

PROGRAMMATION

DEVOIRS

Exercice 1 :

159 Issues: 🚩 1 🏆 158

```
> let chaine1 = "Bonjour";  
⏪ undefined  
> chaine1[0] = 'K'  
⏪ 'K'  
> chaine1  
⏪ 'Bonjour'  
>
```

On peut constater que le changement de caractère n'a pas eu lieu. Il n'est pas possible de changer les caractères d'une chaîne avec ce code tout simplement à cause de la fixité des caractères en JavaScript.

Exercice 2 :

```
> 10 == '10'  
⏪ true  
> 10 === '10'  
⏪ false  
>
```

On peut remarquer qu'avec l'opérateur « == » on obtient une réponse « true » et que pour l'opérateur « === » on obtient une réponse « false ».

Les deux réponses sont différentes car les deux opérateurs n'ont pas la même fonction. En effet, l'opérateur « == » a pour but de convertir les deux valeurs à une valeur commune puis de les comparer entre elle.

Contrairement à l'opérateur « === » qui lui n'effectue aucune conversion des valeurs. Il ne fait que comparer les valeurs ainsi que les types de valeurs.

Exercice 3 :

```
> let texte = "C'était le meilleur des temps, c'était le pire des temps ; \
c'était l'âge de la sagesse, c'était l'âge de la folie ; c'était \
l'époque de la foi, c'était l'époque de l'incrédulité ; c'était la \
saison de la Lumière ; c'était la saison de l'Obscurité ; c'était \
le printemps de l'espoir, c'était l'heure du désespoir ; nous \
avons tout devant nous, nous n'avons rien devant nous ; \
nous devons tous aller directement au Ciel, nous devons \
tous prendre l'autre chemin ; bref, l'époque était tellement \
différente de celle que vous vivons aujourd'hui que quelques- \
unes des plus tapageuses autorités ne parlaient d'elles, que \
ce fut en bien ou en mal, qu'au superlatif.";

< undefined

> let longueur = texte.length;

< undefined

> texte.length

< 662

> let mots = texte.match(/\b\w{6,}\b/g);

< undefined

> mots

< (25) ['meilleur', 'sagesse', 'saison', 'saison', 'Obscurit', 'printemps', 'espoir',
'sespoir', 'avons', 'devant', 'avons', 'devant', 'devions', 'directement', 'devion
s', 'prendre', 'chemin', 'tellement', 'vivons', 'aujourd', 'quelques', 'tapageuses',
'autorit', 'parlaient', 'superlatif']

>
```

- Pour cette question, il a fallu déclarer et initialiser cette phrase en respectant les retours à la ligne. Pour cela il faut ajouter « \ » à la fin de chaque retour à la ligne.
- La longueur de cette phrase est de 662. On l'a obtenu en définissant une variable let que l'on a nommé « longueur » avec comme contenu « texte.length » ce qui nous donne la longueur de la phrase.
- Pour compter le nombre de mots qui ont plus de 5 lettres et qui ne sont pas suivis d'espace, il faut utiliser une regex. En l'occurrence dans notre cas c'était la regex : `/\b\w{6,}\b/g`
Il a fallu d'abord déclarer et initialiser la variable let que l'on nomme « mots », ce qui nous donne ensuite le résultats de la capture d'écran ci-dessus. Il y a 25 mots.

```

> let texte = texte.replace(/\b[Cc]'était\b/g, '');
< undefined
> texte
< " le meilleur des temps, le pire des temps ; l'âge de la sagesse, l'âge de la foli
ie ; l'époque de la foi, l'époque de l'incrédulité ; la saison de la Lumière ; l
a saison de l'Obscurité ; le printemps de l'espoir, l'heure du désespoir ; nous av
ions tout devant nous, nous n'avions rien devant nous ; nous devons tous aller dire
ctement au Ciel, nous devons tous prendre l'autre chemin ; bref, l'époque était tel
lement différente de celle que vous vivons aujourd'hui que quelques- unes des plus t
apageuses autorités ne parlaient d'elles, que ce fut en bien ou en mal, qu'au superl
atif."

```

>

- d) Pour retirer les [Cc]'était de la phrase il nous a fallu utiliser « texte.replace » suivi de la regex :
 \b[Cc]'était\b/g, ''

```

> let texte = texte.replace(/^\s+/, '');
< undefined
> texte
< "le meilleur des temps, le pire des temps ; l'âge de la sagesse, l'âge de la foli
e ; l'époque de la foi, l'époque de l'incrédulité ; la saison de la Lumière ; la
saison de l'Obscurité ; le printemps de l'espoir, l'heure du désespoir ; nous avio
ns tout devant nous, nous n'avions rien devant nous ; nous devons tous aller direct
ement au Ciel, nous devons tous prendre l'autre chemin ; bref, l'époque était telle
ment différente de celle que vous vivons aujourd'hui que quelques- unes des plus tap
ageuses autorités ne parlaient d'elles, que ce fut en bien ou en mal, qu'au superlat
if."
> let texte = texte.replace(/^\w/, (match) => match.toUpperCase());
< undefined
> texte
< "Le meilleur des temps, le pire des temps ; l'âge de la sagesse, l'âge de la foli
e ; l'époque de la foi, l'époque de l'incrédulité ; la saison de la Lumière ; la
saison de l'Obscurité ; le printemps de l'espoir, l'heure du désespoir ; nous avio
ns tout devant nous, nous n'avions rien devant nous ; nous devons tous aller direct
ement au Ciel, nous devons tous prendre l'autre chemin ; bref, l'époque était telle
ment différente de celle que vous vivons aujourd'hui que quelques- unes des plus tap
ageuses autorités ne parlaient d'elles, que ce fut en bien ou en mal, qu'au superlat
if."
> |

```

- e) Pour retirer l'espace au début de la phrase il faut utiliser texte.replace suivi de la regex :
 /^\s+/, ''
- f) Pour mettre la majuscule au début de la phrase il faut utiliser texte.replace suivi de la regex :
 /^\w/ suivi de de la fonction match.toUpperCase

Exercice 4 :

```
> let data = "Lepautre, Jean/MALE/Paris/17051999"
< undefined
> let slash = data.indexOf('/', data.indexOf('/') + 1);
< undefined
> slash
< 19
> |
```

Le deuxième « / » se trouve en 19^{ème} position dans la chaîne de caractère.

```
> let dateNaissance = data.substr(-8);
< undefined
> let jour = dateNaissance.substr(0, 2);
< undefined
> let mois = dateNaissance.substr(2, 2);
< undefined
> let annee = dateNaissance.substr(4);
< undefined
> let dateconcatene = `${annee}-${mois}-${jour}`;
< undefined
> dateconcatene
< '1999-05-17'
> |
```

Pour cette partie, il a fallu extraire chaque élément de la date de naissance pour les placer dans de nouvelles variables. Puis il a fallu concaténer les variables pour former une nouvelle date de naissance au format AAAA/MM/JJ.