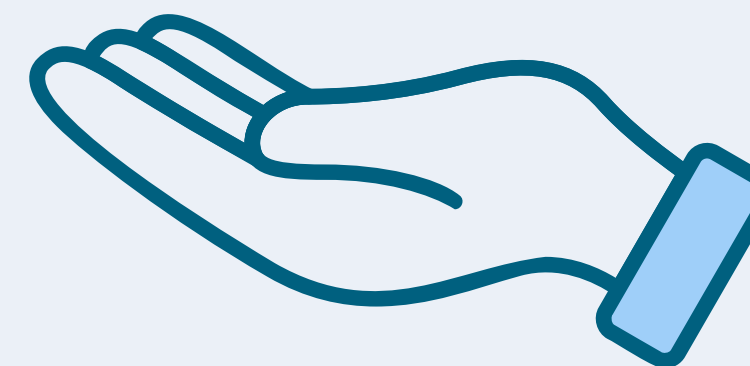




# INTRODUCTION À JAVASCRIPT

Les bases du langage

Laëtitia DELOIRS



# AU PROGRAMME...

- Présentations
- 1. Introduction
- 2. Manipulation des variables
- 3. Chaînes de caractères
- 4. Calculs et opérateurs

# 1. INTRODUCTION À JAVASCRIPT

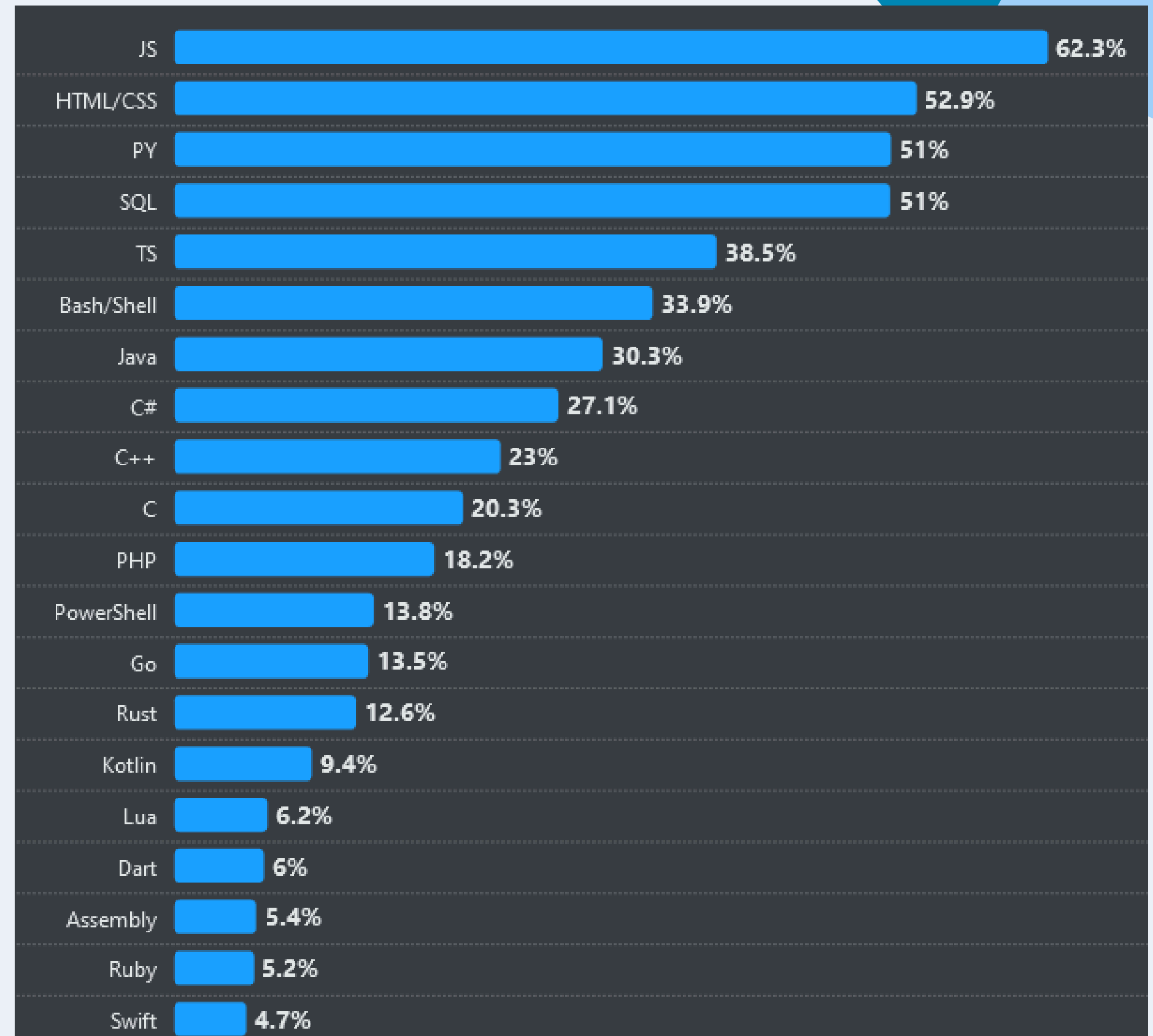
# C'EST QUOI JAVASCRIPT ?

- Langage de programmation créé par Brendan Eich en 1995
- Interprété et exécuté côté client : par le navigateur (Chrome, Firefox, Opera, Safari, Edge, etc.)
- Peut être également exécuté côté serveur : avec Node.js
- Utilisé pour rendre les pages web interactives (interaction avec le DOM) => contrôle des éléments HTML et CSS

=> l'utilisateur va pouvoir interagir avec notre page web

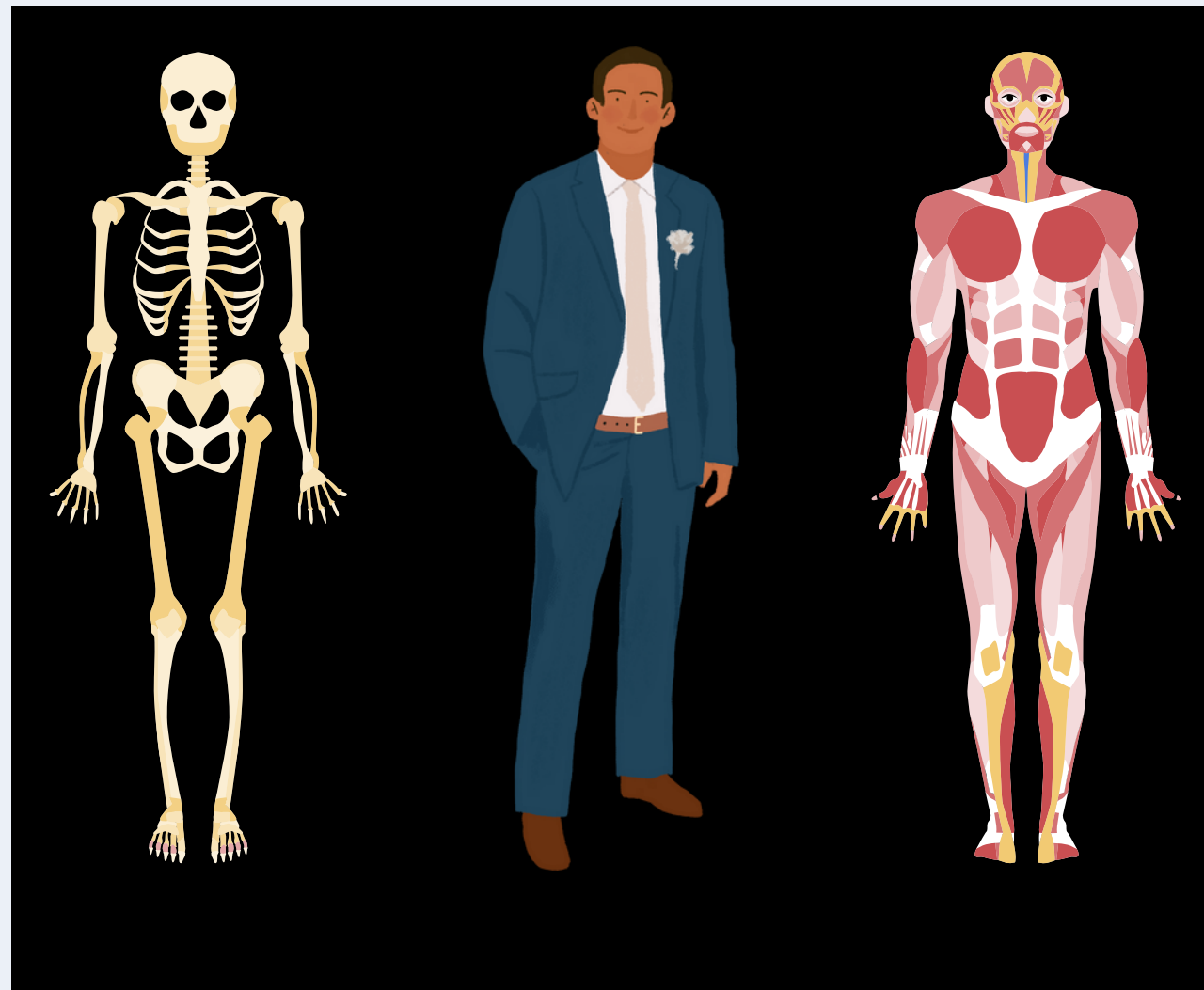
## LES ENJEUX DE JS AUJOURD'HUI

- Langage essentiel pour le développement front-end, largement répandu
- Grande communauté
- Beaucoup de ressources
- à la base de plusieurs frameworks : React, Vue, Angular...



<https://survey.stackoverflow.co/2024/technology#most-popular-technologies-language>

# JAVASCRIPT, HTML ET CSS



HTML : la  
structure

CSS : le  
style

JS :  
interactions  
et logique

# AVANT DE COMMENCER

Prérequis : HTML + CSS

SetUp :

- Un navigateur (Chrome, Firefox, Opera, Safari, Edge, etc.) à jour
- Un éditeur de code (VsCode, NotePad++, Sublime Text, etc.)
- (l'extension de VsCode "Live Server", pas obligatoire mais pratique)

# COMMENT INTÉGRER DU JAVASCRIPT DANS UNE PAGE HTML ?

- Un fichier JS ne s'exécute pas tout seul : il faut le lancer depuis le fichier HTML
- On utilise la balise **<script>** pour intégrer ou lier du code JS à une page HTML



# COMMENT INTÉGRER DU JAVASCRIPT DANS UNE PAGE HTML ?

- Un fichier JS ne s'exécute pas tout seul : il faut le lancer depuis le fichier HTML
- On utilise la balise **<script>** pour intégrer ou lier du code JS à une page HTML
- Deux méthodes :

- Dans le <head> du fichier HTML, une balise <script> vide + un attribut src avec le chemin vers notre fichier.js : (recommandé)

+ Utilisation de l'attribut async

```
<head>  
  <meta charset="UTF-8">  
  <title>Mon site web</title>  
  <script src="test.js"></script>  
</head>
```

# COMMENT INTÉGRER DU JAVASCRIPT DANS UNE PAGE HTML ?

- Un fichier JS ne s'exécute pas tout seul : il faut le lancer depuis le fichier HTML
- On utilise la balise **<script>** pour intégrer ou lier du code JS à une page HTML
- Deux méthodes :
  - Directement dans le corps du HTML, dans une balise script, sans attribut src (peu recommandé) :

```
<body>  
  <script> console.log("Hey !!") </script>  
</body>
```

## **2. MANIPULATION DES VARIABLES**

# C'EST QUOI UNE VARIABLE ?

- Une “boîte” qui permet de stocker des données
- Elle a un nom, un type et une valeur
- Elle sert à stocker des données de différents types (nombre, chaîne de caractères, tableau, objet,...) pour les utiliser et les modifier à différents moments dans un programme
- Déclaration ( `let a;` ), affectation ( `a=5;` ), initialisation ( `let b=12;` )

# LES DIFFÉRENTS TYPES DE DONNÉES

Les types les plus utilisés : **Number**, **String**, **Boolean**, **Null**, **Undefined**, **Object**

Il existe également les types **Symbol** et **BigInt**

# COMMENT NOMMER UNE VARIABLE ?

- Une ou plusieurs lettres
- Peut contenir des chiffres, mais doit commencer obligatoirement par une lettre
- Pas de ponctuation
- Court, (en anglais ? 😊) et explicite
- écriture en camelCase

```
let myVariableCamelCase = "Hello !"
```

# COMMENT DÉCLARER UNE VARIABLE ?

- On utilise principalement ***let*** et ***const***
- ***const*** est réservé aux constantes qui ne seront pas réaffectées par la suite

(on peut rencontrer *var* dans du code existant, mais ce mot clé n'est plus utilisé dans le code récent)

# 3. CHÂÎNES DE CARACTÈRES



# COMMENT ÉCRIRE UNE CHAÎNE (STRING)

- On utilise les “doubles quotes” ou ‘simples quotes’
- Une lettre ou plusieurs, mot(s), phrase(s) ou paragraphe(s)
- La **concaténation** avec +
- L’échappement de caractère avec \ (antislash : altgr+8)
- Interpolation avec les `backticks` (altgr+7) : le template string

# MÉTHODES UTILES

- .length : connaître la longueur d'une chaîne
- Connaître le premier caractère d'une string avec [0]
- toUpperCase() : convertir en majuscules
- toLowerCase() : convertir en minuscules

# MÉTHODES UTILES (SUITE)

- `replace()` : prend en 1er paramètre la chaîne à remplacer, et en 2e la chaîne remplaçante
- `slice()` : prend en paramètre la ou les positions de découpe et retourne une nouvelle chaîne
- `split()` : prend en paramètre l'élément séparateur, et retourne un tableau de sous-chaînes

# 4. CALCULS ET OPÉRATEURS

# CALCULS ET OPÉRATEURS

## LES OPÉRATEURS ARITHMÉTIQUES

- + : addition
- - : soustraction
- \* : multiplication
- / : division
- % : modulo



## LES OPÉRATEURS D'AFFECTATION

## LES OPÉRATEURS LOGIQUES

**Mêmes règles qu'en mathématiques :**

- multiplication et division prioritaires
- utilisation des parenthèses

## LES OPÉRATEURS ARITHMÉTIQUES

- `+` : addition
- `-` : soustraction
- `*` : multiplication
- `/` : division
- `%` : modulo

## LES OPÉRATEURS D'AFFECTATION

- `=` : affectation simple
- `+=`
- `-=`
- `*=`
- `/=`

## LES OPÉRATEURS LOGIQUES

- `&&` : ET
- `||` : OU

## LES OPÉRATEURS D'INCRÉMENTATION

- `++` : augmenter de 1
- `--` : diminuer de 1

# MÉTHODES UTILES

- `toFixed()` : arrondit un nombre au nombre spécifié de décimales
- `Math.floor()` : arrondit un nombre à l'entier inférieur

# 5. CONDITIONS





# INTRO AUX CONDITIONS EN JS

Les conditions permettent d'exécuter un bloc de code uniquement si une certaine condition est remplie.

```
if(condition){  
  //code exécuté si la condition est vraie  
}
```

# STRUCTURE

- on utilise **if** pour tester une première condition
- (puis **else if** pour tester d'autres conditions si la première est fausse)
- (puis **else** pour définir le cas par défaut si aucune condition n'est vraie)

Exemple 1 : renvoyer un message en fonction de l'âge

# LES OPÉRATEURS DE COMPARAISON

- `==` : égalité de valeurs
- `===` : égalité stricte (valeur et type)
- `!=` : Différence de valeurs
- `!==` : Différence stricte (valeur ou type)
- `>` : Strictement supérieur à
- `<` : Strictement inférieur à
- `>=` : Supérieur ou égal
- `<=` : Inférieur ou égal

# OPÉRATEURS LOGIQUES

- `&&` : ET  $\Rightarrow$  les deux conditions doivent être vraies
- `||` : OU  $\Rightarrow$  l'une des deux au moins (ou les deux) doit être vraie
- `!` : NON  $\Rightarrow$  inverse la valeur de vérité

Exemple2

# OPÉRATEUR TERNAIRE

Manière plus rapide et concise d'écrire une condition **if** simple.

```
condition ? valeurSiVraie : valeurSiFausse;
```

# SWITCH CASE

Le switch est utile lorsqu'on veut tester plusieurs valeurs d'une variable et exécuter un code différent selon la valeur trouvée.

```
✓ switch (variable) {  
✓     case valeur1:  
✓         // Code si variable vaut valeur1  
✓         break;  
✓     case valeur2:  
✓         // Code si variable vaut valeur2  
✓         break;  
✓     default:  
✓         // Code par défaut si aucune valeur ne correspond  
✓         break;  
}
```

# 6. TABLEAUX



# C'EST QUOI UN TABLEAU ?

Un tableau est une liste ordonnée d'éléments, qui peuvent être des nombres, des chaînes, des objets, ou même d'autres tableaux.

On l'encadre de crochets, et les éléments sont séparés par une virgule.

```
let fruits = ["pomme", "banane", "orange"];
```



# COMMENT ACCÉDER AUX ÉLÉMENTS ?

```
let fruits = ["pomme", "banane", "orange"];
```

On accède aux éléments par leur index, qui commence à zéro

```
console.log(fruits[0]); // "pomme"  
console.log(fruits[2]); // "orange"
```

# MÉTHODES UTILES

- `push()` : ajouter un élément à la fin
- `pop()` : supprimer le dernier élément
- `shift()` : supprimer le premier élément
- `unshift()` : ajouter un élément au début