

Configurations

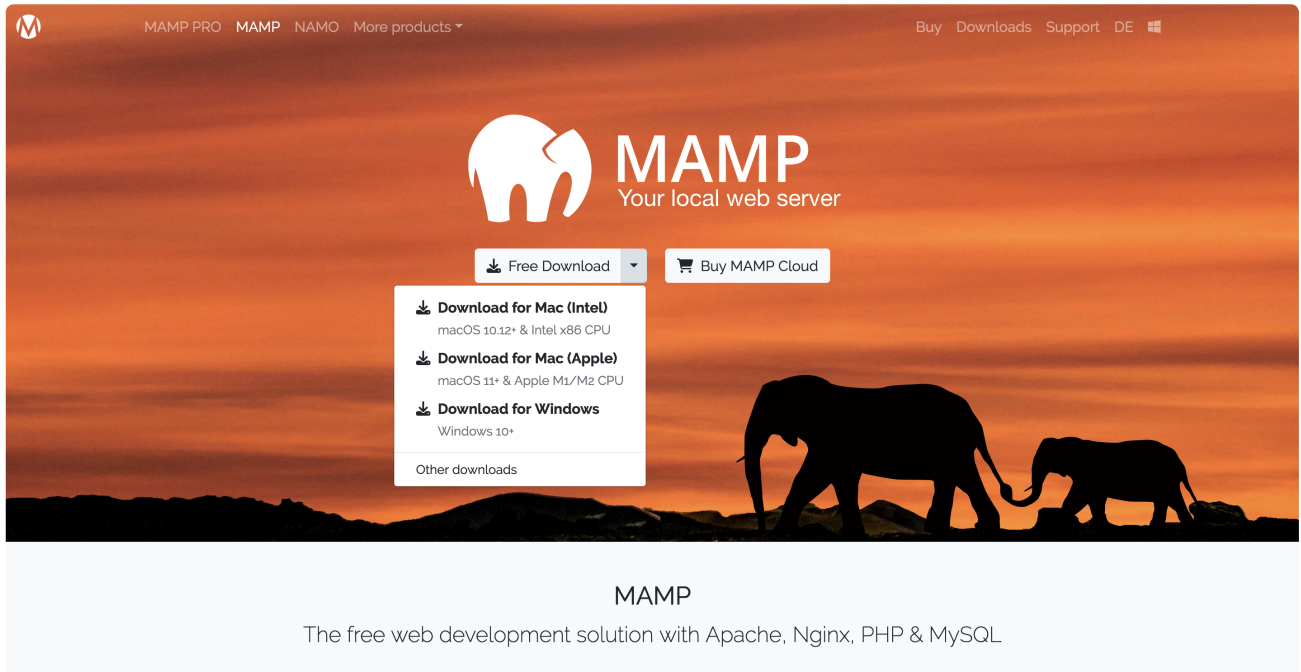
Commencez par créer un dossier nommé `public_html`.

Dans ce dossier créez le fichier `index.php` dont le contenu est le suivant :

```
<h1>
    <?php echo "Ça marche !" ?>
</h1>
```

Installer MAMP

<https://www.mamp.info/en/mamp/mac/>



Changer le dossier racine

Dans Préférences -> Server -> Choose... et sélectionner votre dossier `public_html` que vous venez de créer.

Lancer le serveur

Cliquez maintenant sur le bouton Start.

Dans votre navigateur, tapez `localhost:8888/index.php`.

Normalement, il doit y avoir "Ça marche !" qui s'affiche.

Afficher les erreurs

- Pour afficher les informations de votre version de PHP, aller à la page `http://localhost:8888/MAMP/phpinfo.php`.
- Trouvez la ligne `Loaded Configuration File` : c'est l'endroit où se trouve le fichier de configuration du PHP de votre machine, un fichier qui s'appelle `php.ini`.
- Ouvrez ce fichier dans un éditeur de texte.
- trouvez la ligne où la variable `error_reporting` est initialisée et vérifiez que sa valeur est bien `E_ALL`
- trouvez la ligne où `display_errors` est initialisée et remplacer `Off` par `On`
- enregistrez le fichier

Maintenant on pourra voir les erreurs directement sur le navigateur !

Exercice 1 : afficher son nom dynamiquement

Créez un fichier `ma_page.php` dans lequel vous allez créer une page qui affiche dans un titre `<h1>` votre nom et prénom qui sont stockées dans des variables.

Exercice 2 : afficher la date et l'heure

Retrouvez l'exemple du cours qui affiche la date et l'heure, et ajoutez le morceau de code à votre page.

Jeter un coup d'oeil à la page de documentation, et testez d'autres formats d'affichage.

Exercice 3 : afficher la date en français

Le problème de la fonction d'affichage du PHP est que si on lui demande le jour et/ou le mois en toute lettre, on nous le donne en anglais.

Question 1

Trouvez dans la documentation le format à donner à la fonction `date` pour obtenir seulement le jour en tout lettre en anglais.

Question 2

Trouvez dans la documentation le format à donner à la fonction `date` pour obtenir seulement le mois en tout lettre en anglais.

Question 3

Créez un tableau `$days_to_jours` où chaque clé est un nom de jour en anglais et la valeur est son équivalent en français.

Question 4

Créez un tableau `$months_to_mois` où chaque clé est un nom d'un mois en anglais et la valeur est son équivalent en français.

Question 5

Écrivez maintenant une page qui affiche la date en français au format `Jeudi 13 Juin 2024`.

Exercice 4 : êtes-vous majeur ?

Dans une variable `$age` stockez un âge. Disons que `$age` vaut `16`, alors la page devra afficher :

```
Vous avez 16 ans. Vous êtes mineur. Vous n'avez pas le droit
d'accéder à cette page.
```

et si `$age` vaut `44` :

```
Vous avez 44 ans. Vous êtes majeur. Bienvenue sur cette page.
```

Écrivez un script qui effectue ceci.

Exercice 5 : array en liste

Dans cet exercice vous allez devoir vous rappeler de comment on écrit une liste en HTML !

Créez un tableau `$arr = array("oui", "non", "peut-être", "pouvez-vous répéter la question ?")`.

Question 1

Appellez la fonction `print_r($arr)`. Qu'est-ce qui est affiché ?

Question 2

Avec une boucle `for`, affichez `$arr` comme une liste non-ordonnée où chaque élément (ou item en anglais...) est une case du tableau.

Question 3

Même question mais avec une boucle `foreach`.

Exercice 6 : array en table

Recréez le tableau `$fruits` vu dans le cours.

Question 1

Appelez la fonction `print_r($fruits)`. Qu'est-ce qui est affiché ? Quelle boucle faudra-t-il utiliser ?

Question 2

Avec une boucle, créez une table pour qu'elle ressemble à quelque chose comme ceci :

fruit	quantité
pomme	3
kiwi	2
banane	9
fraise	300

Exercice 7 : compter les valeurs

Créez le tableau suivant : `$arr = array(1, 2, 1, 3, 3, 1, 4, 2, 1)`.

Créez maintenant un tableau `$nb_valeur` où une clé est un des entiers qui apparaît dans `$arr` et sa valeur est le nombre de fois que cette valeur apparaît dans `$arr`.

Par exemple, `$nb_valeur[1]` doit valoir `4`.

Exercice 8 : lister les fichiers de `public_html`

PHP propose une fonction `scandir` qui prend en argument le chemin vers un dossier et qui renvoie un array avec le nom de chaque fichier et dossier à l'intérieur de ce dossier.

Question 1

Appelez `scandir` avec en argument votre dossier `public_html` et regardez ce qui est renvoyé avec `print_r`.

Question 2

Pour chaque élément renvoyé par `scandir` appelez la fonction `pathinfo` avec cet élément en argument.

Affichez ce que renvoie `pathinfo`.

Comment pourrait-on vérifier qu'un fichier est un fichier PHP ou HTML ?

 Vous pouvez ignorer les dossiers en utilisant la fonction `is_dir`

Question 3

Pour chaque fichier PHP ou HTML que vous avez trouvé avec `scandir`, ajouter une ligne dans une liste non-ordonnée pour faire en sorte que quand on clique dessus, le fichier en question est ouvert.

Exercice 9 : afficher un fichier `.ini`

Au début du TP, on a modifié un fichier qui s'appelait `php.ini`.

Question 1

Retrouvez le chemin vers ce fichier et donnez-le en argument à la fonction `parse_ini_file` et affichez ce qu'elle retourne avec `print_r`.

Question 2

Écrivez un script qui affiche si oui ou non `display_errors` est activé.

Exercice 10 : initiation aux prochains cours...

PHP fournit un certain nombre de variable prédéfinies dont on va beaucoup se servir par la suite. Un d'entre eux est notamment le tableaux `$_SERVER`.

Afficher le contenu de `$_SERVER`.

Est-ce que vous reconnaissez quelques valeurs ?

Exercice 11 : la table de 2

Comme on avait fait en JavaScript, écrivez un script qui affiche la table de 2 sous forme de table.