

Chapitre 17 . Bases du langage JavaScript

Le JavaScript est un langage dit « facile à apprendre, difficile à maîtriser ». Cela est d'autant plus vrai que ses applications sont de plus en plus variées et que son panel de fonctionnalités ne cesse de s'élargir depuis quelques années.

Ces particularités rendent le JavaScript à la fois incontournable et véritablement excitant mais en font également l'un des langages les plus durs (si ce n'est le plus dur) à maîtriser complètement.

Pas d'inquiétude cependant : 90% du JavaScript est relativement simple à comprendre et à apprendre et ce sont ces 90% qui vont être le plus souvent utilisés et peu nombreux sont les développeurs qui maîtrisent les 10% restants.

1- LES VARIABLES :

Déclaration des variables :

- Lorsqu'une variable est utilisée la première fois, il est nécessaire de la déclarer avec les mots clés `var` ou `let`
- On ne peut pas déclarer la même variable une seconde fois.
- En déclarant une variable avec `const`, il n'est plus possible de modifier sa valeur.

Type des variables : On retrouve les mêmes types de variables que ceux rencontrés en Python.

Python	JavaScript
<pre>>>> str(20)+'24' '2024'</pre>	<pre>>> String(20)+"24" ← "2024" ou >> a=20 ; a.toString()+"24" ← "2024"</pre>
<pre>>>> int('2')+2 4</pre>	<pre>>> parseInt('2')+2 ← 4</pre>
<pre>>>> float('2')+2 4.0</pre>	<pre>>> parseFloat('2')+2 ← 4</pre>
<pre>>>> type(True) <class 'bool'></pre>	<pre>>> typeof(true) ← "boolean"</pre>
<pre>>>> len("nsi") 3</pre>	<pre>>> 'nsi'.length</pre>
<pre>>>> a = 2024 >>> a == "2024" False</pre>	<pre>>> a=2024 ← 2024 >> a == "2024" ← true >> a === "2024" ← false</pre>

Exemples de déclarations de variables :

<pre>let nom = "Pierre"; nom = "Bonjour "+ nom ; console.log(nom) ;</pre> <p><i>Résultat dans la console :</i></p> <p>Bonjour Pierre</p>	<pre>let nom ; nom = "Pierre"; nom = "Bonjour "+ nom ; console.log(nom) ;</pre> <p><i>Résultat dans la console :</i></p> <p>Bonjour Pierre</p>
<pre>const nom = "Pierre"; nom = "Bonjour "+ nom ; console.log(nom) ;</pre> <p><i>Résultat dans la console :</i></p> <p>! ▶ Uncaught TypeError: invalid assignment to const 'nom'</p>	<pre>const nom ; nom = "Pierre"; console.log(nom) ;</pre> <p><i>Résultat dans la console :</i></p> <p>! Uncaught SyntaxError: missing = in const declaration</p>

2- BOUCLES FOR :

Python	JavaScript
<pre>i = 2024 for i in range(20) : print(i) print(i)</pre>	<pre>let i=2024; for (let i=0 ; i<20 ; i+=1){ console.log(i); } console.log(i);</pre>

3- BOUCLES WHILE :

Python	JavaScript
<pre>i = 0 while i < 20 : print(i) i = i + 1</pre>	<pre>let i=0; while(i < 20){ console.log(i); i = i + 1; }</pre>

4- STRUCTURE IF ELSE :

Python	JavaScript
<pre>a,b = 1,2 if a == b : print('Même valeur') elif a > b and (a!=0 or b!=0) : print('Plus grand et différent de 0') else : print('Plus petit')</pre>	<pre>let a = 1 , b = 2; if(a == b){ console.log('Même valeur'); }else if(a > b && (a!=0 b!=0)){ console.log('Plus grand et différent de 0'); }else { console.log('Plus petit'); }</pre>

5- FONCTIONS – PORTEE DES VARIABLES :

Python	JavaScript
<pre>a = 0 def change() : print(a) change() print(a)</pre>	<pre>let a = 0 function change(){ console.log(a) } change() console.log(a)</pre>
<pre>a = 0 def change() : a = a + 1 print(a) change() print(a)</pre> <p style="color: red; text-align: center;">a = a + 1 UnboundLocalError:</p>	<pre>let a = 0 function change(){ a = a + 1 console.log(a) } change() console.log(a)</pre>
<pre>a = 0 def change() : a = 0 a = a + 1 print(a) change() print(a)</pre>	<pre>let a = 0 function change(){ let a = 0 a = a + 1 console.log(a) } change() console.log(a)</pre>

6- FONCTIONS – PARAMETRES ET VALEURS DE RETOUR :

Python	JavaScript
<pre>def change(x,y) : a,b = 1,2 c = a+b+x+y return c a = change(3,4) print(a)</pre>	<pre>function change(x,y){ let a = 1 , b = 2 let c = a+b+x+y return c } let a = change(3,4) console.log(a)</pre>
<pre>>>> 10</pre>	<pre>▶ GET 10</pre>

Remarque : En JavaScript, on ne peut retourner qu'une seule variable.

7- LES LISTES :

Les listes de Python ont leur équivalent en JavaScript. On les appelle alors les « tableaux ».

Python	JavaScript
<pre>tab = ['lundi', 'mardi', 'mercredi'] print(tab[1]) tab.append('jeudi') print(len(tab)) print(tab) tab[4] = 'vendredi' print(tab)</pre>	<pre>let tab = ['lundi', 'mardi', 'mercredi'] console.log(tab[1]) tab.push('jeudi') console.log(tab.length) console.log(tab) tab[4] = 'vendredi' console.log(tab)</pre>
<p><u>Résultat dans la console :</u></p> <pre>>>> (executing file "coursJs.py") mardi 4 ['lundi', 'mardi', 'mercredi', 'jeudi'] tab[4] = 'vendredi' IndexError: list assignment</pre>	<p><u>Résultat dans la console :</u></p> <pre>▶ GET http://localhost:8080/coursJs.html mardi 4 ▶ Array(4) ["lundi", "mardi", "mercredi", "jeudi"] ▶ Array(5) ["lundi", "mardi", "mercredi", "jeudi", "vendredi"]</pre>

Remarque : En JavaScript, les listes sont des objets appelés « Array ». On peut y ajouter des éléments directement. Par exemple si le contenu de tab est

▶ Array(5) ["lundi", "mardi", "mercredi", "jeudi", "vendredi"], l'exécution de

tab[6] = 'dimanche' modifiera le contenu de tab de la façon suivante :

▶ Array(7) ["lundi", "mardi", "mercredi", "jeudi", "vendredi", <1 empty slot>, "dimanche"]

8- PARCOURS DE LISTES :

Python	JavaScript
<pre>i = 0 for elt in tab : print(i , ' ', elt) i = i + 1 for i in range(len(tab)) : print(i , ' ', tab[i])</pre>	<pre>for(let cle in tab) { console.log(cle , ' ', tab[cle]) } for(let i = 0 ; i < tab.length ; i++){ console.log(i , ' ', tab[i]) }</pre>
<p><u>Résultat dans la console :</u></p> <pre>0 lundi 1 mardi 2 mercredi 3 jeudi</pre>	<p><u>Résultat dans la console :</u></p> <pre>0 lundi 1 mardi 2 mercredi 3 jeudi 4 vendredi 5 undefined 6 dimanche</pre>

9- LES DICTIONNAIRES :

Les dictionnaires de Python ont leur équivalent en JavaScript. On les appelle alors les « tableaux associatifs ».

Python	JavaScript
<pre>ferie = {'Paques': '31-03-2024', 'Aid': '9-04-2024'} print(ferie) ferie['Noel'] = '25-12-2024' print(ferie) for cle in ferie : print(cle , ' ', ferie[cle])</pre>	<pre>let ferie = {'Paques': '31-03-2024', 'Aid': '9-04-2024'} console.log(ferie) ferie['Noel'] = '25-12-2024' console.log(ferie) for(let cle in ferie) { console.log(cle , ' ', ferie[cle]) }</pre>
<p><u>Résultat dans la console :</u></p> <pre>{'Paques': '31-03-2024', 'Aid': '9-04-2024'} {'Paques': '31-03-2024', 'Aid': '9-04-2024', 'Noel': '25-12-2024'} Paques 31-03-2024 Aid 9-04-2024 Noel 25-12-2024</pre>	<p><u>Résultat dans la console :</u></p> <pre>» Object { Paques: "31-03-2024", Aid: "9-04-2024" } » Object { Paques: "31-03-2024", Aid: "9-04-2024", Noel: "25-12-2024" } Paques 31-03-2024 Aid 9-04-2024 Noel 25-12-2024</pre>

On peut également créer le dictionnaire précédent à partir d'un dictionnaire vide. Cela donnera :

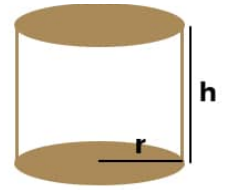
Python	JavaScript
<pre>ferie = {} ferie['Aid'] = '9-04-2024' ferie['Paques'] = '31-03-2024' ferie['Noel'] = '25-12-2024'</pre>	<pre>let ferie = {} ferie['Aid'] = '9-04-2024' ferie['Paques'] = '31-03-2024' ferie['Noel'] = '25-12-2024'</pre>

10- EXERCICES :

a. FONCTION QUI CALCULE UN VOLUME :

Le volume d'un cylindre est donné par la relation ci-contre avec $\pi \approx 3.14$.

$$\text{volume} = \pi \times r^2 \times h$$



Le code Python ci-dessous permet de calculer ce volume pour un cylindre de 1 m de rayon et 2 m de hauteur.

```
def volumeCylindre(r,h) :
    if r < 0 or h < 0 :
        return "les valeurs doivent être positives"
    else :
        volume = 3.14 * r**2 * h
        message = str(volume) + " m3"
        return message

# Programme principal
vol = volumeCylindre(1,2)
print(vol)
```

Question : Ecrire ce code en Javascript.

b. FONCTION QUI CALCULE UNE MOYENNE PONDEREE:

Le code Python ci-dessous permet de calculer la moyenne pondérée d'un élève. Dans les listes, on trouve une note sur 20 et son coefficient.

```
def moyennePonderee(liste) :
    somme = 0
    sommeCoefficient = 0
    for note in liste :
        somme = somme + note[0] * note[1]
        sommeCoefficient = sommeCoefficient + note[1]
    if sommeCoefficient == 0 :
        return None
    else :
        moyenne = somme / sommeCoefficient
        return moyenne

# Programme principal
moy = moyennePonderee([[8,2] , [12,0] , [13.5,1] , [5,0.5]])
print(moy)
```

Question : Ecrire ce code en Javascript.