

A background image of a savanna landscape with several elephants. One elephant in the foreground is drinking from a watering hole, with its trunk curled and water spraying. Other elephants are visible in the background, some standing and some partially obscured. The scene is set in a dry, open landscape under a bright sky.

Introduction à PHP

<https://github.com/heig-vd-progserv-course/heig-vd-progserv1-course>

Visualiser le contenu complet sur GitHub [à cette adresse](#).

L. Delafontaine, avec l'aide de [GitHub Copilot](#).

Ce travail est sous licence [CC BY-SA 4.0](#).

Retrouvez le contenu complet de cette présentation sur GitHub

Cette présentation est un résumé du contenu complet disponible sur GitHub.

Pour plus de détails, retrouvez le [contenu complet sur GitHub](#) ou en cliquant sur l'en-tête de ce document.

Objectifs (1)

- Expliquer le concept d'architecture client-serveur.
- Lister les outils nécessaires pour écrire et exécuter du code PHP.
- Expliquer comment PHP fonctionne dans un environnement web.
- Expliquer la syntaxe de base de PHP.
- Expliquer les variables en PHP.



Objectifs (2)

- Expliquer les constantes en PHP.
- Expliquer la nature dynamique des variables et constantes en PHP.
- Expliquer les opérateurs en PHP.
- Expliquer Les structures conditionnelles en PHP.
- Rédiger du code PHP simple.



Qu'est-ce que PHP (1)

- Jusqu'ici, vous avez développé des applications Java qui s'exécutent sur une seule machine.
- Avec PHP, plusieurs personnes vont pouvoir accéder à une application depuis leur navigateur web.



Qu'est-ce que PHP (2)

- PHP est un langage de programmation datant de 1994.
- Très utilisé pour le développement web.
- Basé sur une architecture client-serveur.
- Actuellement à la version 8.5.
- Un des langages les plus utilisés pour le développement web.

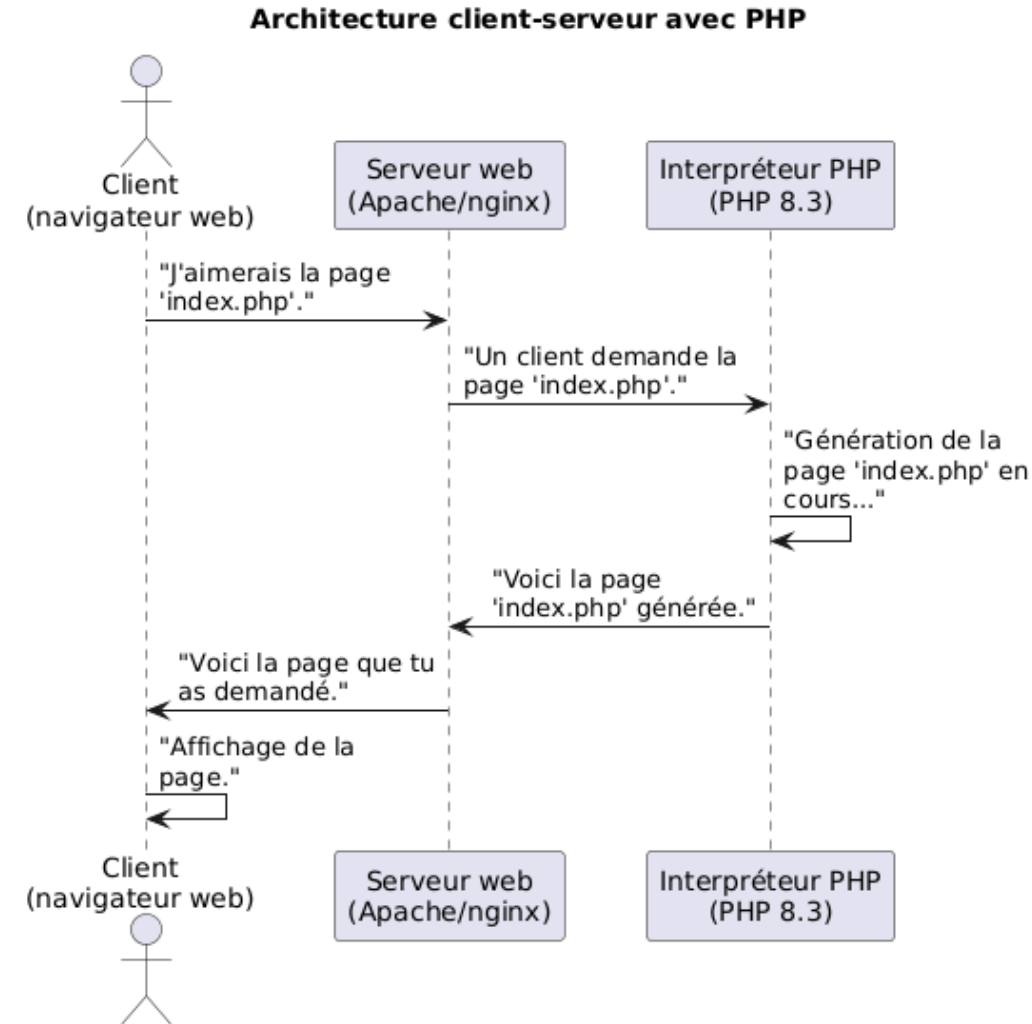


Applications web et architecture client-serveur

- La plupart des applications web modernes reposent sur une architecture dite "*client-serveur*" :
 1. Un client (navigateur web) envoie une requête à un serveur.
 2. Le serveur répond aux requêtes des différents clients.
 3. Le client affiche le résultat de la requête.
- PHP repose sur cette même architecture.

Comment fonctionne PHP

- PHP fonctionne grâce aux outils suivants :
 - Un serveur web.
 - PHP installé sur le serveur web.
 - Un navigateur web.
 - Un éditeur de code (pour le développement).



Comment écrire du code PHP (1)

- Code PHP dans un fichier `.php`.
- Le code PHP est écrit entre des balises `<?php` et `?>`.
- Le code PHP peut être mélangé avec du HTML (tout ce que vous avez appris en HTML s'applique).
- Uniquement `<?php` s'il n'y a que du code PHP.



Comment écrire du code PHP (2)

```
<?php
// Code PHP, dans un fichier `.php`
// Ici, il n'y a pas de balise de fermeture PHP
echo "Hello, World!";
```

```
// Équivalent en Java
public class Main {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}
```

Comment écrire du code PHP (3)

```
<!-- Code PHP, dans un fichier `.php` avec de l'HTML -->
<!-- Il y a la balise de fermeture PHP -->
<!DOCTYPE html>
<html>
<head>
    <title>PHP Test</title>
</head>
<body>
    <h1><?php echo "Hello, World!"; ?></h1>
</body>
</html>
```

Comment exécuter du code PHP

- Il faut avoir un serveur web avec PHP installé sur votre machine
- Heureusement, il existe des solutions toutes faites pour cela :
 - [WampServer](#).
 - [MAMP](#).
 - [XAMPP](#).



Syntaxe de base de PHP

- Similaire à Java et d'autres langages de programmation.
- Comme n'importe quelle langue ou langage de programmation, PHP a ses propres règles de syntaxe.
- Il s'agit de les apprendre et de les comprendre pour lire et écrire du code PHP de manière efficace.



Les commentaires

- Comme dans Java, `//` pour un commentaire sur une seule ligne.
- Comme dans Java, `/* ... */` pour un commentaire sur plusieurs lignes.

```
<?php
```

```
// Ceci est un commentaire sur une seule ligne
```

```
/*
```

```
Ceci est un commentaire  
sur plusieurs lignes
```

```
*/
```

Les variables (1)

- Les variables sont des conteneurs pour stocker des données.
- Les variables sont déclarées avec le signe `$`.
- Les variables peuvent contenir des chaînes de caractères, des nombres, des booléens, des tableaux, etc.



Les variables (2)

```
<?php
// Déclaration d'une variable - une variable commence par le signe `$`
$variable = "Hello, World!";

// Affichage de la variable
echo $variable;

// Modification de la variable
$variable = "Goodbye, World!";

// Affichage de la variable modifiée
echo $variable;
```

Les variables (3)

```
// Équivalent en Java
public static void main(String[] args) {
    // Déclaration d'une variable
    String variable = "Hello, World!";

    // Affichage de la variable
    System.out.print(variable);

    // Modification de la variable
    variable = "Goodbye, World!";

    // Affichage de la variable modifiée
    System.out.print(variable);
}
```

Type de données et typage dynamique (1)

- PHP est un langage de programmation à typage dynamique.
- Il n'y a pas besoin de déclarer le type de données d'une variable.
- Le type de données d'une variable est déterminé par la valeur qui lui est assignée.



Type de données et typage dynamique (2)

```
<?php
// Variable de type chaîne de caractères
$variable = "Hello, World!";

// Variable de type nombre
$variable = 42;

// Variable de type nombre flottant
$variable = 3.14;

// Variable de type booléen
$variable = true;
```

Type de données et typage dynamique (3)

```
// Équivalent en Java
public static void main(String[] args) {
    // Variable de type chaîne de caractères
    String variable1 = "Hello, World!";

    // Variable de type nombre
    int variable2 = 42;

    // Variable de type nombre flottant
    double variable3 = 3.14;

    // Variable de type booléen
    boolean variable4 = true;
}
```

Les chaînes de caractères (1)

- Les chaînes de caractères sont déclarées entre des guillemets simples (') ou doubles (").
- Permettent de créer des mots ou des phrases.
- Pour concaténer des chaînes de caractères, on utilise le point (.).



Les chaînes de caractères (2)

```
<?php
$string = "Hello, World!";

echo $string;
```

```
// Équivalent en Java
public static void main(String[] args) {
    String string = "Hello, World!";

    System.out.print(string);
}
```

Les chaînes de caractères (3)

```
<?php
$first = "Hello, ";
$second = "World!";

echo $first . $second;
```

```
// Équivalent en Java
public static void main(String[] args) {
    String first = "Hello, ";
    String second = "World!";

    System.out.print(first + second);
}
```

Les nombres (1)

- Les nombres sont déclarés sans guillemets.
- Il existe deux types de nombres en PHP : les entiers et les flottants.
- Les entiers sont des nombres entiers positifs ou négatifs.
- Les flottants sont des nombres à virgule, eux aussi, positifs ou négatifs.



Les nombres (2)

```
<?php
// Entier
$integer = 42;

// Flottant
$float = 3.14;

// Affichage des nombres
echo "\$integer contains $integer<br>";
echo "\$float contains $float";
```

Les nombres (3)

```
// Équivalent en Java
public static void main(String[] args) {
    // Entier
    int myInteger = 42;

    // Flottant
    double myFloat = 3.14;

    // Affichage des nombres
    System.out.println("myInteger contains " + myInteger);
    System.out.print("myFloat contains " + myFloat);
}
```

Les booléens (1)

- Les booléens sont des valeurs qui peuvent être soit vraies (`true`) soit fausses (`false`).
- Les booléens sont déclarés en PHP avec les mots-clés `true` et `false` .
- Les booléens sont souvent utilisés pour des conditions dans les structures de contrôle conditionnelles.



Les booléens (2)

```
<?php
// Vrai
$doILikeDogs = true;

// Faux
$doILikeHomework = false;

// Affichage des booléens - `false` est affiché comme une chaîne vide
echo "\$doILikeDogs contains $doILikeDogs<br>";
echo "\$doILikeHomework contains $doILikeHomework";
```

Les booléens (3)

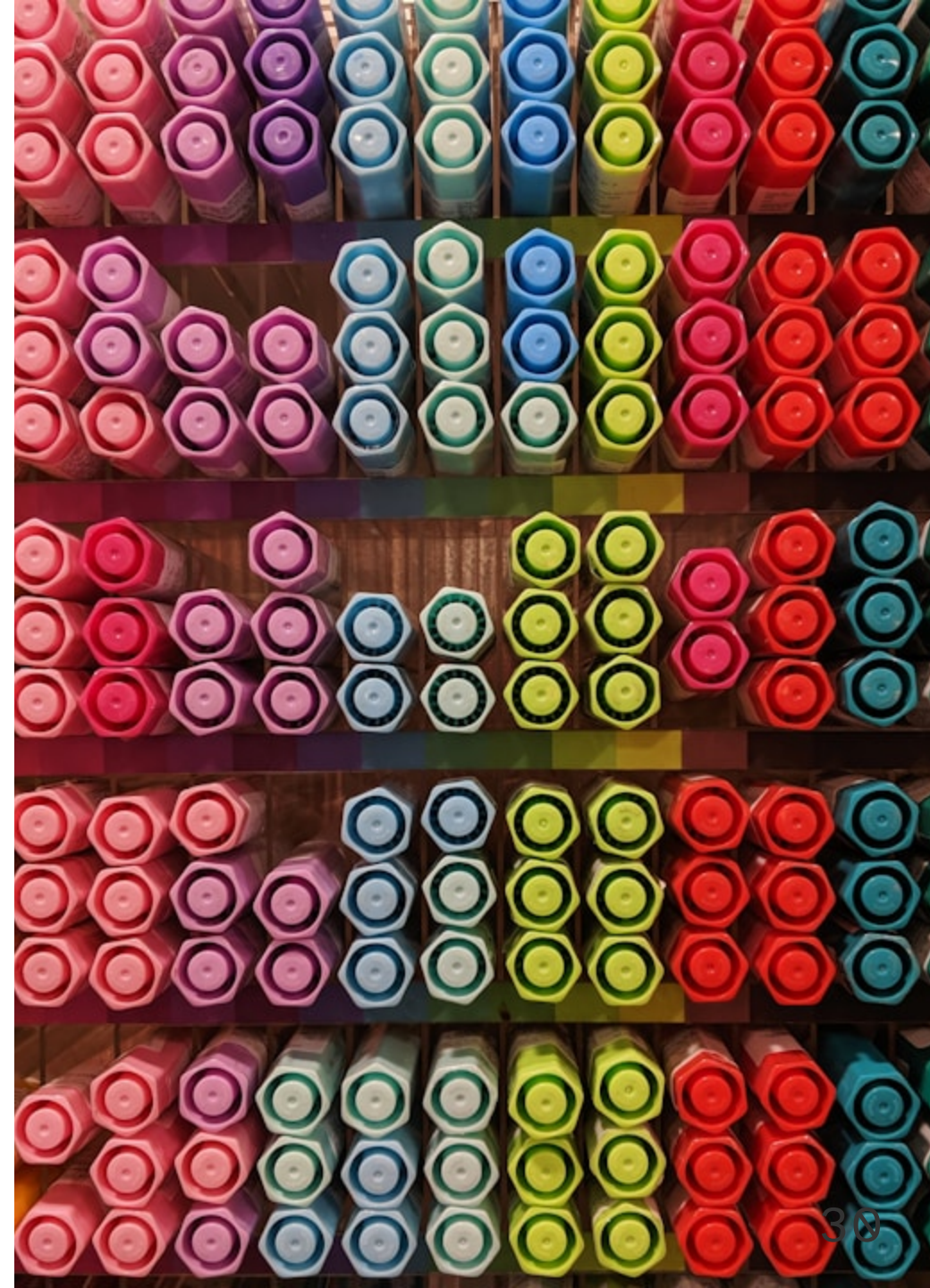
```
// Équivalent en Java
public static void main(String[] args) {
    // Vrai
    boolean doILikeDogs = true;

    // Faux
    boolean doILikeHomework = false;

    // Affichage des booléens
    System.out.println("doILikeDogs contains " + doILikeDogs);
    System.out.print("doILikeHomework contains " + doILikeHomework);
}
```

Les tableaux (1)

- Les tableaux sont des collections de valeurs.
- Les tableaux sont déclarés entre des crochets (`[]`) ou avec la fonction `array()`.
- Les valeurs d'un tableau sont indexées à partir de 0.
- Nous étudierons les tableaux plus en détails dans un prochain cours.



Les tableaux (2)

```
<?php
// Déclaration d'un tableau
$array = ["apple", "banana", "cherry"];

// Affichage de la première valeur du tableau
// Les tableaux sont indexés à partir de 0
echo "$array[0]<br>";

// Déclaration d'un tableau (alternative)
$array = array("apple", "banana", "cherry");

// Affichage de la troisième valeur du tableau
echo "$array[2]";
```

Les tableaux (3)

```
// Équivalent en Java
public static void main(String[] args) {
    // Déclaration d'un tableau
    String[] array = {"apple", "banana", "cherry"};

    // Affichage de la première valeur du tableau
    System.out.println(array[0]);

    // Affichage de la troisième valeur du tableau
    System.out.println(array[2]);
}
```

Les constantes (1)

- Les constantes sont des valeurs qui ne peuvent pas être modifiées.
- Les constantes sont déclarées avec le mot-clé `const` ou avec la fonction `define()`.
- La convention veut que les constantes soient écrites en majuscules.

Les constantes (2)

```
<?php
// Définition d'une constante
const CONSTANT = "Hello, World!";

// Affichage de la constante
echo CONSTANT;

// Tentative de modification de la constante (erreur)
CONSTANT = "Goodbye, World!";

// Définition d'une constante (alternative)
define("CONSTANT", "Hello, World!");
```

Les constantes (3)

```
// Équivalent en Java
public static void main(String[] args) {
    // Définition d'une constante
    final String CONSTANT = "Hello, World!";

    // Affichage de la constante
    System.out.println(CONSTANT);

    // Tentative de modification de la constante (erreur)
    CONSTANT = "Goodbye, World!";
}
```

Les opérateurs (1)

- Permet d'effectuer des opérations sur des variables et des valeurs.
- Opérateurs arithmétiques : `+`, `-`, `*`, `/`, `%` (modulo)
- Opérateurs de comparaison : `===` (égal), `!==` (différent), `>` (supérieur), `<` (inférieur)
- Opérateurs logiques : `&&` (et), `||` (ou), `!` (non/inversion)



Les opérateurs (2)

```
<?php
$sum = 1 + 1; // ` $sum ` contiendra 2
$difference = $sum - 1; // ` $difference ` contiendra 1
$product = 2 * 2; // ` $product ` contiendra 4
$quotient = $product / 2; // ` $quotient ` contiendra 2

echo "Sum: $sum<br>";
echo "Difference: $difference<br>";
echo "Product: $product<br>";
echo "Quotient: $quotient";
```

Les opérateurs (3)

```
// Équivalent en Java
public static void main(String[] args) {
    int sum = 1 + 1; // `sum` contiendra 2
    int difference = 2 - 1; // `difference` contiendra 1
    int product = 2 * 2; // `product` contiendra 4
    int quotient = 4 / 2; // `quotient` contiendra 2

    System.out.println("Sum: " + sum);
    System.out.println("Difference: " + difference);
    System.out.println("Product: " + product);
    System.out.print("Quotient: " + quotient);
}
```

Les opérateurs (4)

```
<?php
// ` $modulo ` contiendra 1
// Il reste 1 après la division de 5 par 2, 5 n'est donc pas pair.
$modulo = 5 % 2;

echo "Modulo: $modulo<br>";

// ` $modulo ` contiendra 0
// Il ne reste rien après la division de 6 par 2, 6 est donc pair.
$modulo = 6 % 2;

echo "Modulo: $modulo";
```

Les opérateurs (5)

```
// Équivalent en Java
public static void main(String[] args) {
    // `modulo` contiendra 1
    // Il reste 1 après la division de 5 par 2, 5 n'est donc pas pair.
    int modulo = 5 % 2;

    System.out.println("Modulo: " + modulo);

    // `modulo` contiendra 0
    // Il ne reste rien après la division de 6 par 2, 6 est donc pair.
    modulo = 6 % 2;

    System.out.print("Modulo: " + modulo);
}
```

Les structures conditionnelles (1)

- Appelées parfois "*structures de contrôle conditionnelles*".
- Permettent de contrôler le flux d'exécution d'un programme.
- Utilisent les opérateurs de comparaison et logiques.
- Elles se composent de `if`, `else`, `elseif` et `switch`.



Les structures conditionnelles (2)

```
<?php
$a = 1;
$b = 2;

if ($a < $b) {
    echo "a is less than b";
} elseif ($a === $b) {
    echo "a is equal to b";
} else {
    echo "a is greater than b";
}
```

Les structures conditionnelles (3)

```
// Équivalent en Java
public static void main(String[] args) {
    int a = 1;
    int b = 2;

    if (a < b) {
        System.out.print("a is less than b");
    } else if (a == b) {
        System.out.print("a is equal to b");
    } else {
        System.out.print("a is greater than b");
    }
}
```

Les structures conditionnelles (4)

```
<?php
// Déclaration de deux variables
$age = 18;
$country = "Switzerland";

// Vérification si `$age` est supérieur ou égal à 18
// et si `$country` est égal à "Switzerland"
if ($age >= 18 && $country === "Switzerland") {
    echo "You are allowed to vote in Switzerland.";
}
```

Les structures conditionnelles (5)

```
// Équivalent en Java
public static void main(String[] args) {
    // Déclaration de deux variables
    int age = 18;
    String country = "Switzerland";

    // Vérification si `age` est supérieur ou égal à 18
    // et si `country` est égal à "Switzerland"
    if (age >= 18 && country.equals("Switzerland")) {
        System.out.print("You are allowed to vote in Switzerland.");
    }
}
```

Les structures conditionnelles (6)

```
<?php
// Déclaration d'une variable
$color = "red";

// Vérification de la variable ` $color `
switch ($color) {
    // Si la variable ` $color ` est égale à "red"
    case "red":
        echo "The color is red.";
        break;
```

```
// Si la variable `$color` est égale à "blue"
case "blue":
    echo "The color is blue.";
    break;
// Par défaut
default:
    echo "The color is neither red nor blue.";
}
```

Les structures conditionnelles (7)

```
// Équivalent en Java
public static void main(String[] args) {
    // Déclaration d'une variable
    String color = "red";

    // Vérification de la variable `color`
    switch (color) {
        // Si la variable `color` est égale à "red"
        case "red":
            System.out.println("The color is red.");
            break;
    }
}
```

```
// Si la variable `$color` est égale à "blue"
case "blue":
    System.out.println("The color is blue.");
    break;
// Par défaut
default:
    System.out.println("The color is neither red nor blue.");
}
}
```

Les structures conditionnelles (8)

- Deux formes d'opérateurs conditionnels :
 - `==` : vérifie l'égalité des valeurs (type non pris en compte).
 - `===` : vérifie l'égalité des valeurs et des types.
- Préférer l'utilisation de `===` pour éviter des erreurs inattendues.



Les structures conditionnelles (9)

```
<?php
// Déclaration de deux variables numériques
$a = 0;
$b = 0;
$c = "0";

// Vérification si `$a` est égal à `$b`)
if ($a == $b) {
    echo "a is equal to b (same value)";
}

// Vérification si `$a` est égal à `$c`)
if ($a == $c) {
    echo "a is equal to c (same value)";
}
```

```
// Vérification si `$a` est égal (identique) à `$b`  
if ($a === $b) {  
    echo "a is identical to b (same value and type)";  
}  
  
// Vérification si `$a` est égal (identique) à `$c`  
if ($a === $c) {  
    echo "a is identical to c (same value and type)";  
} else {  
    echo "a is not identical to c (different type)";  
}
```

Utilisez toujours `===` / `!==` sauf si vous avez une bonne raison de faire autrement !

Il n'existe pas d'équivalent direct en Java pour les opérateurs `==` et `===` de PHP.

Conclusion

- PHP est un langage de programmation très utilisé pour le développement web.
- PHP repose sur une architecture client-serveur.
- PHP est un langage de programmation simple à prendre en main (il suffit de modifier le code, rafraîchir la page et le tour est joué).



Questions

Est-ce que vous avez des questions ?

Mini-projet

Mini-projet

- Application web en PHP pour gérer des animaux de compagnie.
- Permet de mettre en pratique le contenu théorique du cours.
- À réaliser tout au long de l'unité d'enseignement de façon guidée.
- Je suis là pour vous aider si besoin.



Exercices

Exercices

- Permet d'exercer les concepts vus en cours, autant théoriques que pratiques.



À vous de jouer !

- (Re)lire le [support de cours](#).
- Réaliser le [mini-projet](#).
- Faire les [exercices](#).
- Poser des questions si nécessaire.

Pour le mini-projet ou les exercices, n'hésitez pas à vous entraidez si vous avez des difficultés !



Sources (1)

- [Illustration principale](#) par [Richard Jacobs](#) sur [Unsplash](#)
- [Illustration](#) par [Aline de Nadai](#) sur [Unsplash](#)
- [Illustration](#) par [Michiel Leunens](#) sur [Unsplash](#)
- [Illustration](#) par [Taylor Vick](#) sur [Unsplash](#)
- [Illustration](#) par [Aaron Burden](#) sur [Unsplash](#)
- [Illustration](#) par [Fejuz](#) sur [Unsplash](#)
- [Illustration](#) par [Jan Huber](#) sur [Unsplash](#)
- [Illustration](#) par [Kier in Sight Archives](#) sur [Unsplash](#)

Sources (2)

- [Illustration](#) par [Nick Hillier](#) sur [Unsplash](#)
- [Illustration](#) par [Brooke Lark](#) sur [Unsplash](#)
- [Illustration](#) par [Faris Mohammed](#) sur [Unsplash](#)
- [Illustration](#) par [Kenny Eliason](#) sur [Unsplash](#)
- [Illustration](#) par [charlesdeluvio](#) sur [Unsplash](#)
- [Illustration](#) par [Arham Jain](#) sur [Unsplash](#)
- [Illustration](#) par [Alec Favale](#) sur [Unsplash](#)
- [Illustration](#) par [Samuel Girven](#) sur [Unsplash](#)

Sources (3)

- [Illustration](#) par [Nikita Kachanovsky](#) sur [Unsplash](#)