

JavaScript

Déclaration d'une variable

On ne déclare pas le type d'un variable

```
let variable = valeur;  
const constante = valeur; // Déclarer une constante  
let string = "valeur"; let string = 'valeur'; // Déclarer un string  
let boolean = true; let boolean = false; // Déclarer une variable boolean
```

Modifier la valeur d'une variable

=> il suffit de la réaffecter avec le =

Opération arithmétique

```
=> variable++;  
=> variable += valeur;
```

String

Concaténation : var = string1 + string2;

Retourner la valeur d'une var string : var = "Bonjour \${string} !"

Définir un objet

Les objets sont définies dans les fichiers JSON

```
let Objet = {  
    key1 : value1  
    key2 : value2  
    key3 : value3  
};
```

```
let ObjectKey1 = Objet.key1; // Accéder à une valeur d'un clé d'un objet  
let ObjectKey2 = Objet["key2"]; // Accéder à une valeur d'un clé d'un objet
```

Les classes

```
class Book {  
    constructor(title, author, pages) {  
        this.title = title;  
        this.author = author;  
        this.pages = pages;  
    }  
}
```

Définir une méthode de classe :

```
Nom_Methode(arg1, arg2, ...) {  
    ...  
    return 0;  
}
```

Définir une méthode statique de classe :

```
static Nom_Methode(arg1, arg2, ...) {  
    ...  
    return 0;  
}
```

Listes / Tableau

```
let guests = [ ]; // Créer une liste vide  
let guests = ["Sarah Kate", "Audrey Simon", "Will Alexander"]; // Liste initialisée  
// Accéder aux valeurs d'un tableau (indice de la liste commence de 0)  
let firstGuest = guests[0]; // "Sarah Kate"  
let thirdGuest = guests[2]; // "Will Alexander"  
// Taille d'une liste  
let howManyGuests = guests.length; // 3  
// Ajouter l'élément à la fin du tableau  
guests.push("Tao Perkington"); // ajoute "Tao Perkington" à la fin du tableau guests  
// Ajouter l'élément au début du tableau  
guests.unshift("Tao Perkington"); // Ajoute au début du tableau  
// Supprimer le dernier élément du tableau  
guests.pop(); // supprimer le dernier élément du tableau
```

Conditionnel

```
if (myBoolean) { ...  
} else if (Boolean) { ...  
} else { ...  
}
```

Opérateurs logiques

== / != : vérifie si les valeurs sont égales sans vérifier le type
=== / !== : vérifie l'égalité des valeurs et des types
&& : et
|| : ou
! : negation

Switch

```
switch (variable) {  
  case valeur1:  
    ...  
  break;  
  case valeur2:  
    ...  
  break;  
  case valeur3:  
    ...  
  break;  
  default:  
    ...  
}
```

Les boucles

=> boucle for

```
const numberOfPassengers = 10;  
for (let i = 0; i < numberOfPassengers; i++) {  
  console.log("Passager embarqué !");  
}  
  
for (let i in passengers) { // Parcourir les indices d'un tableau  
  console.log("Embarquement du passager " + passengers[i]);  
}  
  
for (let passenger of passengers) { // Parcourir les éléments d'un tableau  
  console.log("Embarquement du passager " + passenger);  
}
```

=> boucle while

```
while (condition) {  
  ...  
}
```

Gestion des exceptions

```
try {  
  ...  
} catch {  
  ...  
}
```

Les fonctions

```
function afficherDeuxValeurs(valeur1, valeur2) {  
    ...  
}
```

```
const Nom_fonction = (arg1, arg2) => {  
    // instruction de la fonction  
}
```