FACULTE DES SCIENCES AIN CHOCK
UNIVERSITE HASSAN II DE CASABLANCA

Département: Mathématiques & Informatique

Séance 8:
Les chaînes

Licence : Physique Chimie

Pr: Youssef Ouassit

Les str

Définition d’une chaîne de caractères

Le type **str** en Python est une séquence d'unités de caractères Unicode.
Les chaînes de caractères sont utilisées pour stocker et manipuler du texte.

Les chaînes de caractères sont définies en utilisant des guillemets simples ('), des guillemets doubles ("), ou des triples guillemets (''' ou """) pour les chaînes multilignes.

Les chaînes de caractères

Déclaration

```
# Définir une chaîne

chaine1 = 'Bonjour'

chaine2 = "le monde"

chaine3 = '''Ceci est une chaîne qui s'étend sur
plusieurs lignes.'''
```

Les chaînes de caractères

Opérateurs de base

Opérateur d'Appartenance **in**

```
ch = "pomme"
print("po" in ch)  # True
print("ora" in ch)  # False
```

Opérateur de concaténation **+**

```
ch1 = "Bonjour "
ch2 = "Ahmed"
ch3 = ch1 + ch2  # "Bonjour Ahmed"
```

Les chaînes de caractères

Opérateurs de base

Opérateur de répétition *

```
ch = "bon"
resultat = ch * 2  # "bonbon"
```

Opérateur d’indexation []

```
ch = "pomme"
print(ch[0])  # 'p'
print(ch[-1])  # 'e'
```

Opérateurs de base

Opérateur de Slicing **[debut : fin : pas]**

```
ch = "Bonjour"
sous_chaine = ch[:3]  # Bon
sous_chaine2 = ch[1:3]   # on
sous_chaine3 = ch[::2]  # Bnor
sous_chaine4 = ch[:]  # Bonjour
```

Les chaînes de caractères

Les méthodes

Méthode : `upper`

Description : Retourne une nouvelle chaîne où tous les caractères sont convertis en majuscules.

Syntaxe : `str.upper()`

Exemple :

```python
texte = "bonjour"
print(texte.upper())  # Affiche : BONJOUR
```

Les chaînes de caractères

Les méthodes

Méthode : `lower`

Description : Retourne une nouvelle chaîne où tous les caractères sont convertis en minuscules.

Syntaxe : `str.lower()`

Exemple :

```python
texte = "BONJOUR"
print(texte.lower())  # Affiche : bonjour
```

Les chaînes de caractères

Les méthodes

Méthode : `` `replace` ``

Description : Remplace toutes les occurrences d'une sous-chaîne par une autre dans la chaîne d'origine.

Syntaxe : `` `str.replace(old, new[, count])` ``

Exemple :

```python
texte = "bonjour monde"
print(texte.replace("monde", "tout le monde"))  # Affiche : bonjour tout le monde
```

Les chaînes de caractères

Les méthodes

Méthode : `split`

Description : Divise la chaîne en une liste de sous-chaînes en utilisant le séparateur spécifié. Si aucun séparateur n'est fourni, les espaces sont utilisés par défaut.

Syntaxe : `str.split([separator[, maxsplit]])`

Exemple :

```python
texte = "bonjour tout le monde"
print(texte.split())  # Affiche : ['bonjour', 'tout', 'le', 'monde']
```

Les chaînes de caractères

Les méthodes

Méthode : `` `find` ``

Description : Retourne l'index de la première occurrence de la sous-chaîne spécifiée dans la chaîne. Retourne `` `-1` `` si la sous-chaîne n'est pas trouvée.

Syntaxe : `` `str.find(sub[, start[, end]])` ``

Exemple :

python — Copier le code

```python
texte = "bonjour"
print(texte.find("jour"))  # Affiche : 3
```

Les chaînes de caractères

Les méthodes

Méthode : `count`

Description : Retourne le nombre de fois qu'une sous-chaîne apparaît dans la chaîne.

Syntaxe : `str.count(sub[, start[, end]])`

Exemple :

```python
texte = "bonjour tout le monde"
print(texte.count("o"))  # Affiche : 3
```

Les chaînes de caractères

Les fonctions

On retrouve les même fonctions déjà vu :

- len
- max
- min
- …

Des fonctions spécifiques aux chaînes de caractères:

- **ord** : Retourne le code Unicode d'un caractère.
- **chr** : Retourne le caractère correspondant à un code Unicode
- **eval :** Évalue une chaîne comme une expression Python.
- **input** : Demande une entrée utilisateur.