

Formation

PHP

PHP

Créé par : Rasmus Lerdorf, en 1994

A ses débuts : Personal Home Page Tools

Puis renommé en **PHP: Hypertext Preprocessor**

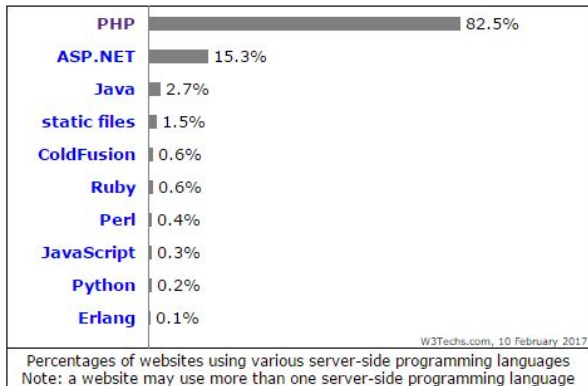
Fondamentaux du langage

- Langage de programmation **interprété** ≠ compilé
- Nécessite un serveur HTTP pour être lu dans un navigateur : Serveur interne de PHP, apache2, Nginx ...
- Peut s'exécuter en ligne de commande exemple : **php nomfichier.php**
- Version **stable** actuelle : **8.4**
- Balisage du code PHP :

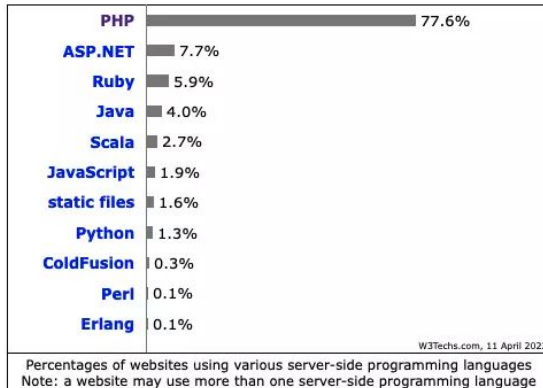
```
<?php  
    echo 'Hello world';  
?>
```

Usage dans le monde (langages de programmation côté serveur)

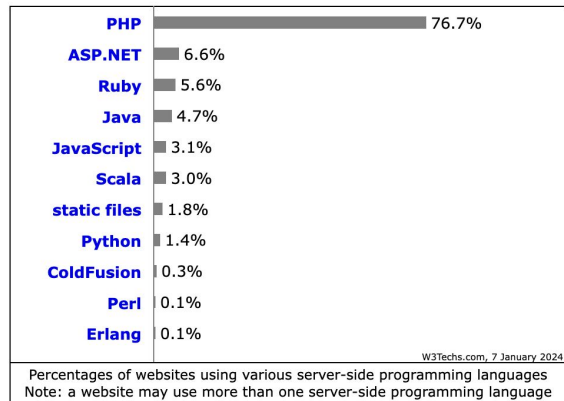
Utilisation de PHP en 2017



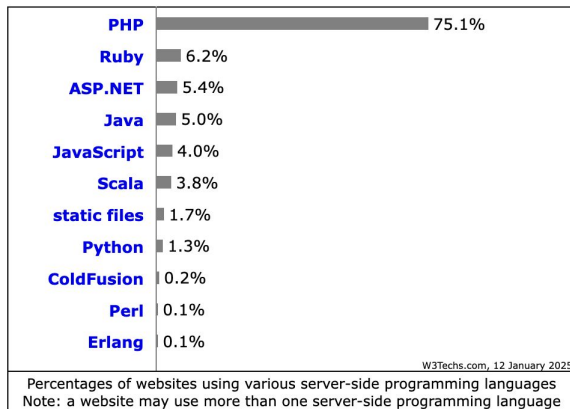
Utilisation de PHP en 2022



Utilisation de PHP en 2023



Utilisation de PHP en 2024



Usage dans le monde (langages de programmation côté serveur)

Sources :

- https://w3techs.com/blog/entry/web_technologies_of_the_year_2024
- https://w3techs.com/technologies/overview/programming_language

PHP dans les projets



Fait tourner à lui seul 43% des sites dans le monde !



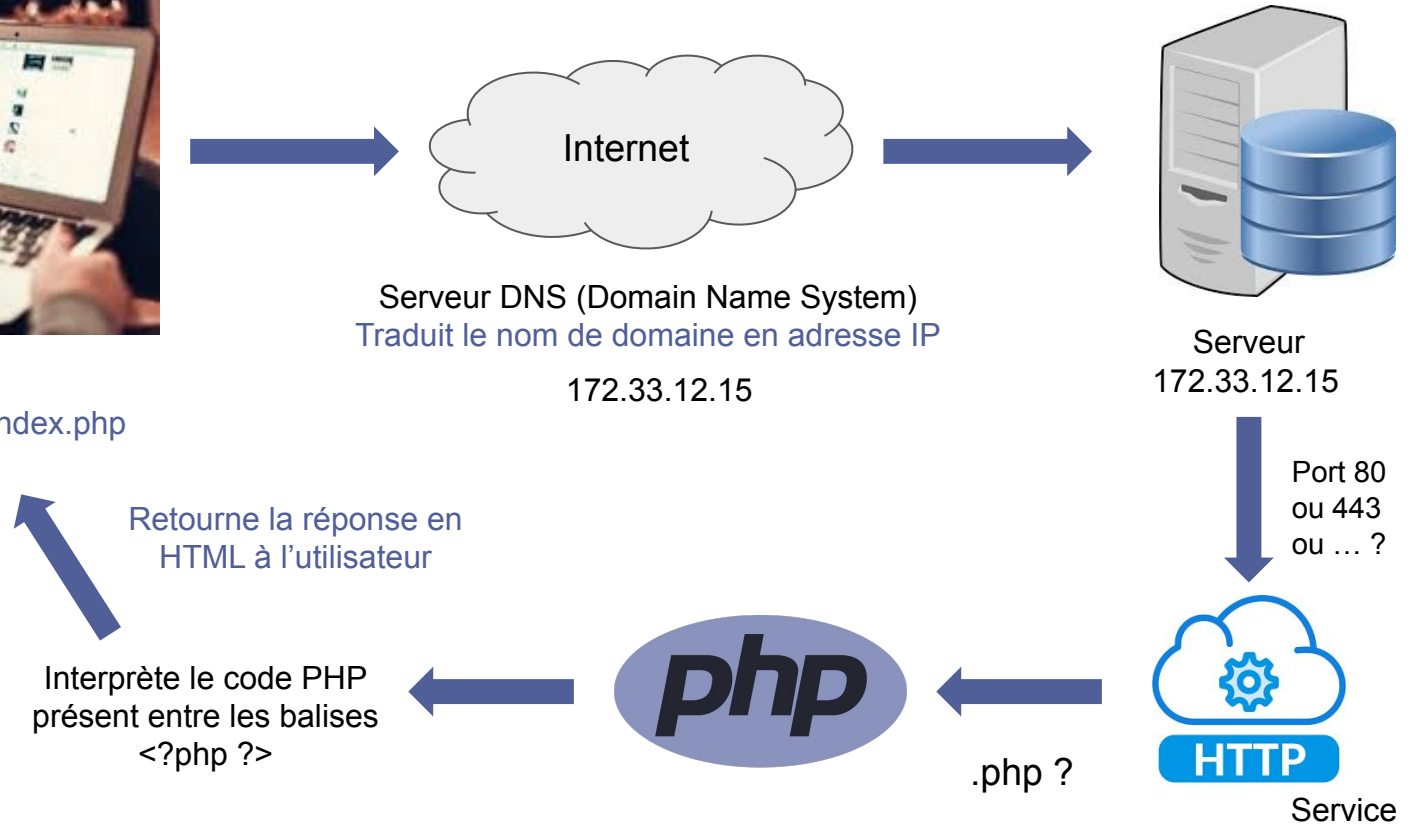
PrestaShop



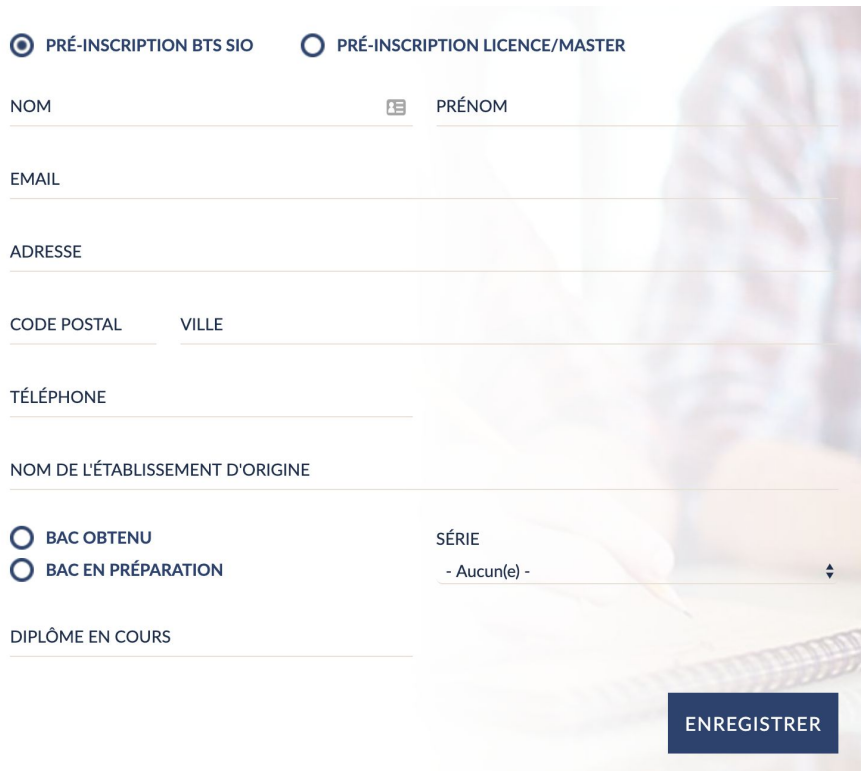
Quel rôle à PHP concrètement quand vous allez sur un site Web ?




Je me rends sur
<http://www.monsite.com/index.php>



Fonctionnement classique d'un site en avec PHP et base de données (formulaire)



☒ PRÉ-INSCRIPTION BTS SIO ☐ PRÉ-INSCRIPTION LICENCE/MASTER

NOM  PRÉNOM

EMAIL

ADRESSE

CODE POSTAL VILLE

TÉLÉPHONE

NOM DE L'ÉTABLISSEMENT D'ORIGINE

☒ BAC OBTENU ☐ BAC EN PRÉPARATION

SÉRIE

DIPLOME EN COURS

ENREGISTRER

1. Formulaire en HTML ;
2. Traitement en PHP au clic sur le bouton « Enregistrer » ;
3. Enregistrement des données soumises par l'utilisateur dans la BDD.

Installation d'un environnement Web



lamp



Mamp



Wamp /Xampp

<https://doc.ubuntu-fr.org/lamp>

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

<http://www.wampserver.com/>

Contient :

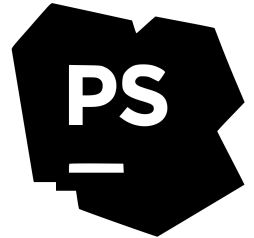
- Apache (serveur web)
- Mysql (base de données)
- PHP (langage de programmation)
- PhpMyAdmin (visualisation de la BDD)

Installer un IDE peut vous permettre de gagner du temps

Environnement de développement intégré: EDI en français

Les avantages :

- Autocomplétion ;
- Lignes colorisées ...
- Outils d'analyse de code
- Moteur de recherche intégré
- Contrôle de versions



Fondamentaux du langage

Faire cohabiter le PHP et le HTML

Insérer du code PHP dans du HTML c'est possible :

```
<h1>Bonjour <?php echo $prenom; ?></h1>
```

Insérer du HTML dans du PHP c'est possible :

```
<?php  
    echo "<b>Bonjour</b>". " ".$prenom. " ".$nomDeFamille;  
?>
```

La concaténation en PHP se fait avec le symbole « . ».

Fondamentaux du langage

La déclaration de variables

// Ceci est un commentaire et ne sera pas affiché dans la page.

\$var = 'Hello World'; // Chaînes de caractères (String)

\$var = 12; // Entier (Integer)

\$var = true; // Boolean

\$var = 1.20; // Float

\$var = null;

\$array = []; // Tableaux

\$var = new Toto(); // Les objets (qu'on verra en 2ème année)

Note : Pas de typage lors de la création d'une variable.
Les variables sont toujours préfixées par un \$.

Fondamentaux du langage

La déclaration de variables

```
<?php
```

```
// Déclaration d'une variable de type string
```

```
$prenom = "Jeff";
```

```
$nomDeFamille = "Martins";
```

```
// Déclaration d'une variable de type integer
```

```
$age = 25;
```

```
// Déclaration d'une variable de type float
```

```
$argentDePoche = 12.5;
```

```
// Réassignation d'une variable
```

```
$nomDeFamille = "Jacquelot";
```

```
// Variable de type booléenne
```

```
$estMajeur = false;
```

```
// On redéfinit la valeur de la variable "estMajeur"
```

```
$estMajeur = true;
```

Fondamentaux du langage

Afficher la valeur d'une variable / ce qu'elle contient :

`echo $var;` // Affiche une chaîne de caractères (valeur de \$var)

`print ($var);` // Affiche une chaîne de caractères (valeur de \$var)

`print_r ($var);` // Informations (debug) liées à \$var

`var_dump ($var);` // Informations (debug) liées à \$var

```
# jeffjacquelot at jacquelotjeff.local in ~/Desktop/bts1 [15:45:43]
→ php index.php
1
1
1
int(1)
```

```
<?php
    $var = ['1', '2', '3'];
    echo $var;
?>
```

```
# jeffjacquelot at jacquelotjeff.local in ~/Desktop/bts1 [15:29:55]
→ php index.php
PHP Notice:  Array to string conversion in /Users/jeffjacquelot/Desktop/bts1/index.php on line 6
Array
```

Fondamentaux du langage

Déterminer si une variable existe ou non avec **isset**

Retournera true si la variable existe.

```
$variable = null;  
$variable = false;
```

```
if (isset($variable)) {  
    echo 'La variable est définie';  
} else {  
    echo 'la variable n'est pas définie.';  
}
```

Tester la condition
ci-dessous en changeant la
valeur de la variable
\$variable.

Fondamentaux du langage

Déterminer si une variable est vide ou non avec **empty**

Retournera true si la variable est vide.

```
$variable = null;  
$variable = false;  
$variable = [];
```

```
if (empty($variable)) {  
    echo 'La variable est vide';  
} else {  
    echo 'la variable n'est pas vide.';  
}
```

Tester la condition
ci-dessous en changeant la
valeur de la variable
\$variable.

Les conditions

Fondamentaux du langage

Les conditions

```
<?php
```

```
if ($condition) { // Ma condition doit être vraie pour  
    passer (true)  
    // Mon code  
} elseif ($condition2) {  
    // Mon code  
} else {  
    echo 'Je ne suis pas passé dans mes 2 conditions';  
}
```

```
?>
```

```
<?php
```

```
$genre = 'homme';  
  
if ($genre == 'homme') {  
    echo 'Bonjour Monsieur';  
} else {  
    echo 'Bonjour Madame';  
}
```

```
?>
```

Bonjour Monsieur

Fondamentaux du langage

Les conditions

```
<?php
```

```
    $gender = 'homme';
```

```
switch ($gender) {
```

```
    case 'femme':
```

```
        echo 'Je suis une femme';
```

```
        # Mon code
```

```
        break;
```

```
    case 'Homme':
```

```
        echo 'Je suis un grand Homme ;)';
```

```
        # Mon code
```

```
        break;
```

```
    case 'homme':
```

```
        echo 'Je suis un homme (simplement ...)';
```

```
        break;
```

```
    default:
```

```
        echo 'Que puis-je être ? :o';
```

```
        break;
```

```
}
```

```
?>
```

```
<?php
```

```
    $var = 1;
```

```
    if ($var) {
```

```
        echo "OK cond 1 \n";
```

```
    }
```

```
    if ($var == 1) {
```

```
        echo "OK cond 2\n";
```

```
    }
```

```
    if ($var === 1) {
```

```
        echo "OK cond 3 \n";
```

```
    }
```

```
    if ($var == true) {
```

```
        echo "OK cond 4 \n";
```

```
    }
```

```
    if ($var === true) {
```

```
        echo "Ok cond 5 \n";
```

```
    }
```

```
?>
```

Fondamentaux du langage

Pourquoi il faut faire des comparaisons strictes

```
<?php
```

```
$argent = false; // Peut varier entre false/0/12/13  
etc.  
// Donne de l'argent de poche toutes les semaines  
if ($argent === 0) {  
    echo "Je n'ai plus d'argent.";  
} elseif ($argent > 0) {  
    echo "J'ai de l'argent.";  
} elseif ($argent === false) {  
    echo "On ne m'a pas encore donné d'argent.";  
} else {  
    echo "/";  
}
```



Tester le script en faisant varier les “===” en “==” et en variant la valeur de la variable \$argent.

?>

Si dans cet exemple nous n'avions pas fait de comparaison stricte pour le cas où \$argent vaut false, on aurait eu le message “*Je n'ai plus d'argent*” au lieu d'avoir le message “*On ne m'a pas encore donné d'argent.*”

Fondamentaux du langage

L'expression match

```
<?php
```

```
$genre = 'autre';
```

```
$instruction = match ($genre) {  
    'homme' => 'Bonjour je suis un homme',  
    'femme', 'fille' => 'Bonjour je suis une femme',  
    default => 'Bonjour',  
};
```

```
echo $instruction;
```

Note :

- Disponible sur PHP >= 8
- Les comparaisons sont “strictes” (opérateur ===)

Fondamentaux du langage

Les conditions : Exercice « Enregistrer »

```
<?php
    // La variable numeroDePage va prendre une valeur aléatoire entre 1 et 5.
    $numeroDePage = rand(1, 5);
?>
```

Grâce aux conditions vues dans le cours et du code ci-dessus, afficher un super héros selon le numéro de page.

Exemple : Si le numéro de page est 3, afficher le super héros « Loki ».

Fondamentaux du langage

Les conditions : Exercice « Supers héros »

Loki, héros ou méchant ?



Loki est une divinité et un super-vilain évoluant dans l'univers Marvel de la maison d'édition Marvel Comics. Créé par les scénaristes Stan Lee et Larry Lieber et le dessinateur Jack Kirby, le personnage de fiction apparaît pour la première fois dans le comic book Journey into Mystery #85 en octobre 1962.

Pouvoirs spéciaux : Force, durabilité et longévité surhumaines (pouvoirs asgardiens);

[> Autre personnage](#)

Note : Rafraichissez le navigateur pour voir les caractéristiques d'un autre personnage :)

Fondamentaux du langage

Les conditions : Exercice « Supers héros »

Bonus pour ceux qui terminent avant :

- Passer les différents éléments de la page en variable :

Au lieu d'afficher tout le contenu au format HTML selon les conditions, assignez aux variables les informations sur le super héros.

Ensuite afficher le contenu de la page.

\$nom

\$image

\$description

\$pouvoirs

Les tableaux

Fondamentaux du langage

Travailler avec les tableaux, exemple concret

```
<?php
```

```
$wishList = [];
```

```
// J'insère une PS5
```

```
$wishList['playstation'] = 'PS5';
```

```
// J'insère une Nintendo Switch
```

```
array_push($wishList, 'Nintendo Switch');
```

```
var_dump($wishList);
```

```
if (in_array('Xbox One Titan', $wishList) === false) {
```

```
    $wishList[] = 'Xbox One Titan';
```

```
    echo "La Xbox One Titan a été ajoutée dans la wishlist.";
```

```
}
```

```
var_dump($wishList);
```

```
?>
```

Fondamentaux du langage

Transformer un tableau en chaîne de caractères avec **implode**

```
$ingredients = ['tomate', 'citron', 'concombre', 'oeuf', 'soja'];
```

```
$mesIngredientsEnFormatTexte = implode(",", $ingredients);
```

```
echo $mesIngredientsEnFormatTexte;
```

tomate, citron, concombre, oeuf, soja

Note : Le premier paramètre de la fonction implode permet de définir le/les caractères qui vont séparer les éléments de mon tableau.

Fondamentaux du langage

Transformer une chaîne de caractères en tableau avec **explode**

```
$ingredients = 'tomate, citron, concombre, oeuf, soja';
```

```
$mesIngredientsEnTableau = explode(",", $ingredients);
```

```
var_dump($mesIngredientsEnTableau);
```

```
array(5) { [0]=> string(6) "tomate" [1]=> string(7) " citron" [2]=> string(10) " concombre" [3]=> string(5) " oeuf" [4]=> string(5) " soja" }
```

Note : Le premier paramètre de la fonction `explode` permet de définir le/les caractères qui vont servir à séparer les éléments dans le tableau.

Fondamentaux du langage

Les tableaux

```
<?php
```

```
// Mon tableau de résolution 2024 !
```

```
$resolutions2024 = [  
    'Faire mon Blog',  
    'Apprendre la couture',  
    'Passer le permis',  
    'Manger plus sainement',  
    'Arrêter la cigarette',  
];
```

```
var_dump($resolutions2024);
```

```
?>
```

```
<?php
```

```
// Tableau associatif (clés/valeurs)
```

```
$resolutions2024Associatif = [  
    1 => 'Faire mon Blog',  
    5 => 'Apprendre la couture',  
    2 => 'Passer le permis',  
    3 => 'Manger plus sainement',  
    4 => 'Arrêter la cigarette',  
];
```

```
var_dump($resolutions2024Associatif);
```

```
?>
```

? La génération des clés dans les tableaux commence toujours par 0. (0, 1, 2, 3, etc.)

Fondamentaux du langage

Les tableaux - Multi dimensions

```
array(2) {  
  [2017]=>  
    array(4) {  
      [0]=>  
        string(17) "Arrêter de boire"  
      [1]=>  
        string(17) "Arrêter de fumer"  
      [2]=>  
        string(23) "Apprendre le Javascript"  
      [3]=>  
        string(14) "Faire du sport"  
    }  
  [2018]=>  
    array(4) {  
      [0]=>  
        string(14) "Faire mon Blog"  
      [1]=>  
        string(16) "Passer le permis"  
      [2]=>  
        string(21) "Manger plus sainement"  
      [3]=>  
        string(20) "Apprendre la couture"  
    }  
}
```

<?php

// Mon tableau de résolutions par années

```
$mesResolutions = [  
    2022 => [  
        'Arrêter de boire',  
        'Arrêter de fumer',  
        'Apprendre le Javascript',  
        'Faire du sport',  
    ],  
    2023 => [  
        'Faire mon Blog',  
        'Passer le permis',  
        'Manger plus sainement',  
        'Apprendre la couture',  
    ],  
];  
var_dump($mesResolutions);
```

?>

Fondamentaux du langage

Les tableaux

```
<?php
```

```
    $prenoms = ['Jenny', 'Grégoire'];
```

```
    echo $prenoms;
```

```
?>
```

Notice: Array to string conversion in `/Users/jeffjacquelot/Projects/bts1-2018/index.php` on line 4
Array

Fondamentaux du langage

Les tableaux

Afficher les valeurs d'un tableau
à l'aide de leurs clés

Fondamentaux du langage

Les tableaux

En se basant sur la clé de l'élément

Imaginons le tableau suivant :

```
<?php
    $prenoms = ['Jenny', 'Grégoire'];
?>
```

En ne spécifiant pas de clé, PHP va se charger d'associer les clés à nos valeurs :

```
array(2) {
    [0]=> string(5) "Jenny"
    [1]=> string(9) "Grégoire"
}
```

Fondamentaux du langage

Les tableaux

Si on souhaite afficher la valeur “Jenny”, on va donc utiliser sa clé associée en suivant ce format :

```
<?php
    echo $prenoms[0]; // Ou 0 est l'indice de la valeur Jenny.
?>
```

Dans un tableau associatif :

```
<?php
    $prenoms = ['1er élément' => 'Jenny', '2ème élément' => 'Grégoire'];

    echo $prenoms['2ème élément']; // On sait que Grégoire a la clé "2ème élément", on l'utilise
    donc pour récupérer sa valeur.
?>
```

Fondamentaux du langage

Les tableaux

Parcourir les tableaux

Fondamentaux du langage

Les structures de contrôle : **for**

```
<?php
```

```
$prenoms = ['Jenny', 'Grégoire'];  
$nbPrenoms = count($prenoms);
```

```
for ($i = 0; $i < $nbPrenoms; $i++) {  
    echo $prenoms[$i];  
    echo '<br />';  
}
```

```
?>
```

Jenny
Grégoire

On **initialise une variable** `$i` à la valeur 0.

Cette expression est exécutée avant le début de la boucle.

On définit la condition pour savoir si la boucle doit continuer. **Si l'expression vaut TRUE, alors la boucle continue.** Cette expression est réévaluée à chaque itération.

Ici tant que `$i < $nbPrenoms`, alors on réitère.

Cette expression est exécutée à chaque fin de boucle.

Ici, on **met à jour la variable** `$i` : `$i = $i + 1`;

Fondamentaux du langage

Les structures de contrôle : **foreach**

... Fournit une façon simple de parcourir des tableaux

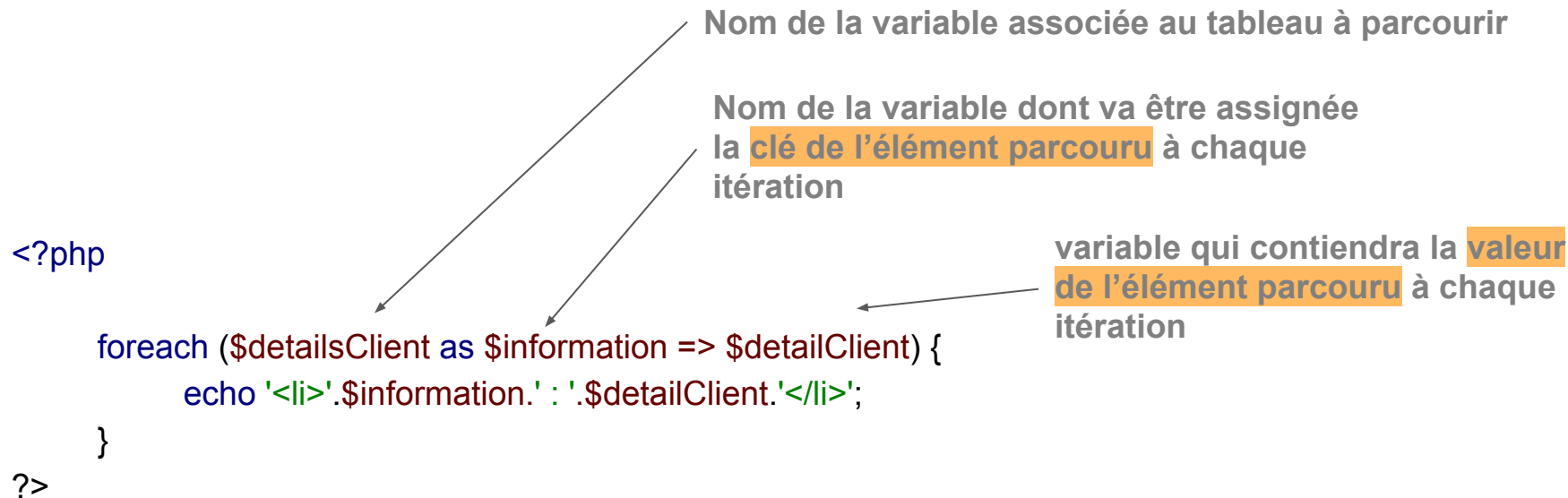
```
<?php
    $detailsClient = [
        'Prénom' => 'William',
        'Nom' => 'Dranes',
        'Date de naissance' => '12 octobre 1998'
    ];

    foreach ($detailsClient as $information => $detailClient) {
        echo '<li>'.$information.' : '.$detailClient.'</li>';
    }
?>
```

- Prénom : William
- Nom : Dranes
- Date de naissance : 12 octobre 1998

Fondamentaux du langage

Les structures de contrôle : **foreach**, explications en détails



Fondamentaux du langage

Les structures de contrôle : **foreach**, la clé peut être omise dans le parcours

```
<?php
```

```
$mesParfums = [  
    'One Million',  
    'Dior Homme Intense',  
    'Azaro'  
];
```

```
foreach ($mesParfums as $monParfum) {  
    echo '<li>'.$monParfum.'</li>';  
}
```

```
?>
```

Ici on n'a pas besoin de récupérer pour chaque itération la clé de l'élément parcouru.

La variable `$monParfum` contiendra la valeur de l'élément parcouru à chaque itération.

- One Million
- Dior Homme Intense
- Azaro

Fondamentaux du langage

La boucle **while**

PHP exécute l'instruction tant que l'expression de la boucle *while* est évaluée comme **TRUE**. La valeur de l'expression est vérifiée à chaque début de boucle.

```
<?php
```

```
    $age = 12;
```

```
    $agePourConduire = 18;
```

```
    while ($age < $agePourConduire) {
```

```
        echo "J'ai $age ans, je ne peux donc pas conduire.";
```

```
        echo '<br />';
```

```
        $age++; // Incrémentation de la variable de 1, également possible d'écrire $age = $age + 1;
```

```
    }
```

```
    echo "J'ai maintenant $age ans. Je peux donc conduire ! YEAH";
```

```
?>
```

J'ai 12 ans, je ne peux donc pas conduire.

J'ai 13 ans, je ne peux donc pas conduire.

J'ai 14 ans, je ne peux donc pas conduire.

J'ai 15 ans, je ne peux donc pas conduire.

J'ai 16 ans, je ne peux donc pas conduire.

J'ai 17 ans, je ne peux donc pas conduire.

J'ai maintenant 18 ans. Je peux donc conduire ! YEAH

Les fonctions

Fondamentaux du langage

Les fonctions

```
<?php
```

Signature

Nom de la fonction

Paramètres

```
function addition($number1, $number2)
{
    return $number1 + $number2;
}
```

```
echo addition(1, 2); // 3
```

Exemple :

```
<?php
```

```
function nomDuSite()
```

```
{
```

```
    return "Jeff Martins-Jacquelot.";
```

```
}
```

```
?>
```

```
<div style="float: left; width: 200px;">
```

```
<?php echo nomDuSite(); ?>
```

```
</div>
```

```
<div style="float: right;">
```

```
<?php echo nomDuSite(); ?>
```

```
</div>
```

Fondamentaux du langage

Les fonctions

```
<?php
    function sePresente($genre)
    {
        if ($genre === 'homme') {
            return "<p style='color: blue;'>Bonjour Monsieur</p>";
        }

        return "<p style='color: pink;'>Bonjour Madame</p>";
    }

    $genre = 'homme';

    echo sePresente($genre);
?>
```

Fondamentaux du langage

Les fonctions

Créer un tableau PHP dans lequel vous renseignez les informations suivantes :

- Nom
- Prénom
- Âge
- Plat préféré
- Marque de voiture préférée

Créer une fonction qui prend en argument le tableau créé juste avant.

Cette fonction devra afficher les informations du tableau dans votre navigateur :

Salut Je m'appelle **Jeff Jacquelot**.

J'ai **23** ans.

Mon plat préféré est les **lasagnes** (Miam !)

Ma marque de voiture préférée est **Peugeot** bien évidemment !

Fondamentaux du langage

Exercice sur les tableaux et fonctions

```
$filmsEtSeries = [  
  ['nom' => 'The walking Dead', 'type' => 'Série'],  
  ['nom' => 'Breaking Bad', 'type' => 'Série'],  
  ['nom' => 'Expandables', 'type' => 'Film'],  
  ['nom' => 'La tour infernale', 'type' => 'Film'],  
  ['nom' => 'Vikings', 'type' => 'Série'],  
  ['nom' => 'Harold et Kumar', 'type' => 'Film'],  
  ['nom' => 'Game of thrones', 'type' => 'Série'],  
];
```

1. Créer une fonction qui prend en argument un tableau de séries et de films (comme l'exemple ci-dessus).
2. Cette fonction devra retourner sous forme de tableaux les éléments triés dans 2 tableaux distincts afin d'avoir le résultat suivant (code source de la page) :

```
array(2) {  
  ["series"]=>  
    array(4) {  
      [0]=>  
        string(16) "The walking Dead"  
      [1]=>  
        string(12) "Breaking Bad"  
      [2]=>  
        string(7) "Vikings"  
      [3]=>  
        string(15) "Game of thrones"  
    }  
  ["films"]=>  
    array(3) {  
      [0]=>  
        string(11) "Expandables"  
      [1]=>  
        string(17) "La tour infernale"  
      [2]=>  
        string(15) "Harold et Kumar"  
    }  
}
```

Ne pas recopier bêtement le code que l'on trouve sur Internet



Formulaires

Transmettre des données au serveur

Soumettre des données au serveur Web

2 méthodes sont définies par le protocole HTTP pour nous permettre d'envoyer des informations à notre serveur Web : **GET** et **POST**

GET : les données transmises sont accessibles de l'url du navigateur

POST : les données sont transmises dans le body de la requête envoyée au serveur (Form Data)

Transmettre des données au serveur

Envoyer des informations en passant par la méthode GET

Situation concrète : Vous souhaitez acheter le nouvel Iphone 14 pro max, pour avoir la classe ...

Toutes nos catégories ▼

iphone 14 pro



Je valide ma recherche

Lien dans le navigateur :

https://www.amazon.fr/s?k=iphone+14+pro&__mk_fr_FR=%C3%85M%C3%85%C5%BD%C3%95%C3%91&ref=nb_sb_noss_2

!! Si vous copiez le lien et le collez dans un autre onglet, vous arriverez sur la même page.
Ça tombe bien, c'est le but :)

Transmettre des données au serveur

Envoyer des informations en passant par la **méthode POST**

Situation concrète : Les formulaires de connexion, connectons-nous sur Amazon !



Identifiez-vous

Adresse e-mail ou numéro de téléphone
portable

admin123456@amazon.com

Mot de passe

[Mot de passe oublié](#)

...

Identifiez-vous

☐ Maintenir ma session ouverte. [Détails](#)

Nouveau chez Amazon ?

Créer votre compte Amazon

Je le veux vraiment ce nouvel Iphone !

/!\ Avant de vous connecter, faites un clic droit
sur le page du navigateur, onglet Networks

▼ General

Request URL: https://www.amazon.fr/ap/signin

Request Method: POST

Status Code: 🟡 302

Remote Address: 104.124.210.230:443

Referrer Policy: no-referrer-when-downgrade

Transmettre des données au serveur

Envoyer des informations en passant par la méthode POST

Vous souvenez-vous, on avait dit qu'il fallait que notre site internet utilise le protocole https ?

Si vous n'utilisez pas le protocole HTTPS, vos données seront visibles au moment de l'envoi au serveur.

D'ailleurs, on peut y accéder en clair au moment de l'envoi, comme le montre la capture ci-dessous :

The screenshot displays a network traffic capture tool interface. On the left, a list of network requests is shown, including a 'signin /ap' request. The 'Headers' tab is selected, showing the request details. The request body is visible in plain text, containing the email 'azerty123@gmail.com' and the password '123'. A red box highlights the request body content.

Name	Headers	Preview	Response	Co
signin /ap	LB33SbwjNB_-hPCKF8m4jQzCw9-12J9CsU XKUGXj4fUgnr5MrA1K8EC0VbKPm10SM545 Ghcg0tQ2USbH_KrjVUnhWZKy10pR-kjqxB U13heU6t81406q0D6D4mUV5u67nj2s5wx MV4hTR97m5dzGrb0TeHsWwCUhN3iuByjaL BZfRyTz4bcfVn7LTuAFgAe80rI11PD4kdc fbhR2yXJIG6rr0kXUEY1fDWe9vh7fsEDeS 30w			

email: azerty123@gmail.com
email: azerty123@gmail.com
password: 123

46 requests 638 KB transferred 1.9 MB resource

Les formulaires

Envoyer des données au serveur depuis un formulaire en passant par la **méthode GET**

```
<form action="#" method="GET">  
  <label for="identifiant">Identifiant :</label>  
  <input type="text" id="identifiant" name="identifiant">  
  
  <label for="motdepasse">Mot de passe :</label>  
  <input type="password" id="motdepasse" name="motdepasse">  
  
  <input type="submit">  
</form>
```

Identifiant : Mot de passe :

Les formulaires

Envoyer des données au serveur depuis un formulaire en passant par la **méthode GET**

Sur quelle page PHP envoyer les données ?
correspond à la page actuelle.

Quelle est la **méthode utilisée** pour envoyer les données ? GET ou POST.

`<form action="#" method="GET">`
 `<label for="identifiant">Identifiant :</label>`
 `<input type="text" id="identifiant" name="identifiant">`

Chaque champ doit avoir un **attribut name** afin de pouvoir récupérer sa valeur côté serveur.

`<label for="motdepasse">Mot de passe :</label>`
 `<input type="password" id="motdepasse" name="motdepasse">`

`<input type="submit">`
`</form>`

Les formulaires

Envoyer des données au serveur depuis un formulaire en passant par la **méthode GET**

```
<form action="#" method="GET">
  <label for="identifiant">Identifiant :</label>
  <input type="text" id="identifiant" name="identifiant">

  <label for="motdepasse">Mot de passe :</label>
  <input type="password" id="motdepasse" name="motdepasse">

  <input type="submit">
</form>
```

Au clic sur le bouton Submit, le
formulaire **va être soumis**.



The screenshot displays the 'General' tab of a browser's developer tools network panel. It shows a GET request to the URL `http://localhost:8000/formulaire.php?identifiant=admin&motdepasse=admin`. The status is 200 OK. Below this, the 'Response Headers' tab is selected, showing headers like 'Connection: close', 'Content-type: text/html; charset=UTF-8', and 'X-Powered-By: PHP/7.1.5'. At the bottom, the 'Request Headers (11)' tab is selected, showing the 'Query String Parameters' as `identifiant: admin` and `motdepasse: admin`.

General	
Request URL:	http://localhost:8000/formulaire.php?identifiant=admin&motdepasse=admin
Request Method:	GET
Status Code:	200 OK
Remote Address:	::1:8000
Referrer Policy:	no-referrer-when-downgrade

Response Headers		view source
Connection:	close	
Content-type:	text/html; charset=UTF-8	
Date:	Sun, 14 Jan 2018 19:44:35 +0000	
Host:	localhost:8000	
X-Powered-By:	PHP/7.1.5	

Request Headers (11)		
Query String Parameters		
identifiant:	admin	
motdepasse:	admin	

Les formulaires

Et l'envoi de données au serveur depuis la **méthode POST**, ça marche comment ?

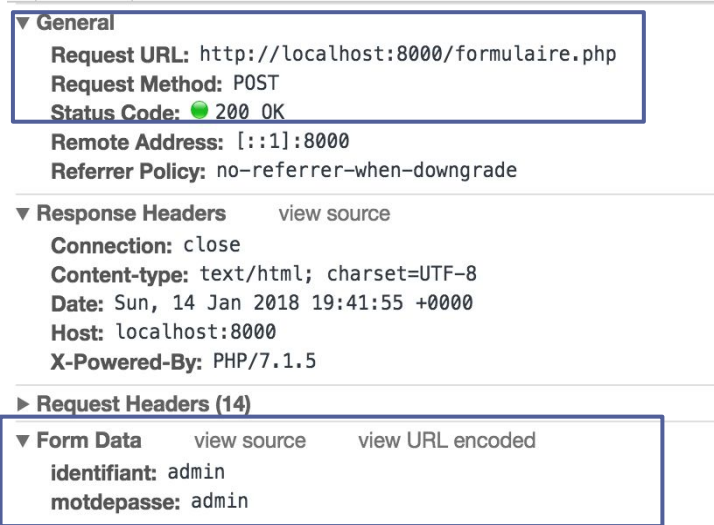
C'est la même chose, sauf que la méthode change :)

```
<form action="#" method="POST">
  <label for="identifiant">Identifiant :</label>
  <input type="text" id="identifiant" name="identifiant">

  <label for="motdepasse">Mot de passe :</label>
  <input type="password" id="motdepasse" name="motdepasse">

  <input type="submit">
</form>
```

Au clic sur le bouton Submit, le
formulaire **va être soumis**.



The screenshot shows the 'General' tab of a network request. The request is a POST to 'http://localhost:8000/formulaire.php' with a status of 200 OK. The 'Response Headers' tab shows the response is text/html. The 'Request Headers' tab shows the form data: 'identifiant: admin' and 'motdepasse: admin'.

General	
Request URL:	http://localhost:8000/formulaire.php
Request Method:	POST
Status Code:	200 OK
Remote Address:	::1:8000
Referrer Policy:	no-referrer-when-downgrade

Response Headers	
Connection:	close
Content-type:	text/html; charset=UTF-8
Date:	Sun, 14 Jan 2018 19:41:55 +0000
Host:	localhost:8000
X-Powered-By:	PHP/7.1.5

Request Headers (14)	
Form Data	
identifiant:	admin
motdepasse:	admin

Les formulaires

Récupérer les informations transmises par la méthode GET

Comment récupérer les valeurs transmises par la méthode GET côté PHP ?

PHP va construire un tableau associatif avec les valeurs transmises par le formulaire et les injecter dans une variable Superglobale de PHP appelée `$_GET`.

```
<?php
    var_dump($_GET);
?>
```


Les formulaires

Récupérer les informations transmises par la méthode POST

Comment récupérer les valeurs transmises par la méthode POST côté PHP ?

PHP va construire un tableau associatif avec les valeurs transmises par le formulaire et les injecter dans une variable Superglobale de PHP appelée `$_POST`.

```
<?php
    var_dump($_POST);
?>
```

Les formulaires

Petite appartée ... On a parlé de variables Superglobales, qu'est-ce que c'est que ça ?

Les Superglobales sont des **variables internes qui sont toujours disponibles**, quel que soit le contexte.

Voici la liste (n'hésitez pas à les vardumper !)

- \$GLOBALS (<https://www.php.net/manual/fr/reserved.variables.globals.php>)
- \$_SERVER (<https://www.php.net/manual/fr/reserved.variables.server.php>)
- \$_GET (<https://www.php.net/manual/fr/reserved.variables.get.php>)
- \$_POST (<https://www.php.net/manual/fr/reserved.variables.post.php>)
- \$_FILES (<https://www.php.net/manual/fr/reserved.variables.files.php>)
- \$_COOKIE (<https://www.php.net/manual/fr/reserved.variables.cookies.php>)
- \$_SESSION (<https://www.php.net/manual/fr/reserved.variables.session.php>)
- \$_REQUEST (<https://www.php.net/manual/fr/reserved.variables.request.php>)
- \$_ENV (<https://www.php.net/manual/fr/reserved.variables.environment.php>)

Les Super Héros n'ont qu'à bien se tenir ...



Les formulaires

Petit exemple, affichage d'une information transmise avec la méthode GET

Créez un petit fichier nommé exercice-get.php, et mettez le dans votre répertoire Web.

Insérer le code suivant dans le fichier :

```
<?php
```

```
if (isset($_GET['prenom'])) {  
    echo '<h3>Bienvenue ' . $_GET['prenom'] . '</h3>';  
}
```

```
?>
```

Depuis votre navigateur, rendez-vous sur <https://localhost/exercice-get.php?prenom=Jeff>

Les formulaires

Exercice : création d'un formulaire de contact

But: Créer un formulaire de contact sur une page (image 1). Envoyer le formulaire sur une page de traitement (traitement.php par exemple).

Cette page va récupérer et afficher les données saisies dans le formulaire (image 2)

Contactez-moi !

Lorem ipsum dolor sit amet, consectetur adipisicing elit. A
nisi voluptas. Ad aperiam doloremque laborum nulla place

Votre nom:

Votre prénom:

Votre N° de téléphone:

Envoyer

image 1

Contactez-moi !

Merci de m'avoir contacté.

Voici les informations que vous m'avez transmises :

- Nom : Martins
- Prénom : Jeff
- N° De téléphone : 0000000000

[Retourner au formulaire](#)

image 2

Les formulaires

Exercice : création d'un formulaire de contact (avancé)

But: Créer un formulaire de contact sur une page (image 1). Envoyer le formulaire sur une page de traitement (traitement.php par exemple).

Cette page va récupérer et afficher les données saisies dans le formulaire (image 2)

Contactez-moi !

Lorem ipsum dolor sit amet, consectetur adipisicing elit. A
nisi voluptas. Ad aperiam doloremque laborum nulla place

Votre nom:

Votre prénom:

Votre N° de téléphone:

Envoyer

image 1

Contactez-moi !

Merci de m'avoir contacté.

Voici les informations que vous m'avez transmises :

- Nom : Martins
- Prénom : Jeff
- N° De téléphone : 0000000000

[Retourner au formulaire](#)

image 2

Les formulaires

Sécuriser l'envoi des données

Convertir les caractères spéciaux en entités HTML :

```
<?php
    $valeur = '<script>alert("HELLO WORLD");</script>';
    echo htmlspecialchars($valeur);
?>
```

```
<script>alert("HELLO WORLD");</script>
```

(Ne s'exécutera pas dans le navigateur)

Valeur (les caractères spéciaux ont été transformés en entité HTML) :

```
"&lt;script&gt;alert(&quot;HELLO WORLD&quot;);&lt;/script&gt;,"
```

Documentation :

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

Les formulaires

Utiliser les filtres de validation

Tester la validité d'un email :

```
<?php
    $filterEmail = filter_var('montexte', FILTER_VALIDATE_EMAIL);
    var_dump($filterEmail);

?>
```

bool(false)

Documentation :

<https://www.php.net/manual/fr/function.filter-var.php>

Liste des filtres de validation : <https://www.php.net/manual/fr/filter.filters.validate.php>

Autres filtres de nettoyage : <https://www.php.net/manual/fr/filter.filters.sanitize.php>

Les fonctions require et include

Fichier index.php

```
<?php  
    require_once('includes/header.php');  
?>
```

Voici le contenu de mon site Internet :D

Fichier includes/header.php

```
<h1>BIENVENUE SUR MON SITE WEB</h1>
```

BIENVENUE SUR MON SITE WEB

Voici le contenu de mon site Internet :D

Interagir avec la base de
données

Interactions avec la base de données

Petits rappels sur PHPMYAdmin

Logiciel de gestion de base de données Mysql

phpMyAdmin

Récentes Préférées

Nouvelle base de données

- information_schema
- mysql
- performance_schema
- sys

Serveur: db

Bases de données SQL État Comptes d'utilisateurs Export Import Paramètres Réplication plus

Bases de données

Créer une base de données

Nom de base de données Interclassement Créer

Base de données	Interclassement	Action
<input type="checkbox"/> information_schema	utf8_general_ci	Vérifier les privilèges
<input type="checkbox"/> mysql	latin1_swedish_ci	Vérifier les privilèges
<input type="checkbox"/> performance_schema	utf8_general_ci	Vérifier les privilèges
<input type="checkbox"/> sys	utf8_general_ci	Vérifier les privilèges
Total: 4	latin1_swedish_ci	

☐ Tout cocher Pour la sélection : [Supprimer](#)

Note: L'activation des statistiques peut causer un trafic important entre le serveur web et le serveur MySQL.

- [Activer les statistiques](#)

Se rendre sur <http://localhost/phpmyadmin>

Interactions avec la base de données

Bonnes pratiques

Encodage : utf8_general_ci

Séparation des termes par des « _ » au lieu du camelCase

Clés étrangères terminées par « _id »

Tout en français, ou tout en anglais

Interactions avec la base de données

Se connecter à notre base de données

Avec mysqli

```
<?php
```

```
$con = mysqli_connect("localhost","root","", 'ma_base'); mysqli_set_charset($con, "utf8");
```

```
if (mysqli_connect_errno()) {  
    echo "Failed to connect to MySQL: " . mysqli_connect_error();  
}
```

Interactions avec la base de données

Se connecter à notre base de données

Avec PDO

Avantage : fonctionne avec plusieurs SGBD



CUBRID (PDO)
MS SQL Server (PDO)
Firebird (PDO)
IBM (PDO)
Informix (PDO)
MySQL (PDO)
MS SQL Server (PDO)
Oracle (PDO)
ODBC et DB2 (PDO)
PostgreSQL (PDO)
SQLite (PDO)
4D (PDO)

<?php

```
$user = "root";  
$pass = "root";  
$dbName = "monapp";  
$con = new \PDO("mysql:host=127.0.0.1;dbname=$dbName;charset=UTF8", $user, $pass);
```

Interactions avec la base de données

Mettons en place la connexion à la base de données

Fichier includes/db.php

```
<?php
    $user = "root";
    $pass = "root";
    $dbName = "monapp";
    $con = new PDO("mysql:host=127.0.0.1;dbname=$dbName;charset=UTF8", $user, $pass);
?>
```

Fichier index.php

```
<?php
    require_once 'includes/db.php';

    $query = $con->prepare("SELECT * FROM users");
    $query->execute();

    $users = $query->fetchAll();
?>
```

Interactions avec la base de données

Récupération des résultats une base de données

SELECT

Si vous ne mettez pas de `PDO::FETCH_ASSOC` en argument de la fonction, le tableau contiendra pour chaque résultat une clé incrémentée (voir doc).

<?php

```
$user = "root";  
$pass = "root";  
$dbName = "portfolio";
```

```
$connexion = new \PDO("mysql:host=127.0.0.1;dbname=$dbName;charset=UTF8", $user, $pass);
```

```
$requete = $connexion->prepare("SELECT * FROM contact");  
$requete->execute();
```

```
$contacts = $requete->fetchAll(\PDO::FETCH_ASSOC);
```

?>

Interactions avec la base de données

Insertions en base de données

INSERT

<?php

```
$query = $con->prepare('INSERT INTO users (username, password) VALUES (:username, :password)');  
$query->bindParam(':username', $username);  
$query->bindParam(':password', $password);
```

```
$username = 'Jeff';  
$password = '123456';  
$query->execute();
```

```
// Debugger les erreurs :  
$query->errorInfo();
```

?>

En utilisant les bindParams, les instructions SQL sont envoyées et analysées par le serveur de base de données séparément de tous vos paramètres.

=> Impossible de faire des injections SQL ! (et ça c'est cool)

Interactions avec la base de données

Topo sur les injections SQL

“L'**injection SQL** directe **est** une technique où un pirate modifie une requête **SQL** existante pour afficher des données cachées, ou pour écraser des valeurs importantes, ou encore exécuter des commandes dangereuses pour la base.”

Interactions avec la base de données

Insertions en base de données

DELETE

```
<?php

$query = $con->prepare('DELETE FROM users WHERE username = :username');
$query->bindParam(':username', $username);
$username = 'Jeff';
$query->execute();
```

```
?>
```

Interactions avec la base de données

Insertions en base de données

UPDATE

```
<?php

$query = $con->prepare('UPDATE users SET username = :username WHERE id = :id');
$query->bindParam(':username', $username);
$query->bindParam(':id', $id);
$username = 'Jeff';
$id = 1;
$query->execute();

?>
```

TP - Interaction avec la base de données

Insertions en base de données

Création d'un "blog" qui présente une liste d'articles.

Page liste des articles :

Liste des articles

[Nouveau](#)

Portfolio 2019

Nouveau portfolio réalisé avec PHP et base de données en SQL

>> [Voir l'article](#)

>> [Éditer l'article](#)

TP - Interaction avec la base de données

Insertions en base de données

Création d'un "blog" qui présente une liste d'articles.

Page ajout d'un article :

Ajout d'un article

Titre de l'article

Contenu de l'article

Catégorie

Développement

Ajouter !

TP - Interaction avec la base de données

Insertions en base de données

Création d'un "blog" qui présente une liste d'articles.

Page vue d'un article :

Portfolio 2019

Nouveau portfolio réalisé avec PHP et base de données en SQL

Développement

[Retour à la liste](#)

TP - Interaction avec la base de données

Insertions en base de données

Création d'un "blog" qui présente une liste d'articles.

Page édition d'un article :

Edition d'un article

Titre de l'article

Nouveau portfolio réalisé avec PHP et base de données en SQL

Contenu de l'article

Catégorie

Interactions avec la base de données

Hasher les mots de passe

L'algorithme md5

Méthode pas du tout sécurisée ! Des méthodes existent pour obtenir le mot de passe

Les algorithmes argon2i, bcrypt

Méthodes sécurisées !

hash

(PHP 5 >= 5.1.2, PHP 7, PHP 8, PECL hash >= 1.1)

hash — Generate a hash value (message digest)

Description

```
hash(  
    string $algo,  
    string $data,  
    bool $binary = false,  
    array $options = []  
): string
```


Interactions avec la base de données

Exercice

Créer une table dans votre base de données “users”

Structure:

- `id` (AI, primary)
 - `first_name` (varchar, 255)
 - `last_name` (varchar, 255)
 - `username` (varchar, 255)
 - `password` (varchar, 255)
-
- Faire un formulaire d'inscription
 - Enregistrer le mot de passe en **Argon2id**
 - Faire un formulaire de connexion
 - Récupérer le mot de passe saisi par l'utilisateur, le hasher en Argon2id et vérifier que c'est le bon mot de passe
 - Quand la personne est connectée : faire un bouton de suppression de compte : supprimer son compte

Les cookies

Signature de la fonction :

```
setcookie(  
    string $name,  
    string $value = "",  
    int $expires_or_options = 0,  
    string $path = "",  
    string $domain = "",  
    bool $secure = false,  
    bool $httponly = false  
): bool
```

← Nom du cookie
← Valeur à stocker
← Date d'expiration
← Chemin du stockage
← Spécification du sous domaine
← Si secure = true : sera transmis uniquement en https
← Empêche l'utilisation du cookie en Javascript

- Valeurs stockées sur le **PC du client**
- Donc -> Accessible facilement
- On peut donner une valeur d'expiration
- Utile pour stocker les données non sensible



Documentation complète :

<https://www.php.net/manual/en/function.setcookie.php>

Les cookies, exemple d'utilisation

```
<?php
```

```
$int = 50000;  
setcookie("mon-cookie", "ma-valeur", time() + $int, " ", " ", true, true);
```

```
echo $_COOKIE["mon-cookie"];
```

```
?>
```



Les cookies

Exercice

Customiser le site !

Couleur du texte : Bleu 

- Créer un formulaire en HTML
- Créer une liste déroulante
- Mettre plusieurs couleurs en option au choix
- A l'envoi du formulaire, stocker dans un cookie la couleur sélectionnée par la personne
- Changer la couleur du texte grâce au cookie sur le site Internet



Les cookies

Exercice : création d'un mode sombre

- Créer un formulaire en HTML
- Créer un bouton “switch” permettant de passer du mode sombre à clair
- A l'envoi du formulaire, stocker dans un cookie la couleur sélectionnée par la personne
- Changer les couleurs du site en fonction du mode



Les sessions

Les sessions permettent de stocker temporairement sur le serveur (par défaut) des informations liées aux internautes.

Elles sont notamment utilisées pour :

- Garder un utilisateur authentifié sur le site
- Stocker des informations redondantes (non sensibles). Exemple : email, nom, prénom
- Garder en mémoire le panier d'un utilisateur
- Stocker des notifications (messages flash)
- [...]

Les sessions

Schématisation du fonctionnement



<https://www.leboncoin.fr/index.php>

COOKIE avec identifiant unique de
session : 57894483422

COOKIE 57894483422



Serveur Web

Session 57894483422 = [
 "username": Jeff,
 "id": 12,
]

Les sessions

<?php

```
/*Pour prendre en compte les sessions,  
vous devez démarrer une session */  
session_start();
```

```
$monUtilisateurConnecte = [  
    'id' => 123,  
    'nom' => 'Martins',  
    'prenom' => 'Jeff'  
];
```

```
$_SESSION["user"] = $monUtilisateurConnecte;
```

```
var_dump($_SESSION['user']);
```

?>

- Valeurs stockées sur le serveur
- Donc -> "Sécurisé"
- On ne peut pas donner une valeur d'expiration : la session expire dans 24 minutes par défaut (dépend de la configuration du serveur (php.ini))

Supprimer une session :
`session_destroy();`

Les sessions

Exercice

Objectifs de la séance

- Stocker dans une session les identifiants d'une personne
- Utiliser la session et la supprimer (exercice précédent sur la l'inscription/connexion/Déconnexion/Suppression de compte)

Si session existe : utilisateur est connecté : afficher un message de bienvenue avec le nom de l'utilisateur (stocké dans la session) + date de connexion.

Créer un bouton de déconnexion : doit supprimer la session de l'utilisateur