
I2 - Afficher une phrase dynamique

On veut souvent afficher une phrase dont le contenu va varier en fonction des valeurs de nos variables. C'est ce qu'on appelle une **phrase dynamique** par opposition à une **phrase statique** qui restera toujours la même.

Imaginons que nous ayons trois variables `nom = "Hugo"`, `prenom = "Victor"` et `age = 83` et que nous voulions afficher la phrase `"Bonjour Victor Hugo. Tu as 83 ans."`.

Un simple

```
print("Bonjour Victor Hugo. Tu as 83 ans.")
```

suffirait mais on devra alors modifier cette instruction si les valeurs des variables changent.

Pour que ce changement se fasse automatiquement, on a quatre solutions :

► **Utiliser plusieurs paramètres dans la fonction `print` :**

```
>>> print("Bonjour", prenom, nom, "! Tu as", age, "ans.")
Bonjour Victor Hugo. Tu as 83 ans.
```

On remarque alors que Python met automatiquement des espaces entre les différentes parties. Pour modifier ce comportement, on peut ajouter un argument optionnel à la fonction `print` :

```
>>> print("Bonjour", prenom, nom, "! Tu as", age, "ans.", sep='')
BonjourVictorHugo! Tu as83ans.

>>> print("Bonjour", prenom, nom, "! Tu as", age, "ans.", sep='-')
Bonjour-Victor-Hugo-! Tu as-83-ans.
```

► **Utiliser le typage (conversion de type) et la concaténation :**

```
>>> print("Bonjour " + prenom + nom + " ! Tu as " + str(age) + " ans.")
Bonjour Victor Hugo. Tu as 83 ans.
```

On remarque que l'on doit alors gérer nous même les espacements entre les morceaux au risque d'avoir un résultat qui n'est pas satisfaisant :

```
>>> print("Bonjour" + prenom + nom + " ! Tu as" + str(age) + "ans.")
BonjourVictorHugo! Tu as83ans.
```

► **Utiliser la fonction `format` :**

```
>>> print("Bonjour {} {} ! Tu as {} ans.".format(prenom, nom, age))
Bonjour Victor Hugo. Tu as 83 ans.
```

Ici les accolades représentent les endroits où les valeurs des variables vont être placées. Entre les parenthèses après `format`, on écrit le nom de ces variables dans l'ordre d'apparition dans la phrase.

► **Utiliser les f-string (sous Python 3.6 et supérieur) :**

```
>>> print(f"Bonjour {prenom} {nom} ! Tu as {age} ans.")
Bonjour Victor Hugo. Tu as 83 ans.
```

Ici, ce qui est entre accolades ne sera pas affiché directement mais évalué puis affiché. C'est la façon la plus simple et logique de procéder.

Remarque : On peut encore améliorer les exemples précédents en remarquant qu'on n'est pas sûr que `age` soit supérieur ou égal à 2 auquel cas on aura une faute d'orthographe au mot "`ans`". Pour résoudre ce problème, on peut définir une variable `s` qui vaudra "`s`" si `age` est au moins égal à 2 et qui vaudra "" sinon :

```
if age >= 2:
    s = "s"
else:
    s = ""
print(f"Bonjour {prenom} {nom} ! Tu as {age} an{s}.")
```

puis

```
>>> age = 1
>>> (executing file "<tmp 1>")
Bonjour Victor Hugo. Tu as 1 an.
```

Erreurs fréquentes

- Ne pas faire la différence entre le nom d'une variable et sa valeur

Par exemple

```
>>> print("Bonjour prenom nom ! Tu as age ans.")
Bonjour prenom nom ! Tu as age ans.
```