

## TD 1 - Strings

Ferdinand Mom

April 2019

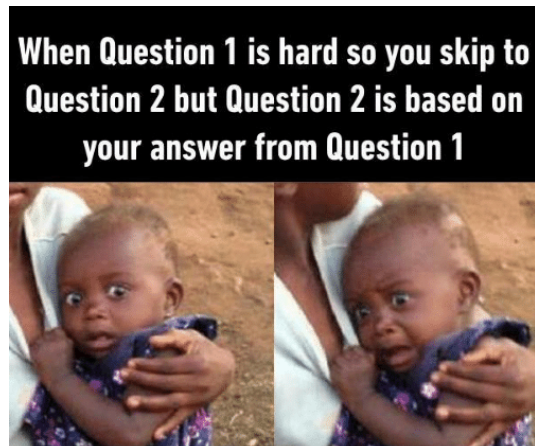


Figure 1: Meme de qualité



# 1 Classiques

## Exercice 1.1: myLen

Créez la fonction **myLen(s)** qui retourne la longueur de la string *s*.

```
res = myLen("Ferd")
print(res)
>>> 5
```

## Exercice 1.2: Frequency

Créez la fonction **frequency(c, s)** qui retourne la fréquence du caractère *c* dans *s*.

```
res = frequency("a", "")
print(res)
>>> ("a", 0)

res = frequency("a", "Saalut")
print(res)
>>> ("a", 2)
```

## Exercice 1.3: isVowel

Créez la fonction **isVowel(c)** qui retourne *True* si *c* est une voyelle. Ce dernier peut être en minuscule tout comme en majuscule.

```
res = isVowel('a')
print(res)
>>> True

res = isVowel('A')
print(res)
>>> True

res = isVowel('z')
print(res)
>>> False
```

## Exercice 1.4: countVowel

Créez la fonction **countVowel(s)** qui retourne le nombre de voyelles dans *s*.

```
res = countVowel("sAlut")
print(res)
>>> 2
```

**Exercice 1.5: noVowel**

Créez la fonction **noVowel(s)** qui retourne une string sans les voyelles de *s*.

```
res = noVowel("AeIouy")
print(res)
>>> ""

res = noVowel("la balle au bond rebondit")
print(res)
>>> "l bll bnd rbndt"
```

**Exercice 1.6: mysteryWord**

Créez la fonction **mysteryWord(s)** qui retourne une string avec les voyelles de *s* remplacées par des "-".

```
res = mysteryWord("AeIouy")
print(res)
>>> "-----"

res = mysteryWord("la balle au bond rebondit")
print(res)
>>> "l- b-ll- -- b-nd r-b-nd-t"
```

**Exercice 1.7: isPalindrom**

Créez la fonction **isPalindrom(s)** qui retourne *True* si *s* est un palindrome.

```
res = isPalindrom("")
print(res)
>>> True

res = isPalindrom("je ne suis pas un palindrome")
print(res)
>>> False

res = isPalindrom("aba")
print(res)
>>> True
```



### Exercice 1.8: deleteChar

Créez la fonction **deleteChar(c, s)** qui retourne une string sans le caractère *c* de la string *s*.

```
res = deleteChar('a', "")
print(res)
>>> ""

res = deleteChar('a', "aaaaaa")
print(res)
>>> ""

res = deleteChar('u', "crouu crouuu")
print(res)
>>> "cro cro"
```

### Exercice 1.9: deleteFirstChar

Créez la fonction **deleteFirstChar(c, s)** qui retourne une string sans la première occurrence de *c* dans *s*.

```
res = deleteFirstChar('a', "")
print(res)
>>> ""

res = deleteFirstChar('a', "aaaaaa")
print(res)
>>> "aaaaa"

res = deleteFirstChar('u', "crouu crouuu")
print(res)
>>> "crou crouuu"
```

### Exercice 1.10: deleteLastChar

Créez la fonction **deleteLastChar(c, s)** qui retourne une string sans la dernière occurrence de *c* dans *s*.

```
res = deleteLastChar('a', "")
print(res)
>>> ""

res = deleteLastChar('a', "aaaaaa")
print(res)
>>> "aaaaa"

res = deleteLastChar('u', "crouu crouuu")
print(res)
>>> "crouu crouu"
```

**Exercice 1.11: subWord**

Créez la fonction **subWord**(sw, s) qui retourne *True* si *sw* est un sous-mot de *s*.

```
res = subWord("al", "salut")
print(res)
>>> True

res = subWord("sal", "glouglou")
print(res)
>>> False
```

**Exercice 1.12: capitalize**

Créez la fonction **capitalize**(s) qui retourne une string en majuscule. Utilisez **ord(c)** pour obtenir la valeur ASCII d'un caractère et **chr(n)** pour transformer une valeur ASCII en caractère. Gérez les possibles exceptions.

**Tips:**

- Pour lever une exception, utilisez le code suivant:

```
try:
    raise Exception("Error")
except Exception as e:
    return "Input cannot be capitalized."
```

```
res = capitalize("")
print(res)
>>> ""

res = capitalize("salut")
print(res)
>>> SALUT

res = capitalize("1234")
print(res)
>>> "Input cannot be capitalized."
```

True knowledge exists in knowing that you know nothing.

- Socrates