

TD optionnel : Aller plus loin en développement Web

V2.2.2



Cette œuvre est mise à disposition selon les termes de la [licence Creative Commons Attribution – Pas d'Utilisation Commerciale – Partage à l'Identique 3.0 non transposé](#).

Document en ligne : www.mickaël-martin-nevot.com

1 Généralités

Vous devez avoir terminé l'ensemble des TD de cet enseignement avant de réaliser ce TD optionnel.

Avec l'ensemble de la formation que vous avez reçu en PHP (et en base de données), vous devriez maintenant être capable de réaliser un certain nombre d'implémentations dont celles ci-dessous.

Choisissez plusieurs de ces exercices (ou d'autres) en fonction de votre goût, de vos connaissances, de votre temps et essayez de les réaliser. Les possibilités de PHP sont très nombreuses et, avec ces notions de bases, vous pouvez aborder de nouvelles notions plus complexes et comprendre les scripts disponibles sur le Web.

Vous visualiserez systématiquement votre travail dans différents navigateurs Web.

N'oubliez pas de faire des recherches sur le Web à chaque fois que cela est nécessaire en prenant soin de vérifier que les informations trouvées soient correctes.

Vous trouverez la boîte à outils ainsi que l'ensemble des documents nécessaires à la réalisation de ce TD sur le site Web de l'enseignant.

2 Approfondissement des TD

2.1 Dynamisation du code source

Réécrivez votre/vos page(s) Web qui contient/contiennent le menu de votre site Web en utilisant des variables pour les intitulés des liens ainsi que les liens correspondants.

Par exemple, vous pouvez écrire au début du fichier quelque chose du genre :

```
<?php
$rub1 = 'Rubrique 1';
$rub1addr = 'page1.html';
```

```
?>
```

Puis, à la place de : `Rubrique 1`, vous pouvez écrire :

```
<a href="<?php echo $rubladdr; ?>" target="centre"><?php echo $rub1; ?></a>
```

De cette manière, vous pouvez dynamiser l'ensemble du code source de votre site Web.

2.2 Enregistrement d'un formulaire dans un fichier

Afin d'enregistrer les données renseignées d'un formulaire dans un fichier, commencez par créer le fichier `data.txt` dans le même répertoire que `data-processing.php` (en n'oubliant pas de lui donner les droits en écriture nécessaires au besoin).

Ajoutez un bouton submit dont le nom est « action » et la valeur « rec ». Dans le fichier `data-processing.php`, ajoutez une condition de test de la valeur du bouton pressé ; par exemple :

```
<?php
    if($action == 'mailer')
    {
        ...
    }
    elseif($action == 'rec')
    {
        ...
    }
    else
    {
        echo '<br/><strong>Bouton non géré ! </strong><br/>';
    }
?>
```

Ouvrez le fichier `data.txt` en écriture :

```
<?php
    $file= 'data.txt';
    if(!($file = fopen($file, 'a')))
    {
        echo 'Erreur d\'ouverture';
        exit();
    }
?>
```

Puis, écrivez les résultats dans le fichier ; par exemple :

```
<?php
    fputs($file, 'id : ' . $id . ', email : ' . $email . "\n");
?>
```

Enfin, fermez le fichier :

```
<?php
    fclose($file);
?>
```

Vous pouvez vérifier que l'enregistrement a bien fonctionné avec un éditeur de texte.

Créez ensuite un fichier (qui inclut le fichier `utils.inc.php`) permettant la lecture du contenu de `data.txt`.

Ouvrez le fichier en lecture; par exemple :

```
<?php
$file = 'data.txt';
if(!($file = fopen($file, 'r')))
{
    echo 'Erreur de lecture';
    exit();
}
?>
```

Puis, parcourez le fichier ligne par ligne :

```
<?php
echo 'Liste des utilisateurs : <br/>';
$cpt = 1;
while(!feof($file))
{
    $line = fgets($file, 255);
    echo 'Utilisateur n ' . $cpt . ' : ' . $line . '<br/>';
    ++$cpt;
}
?>
```

Enfin, fermez le fichier.

2.3 Base de données

2.3.1 Modifications en base de données

Ajoutez à la table `user` un champ numérique correspondant au nombre de connexions d'un utilisateur à votre site Web.

Modifiez votre code source afin que ce champ soit incrémenté d'un à chaque fois qu'un utilisateur s'authentifie (il ne doit donc pas être modifiable par un formulaire)

2.3.2 Mot de passe multiple dans la base de données

Modifiez votre code source (si nécessaire) pour avoir plusieurs utilisateurs et testez.

2.3.3 Mot de passe multiple avec accès différencié dans la base de de données

Vous pouvez aussi donner des droits différents à chaque utilisateur et donc leur donner des accès différenciés. Par exemple, faites en sorte que le premier utilisateur soit redirigé vers une page Web donnée alors que le second, par le même formulaire, soit redirigé vers une page Web différente.

2.3.4 Mise en page de données issues d'une base de données

Dans un nouveau fichier, récupérez les données des utilisateurs inscrits en base de données et affichez-les dans un tableau de manière séquentielle en affichant une balise `<table>` (en n'oubliant pas de mettre une bordure en CSS) avant la boucle et les balises `<tr><td>` au début de chaque itération, les balises `</td></tr>` en fin de chaque itération et la balise `</table>` après la boucle.

Ajoutez ensuite la fonction `use_color()` dans le fichier `utils.inc.php` et utilisez-là de manière à

coloriser alternativement le fond des lignes du tableau :

```
<?php
function use_color()
{
    static $color;
    if($color == '#7CCA62')
    {
        $color = '#B0DFA0';
    }
    else
    {
        $color = '#7CCA62';
    }
    return ($color);
}
?>
```

Remarque

Une variable locale est détruite à la sortie de la fonction où elle est déclarée, sauf si le mot clef `static` est utilisé (car dans ce cas, entre deux appels à la fonction, l'ancienne valeur est préservée), `use_color()` renverra donc une couleur vert foncé puis à l'appel suivant une couleur vert clair et ainsi de suite.

Faites en sorte de représenter l'affichage dans un tableau du genre :

Tableau 1 – Exemple d'affichage de données issues d'une base de données en tableau

Id	E-mail	Pays	Date	Nb de connexions
1	email@fai.fr	France	2020/10/01	4
3	email@fai.fr	USA	2020/10/29	5
15	email@fai.org	France	2020/12/10	2
...				

2.4 Représentation graphique des données statistiques

2.4.1 Histogrammes

2.4.1.1 Histogrammes : un première approche

Il est possible d'ajouter des histogrammes (graphiques en barre) de manière assez simple en utilisant presque uniquement HTML. La balise `` possède les attributs `height` et `width`. Ainsi `` affiche l'image avec sa taille d'origine mais `` affiche l'image avec une taille de 5 px en hauteur sur 20 px en largeur.

Créez l'image `rg.jpg`, qui n'est qu'un simple carré rouge de 2 px sur 2 px et affichez à la suite du tableau une barre rouge de 5 px de hauteur sur 120 px de largeur en « agrandissant » l'image : cela vous permet de créer un histogramme.

À la place de l'affichage du nombre de connexions dans le tableau, affichez l'image `rg.jpg` avec comme largeur la valeur de ce champ :




```
<?php
echo '<td></td>';
```

?>

Selon l'échelle des valeurs du champ choisie, le résultat peut ne pas être très probant (les images n'auront pas de différences importantes de largeur). Modifiez l'affichage en multipliant la valeur par une constante de sorte de rendre cette différence mieux visible :

```
<?php
    $width = 100 * $login_nb;
?>
```

Tableau 2 – Exemple d'histogrammes

Id	E-mail	Pays	Date	Nb de connexions
1	email@fai.fr	France	2020/10/01	
3	email@fai.fr	USA	2020/10/29	
15	email@fai.org	France	2020/12/10	
...				

Récupérez le nombre d'inscription par jour avec une requête du genre :

```
SELECT date, COUNT(*) as nb FROM user GROUP BY date
```

Attention

Pensez à utiliser le renommage de champ (en utilisant le mot clef `as`) dans une requête : ce n'est pas obligatoire en SQL mais nécessaire pour pouvoir y accéder après l'utilisation de la fonction `mysql_fetch_assoc()`.



Ajoutez un second tableau indiquant le nombre d'inscriptions par jour :

Tableau 3 – Exemple de tableau d'affichage du nombre d'inscriptions par jour

Date	Nombre d'inscriptions
2020/10/01	2
2020/10/29	1
...	

Puis modifiez ce tableau pour obtenir un histogramme vertical du genre :

Tableau 4 – Exemple d'histogramme vertical

Nombre de vote			
Date du vote	2020/10/01	2020/10/29	...

Entraînez-vous en faisant un regroupement similaire par civilité et par pays

2.4.1.2 Autre approche pour les histogrammes

Il existe une autre manière de faire des histogrammes : en créant une image dynamiquement (à la demande).

Quand un utilisateur entre l'URL d'une page Web (en PHP), le serveur Web interprète le fichier et envoie à l'utilisateur (à son navigateur Web) le résultat. Sans consigne particulière ce résultat est un

fichier au format HTML, avec l'instruction `header("Content-type: image/jpeg");` ; le résultat est, à la place, un fichier à la norme JPEG. Prenez garde de ne pas mélanger les deux car si dans un fichier avec un entête de type JPEG vous mettez des balises HTML : le résultat sera pas celui escompté !

Créez le fichier `image.php` de la manière suivante :

```
<?php
header("Content-type: image/jpeg");
// Crée une image de 200 pixels de large par 300 de haut.
$image = imagecreate(200, 300);
// Crée une couleur bleu utilisable dans l'image.
$bgcolor = imagecolorallocate($image, 0x00, 0x00, 0xFF);
// Utilise la couleur bleu créée comme couleur de fond.
imagefill($image, 0, 0, $bgcolor);
// Crée le fichier jpeg à envoyer à partir de l'image créée.
imagejpeg($image);
?>
```

Ajoutez sous vos tableaux une balise `image` faisant référence au fichier `image.php` au lieu d'utiliser classiquement un réel fichier d'image :

```
<br/>
```

Affichez dans votre navigateur Web la page `image.php`, puis votre page Web contenant les tableaux.

Remarques

Vous pouvez visualiser l'image soit par la page Web contenant la balise `` soit directement par `image.php`. Choisissez plutôt le second choix durant la phase de développement. En effet, dans le premier cas, si l'image n'est pas correctement créée, le navigateur affiche simplement un signe signifiant l'impossibilité de trouver/charger le fichier (le plus souvent une croix) à la place de celle-ci. Dans le second cas, l'image est affichée ou un message d'erreur apparaît.

2.4.1.3 Histogramme : cas pratique

Réalisez un histogramme à partir de données fixées à l'avance :

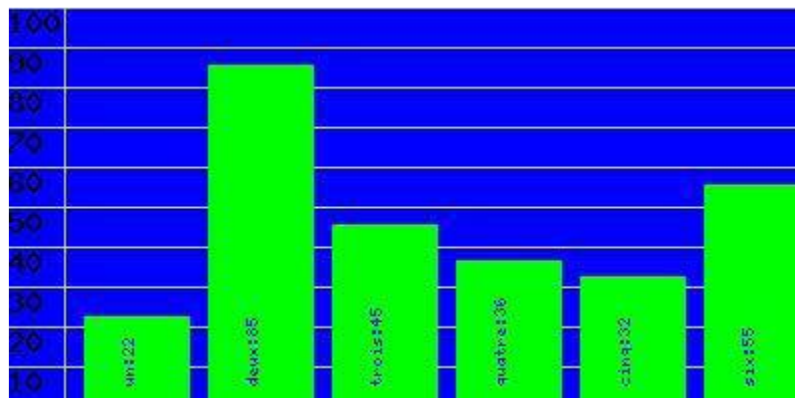


Figure 1 – Exemple d'histogramme

Pour cela, créez l'image puis fixez quelques valeurs afin de faire un histogramme de 400 px de largeur et de 200 px de hauteur avec une taille de police de cinq. Cet histogramme doit contenir six barres dont les titres et les valeurs sont dans deux tableaux distincts :

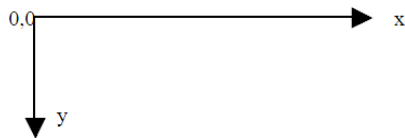
```
<?php
    $width = 400;
    $height = 200;
    $scale = 2;
    $font = 5;
    $nb = 6;
    $data = array('22', '85', '45', '36', '32', '55');
    $titles = array('un', 'deux', 'trois', 'quatre', 'cinq', 'six');
?>
```

Créez quatre couleurs, une pour le fond, une pour la grille, une pour les barres et une pour le texte, avant de colorier le fond de l'image.

Affichez maintenant la grille de fond et la légende de 10 à 100 :

```
<?php
    // Cette fonction dessine une ligne !
    imageline($image, $fromx, $fromy, $tox, $toy, $color);
?>
```

En PHP (et en informatique de manière générale), les représentations graphiques se font dans un repère orthonormé indirect :



Le point 0,0 se trouve en haut à gauche et plus le x augmente plus on se déplace à droite, plus le y augmente plus on descend !

Figure 2 – Repère orthonormé indirect

Les ordonnées des barres prennent donc leurs valeurs entre $y = 0$ et $y = hauteur - 1$. Par exemple, le rapport entre l'ordonnée et la hauteur d'une barre peut-être de deux (une ordonnée de 22 correspondant alors à une barre de 44 px de hauteur).

Les abscisses des barres prennent leur valeur entre $x = 0$ et $x = largeur - 1$. L'abscisse doit apparaître dans le libellé de la barre correspondante, écrit avec une police traille cinq et avec au maximum trois caractères (pour $x = 100$). La fonction `imagefontwidth()` permet de connaître la largeur d'un caractère écrit dans une police donnée. Calculez la largeur réelle du libellé grâce à elle :

```
<?php
    $labelwidth = imagefontwidth($font) * 3 + 1;
?>
```

Puis, écrivez le libellé en utilisant la fonction `imagestring()` et en prenant soin d'aligner convenablement (par rapport au bas) les libellés et les barres :

```
<?php
    // Écrit le texte à partir du point (x, y).
    imagestring($image, $fontnb, $x, $y, $text, $color);
?>
```

Tracez maintenant les barres, après avoir calculé leur largeur :

```
<?php
    $reewidth = (($width - $labelwidth) / $nb) - 10;
?>
```

Colorez ensuite l'intérieur des barres avec la fonction `imagefilledrectangle()` :

```
<?php
    imagefilledrectangle($image, $leftx, $stopy, $rightx, $stoty, $reccolor);
?>
```

Notez qu'il vous est aussi possible de dessiner ces barres de cette manière par exemple :

```
<?php
for($i = 0 ; $i < $nb ; ++$i)
{
    $leftx = $labelwidth + (($i + 1) * 10) + ($i * $reewidth);
    $rightx = $leftx + $reewidth;
    $stoty = $height - 1;
    $stopy = $stoty - ($data[$i] * $scale);
    imagefilledrectangle($image, $leftx, $stopy, $rightx, $stoty, $reccolor);
}
?>
```

Écrivez dans la barre le nom de la donnée en utilisant les fonctions `imagestringup()` et `imagefontheight()` :

```
<?php
    $labelx = $leftx + (($reewidth) / 2) - (imagefontheight($font) / 2);
    $labeledy = $stoty - 10; // Pour ne pas écrire immédiatement en bas.
    imagestringup($image, $font, $labelx, $labeledy, $titles[$i] . ':' . $data[$i]
], $textcolor);
?>
```

Enfin, n'oubliez pas de générer le fichier :

```
<?php
    imagejpeg($image);
?>
```

2.4.1.4 Histogramme : mise en application

Créez une nouvelle requête de regroupement (par exemple le nombre d'inscriptions par jour et par pays). Remplissez un tableau et réalisez l'histogramme associé de la manière de votre choix.

2.4.2 Diagramme circulaire

Pour créer une image de diagramme circulaire (ou camembert), le principe est le même que pour les histogrammes : il s'agit simplement d'afficher un cercle à place des barres.

Dessinez un diagramme circulaire représentant cette répartition :

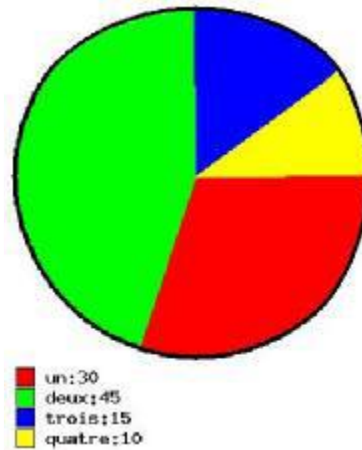


Figure 3 – Exemple de diagramme circulaire

Pour cela, commencez donc par modifier l'entête :

```
<?php
header('Content-type: image/jpeg'); // Au départ.
?>
```

Fixez ensuite quelques valeurs :

```
<?php
$diameter = null; // Diamètre du cercle.
$font = null; // Police de la légende.
$fontheight = null; // Calculé avec imagefontheight.
$nb = null; // Nombre de valeurs à mettre dans le cercle.
$width = $diameter + 20; // Pour les marges.
$height = $diameter + 40 + ($nb * $fontheight); //Pour les marges.
$xcentre = $diameter / 2 + 10;
$ycentre = $diameter / 2 + 10;
?>
```

Ensuite, initialisez la variable `$image` avec `imagecreate()` et définissez `$bgcolor`, `$bordercolor` et `$textcolor` avec `imagecolorallocate()`. Ensuite, définissez six variables correspondant à six couleurs différentes pour le diagramme circulaire : `$c1`, `$c2`, `$c3`, `$c4`, `$c5`, `$c6` et `$nbcolor`. Remplissez ensuite le fond de l'image en utilisant `imagefill($image, 0, 0, $bgcolor)`.

Pour chaque valeur donnée (initialement `$degree = 0`), préparez le calcul de la part du diagramme circulaire correspondante :

```
<?php
$initdegree = round($degree);
$degree += (($diagramvalue / 100) * 360);
$finaldegree = round($degree);
// On choisit une couleur pour la part du diagramme circulaire.
// $i vaut 0 pour la première valeur, 1 pour la suivante, etc.
switch($i)
{
    case 0:
        $slice_color = $c1;
        break;
    case 1:
        $slice_color = $c2;
```

```

        break;
    case 2:
        $slice_color = $c3;
        break;
    case 3:
        $slice_color = $c4;
        break;
    case 4:
        $slice_color = $c5;
        break;
    case 5:
        $slice_color = $c6;
        break;
}
?>

```

Tracez chaque arc de cercle correspondant à la part du diagramme circulaire :

```

<?php
    imagearc($image, $centerx, $centery, $diameter, $diameter, $initdegree, $finaldegree, $slice_color);
?>

```

Pour chaque arc de cercle, calculez les coordonnées de la première extrémité :

```

<?php
    // La fonction floor(...) permet d'obtenir l'arrondi par défaut.
    $arcx = floor($centerx + cos(deg2rad($initdegree)) * ($diameter / 2));
    $arcy = floor($centery + sin(deg2rad($initdegree)) * ($diameter / 2));
?>

```

Pour chaque arc de cercle, tracez un segment du centre du cercle jusqu'à la première extrémité :

```

<?php
    imageline($image, $centerx, $centery, $arcx, $arcy, $slice_color);
?>

```

Idem, du centre du cercle jusqu'à la seconde extrémité de l'arc :

```

<?php
    $arcx = floor($centerx + cos(deg2rad($finaldegree)) * ($diameter / 2));
    $arcy = floor($centery + sin(deg2rad($finaldegree)) * ($diameter / 2));
    imageline($image, $centerx, $centery, $arcx, $arcy, $slice_color);
?>

```

Testez :

```

<?php
    imagejpeg($image);
?>

```

Pour chaque arc de cercle, calculez son milieu :

```

<?php
    $arcx = floor($centerx + cos(deg2rad(round((($finaldegree - $initdegree) / 2) + $initdegree))) * ($diameter / 2));
    $arcy = floor($centery + sin(deg2rad(round((($finaldegree - $initdegree) / 2) + $initdegree))) * ($diameter / 2));
?>

```

Puis, pour chaque arc de cercle, calculez le centre de la part (c.-à-d. le milieu du segment délimité par le point précédent et le centre du cercle) :

```
<?php
    if($arcx > $centerx)
    {
        $midx = round(($arcx - $centerx) / 2 + $centerx);
    }
    else
    {
        $midx = round(($centerx - $arcx) / 2 + $arcx);
    }
    if($arcy > $centery)
    {
        $midy = round(($arcy - $centery) / 2 + $centery);
    }
    else
    {
        $midy = round(($centery - $arcy) / 2 + $arcy);
    }
?>
```

Remplissez chaque part :

```
<?php
    imagefilltoborder($image, $midx, $midy, $slice_color, $slice_color);
?>
```

Testez, puis tracez le cercle en noir :

```
<?php
    imagearc($image, $centerx, $centery, $diameter, $diameter, 0, 180, $border_color);
    imagearc($image, $centerx, $centery, $diameter, $diameter, 180, 360, $border_color);
?>
```

Agrandissez le cercle :

```
<?php
    imagearc($image, $centerx, $centery, $diameter + 7, $diameter + 7, 0, 180, $border_color);
    imagearc($image, $centerx, $centery, $diameter + 7, $diameter + 7, 180, 360, $border_color);
?>
```

Puis, remplissez l'espace entre les deux cercles :

```
<?php
    imagefilltoborder($image, $centerx + floor($diameter / 2) + 2, $centery, $border_color, $border_color);
?>
```

Pour chaque valeur donnée, déterminez la couleur (de la même manière que pour le diagramme circulaire) puis calculez l'ordonnée de la légende correspondante :

```
<?php
    $line = $diameter + 20 + ($fontheight + 2) * $i;
?>
```

Puis, tracez le rectangle de la légende et « remplissez-le » :

```
<?php
    imagerectangle($image, 10, $line, 10 + $fontheight, $line + $fontheight, $b
ordercolor);
    imagefilltoborder($image, 12, $line + 2, $bordercolor, $sliceolor);
?>
```

Écrivez le texte de la légende :

```
<?php
    imagestring($image, $font, 20 + $fontheight, $line, ' texte de la légende '
, $textcolor);
?>
```

Testez avec des valeurs fixes puis affichez le diagramme circulaire correspondant à la répartition calculée à partir des données de votre base de données.