



1

Définissez par écrit ce que vous voulez faire : ce qui se conçoit bien s'énonce clairement !

2

Intégrez le design de votre objectif en HTML et CSS, sans Javascript. Ne faites pas tout d'un coup : segmenter la difficulté, c'est la clé.

3

Découpez le plus possible la logique de votre code en sous-logiques.

4

Ecrivez chaque bout de code l'un après l'autre en prenant soin de bien les tester à chaque fois.

5

Indentez votre code et commentez-le au fur et à mesure pour qu'il reste facilement maintenable et évolutif.

Définitions

Variable

Association d'un nom à une valeur. Elle est déclarée via un mot-clé et peut prendre plusieurs formes : texte, nombre, booléen, etc.

Indentation

Mise en forme qui facilite la lecture des lignes de code.

Opérateurs logiques

Caractères spéciaux qui permettent de lier plusieurs conditions ou de comparer des valeurs.

&&

||

<

>

>=

<=

==

===

!=

POO (Programmation Orientée Objet)

Modèle qui consiste à définir et faire interagir des éléments que l'on appelle "objets".

Objet

Représentation qui se rattache au monde physique : un livre, une page de livre, une lettre.

Code

javascript

```
1 class Person{
2   constructor(nom, age = 0){
3     //On récupère les informations de la personne en interne
4     this.nom = nom;
5     this.age = age;
6     this.isMajeur = (this.age >= 18);
7   }
8
9   salutation() {
10    /* On écrit un message de bienvenue
11     personnalisé selon qu'il est majeur ou non */
12    if(this.isMajeur) {
13      console.log("Bonjour " + this.nom + ", tu as " + this.age + " ans donc tu es
14      majeur.");
15    }else{
16      console.log(`Bonjour ${this.nom}, tu as ${this.age} ans donc tu es mineur.`);
17    }
18  }
19  const person1 = new Person("Jean", 22);
20  person1.salutation(); // Affichera dans la console 'Bonjour Jean, tu as 22 ans donc tu es
21  majeur.'
22  const person2 = new Person("Luc", 12);
23  person2.salutation(); // Affichera dans la console 'Bonjour Luc, tu as 12 ans donc tu es
24  mineur.'
```

Bonnes pratiques

- ✓ Nommer les variables et les fonctions de manière explicite.
- ✓ Coder en POO pour regrouper les blocs de logique.
- ✓ Pratiquer avec des projets personnels, l'expérience est la clé du succès.
- ✓ Prendre en compte la portée des variables.
- ✓ Analyser les erreurs dans la console.
- ✓ Lire les documentations.

Erreurs classiques

- ✗ Faire des fonctions trop longues qui font trop de choses.
- ✗ Utiliser le mauvais type de variable (on ne peut pas faire d'opération mathématique sur des chaînes de caractères).