

# Les chaînes de caractères

*Attribution - Partage dans les Mêmes Conditions :*  
<http://creativecommons.org/licenses/by-sa/3.0/fr/>

# Table des matières

<b>Introduction</b>	<b>3</b>
<b>I - Introduction aux chaînes de caractères</b>	<b>4</b>
<b>II - Exercice : Appliquer la notion</b>	<b>6</b>
<b>III - Opérations sur les chaînes</b>	<b>7</b>
<b>IV - Exercice : Appliquer la notion</b>	<b>10</b>
<b>V - Concaténation</b>	<b>11</b>
<b>VI - Exercice : Appliquer la notion</b>	<b>14</b>
<b>VII - Les caractères spéciaux</b>	<b>15</b>
<b>VIII - Exercice : Appliquer la notion</b>	<b>18</b>
<b>IX - Essentiel</b>	<b>19</b>
<b>X - Quiz</b>	<b>20</b>
<b>XI - Exercice : Défi final</b>	<b>24</b>
<b>Conclusion</b>	<b>26</b>
<b>Solutions des exercices</b>	<b>27</b>

## Introduction

Manipuler des nombres c'est bien, mais manipuler du texte c'est encore plus intéressant. En effet, beaucoup de programmes que l'on utilise vont avoir une interface avec l'utilisateur, et cette interface comporte quasi-systématiquement du texte. Que ce soit pour afficher le texte d'un bouton ou pour faire un traitement statistiques sur les mots d'un livre, la chaîne de caractères est un type de variable important. Comme pour les nombres il est possible de réaliser différentes opérations, que nous allons aborder, sur ce type de variables.

# I Introduction aux chaînes de caractères

## Objectif

- Savoir comment définir et utiliser une chaîne de caractères.

## Mise en situation

Comme son nom l'indique, une chaîne de caractères est une simple liste de un ou plusieurs caractères. En mémoire, les caractères sont stockés les uns à la suite des autres, en utilisant un encodage pour les convertir en nombres. Tout cela est géré par le langage de programmation, et au niveau du code source il suffit simplement d'initialiser une variable en lui assignant le texte que l'on veut y mettre en valeur. Ce texte devra uniquement être entouré de guillemets doubles ou simples, pour signifier que c'est une variable de type chaîne de caractères.

### Chaîne de caractères

Az Définition

Une chaîne de caractères est une liste ordonnée de zéro, un ou plusieurs caractères. Elle s'écrit en général dans les langages de programmation entre guillemets doubles ou simples : " ou ' .

Exemple

```
1 """Python."""
2 str = 'Bonjour'
3 print(str)

1 /** JavaScript */
2 const str = 'Bonjour'
3 console.log(str)
```

Les deux programmes affichent Bonjour.

### Choix des guillemets

Méthode

- En JavaScript, il est possible d'utiliser les guillemets doubles ou simples, mais il est d'usage de placer les chaînes de caractères entre guillemets simples lorsque que les chaînes ne possèdent pas de caractère ' .
- En Python, il est également possible d'utiliser les guillemets doubles ou simples. Il n'y a pas de préférence officielle.

Dans tous les cas on agit avec cohérence pour l'ensemble du programme :

- que des guillemets doubles,
- que des guillemets simples,
- que des guillemets simples sauf quand il y a un caractère guillemet simple,
- que des guillemets doubles sauf quand il y a un caractère guillemet double,

- etc.

### Ressemblances entre chaînes et tableaux de caractères

⊕ Complément

Un chaîne de caractères peut être vue comme un tableau de caractères de dimension 1.

Certaines fonctionnalités sont identiques entre les chaînes et les tableaux de caractères, par exemple `.length` renvoie dans les deux cas le nombre d'éléments, ou de caractères.

## À retenir

Les chaînes de caractères et les tableaux de caractères sont déclarés et initialisés différemment, ils peuvent tous deux utiliser certaines fonctions mais ne sont pas parfaitement équivalents.

## II Exercice : Appliquer la notion

Pour les questions suivantes, utiliser la déclaration de variables ci-dessous comme base.

```
1 /** JavaScript */  
2 const title = 'Harry Potter'
```

### Question 1

[solution n°1 p. 27]

Afficher la lettre y à partir de la variable `title`.

#### Indice :

La syntaxe `title[indice]` permet de récupérer un caractère particulier. Les indices commencent à 0, et y est le cinquième caractère.

### Question 2

[solution n°2 p. 27]

Afficher la longueur de `title`.


# III Opérations sur les chaînes

## Objectifs

- Savoir manipuler des chaînes de caractères ;
- Savoir extraire des informations d'une chaîne de caractères.

## Mise en situation

Il existe beaucoup de fonctions de manipulations de chaînes, voici les plus utiles.

 Rappel

```
1 /** JavaScript */
2 const citation = 'Par une soirée extrêmement chaude du début de juillet...'
3 console.log(citation.length)
4 console.log(citation[4])
5 console.log(citation.substring(8, 14))
6
```

```
1 """Python."""
2 citation = 'Par une soirée extrêmement chaude du début de juillet...'
3 print(len(citation))
4 print(citation[4])
5 print(citation[8:14])
6
```

Ces deux programmes retournent :

```
1 56
2 u
3 soirée
```

## Mettre une chaîne en majuscule

 Méthode

Pour mettre une chaîne en majuscule on utilise la méthode `.toUpperCase()` en JavaScript. En Python, on utilise la méthode `.upper()`.

 Exemple

```
1 """Python."""
2 str = 'Bonjour tout le monde'
3 up = str.upper()
4 print(up)

1 /** JavaScript */
2 const str = 'Bonjour tout le monde'
3 const up = str.toUpperCase()
4 console.log(up)
```

Les deux programmes affichent BONJOUR TOUT LE MONDE.

## Vérifier si une sous chaîne est contenue dans une chaîne

[🔗 Méthode](#)

Pour vérifier si une sous chaîne est incluse dans une chaîne on utilise la méthode `.includes(subString)`.

En Python on utilise l'opérateur `in`.

[👁 Exemple](#)

```

1 """Python."""
2 str = 'Bonjour tout le monde'
3 include = 'Bonjour' in str
4 print(include)

1 /** JavaScript */
2 const str = 'Bonjour tout le monde'
3 const include = str.includes('Bonjour')
4 console.log(include)

```

Les deux programmes renvoient la valeur booléenne `true`.

## Tester le début d'une chaîne

[🔗 Méthode](#)

Pour tester si une sous-chaîne se trouve au début d'une chaîne on utilise la méthode `.startsWith(subString)` en JavaScript.

En Python, on utilise la méthode `startswith(sub_string)`.

[👁 Exemple](#)

```

1 """Python."""
2 str = 'Bonjour tout le monde'
3 start = str.startswith('tout')
4 print(start)

1 /** JavaScript */
2 const str = 'Bonjour tout le monde'
3 const start = str.startsWith('tout')
4 console.log(start)

```

Les deux programmes renvoient la valeur booléenne `false`.

## Parcourir une chaîne

[⊕ Complément](#)

En JavaScript, la structure (`for...of`) permet de parcourir tous les éléments d'une liste, comme un tableau ou une chaîne de caractères. Il s'agit d'un type particulier de boucle qui commence à l'indice 0 de la chaîne et s'arrête à la fin.

En Python, la structure équivalente est `for in`.

Cette fonction est notamment utilisée lorsqu'on veut appliquer le même traitement à tous les éléments de la chaîne.



```
1 /** JavaScript */
2 const citation = 'Engage le jeu que je le gagne'
3 for (const char of citation) {
4   console.log(char)
5 }

1 """Python."""
2 citation = 'Engage le jeu que je le gagne'
3 for char in citation:
4   print(char)
```

## Documentation

[⊕ Complément](#)

La documentation MDN liste toutes les fonctions disponibles comme, par exemple, *replace* qui permet de faire du remplacement de caractères.

[developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Objets\\_globaux/String](https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Objets_globaux/String)

## À retenir

Il existe de nombreuses méthodes pour manipuler les chaînes de caractères, elles ne s'utilisent pas de la même façon en JavaScript et en Python.

Il est préférable d'utiliser les guillemets simples.

## IV Exercice : Appliquer la notion

Pour les questions suivantes, on utilisera les déclarations ci-dessous qui définissent deux titre de livres adaptés au cinéma :

```
1 /** JavaScript */  
2 const title1 = 'Alice au pays des merveilles'  
3 const title2 = 'Charlie et la chocolaterie'
```

### Question 1

[solution n°3 p. 27]

Vérifier si la chaîne `title1` contient la sous chaîne `des` à l'aide d'une fonction.

### Question 2

[solution n°4 p. 27]

Tester si la chaîne `title2` commence par la lettre `charl` à l'aide d'une fonction.

### Question 3

[solution n°5 p. 27]

Afficher les chaîne `title1` et `title2` de façon à ce qu'elles ne contiennent que des majuscules à l'aide d'une fonction.

# V Concaténation

## Objectif

- Savoir à quoi correspond la concaténation et comme l'utiliser.

## Mise en situation

Une opération de manipulation de texte que nous n'avons pas abordée et qui semble pourtant naturelle est la concaténation. C'est l'opération qui consiste à raccorder entre elles 2 chaînes de caractères pour n'en former qu'une seule. Imaginez que votre programme demande le nom de votre utilisateur, le stocke dans une variable, puis veuille afficher un message de bienvenue. La chaîne de caractères que vous allez afficher sera la concaténation de votre message générique et du nom de l'utilisateur.

### La concaténation

Az Définition

La concaténation est une opération de « collage » de chaînes de caractères. Il s'agit tout simplement d'ajouter une chaîne à la suite d'une autre.

La concaténation s'effectue grâce à l'opérateur `+`. Elle place les chaînes les unes à la suite des autres, sans les séparer avec des espaces : c'est au programmeur de s'en occuper.

On peut concaténer un nombre quelconque de chaînes.

Exemple

```
1 /** JavaScript */
2 const str1 = 'miel'
3 const str2 = 'de'
4 const str3 = 'printemps'
5 const label = str1 + str2 + str3
6 console.log(label)
```

```
1 """Python."""
2 str1 = 'miel'
3 str2 = 'de'
4 str3 = 'printemps'
5 label = str1 + str2 + str3
6 print(label)
```

Les deux programmes affichent `mieldeprintemps`.

Syntaxe

Il est possible d'afficher directement la concaténation de chaînes avec l'instruction d'impression sur la sortie standard.

- en JavaScript : `console.log(str1, str2, ...)`
- en python : `print(str1, str2, ...)`

Les chaînes sont dans ce cas **automatiquement espacées** les unes des autres.

 Exemple

```
1 """Python."""
2 str1 = 'pirates'
3 str2 = 'des'
4 str3 = 'Caraïbes'
5 print(str1, str2, str3)

1 /** JavaScript */
2 const str1 = 'pirates'
3 const str2 = 'des'
4 const str3 = 'Caraïbes'
5 console.log (str1, str2, str3)
```

Les deux programmes affichent pirates des Caraïbes.

## Fonction de concaténation

 Méthode

Il existe plusieurs façon de concaténer des chaînes. On peut utiliser l'opérateur `+` mais cela peut créer de la confusion avec les additions entre nombres.

- En JavaScript on utilise la méthode `.concat()` qui fait la concaténation de chaînes de caractères : la première chaîne est celle qui appelle la méthode `.concat()` et les autres sont placées entre les parenthèses.

```
str1.concat(' ', str2, ...)
```

- En Python, il est préférable d'utiliser la méthode `' '.join([])` qui concatène plusieurs chaînes en les séparant avec le caractère placé entre guillemets simples.

```
' '.join([str1, str2, ...])
```

 Exemple

```
1 """Python."""
2 str1 = 'pirates'
3 str2 = 'des'
4 str3 = 'Caraïbes'
5 title = ' '.join([str1, str2, str3])
6 print(title)

1 /** JavaScript */
2 const str1 = 'pirates'
3 const str2 = 'des'
4 const str3 = 'Caraïbes'
5 const title = str1.concat(' ', str2, ' ', str3)
6 console.log (title)
```

Les deux programmes affichent pirates des Caraïbes.

## Littéraux de gabarits en JavaScript

[⊕ Complément](#)

Les littéraux de gabarits permettant d'intégrer des chaînes de caractères multi-lignes ou d'insérer des valeurs de variables directement à l'intérieur de chaînes de caractères.

[developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Litt%C3%A9raux\\_gabarits](https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Litt%C3%A9raux_gabarits)<sup>1</sup>

## À retenir

La concaténation consiste à « coller » des chaînes les unes à la suite des autres. Il est possible de concaténer des chaînes avec l'opérateur `+` ou d'utiliser les méthodes dédiées à la concaténation. Le choix est laissé au développeur, selon le contexte et la clarté du code.

---

<sup>1</sup>. [https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Litt%C3%A9raux\\_gabarits](https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Litt%C3%A9raux_gabarits)

# VI Exercice : Appliquer la notion

Pour répondre aux questions suivantes, utiliser les définitions ci dessous :

```
1 /** JavaScript */  
2 const name = 'X-men'  
3 const movie1 = 'Le commencement'  
4 const movie2 = 'Apocalypse'  
5 const movie3 = 'Dark phoenix'
```

## Question 1

[solution n°6 p. 28]

Concaténer les deux premières chaînes, en les espaçant, avec la méthode de votre choix. Le programme doit afficher « *X-men Le commencement* ».

## Question 2

[solution n°7 p. 28]

Concaténer les chaînes 1 et 3 en les séparant avec le symbole : en utilisant la méthode de votre choix. Le programme doit afficher « *X-men:Apocalypse* ».

## Question 3

[solution n°8 p. 28]

Concaténer toutes les chaînes de façon à afficher : *X-men : Le commencement, Apocalypse, Dark phoenix.*

### Indice :

Attention à la différence de séparateurs entre les différences chaînes.

# VII Les caractères spéciaux

## Objectif

- Comprendre la spécificité des caractères spéciaux et leur utilisation.

## Mise en situation

Comment afficher la chaîne de caractères `c'est l'été` ?

L'affichage de chaîne est une opération simple qui peut se complexifier dès que l'on souhaite afficher des caractères particuliers comme les apostrophes, les guillemets, les tabulations ou les antislash.

### Les caractères spéciaux

Az Définition

Certains caractères ont une signification particulière, lorsque le compilateur en rencontre un, il effectue un traitement particulier, c'est le cas des guillemets simples ou doubles qui sont considérés comme le début et la fin d'une chaîne de caractères.

### Erreur : afficher une chaîne contenant des apostrophes

Exemple

```
1 """Python."""
2 str1 = 'c'est l'été'
3 print(str1)

1 /** JavaScript */
2 const str1 = 'c'est l'été'
3 console.log(str1)
```

Ces deux programmes ne fonctionnent pas parce que l'interpréteur considère qu'il y a deux chaînes et des caractères entre les deux, et donc qu'il y a une erreur de syntaxe.

## Solution

Il existe deux façon de résoudre ce problème.

1. Il est possible de placer la chaîne entre guillemets doubles, les apostrophes ne seront donc pas considérées comme des délimiteurs de chaîne. Cependant, cela ne permet pas de résoudre le problème si on veut une chaîne contenant à la fois des guillemets simples et des guillemets doubles.
2. L'autre méthode est donc de signaler qu'on veut insérer un caractère spécial en plaçant un **antislash** avant.

👁 Exemple

```

1 """Python."""
2 message = 'c'est l'été'
3 print(message)
4 message = "c'est l'été"
5 print(message)
6
1 /** JavaScript */
2 let message = 'c'est l'été'
3 console.log(message)
4 message = "c'est l'été"
5 console.log(message)

```

Les deux programmes affichent c ' est l ' été.

### Afficher le caractère antislash

⚠ Attention

Pour afficher le caractère antislash il faut indiquer au compilateur qu'il ne s'agit pas d'un caractère spécial. Il faut donc le précéder d'un antislash : `\\`.

### Les caractères spéciaux de mise en page

📖 Syntaxe

Pour demander au compilateur d'aller à la ligne on utilise `\n`, pour demander à faire une tabulation on utilise `\t`.

Il existe beaucoup d'autres caractères spéciaux de ce types, mais ceux-là sont les plus utilisés.

### Encodage des caractères

📅 Rappel

Étant donné que les ordinateurs ne peuvent comprendre que le langage binaire chaque caractère doit être associé à un nombre binaire particulier. Au début de l'informatique des américains ont créé le code ASCII (*American Standard Code for Information Interchange*) qui permet d'écrire n'importe quel texte en anglais.

Cependant, cet encodage ne permet pas d'écrire certains caractères d'autres langues (par exemple les caractères accentués français). Plusieurs autres encodages ont donc été créés et aujourd'hui l'encodage UTF-8, qui permet d'écrire presque la totalité des caractères utilisés dans le monde, est le plus répandu.

### Tous les caractères spéciaux en JavaScript

⊕ Complément



- `\0` Octet null
- `\b` Retour arrière
- `\f` Saut de page
- `\n` Nouvelle ligne
- `\r` Retour chariot



- \t Tabulation
- \v Tabulation verticale
- \' Apostrophe ou guillemet droit simple
- \" Guillemet droit double
- \\ Barre oblique inversée



[developer.mozilla.org/fr/docs/Web/JavaScript/Guide/Types\\_et\\_grammaire](https://developer.mozilla.org/fr/docs/Web/JavaScript/Guide/Types_et_grammaire)

## À retenir

Certains caractères sont considérés comme spéciaux et associés à une fonction particulière. Pour les utiliser dans un autre contexte, par exemple pour les afficher il faut indiquer au compilateur que le caractère en question n'est pas spécial, il faut donc placer un antislash avant.

## VIII Exercice : Appliquer la notion

En affichant certains caractères à un emplacement précis il est possible de réaliser de magnifiques images.

Dans l'art ASCII, on utilise seulement les caractères de la table ASCII ce qui permet de générer des images en noir et blanc.

Art ASCII<sup>2</sup>

### Question

[solution n°9 p. 28]

Écrire un programme qui affiche le smiley japonais *content* : ^\_^.

Puis afficher deux rectangles de même largeur l'un en dessous de l'autre, le premier dont les cotés sont dessinés avec des traits, le second avec des croisillons (#).

Chaque rectangle doit être affiché avec une seule instruction `console.log`.

### Indice :

On utilisera le caractère spécial `\n` pour retourner à la ligne.

---

<sup>2</sup> [https://fr.wikipedia.org/wiki/Art\\_ASCII](https://fr.wikipedia.org/wiki/Art_ASCII)

# IX Essentiel



C |

 D \*

### Exercice

Quelle solution permet de concaténer des chaînes de caractères en JavaScript ?

 A Utiliser l'opérateur +

 B Utiliser la méthode `.concat()`
 C Utiliser la méthode `.join()`

### Exercice

Qu'affiche le programme suivant ?

```
1 const nbPants = 3
2 const nbSkirts = 5
3 console.log('Nombre de vêtements: ' + nbPants + nbSkirts)
4 console.log('Nombre de vêtements: ' + (nbPants + nbSkirts))
```

 A Nombre de vêtements: 35 Nombre de vêtements: 8

 B Nombre de vêtements: 35 Nombre de vêtements: 35

 C Nombre de vêtements: 8 Nombre de vêtements: 8

 D Nombre de vêtements: 8 Nombre de vêtements: 35

### Exercice

Quelle instruction affiche la citation *"Rien ne sert de courir, il faut partir à point"* (entre doubles apostrophes) suivie du nom de son auteur (*Jean de La Fontaine*), séparés par une virgule ?

 A `console.log("Rien ne sert de courir, il faut partir à point", Jean de la Fontaine")`
 B `console.log('Rien ne sert de courir, il faut partir à point' , Jean de la Fontaine')`
 C `console.log("'Rien ne sert de courir, il faut partir à point', Jean de la Fontaine")`

**D**
 console.log('"Rien ne sert de courir, il faut partir à point",  
Jean de la Fontaine')
**E**
 console.log("\"Rien ne sert de courir, il faut partir à point\",  
Jean de la Fontaine")

## Exercice

Qu'affiche l'instruction suivante ?

```
1 /** JavaScript */
2 console.log('c\'est \n\'importe quoi')
```

**A**

« c'est n'importe quoi »

**B**

« c'est 'n'importe quoi »

**C**

« c'est 'importe quoi »

**D**

« c'est nimporte quoi »

## Exercice 11 : Quiz - Code

[solution n°12 p. 32]

### Exercice

```
1 /** JavaScript */
2 const music1 = 'Alors, alors'
3 const music2 = 'Poker face'
4 const music3 = 'Gangnam Style'
5
```

A partir des déclarations ci-dessus quelle instruction affiche le mot « *Poker* » ?

**A**

console.log(music2.substring(0,5))

**B**

console.log(music2.substring(1,6))

### Exercice

```
1 /** JavaScript */
2 const music1 = 'Alors, alors'
3 const music2 = 'Poker face'
4 const music3 = 'Gangnam Style'
5
```

A partir des déclarations ci-dessus, quelle instruction permet d'afficher music3 en majuscule ?

**A**

console.log(music3.touppercase())

B console.log(music3.toUpperCase())

C console.log(music3.upperCase())

D console.log(music3.uppercase())

### Exercice

```
1 /** JavaScript */
2 const music1 = 'Alors, alors'
3 const music2 = 'Poker face'
4 const music3 = 'Gangnam Style'
5
```

Quelle instruction permet d'afficher les trois chaînes, espacées et séparées par le symbole & ?

A console.log(music1, music2, music3.concat(' & '))

B console.log(music1, music2, music3.join(' & '))

C console.log(music1.concat(' & ', music2, ' & ', music3))

### Exercice

A partir des déclarations ci-dessous, quelles instructions affichent la longueur de la chaîne book2 ?

```
1 /** JavaScript */
2 const book1 = '1984'
3 const book2 = 'Le meilleur des mondes'
```

A `const lengthBook = console.log('Longueur chaine : ' + book2.length lengthBook)`

B `const lengthBook = console.log('Longueur chaine : ' + book2.length()) lengthBook)`

C `const lengthBook = console.log('Longueur chaine : ' book2.length lengthBook)`

D `const lengthBook = console.log('Longueur chaine : ' book2.length()) lengthBook)`

# XI Exercice : Défi final

La librairie « Au fil des pages » souhaite revoir l'affichage de ses livres.

Écrire un programme qui améliore l'affichage des livres de la librairie, en utilisant les déclarations fournies :

```
1 // Meilleures ventes
2 const bestS1 = 'Le petit Prince'
3 const bestS2 = 'Le mariage de Figaro'
4
5 // Livres de la série Narnia
6 const serie1 = 'Le neveu du magicien'
7 const serie2 = 'Le prince Caspian'
8
9 // Titres sans apostrophe
10 let error1 = 'Le songe d'une nuit d'été'
11 let error2 = 'Le lion, la sorcière blanche et l'armoire magique'
12
13 // Titres les uns à la suite des autres
14 const title = 'PhèdreAndromaqueBritannicus'
```

## Question 1

[solution n°13 p. 34]

Pour commencer, il faut mettre en avant les meilleures ventes.

Afficher les titres des best-sellers en majuscules et précédés de la mention « *Meilleures ventes* » . On doit obtenir l'affichage suivant :

```
1 Meilleures ventes : LE PETIT PRINCE
2 Meilleures ventes : LE MARIAGE DE FIGARO
```

### Indice :

Il faut utiliser la fonction `.toUpperCase()`.

## Question 2

[solution n°14 p. 34]

Ensuite, il faut faciliter la recherche des lecteurs.

Indiquer que certains livres appartiennent à une série en affichant le nom de la série avant celui du livre. On doit obtenir l'affichage suivant :

```
1 Narnia : Le neveu du magicien, Le prince Caspian
```

### Indice :

Il faut faire une concaténation avec la fonction `.concat()` par exemple.

## Question 3

[solution n°15 p. 34]

De plus, il faut corriger des erreurs sur certains titres. En effet, un des éditeurs a eu un problème de clavier : sa touche de guillemets simple ne fonctionnait plus.

On doit obtenir l'affichage suivant :

```
1 Le songe d'une nuit d'été
2 Le lion la sorcière blanche et l'armoire magique
```



**Indice :**

Les guillemets sont des caractères spéciaux, il faut les précéder d'un antislash.

**Question 4**

[solution n°16 p. 34]

Enfin, un autre éditeur a envoyé les titres de ses livres les uns à la suite des autres, collés entre eux. Cela perturbe les lecteurs. Séparer les titres et les afficher les uns à la suite des autres **sans ré-écrire les titres**.

On doit obtenir l'affichage suivant :

```
1 Phèdre
2 Andromaque
3 Britannicus
```

**Indice :**

Il faut extraire des sous-chaînes avec la fonction `.substring(début, fin)`.

## Conclusion

Les chaînes de caractères sont plus complexes que les nombres, mais cela n'empêche pas un grand nombre d'opérations possibles sur celles-ci. Des fonctions de traitement basique du texte sont disponibles dans tous les langages, et des fonctions plus avancées peuvent être trouvées dans des bibliothèques. Une des plus courante est sans doute la concaténation, qui permet de mettre bout à bout 2 chaînes de caractères. De cette manière il est possible d'intégrer du texte dynamique, qui serait stocké dans une variable, au sein d'un message générique, qui sert de modèle.

# Solutions des exercices

## Solution n°1

[exercice p. 6]

```
1 /** JavaScript */
2 const title = 'Harry Potter'
3 console.log(title[4])
```

Les indices commencent à 0, donc l'indice de y est 4.

## Solution n°2

[exercice p. 6]

```
1 /** JavaScript */
2 const title = 'Harry Potter'
3 console.log(title.length)
```

## Solution n°3

[exercice p. 10]

```
1 const title = 'Alice au pays des merveilles'
2 console.log(title.includes('des'))
```

Le programme affiche true.

La fonction `includes()` ne cherche pas des mots, elle cherche des groupes de lettres, qui forment un mot ou pas.

## Solution n°4

[exercice p. 10]

```
1 const title2 = 'Charlie et la chocolaterie'
2 console.log(title2.startsWith('charl'))
```

Le programme affiche false.

La fonction `startsWith()` respecte la casse et différencie les majuscules et les minuscules.

## Solution n°5

[exercice p. 10]

```
1 const title1 = 'Alice au pays des merveilles'
2 const title2 = 'Charlie et la chocolaterie'
3 console.log(title1.toUpperCase())
4 console.log(title2.toUpperCase())
```

## Solution n°6

[exercice p. 14]

```

1 const name = 'X-men'
2 const movie1 = 'Le commencement'
3 const completeName = name.concat(' ', movie1)
4 console.log(completeName)

```

On aurait aussi pu écrire :

```

1 const name = 'X-men'
2 const movie1 = 'Le commencement'
3 const completeName = name + ' ' + movie1
4 console.log(completeName)

```

## Solution n°7

[exercice p. 14]

```

1 const name = 'X-men'
2 const movie2 = 'Apocalypse'
3 const completeName = name.concat(':', movie2)
4 console.log(completeName)

```

On aurait aussi pu écrire :

```

1 const name = 'X-men'
2 const movie2 = 'Apocalypse'
3 const completeName = name + ':' + movie2
4 console.log(completeName)

```

## Solution n°8

[exercice p. 14]

```

1 const name = 'X-men'
2 const movie1 = 'Le commencement'
3 const movie2 = 'Apocalypse'
4 const movie3 = 'Dark phoenix'
5 const list = name.concat(' : ', movie1, ', ', movie2, ', ', movie3, '.')
6 console.log(list)

```

On aurait aussi pu écrire :

```

1 const name = 'X-men'
2 const movie1 = 'Le commencement'
3 const movie2 = 'Apocalypse'
4 const movie3 = 'Dark phoenix'
5 const list = name + ' : ' + movie1 + ', ' + movie2 + ', ' + movie3 + '.'
6 console.log(list)

```

## Solution n°9

[exercice p. 18]

```

1 /** JavaScript */
2 console.log('^_^')
3 console.log('___\n| |\n| |\n___')
4 console.log('###\n# #\n# #\n###')

```

## Solution n°10

### Exercice

Quel est l'encodage (le nombre décimal associé) du caractère 'C' dans la table ASCII ? Et quel est celui du caractère 'c' ?

Table ASCII<sup>4</sup>

**A** 99 et 67

**B** 99 et 99

**C** 67 et 67

**D** 67 et 99

🔍 Les lettres majuscules et minuscules n'ont pas le même encodage, c'est pour cela qu'il est possible de les différencier.

### Exercice

Tous les caractères, lorsqu'ils sont précédés d'un antislash, forment des caractères spéciaux.

**A** Vrai

**B** Faux

🔍 La lettre n n'est pas un caractère spécial mais `\n` déclenche un retour à la ligne dans une chaîne de caractères.  
En revanche, tous les caractères précédés par un antislash ne sont **pas** des caractères spéciaux : `'\9'` n'est pas un caractère spécial.

### Exercice

Une chaîne de caractères peut être assimilée à un tableau de chaînes de caractères de dimension 1.

**A** Vrai

**B** Faux

🔍 C'est vrai, mais beaucoup de méthodes sont spécifiques aux chaînes de caractères et ne s'appliquent pas aux tableaux, comme `toUpperCase()`.

4. [http://www.asciichars.com/\\_site\\_media/ascii/ascii-chars-landscape.jpg](http://www.asciichars.com/_site_media/ascii/ascii-chars-landscape.jpg)

## Solution n°11

### Exercice

Quels symboles peuvent encadrer les chaînes de caractères ?

**A** '

**B** "

**C** |

**D** \*

🔍 On peut utiliser des guillemets simple ou doubles. En JavaScript, on utilise par convention les guillemets simples (*simple quote*), ou apostrophes.

### Exercice

Quelle solution permet de concaténer des chaînes de caractères en JavaScript ?

**A** Utiliser l'opérateur +

**B** Utiliser la méthode `.concat()`

**C** Utiliser la méthode `.join()`

### Exercice

Qu'affiche le programme suivant ?

```
1 const nbPants = 3
2 const nbSkirts = 5
3 console.log('Nombre de vêtements: ' + nbPants + nbSkirts)
4 console.log('Nombre de vêtements: ' + (nbPants + nbSkirts))
```

**A** Nombre de vêtements: 35 Nombre de vêtements: 8

**B** Nombre de vêtements: 35 Nombre de vêtements: 35

**C** Nombre de vêtements: 8 Nombre de vêtements: 8

**D** Nombre de vêtements: 8 Nombre de vêtements: 35



Dans la première instruction d'affichage, les opérations sont effectuées de gauche à droite. L'opérateur `+`, appliqué sur une chaîne et un nombre, effectue une concaténation entre la chaîne et le nombre. Le deuxième `+` s'applique donc une nouvelle fois sur la première chaîne concaténée et le dernier nombre.

Dans la deuxième instruction d'affichage, la première opération effectuée est l'**addition** des deux nombres, à cause des parenthèses. Le premier `+` est donc appliqué à la fin, et concatène la chaîne *Nombre de vêtements* avec le résultat de l'addition.

L'opérateur `+` est donc très pratique pour concaténer des chaînes rapidement mais doit être utilisé avec précaution dans certains cas.

### Exercice

Quelle instruction affiche la citation "*Rien ne sert de courir, il faut partir à point*" (entre doubles apostrophes) suivie du nom de son auteur (*Jean de La Fontaine*), séparés par une virgule ?

**A**

```
console.log("Rien ne sert de courir, il faut partir à point",
Jean de la Fontaine")
```

**B**

```
console.log('Rien ne sert de courir, il faut partir à point' ,
Jean de la Fontaine')
```

**C**

`console.log("Rien ne sert de courir, il faut partir à point",
Jean de la Fontaine")`

**D**

`console.log('Rien ne sert de courir, il faut partir à point",
Jean de la Fontaine')`

**E**

`console.log("\Rien ne sert de courir, il faut partir à point\",
Jean de la Fontaine")`



- Les réponses 1 et 2 qui utilisent un seul type de guillemets vont générer une erreur.
- La réponse 3 fonctionne, mais elle affiche la citation entre simples apostrophes au lieu de doubles.
- Les réponses 4 et 5 sont correctes mais il vaut mieux utiliser la syntaxe de la réponse 4 qui respecte les conventions du JavaScript (c'est à dire placer les chaînes de caractères entre guillemets simples) et évite les caractères d'échappement.

### Exercice

Qu'affiche l'instruction suivante ?

```
1 /** JavaScript */
2 console.log('c\'est \n\'importe quoi')
```

**A** « c'est n'importe quoi »

**B** « c'est 'n'importe quoi »

**C** « c'est 'importe quoi »

**D** « c'est nimporte quoi »

🔍 Les apostrophes sont précédées d'un antislash, donc correctement affichées, le caractère `\n` déclenche un retour à la ligne.

## Solution n°12

[exercice p. 22]

### Exercice

```
1 /** JavaScript */
2 const music1 = 'Alors, alors'
3 const music2 = 'Poker face'
4 const music3 = 'Gangnam Style'
5
```

A partir des déclarations ci-dessus quelle instruction affiche le mot « *Poker* » ?

**A** `console.log(music2.substring(0,5))`

**B** `console.log(music2.substring(1,6))`

🔍 Les indices commencent à 0.

### Exercice

```
1 /** JavaScript */
2 const music1 = 'Alors, alors'
3 const music2 = 'Poker face'
4 const music3 = 'Gangnam Style'
5
```

A partir des déclarations ci-dessus, quelle instruction permet d'afficher `music3` en majuscule ?

**A** `console.log(music3.touppercase())`

**B** `console.log(music3.toUpperCase())`

**C** `console.log(music3.upperCase())`



**D** `console.log(music3.toUpperCase())`

🔍 C'est la méthode `.toUpperCase()` dont le nom est écrit selon la convention de casse *camelCase*.

### Exercice

```
1 /** JavaScript */
2 const music1 = 'Alors, alors'
3 const music2 = 'Poker face'
4 const music3 = 'Gangnam Style'
5
```

Quelle instruction permet d'afficher les trois chaînes, espacées et séparées par le symbole **&** ?

**A** `console.log(music1, music2, music3.concat(' & '))`

**B** `console.log(music1, music2, music3.join(' & '))`

**C** `console.log(music1.concat(' & ', music2, ' & ', music3))`

🔍

- La première instruction ne déclenche pas d'erreur mais ne correspond pas à l'affichage demandé.
- La deuxième instruction déclenche une erreur.

### Exercice

A partir des déclarations ci-dessous, quelles instructions affichent la longueur de la chaîne `book2` ?

```
1 /** JavaScript */
2 const book1 = '1984'
3 const book2 = 'Le meilleur des mondes'
```

**A** `const lengthBook = console.log('Longueur chaine : ' + book2.length)`

**B** `const lengthBook = console.log('Longueur chaine : ' + book2.length())`

**C** `const lengthBook = console.log('Longueur chaine : ' + book2.length)`

**D**

```
const lengthBook = console.log('Longueur chaine : ' +
book2.length() lengthBook)
```



Il ne faut pas mettre de parenthèses après `length`. Il faut concaténer les deux informations pour l'affichage, ici il est plus rapide d'utiliser l'opérateur `+` pour concaténer.

## Solution n°13

[exercice p. 24]

```
1 console.log('Meilleures ventes : ' + bestS1.toUpperCase())
2 console.log('Meilleures ventes : ' + bestS2.toUpperCase())
```

## Solution n°14

[exercice p. 24]

```
1 console.log('Narnia'.concat(' : ', serie1, ', ', serie2))
```

## Solution n°15

[exercice p. 24]

```
1 /** JavaScript */
2 error1 = 'Le songe d\'une nuit d\'été'
3 console.log(error1)
4 error2 = 'Le lion, la sorcière blanche et l\'armoire magique'
5 console.log(error2)
```

## Solution n°16

[exercice p. 25]

```
1 /** JavaScript */
2 const title1 = title.substring(0, 6)
3 console.log(title1)
4 const title2 = title.substring(6, 16)
5 console.log(title2)
6 const title3 = title.substring(16, 27)
7 console.log(title3)
```

