

# Chaine de caractères

## Quelques opérations élémentaires

```
In [1]: # Concaténation

str_a = "Ronaldo fait la passe à "
str_b = "Messi"
str_concat = str_a + str_b
print(str_concat)
```

Ronaldo fait la passe à Messi

```
In [2]: # On peut multiplier les chaines de caractères

str_X = "je veux l'X " * 10
print(str_X)
```

je veux l'X je veux l'X je veux l'X je veux l'X je veux l'X je veux l'X je veux l'X je veux l'X j  
e veux l'X je veux l'X

```
In [3]: # Longueur d'une chaine de caractères

print(f'La longueur de str_a est égale à', len(str_a))
print(f'La longueur de str_b est égale à', len(str_b))
print(f'La longueur de str_concat est égale à', len(str_concat))
```

La longueur de str\_a est égale à 24  
La longueur de str\_b est égale à 5  
La longueur de str\_concat est égale à 29

```
In [5]: # Affecter à une variable la valeur d'un élément d'une liste
# La chaîne de caractère ne change pas !
# indice du premier élément : 0 !

prems_a = str_a[0]
prems_b = str_b[0]
print(f'Le premier élément de str_a est', prems_a, "de la chaîne de caractères ", str_a)
print(f'Le premier élément de str_b est', prems_b, "de la chaîne de caractères ", str_b)
```

Le premier élément de str\_a est R de la chaîne de caractères Ronaldo fait la passe à  
Le premier élément de str\_b est M de la chaîne de caractères Messi

```
In [6]: # indice du dernier élément : len(str)-1 ! ou -1 !
dern_a = str_a[len(str_a)-1]
dern_b = str_b[-1]
print(f'Le dernier élément de "{str_a}" est', dern_a)
print(f'Le dernier élément de "{str_b}" est', dern_b)
```

Le dernier élément de "Ronaldo fait la passe à " est  
Le dernier élément de "Messi" est i

```
In [8]: # Récupérer un "morceau" de chaîne de caractères : str[m:n:pas] (n exclu)
print(str_b, ": str_b")

morceau_1 = str_b[1:4]
print(morceau_1, ": morceau_1")

morceau_2 = str_b[0:len(str_b):2]
print(morceau_2, ": morceau_2")

str_reverse = str_b[::-1] # l'inverse
print(str_reverse, ": str_reverse")
```

Messi : str\_b  
ess : morceau\_1  
Msi : morceau\_2  
isseM : str\_reverse

## Ajouter un élément en fin de chaîne

```
In [12]: # on peut utiliser la concaténation
str_aa = str_a
print(str_aa, " : str_aa")

str_aa = str_aa + "team mpx"
print(str_aa, " : str_aa")
print(str_a, " : str_a")

Ronaldo fait la passe à    : str_aa
Ronaldo fait la passe à team mpx  : str_aa
Ronaldo fait la passe à    : str_a
```

### Remplacer un élément dans une chaîne

```
In [16]: # solution : fonction replace(old, new, max)

remp_a = str_a
remp_a_neymar = remp_a.replace("Ronaldo", "Neymar")
remp_a_u = remp_a.replace("a", "u", 2)

print("remp_a :", remp_a)
print("remp_a_neymar :", remp_a_neymar)
print("remp_a_u :", remp_a_u)
print("str_a :", str_a)

remp_a : Ronaldo fait la passe à
remp_a_neymar : Neymar fait la passe à
remp_a_u : Ronuldo fuit la passe à
str_a : Ronaldo fait la passe à
```

### Insérer un élément (ou plusieurs) dans une chaîne

```
In [18]: # solution : on peut utiliser la concaténation

insert_ele = str_a
print(insert_ele, ": insert_ele")

insert_ele = insert_ele[:2] + "nul" + insert_ele[2:]
print(insert_ele, ": insert_ele")

Ronaldo fait la passe à : insert_ele
Ronulnaldo fait la passe à : insert_ele
```

### Modifier la mise en forme du texte

```
In [19]: # plein de possibilités : https://www.w3schools.com/python/python_ref_string.asp
str_a_min = str_a.lower()
str_a_cap = str_a.upper()

print(str_a_min, ": str_a_min")
print(str_a_cap, ": str_a_cap")

ronaldo fait la passe à : str_a_min
RONALDO FAIT LA PASSE À : str_a_cap
```

### Diviser une chaîne de caractères

```
In [20]: # solution : fonction str.split(sep)
str_a_div = str_a
str_a_div_ = str_a_div.split(" ")
str_a_div_l = str_a_div.split("l")

print(str_a_div_, ": str_a_div")
print(str_a_div_l, ": str_a_div")
print(str_a, ": str_a")

['Ronaldo', 'fait', 'la', 'passe', 'à', ''] : str_a_div
['Rona', 'do fait ', 'a passe à '] : str_a_div
Ronaldo fait la passe à : str_a
```

## Supprimer la partie avant d'une chaîne de caractères

```
In [21]: # solution : fonction str.strip(leading)
str_a_strip = str_a
str_a_ron = str_a_strip.strip("Ronaldo ")
str_a_flp = str_a_strip.strip("fait la passe")
str_a_flp_ = str_a_ron.strip("fait la passe")

print(str_a_ron , ": str_a_ron")
print(str_a_flp, ": str_a_flp")
print(str_a_flp_ , ": str_a_flp_")
print(str_a, ": str_a")

fait la passe à : str_a_ron
Ronaldo fait la passe à : str_a_flp
à : str_a_flp_
Ronaldo fait la passe à : str_a
```

## Joindre des mots issus d'une liste

```
In [23]: # solution : fonction str.join(iterable(liste, tuple etc.))
print(str_a_div_ , ": str_a_div ")
str_a_join = " ".join(str_a_div_)
str_a_join_div = "-".join(str_a_div_)

print(str_a_join, ": str_a_join")
print(str_a_join_div, ": str_a_join_div")

['Ronaldo', 'fait', 'la', 'passe', 'à', ''] : str_a_div
Ronaldo fait la passe à : str_a_join
Ronaldo-fait-la-passe-à- : str_a_join_div
```

## Compter un caractère en particulier

```
In [24]: # solution : fonction str.count(str) ATTENTION : VERIFIEZ BIEN QUE VOUS AVEZ LE DROIT DE L'UTILISER
!
str_a_count_o = str_a.count("o")
str_a_count_fa = str_a.count("fa")

print(str_a_count_o, ": str_a_count_o")
print(str_a_count_fa, ": str_a_count_fa")
print(str_a, ": str_a")

2 : str_a_count_o
1 : str_a_count_fa
Ronaldo fait la passe à : str_a
```