

**Devoir 4:**

1. Initialisez une chaîne de caractère chaine1.


Essayez de modifier l'un des caractères en utilisant son index.

Que se passe-t-il? Pourquoi ?

```
>> let chaine1 = "une chaîne de caractères";  
← undefined  
  
>> chaine1[0] = "U";  
← "U"  
  
>> chaine1;  
← "une chaîne de caractères"
```

Si on pose directement sur la console, rien ne se passe.

```
>> let chaine1 = "une chaîne de caractères";  
    chaine1[0] = "U";  
    console.log("chaine1");
```

 **Uncaught SyntaxError: redeclaration of let chaine1** debugger eval code:1:1  
 <anonymous> debugger eval code:1  
 [\[En savoir plus\]](#)

Si on écrit tout et qu'on compile, un message d'erreur s'affiche.

Dans les deux cas, la cause est la même : les chaînes de caractères sont non mutables donc il est impossible de modifier uniquement l'un des caractères. Par contre, on peut assigner une nouvelle valeur à la chaîne entière.

```
>> let chaine1 = "une chaîne de caractères";
```

```
← undefined
```

---

```
>> chaine1 = "Ceci est " + chaine1;
```

```
← "Ceci est une chaîne de caractères"
```

---

```
>> chaine1;
```

```
← "Ceci est une chaîne de caractères"
```

2. Comparez les résultats de :

10 == '10' et 10 === '10'.

```
>> 10 == '10';
```

```
← true
```

---

```
>> 10 === '10';
```

```
← false
```

Sont-ils similaires ? Pourquoi ?

On obtient des résultats différents à cause de l'opérateur de comparaison.

"===" implique une égalité stricte qui concerne aussi bien la valeur des variables comparées que leur type (ici, 10 et '10' ont le même contenu mais le premier est nombre alors que le second est une chaîne de caractère)

"==", par contre, ne compare que la valeur des variables en les convertissant au préalable pour qu'elles appartiennent au même type (ici 10 sera transformé en '10' ou '10' sera transformé en 10).

3. Soit la phrase:

"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'avions 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."

Déclare-la et initialisez-la en respectant les retours à la ligne.

Quelle est la longueur de cette phrase ?

Trouvez tous les mots de plus de 5 lettres non suivis d'un espace. Combien y en a-t-il ?

Supprimez tous les "[Cc]'était" du texte. Enlevez l'espace supplémentaire en début de phrase et mettez la première lettre de la phrase en majuscule.

```
//déclaration et initialisation de la phrase avec des accents graves
let phrase=`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.`;

//calcul de la longueur de la phrase
let taille = phrase.length;

//Affichage de la longueur de la phrase
console.log(`La phrase contient ${taille} caractères.`);

//Mots de plus 5 lettres non suivis d'un espace
let words = phrase.match(/p{L}{6,}\b(?:[s\p{L}'])/uig);
// p{L} permet de récupérer n'importe quelle lettre et pas seulement les
caractères latins.

console.log(`La phrase compte ${words.length} mots de plus 5 lettres non suivis
d'un espace : ${words}.`);

// Suppression de tous les "[Cc]'était" et de l'espace supplémentaire
let phrase2 = phrase.replace(/c'était/ig,"").substring(1);

// Première lettre de la phrase en majuscule
let phrase3 = phrase2[0].toUpperCase() + phrase2.substring(1);

console.log(phrase3);
```

---

La phrase contient 672 caractères.

---

La phrase compte 4 mots de plus 5 lettres non suivis d'un espace : sagesse, espoir, quelques, superlatif.

---

Le meilleur des temps, le pire des temps ;  
l'âge de la sagesse, l'âge de la folie ;  
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  
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.

---

Autre possibilité pour la déclaration et l'initialisation

4. let data = "Lepautre, Jean/MALE/Paris/17051999" ;

Trouvez la position du deuxième "/".

Les 8 derniers chiffres représentent le jour, le mois et l'année de naissance.

Extraire chaque élément (jour, mois, année) et le stocker dans une nouvelle variable.

Concaténez ces 3 variables pour avoir un format AAAA-MM-JJ.

```
let data = "Lepautre, Jean/MALE/Paris/17051999";

// Position du 2e /
let index2 = data.slice(data.indexOf('/')+1).search('/') + data.indexOf('/')+1;

console.log(`L'indice du deuxième "/" est ${index2}.`);

//Extraction des 8 derniers chiffres et stockage dans une nouvelle variable
let jour = data.substr(data.length-8,2);
let mois = data.substr(data.length-6,2);
let an = data.substr(data.length-4);

//concaténation au format AAAA-MM-JJ
let bon_format= an.concat('-',mois, '-' , jour);

console.log(bon_format);
```

---

L'indice du deuxième "/" est 19.

---

1999-05-17

---