

# Les variables

## C'est quoi une variable ?

Voyez les variables comme un conteneur pour une valeur. Elle peut contenir toute sorte de valeur comme des chaînes de caractères, des nombres, des tableaux ou encore des objets.

Elles doivent impérativement respecter certains critères:

- Le nom doit commencer par une lettre, un signe \$ ou un \_. Il ne doit pas commencer par un numéro.
- Le nom peut contenir des lettres, des chiffres, un signe \$ ou un trait de soulignement \_. Vous ne devez pas utiliser un tiret ou un point.
- Les noms de variables ne peuvent pas contenir d'espace
- On ne peut pas utiliser de **Keyword réservé**

Il existe 3 manière de déclarer une variable en JavaScript : **var**, **let** et **const**

# Utilisation de let

Dans cet exemple: **x**, **y**, et **z**, sont des variables, déclarées avec le mot clé **let**

```
<script>
  let x = 5;
  let y = 6;
  let z = x + y;
  document.getElementById("demo").innerHTML = "La valeur de z est: " + z;
</script>
```

# Utilisation de var

Dans cet exemple: **x**, **y**, et **z**, sont des variables, déclarées avec le mot clé **var**

```
<script>
  var x = 5;
  var y = 6;
  var z = x + y;
  document.getElementById("demo").innerHTML = "La valeur de z est: " + z;
</script>
```

## Quand utiliser JavaScript var ?

Déclarez toujours les variables JavaScript avec **var**, **let** ou **const**.

Le mot-clé **var** est utilisé dans tout le code JavaScript de 1995 à 2015.

Les mots **let** et **const** ont été ajoutés à JavaScript en 2015.

Si vous souhaitez que votre code s'exécute dans un navigateur plus ancien, vous devez utiliser var

# L'utilisation de const

**const** permet de déclarer une variable qui sera en "lecture seul", c'est à dire que sa valeur ne pourra pas être modifié durant l'exécution du script.

Il est impératif d'assigner une valeur lors de la déclaration d'une variable **const**.

```
<script>
  const price1 = 5;
  const price2 = 6;
  let total = price1 + price2;
  document.getElementById("demo").innerHTML = "Le total est: " + total;
</script>
```

Les prix sont déclarés avec le mot-clé const.

Ces prix sont des valeurs **constantes** et ne peuvent être **modifiés**.

# Identifiants JavaScript

Toutes les variables JavaScript doivent être identifiées avec des noms uniques .

Ces noms uniques sont appelés **identifiants** .

Les identifiants peuvent être des noms courts (comme x et y) ou des noms plus descriptifs (age, sum, totalVolume).

Les règles générales de construction des noms de variables (identifiants uniques) sont :

- Les noms peuvent contenir des lettres, des chiffres, des traits de soulignement et des signes dollar.
- Les noms doivent commencer par une lettre
- Les noms peuvent aussi commencer par \$ et \_
- Les noms sont sensibles à la casse (y et Y sont des variables différentes)
- Les mots réservés (comme les mots-clés JavaScript) ne peuvent pas être utilisés comme noms de variable.

A noter que les identifiants (variables) sont sensibles à la casse.

# L'opérateur d'affectation

En JavaScript, le signe égal ( = ) est un opérateur "d'affectation", pas un opérateur "égal à".

L'opérateur « égal à » s'écrit avec un double égal ( == ) en JavaScript

# Types de données JavaScript

Les variables JavaScript peuvent contenir des nombres comme 100 et des valeurs de texte comme « Papi ». En programmation, les valeurs de texte sont appelées chaînes de texte.

```
<script>
  const pi = 3.14;
  let person = "Papi";
  let answer = "Cours de JavaScript";
  document.getElementById("demo").innerHTML = pi + "<br>" + person + "<br>" + answer;
</script>
```

JavaScript peut gérer de nombreux types de données, mais pour l'instant, pensez simplement aux nombres et aux chaînes.

Les chaînes sont écrites entre guillemets simples ou doubles. Les nombres sont écrits sans guillemets.

Si vous mettez un nombre entre guillemets, il sera traité comme une chaîne de texte.

# Commentaires en JavaScript

Les commentaires JavaScript peuvent être utilisés pour expliquer le code JavaScript et le rendre plus lisible.

Les commentaires JavaScript peuvent également être utilisés pour empêcher l'exécution, lors du test de code alternatif.

## **Commentaires sur une seule ligne**

Les commentaires sur une seule ligne commencent par `//`.

Tout texte entre `//` et la fin de la ligne sera ignoré par JavaScript (ne sera pas exécuté).

## **Commentaires multi lignes**

Les commentaires multi lignes commencent par `/*` et se terminent par `*/`.

Tout texte entre `/*` et `*/` sera ignoré par JavaScript.