

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 = "Ceci n'est pas une chaîne de caractères."  
← undefined  
  
>> let chaine2 = chaine1.replace(chaine1[3], "a")  
← undefined  
  
>> console.log(chaine2)  
Ceca n'est pas une chaîne de caractères.  
← undefined
```

⇒ Le quatrième caractère de la `chaine1` (donc le caractère d'indice 3, 'i'), est remplacé par le caractère « a » grâce à la fonction `string.replace`. L'expression `chaine1[3]`, qui cible le caractère auquel est associé l'index entre crochets, peut en effet servir d'argument à cette fonction.

2. Comparez les résultats de :
`10 == '10'` et `10 === '10'`.

Sont-ils similaires ? Pourquoi ?

```
>> 10 == '10'  
← true  
  
>> 10 === '10'  
← false
```

⇒ Non, chaque expression renvoie une valeur différente car les deux opérateurs ne vérifient pas la même chose. L'opérateur d'égalité `'=='` permet de vérifier l'identité d'objets de types différents en les convertissant (donc ici, un nombre et une chaîne de caractères), tandis que l'opérateur d'égalité stricte `'==='` considère que deux objets de types différents seront nécessairement différents.

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'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."

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

Quelle est la longueur de cette phrase ?

```
>> let phrase = "C'était le meilleur des temps, c'était le pire des temps ; \nc'était l'âge de la sagesse, c'était l'âge de la folie ; c'était \nl'époque de la foi, c'était l'époque de l'incrédulité ; c'était la \nsaison de la Lumière ; c'était la saison de l'Obscurité ; c'était \nle printemps de l'espoir, c'était l'heure du désespoir ; nous \navions tout devant nous, nous n'avions rien devant nous ; \nnous devons tous aller directement au Ciel, nous devons \ntous prendre l'autre chemin ; bref, l'époque était tellement \ndifférente de celle que vous vivons aujourd'hui que quelques- \nnunes des plus tapageuses autorités ne parlaient d'elles, que \nce fut en bien ou en mal, qu'au superlatif."
< undefined
>> console.log(phrase.length)
672
< undefined
```

⇒ Cette phrase compte 672 caractères.

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

```
>> let phrase = "C'était le meilleur des temps, c'était le pire des temps ; \nc'était l'âge de la sagesse, c'était l'âge de la folie ; c'était \nl'époque de la foi, c'était l'époque de l'incrédulité ; c'était la \nsaison de la Lumière ; c'était la saison de l'Obscurité ; c'était \nle printemps de l'espoir, c'était l'heure du désespoir ; nous \navions tout devant nous, nous n'avions rien devant nous ; \nnous devons tous aller directement au Ciel, nous devons \ntous prendre l'autre chemin ; bref, l'époque était tellement \ndifférente de celle que vous vivons aujourd'hui que quelques- \nnunes des plus tapageuses autorités ne parlaient d'elles, que \nce fut en bien ou en mal, qu'au superlatif."
< undefined
>> let mots = phrase.match(/[wéàùçâëïô]{5,}\S/gm)
< undefined
>> console.log(mots)
Array(33) [ "meilleur", "temps,", "sagesse,", "époque", "époque", "incrédulité", "saison", "Lumière", "saison", "Obscurité", ... ]
< undefined
>> console.log(mots.length)
33
< undefined
```

⇒ J'ai essayé plusieurs regex (avec \S ou avec [^], en copiant-collant le type d'espace contenu dans le texte au cas où elle serait insécable), mais l'array que j'obtiens contient systématiquement des mots suivis d'une espace malgré tout...

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.

```
>> let phrase2 = phrase.replace(/[[Cc]'était/gm, "")
< undefined
>> console.log(phrase2)

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'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.
< undefined
```

```
>> phrase3 = phrase2.trim()
< "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'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."
```

```

>> let phraseZ = phrase3.replace(phrase3[0], phrase3[0].toUpperCase())
← undefined

>> console.log(phraseZ)
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.
← undefined

```

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

Trouvez la position du deuxième "/".

```

>> let data = "Lepautre, Jean/MALE/Paris/17051999"
← undefined

>> data.search("/Paris");
← 19

```

⇒ Le deuxième slash a pour index 19.

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.

```

>> data.search("1999")
← 30

>> let annee = data.substring(30)
← undefined

>> let mois = data.substring(28,30)
← undefined

>> let jour = data.substring(26,28)
← undefined

>> let ddn = annee.concat("/", mois.concat("/", jour))
← undefined

>> console.log(ddn)
1999/05/17

```