

## DEVOIR 3 JAVASCRIPT

1. Initialiser une constante pi = 3,14

Assignez-lui une nouvelle valeur que se passe t-il?

```
>> const pi = 3.14;
← undefined

>> pi = 2;

! ▶ Uncaught TypeError: invalid assignment to const 'pi'
  <anonymous> debugger eval code:1
  [En savoir plus]
```

⇒ Un message d'erreur s'affiche car il est impossible de modifier la valeur de cette constante.

2. Déclarer et initialiser quatre variables correspondant à des nombres différents:

- une avec var uniquement;
- une avec let uniquement;
- une avec le nom de la variable uniquement;
- une avec le type d'objet.

```
>> let n1 = 1;
← undefined

>> var n2 = 2;
← undefined

>> n3 = 3;
← 3

>> n4 = new Number(4);
← ▶ Number { 4 }
```

Afficher le type de chaque variable, est-ce le même ? Si non, pourquoi selon vous ?

```
>> typeof(n1);
← "number"

>> typeof(n2);
← "number"

>> typeof(n3);
← "number"

>> typeof(n4);
← "object"
```

⇒ Peut-être que la commande new Type() utilisée pour créer la variable n4 permet d'initialiser des variables complexes.

3. Déclarer et initialiser deux variables correspondant à deux nombres différents.

Assigner une nouvelle valeur à l'une des variables en la multipliant par la seconde.

```
>> var a = 45;  
← undefined  
>> var b = 79;  
← undefined  
>> b = a * b;  
← 3555
```

4. Admettons qu'on veuille calculer le périmètre d'un rectangle et qu'on pose:  
let longueur = 30;  
let largeur = 10;  
let perimetre = longueur + largeur \* 2;

Obtient-on le résultat attendu ? Justifiez et corrigez si nécessaire.

```
>> let longueur = 30;  
    let largeur = 10;  
    let perimetre = longueur + largeur * 2;  
  
>> perimetre = longueur + largeur * 2;  
← 50
```

⇒ Non, on obtient un résultat faux. Pour obtenir le bon périmètre, il faut ajouter des parenthèses afin de calculer d'abord la somme de la longueur et de la largeur, puis de la multiplier par deux :

```
>> perimetre = (longueur + largeur) * 2;  
← 80
```