutez l'application sur le serveur.

Quelques remarques :

- Le symbole « & » précédant les variables *intJour*, *intMois* et *intAnnee* dans l'en-tête de la fonction indique que ces dernières seront passées en référence. Notez que le symbole n'apparaît pas à l'appel.
- Même si nous n'avons spécifié que 3 paramètres lors de la déclaration de la fonction, il sera possible d'en passer 4 à l'appel.
- C'est la fonction **func\_num\_args()** qui nous permettra de déterminer le nombre d'arguments saisis à l'appel, alors que la fonction **func\_get\_arg()** nous permettra de récupérer ledit paramètre excédentaire s'il est présent.
- La fonction *convertitSousChaineEnEntier* n'a pas été utilisée pour permettre à la fonction **extraireJJMMAAAA** d'être la plus portable possible. Notez que ce choix est arbitraire.

f. Création de la fonction *moisEnLitteral*

Réf. : <http://php.net/manual/fr/function.ucfirst.php>

Créons maintenant une fonction qui nous permettra de retourner la forme littérale d'un mois (*janvier* à *décembre* ou *Janvier* ou *Décembre*) à partir d'un entier (**1** à **12**).

1. Saisissez le code source ci-dessous à la suite du code déjà saisi...

```

/*
-----|
moisEnLitteral (2018-01-jj)
Scénarios : moisEnLitteral($intMois)           => Première lettre en minuscule
             moisEnLitteral($intMois, $binMajuscule) => En fonction de $binMajuscule
-----|
*/
function moisEnLitteral($intMois, $binMajuscule=false) {
    /* Par défaut, la première lettre du mois s'affiche en minuscule ($binMajuscule=false)
     * Si un deuxième argument est saisi, il déterminera si la première lettre doit
     * s'afficher en majuscule ou non */
    $strMois = "N/A";
    switch ($intMois) {
        case 1 : $strMois = "janvier";break;
        case 2 : $strMois = "février";break;
        case 3 : $strMois = "mars";break;
        case 4 : $strMois = "avril";break;
        case 5 : $strMois = "mai";break;
        case 6 : $strMois = "juin";break;
        case 7 : $strMois = "juillet";break;
        case 8 : $strMois = "août";break;
        case 9 : $strMois = "septembre";break;
        case 10 : $strMois = "octobre";break;
        case 11 : $strMois = "novembre";break;
        case 12 : $strMois = "décembre";break;
    }
    /*
     * if ($binMajuscule)
     *     $strMois = ucfirst($strMois);
     */
    $strMois = $binMajuscule ? ucfirst($strMois) : $strMois;
    return $strMois;
}

```

N/A, N/A, N/A janvier, janvier, Janvier février, février, Février mars, mars, Mars avril, avril, Avril mai, mai, Mai juin, juin, Juin juillet, juillet, Juillet août, août, Août septembre, septembre, Septembre octobre, octobre, Octobre novembre, novembre, Novembre décembre, décembre, Décembre N/A, N/A, N/A
---

```

for ($i=0; $i<=13; $i++) {
    echo moisEnLitteral($i) . ", " . moisEnLitteral($i, false) . ", " .
        moisEnLitteral($i, true) . "<br />";
}
?>

```

2. Exécutez l'application sur le serveur.

Quelques remarques :

- Il est possible de spécifier une valeur par défaut pour un paramètre lors de la déclaration d'une fonction.

Dans l'exemple, la variable *\$binMajuscule* se verra attribuer la valeur **false** si nous n'entrons pas de deuxième argument à l'appel.

- Les caractères spéciaux **&eacute;** et **&ucirc;** doivent être insérés en lieu et place de « é » et « û » pour éviter la sortie ci-dessous :

```

février ⇒ fÃ©vrier
août ⇒ aoÃ»t
décembre ⇒ dÃ©cembre

```

### g. Exercice 1 : Création de la fonction *jourSemaineEnLitteral*

En vous référant à la fonction *moisEnLitteral*, on vous demande de concevoir la fonction **jourSemaineEnLitteral** qui permettra de retourner la forme littérale d'un jour de semaine (*lundi* à *dimanche* ou *Lundi* ou *Dimanche*) à partir d'un entier (1=**lundi** à 7=**dimanche**).

Prototype de la fonction à concevoir :

```
function jourSemaineEnLitteral($intNoJour, $binMajuscule=false)
```

Exemple d'appel

```

for ($i=0; $i<=8; $i++) {
    echo jourSemaineEnLitteral($i) . ", " .
        jourSemaineEnLitteral($i, false) . ", " .
        jourSemaineEnLitteral($i, true) . "<br />";
}

```

Sortie attendue

```

N/A, N/A, N/A
lundi, lundi, Lundi
mardi, mardi, Mardi
mercredi, mercredi, Mercredi
jeudi, jeudi, Jeudi
vendredi, vendredi, Vendredi
samedi, samedi, Samedi
dimanche, dimanche, Dimanche
N/A, N/A, N/A

```

Solution proposée :

```

/*
-----|
jourSemaineEnLitteral (2018-01-jj)
Scénarios : jourSemaineEnLitteral($intNoJour)
             jourSemaineEnLitteral($intNoJour, $binMajuscule)
-----|
*/
function jourSemaineEnLitteral($intNoJour, $binMajuscule=false) {
    /* Par défaut, la première lettre du jour de la semaine s'affiche en minuscule
     * ($binMajuscule=false). Si un deuxième argument est saisi, il déterminera si
     * la première lettre doit s'afficher en majuscule ou non */
    $strJourSemaine = "N/A";
    switch ($intNoJour) {
        case 1 : $strJourSemaine = "lundi"; break;
        case 2 : $strJourSemaine = "mardi"; break;
        case 3 : $strJourSemaine = "mercredi"; break;
        case 4 : $strJourSemaine = "jeudi"; break;
        case 5 : $strJourSemaine = "vendredi"; break;
        case 6 : $strJourSemaine = "samedi"; break;
        case 7 : $strJourSemaine = "dimanche"; break;
    }
    $strJourSemaine = $binMajuscule ? ucfirst($strJourSemaine) : $strJourSemaine;
    return $strJourSemaine;
}

echo "<br />";
for ($i=0; $i<=8; $i++) {
    echo jourSemaineEnLitteral($i) . ", " . jourSemaineEnLitteral($i, false) . ", " .
        jourSemaineEnLitteral($i, true) . "<br />";
}
?>

```

#### h. Exercice 2 : Création de la fonction *er*

On vous demande de concevoir la fonction **er** qui retourne (1) « **1er** » ou « **1<sup>er</sup>** » si le nombre 1 est passé comme argument ou (2) le « **nombre passé en argument** » s'il est différent de 1.

Prototype de la fonction à concevoir :

```
function er($intEntier, $binExposant=true)
```

Exemples d'appel

```

echo "<br />";
echo er(1) . ", ";
echo er(1, false) . ", ";
echo er(1, true) . "<br />";
echo er(31) . ", ";
echo er(31, false) . ", ";
echo er(31, true) . "<br />";

```

Sortie attendue

```

1er, 1er, 1er
31, 31, 31

```



Solution proposée :

```

/*
-----|
er (2018-01-jj)
Scénarios : er($intEntier)
            er($intEntier, $binExposant)
-----|
*/
function er($intEntier, $binExposant=true) {
    return $intEntier . ($intEntier == 1 ? ($binExposant ? "<sup>er</sup>" : "er") : "");
}

echo "<br />";
echo er(1) . ", ";
echo er(1, false) . ", ";
echo er(1, true) . "<br />";
echo er(31) . ", ";
echo er(31, false) . ", ";
echo er(31, true) . "<br />";
?>

```

### i. Exercice 3 : Création de la fonction *extraitJSJJMMAAAA*

On vous demande de créer la fonction **extraitJSJJMMAAAA** qui est une mise à niveau (*upgrade*) de la fonction *extraitJJMMAAAA*. Remarquez les lettres « JS » dans le nom de la fonction (JS=Jour de la semaine).

Prototype de la fonction à concevoir :

```
function extraitJSJJMMAAAA(&$intJourSemaine, &$intJour, &$intMois, &$intAnnee)
```

Exemples d'appel

```

$intJour6;
$intMois6;
$intAnnee6;
$intJourSemaine6;
/* Extraction à partir de la date courante */
echo "<br />";
extraitJSJJMMAAAA($intJourSemaine6, $intJour6, $intMois6, $intAnnee6);
echo "Nous sommes le " . jourSemaineEnLitteral($intJourSemaine6) . " " .
    er($intJour6) . " " . moisEnLitteral($intMois6) . " $intAnnee6.<br /><br />";

$intJour7;
$intMois7;
$intAnnee7;
$intJourSemaine7;
/* Extraction à partir de la date spécifiée */
for ($i=1; $i<=31; $i++) {
    extraitJSJJMMAAAA($intJourSemaine7, $intJour7, $intMois7, $intAnnee7,
        ($i < 10 ? "0" : "") . "$i-01-2018");
    echo "Nous sommes le " . jourSemaineEnLitteral($intJourSemaine7) . " " .
        er($intJour7) . " " . moisEnLitteral($intMois7) . " $intAnnee7.<br />";
}

```

Sortie attendue (exemple)

Nous sommes le vendredi 26 janvier 2018.

Nous sommes le lundi 1<sup>er</sup> janvier 2018.  
 Nous sommes le mardi 2 janvier 2018.  
 Nous sommes le mercredi 3 janvier 2018.  
 Nous sommes le jeudi 4 janvier 2018.  
 Nous sommes le vendredi 5 janvier 2018.  
 Nous sommes le samedi 6 janvier 2018.  
 Nous sommes le dimanche 7 janvier 2018.  
 Nous sommes le lundi 8 janvier 2018.  
 Nous sommes le mardi 9 janvier 2018.  
 Nous sommes le mercredi 10 janvier 2018.  
 Nous sommes le jeudi 11 janvier 2018.  
 Nous sommes le vendredi 12 janvier 2018.  
 Nous sommes le samedi 13 janvier 2018.  
 Nous sommes le dimanche 14 janvier 2018.  
 Nous sommes le lundi 15 janvier 2018.  
 Nous sommes le mardi 16 janvier 2018.  
 Nous sommes le mercredi 17 janvier 2018.  
 Nous sommes le jeudi 18 janvier 2018.  
 Nous sommes le vendredi 19 janvier 2018.  
 Nous sommes le samedi 20 janvier 2018.  
 Nous sommes le dimanche 21 janvier 2018.  
 Nous sommes le lundi 22 janvier 2018.  
 Nous sommes le mardi 23 janvier 2018.  
 Nous sommes le mercredi 24 janvier 2018.  
 Nous sommes le jeudi 25 janvier 2018.  
 Nous sommes le vendredi 26 janvier 2018.  
 Nous sommes le samedi 27 janvier 2018.  
 Nous sommes le dimanche 28 janvier 2018.  
 Nous sommes le lundi 29 janvier 2018.  
 Nous sommes le mardi 30 janvier 2018.  
 Nous sommes le mercredi 31 janvier 2018.

Avant de poursuivre, vous devez savoir que...

- Il est possible de récupérer le jour de la semaine de la date courante en passant l'argument « N » à la fonction **date()**.

Exemple (Date du jour : Jeudi, 25 janvier 2018) :

```
$strJourSemaine = date("N"); var_dump($strJourSemaine);
```

↓

string '4' (length=1) ⇒ Jeudi

- Pour pouvoir récupérer le jour de la semaine correspondant à une date entreposée sous la forme d'une chaîne, vous devez (1) convertir cette dernière en *timestamp Unix* en utilisant la fonction **strtotime()**, puis (2) utiliser la fonction **date()**.

Réf. : <http://php.net/manual/fr/function strtotime.php>

Exemple :

```
$strDate = "3-2-2018"; var_dump($strDate);
$intDate = strtotime($strDate); var_dump($intDate);
$strJourSemaine = date("N", $intDate); var_dump($strJourSemaine);
```

↓

string '3-2-2018' (length=8)  
 int 1517634000  
 string '5' (length=1) ⇒ Samedi

« - » comme séparateur  
 ⇒ jj-mm-aaaa

« / » comme séparateur  
 ⇒ mm/jj/aaaa

Nombre de secondes  
 depuis le 1<sup>er</sup> janvier  
 1970 à 00:00:00 UTC.

Solution proposée :

```

/*
-----|
extraitJSJJMMAAAA (2018-01-jj)
Scénarios : extraitJSJJMMAAAA($intJourSemaine, $intJour, $intMois, $intAnnee)
            extraitJSJJMMAAAA($intJourSemaine, $intJour, $intMois, $intAnnee, strDate)
-----|
*/
function extraitJSJJMMAAAA(&$intJourSemaine, &$intJour, &$intMois, &$intAnnee) {
    /* Par défaut, l'extraction s'effectue à partir de la date courante;
    * autrement elle s'effectue à partir du 5e argument spécifié à l'appel */
    if (func_num_args() == 4) {
        /* Récupération de la date courante */
        $strDate = date("d-m-Y");
        /* Récupération du jour de la semaine */
        $intJourSemaine = date("N");
    }
    else {
        /* Récupération du 5e argument */
        $strDate = func_get_arg(4);
        /* Récupération du jour de la semaine après avoir convertit la date passée
        * sous forme d'une chaîne en timestamp Unix */
        $intJourSemaine = date("N", strtotime($strDate));
    }
    $intJour = intval(substr($strDate, 0, 2));
    $intMois = intval(substr($strDate, 3, 2));
    $intAnnee = intval(substr($strDate, 6, 4));
}

$intJour6;
$intMois6;
$intAnnee6;
$intJourSemaine6;
/* Extraction à partir de la date courante */
echo "<br />";
extraitJSJJMMAAAA($intJourSemaine6, $intJour6, $intMois6, $intAnnee6);
echo "Nous sommes le " . jourSemaineEnLitteral($intJourSemaine6) . " " .
    er($intJour6) . " " . moisEnLitteral($intMois6) . " " $intAnnee6.<br /><br />";

$intJour7;
$intMois7;
$intAnnee7;
$intJourSemaine7;
/* Extraction à partir de la date spécifiée */
for ($i=1; $i<=31; $i++) {
    extraitJSJJMMAAAA($intJourSemaine7, $intJour7, $intMois7, $intAnnee7,
        ($i < 10 ? "0" : "") . "$i-01-2018");
    echo "Nous sommes le " . jourSemaineEnLitteral($intJourSemaine7) . " " .
        er($intJour7) . " " . moisEnLitteral($intMois7) . " " $intAnnee7.<br />";
}
}
?>

```

## 6. Pour terminer

Compressez le dossier **laboratoire02**, puis envoyez-le via LÉA. Avisez le professeur que vous avez terminé.

**FIN DU LABORATOIRE...**