

Paramètres par défaut

Chaines de caractères

Arnaud Sallaberry

arnaud.sallaberry@univ-montp3.fr

Plan

- Paramètres par défaut
- Chaines de caractères

Paramètres par défaut : Principe

- Si aucune valeur n'est passée en paramètre, la valeur par défaut est attribuée
- Si une valeur est passée en paramètre, cette valeur est utilisée pour « remplir » le champ correspondant
- Applicable à tous les types de fonctions (fonctions, méthodes, constructeurs)

Exemples

```
class Personne:
```

```
    def __init__(self, n, p, a):
```

```
        self.nom = n
```

```
        self.prenom = p
```

```
        self.age = a
```

```
    def afficher(self):
```

```
        print(self.nom+" "+self.prenom+" "+self.age)
```

Exemples

```
class Personne:
```

```
    def __init__(self, n="nom inconnu",  
                  p="prenom inconnu", a=-1):
```

```
        self.nom = n
```

```
        self.prenom = p
```

```
        self.age = a
```

```
    def afficher(self):
```

```
        print(self.nom+", "+self.prenom+", "+self.age)
```

Exemples

Code :

```
p = Personne('Sallaberry', 'Arnaud', 29)
p.afficher()
```

Affichage :

```
Sallaberry, Arnaud, 29
```

Code :

```
p = Personne('Sallaberry', 'Arnaud')
p.afficher()
```

Affichage :

```
Sallaberry, Arnaud, -1
```

Exemples

Code :

```
p = Personne (p='Arnaud')  
p.afficher()
```

Affichage :

```
nom inconnu, Arnaud, -1
```

Code :

```
p = Personne (n='Sallaberry', a=29)  
p.afficher()
```

Affichage :

```
Sallaberry, prenom inconnu, 29
```

Exemples

Code :

```
p = Personne()  
p.afficher()
```

Affichage :

```
nom inconnu, prenom inconnu, -1
```

Plan

- Paramètres par défaut
- Chaines de caractères

Décomposer une chaîne

```
s1 = "Einstein Albert UPV"
```

```
mots1 = s1.split()
```

```
mots1 contient la liste : ['Einstein', 'Albert', 'UPV']
```

```
mots1[0] contient la chaîne 'Einstein'
```

```
mots1[1] contient la chaîne 'Albert'
```

```
mots1[2] contient la chaîne 'UPV'
```

```
s2 = "Einstein;Albert;UPV"
```

```
mots2 = s2.split(";")
```

```
mots2 contient la liste : ['Einstein', 'Albert', 'UPV']
```

Recomposer une chaîne

```
mots = ['Einstein', 'Albert', 'UPV']
```

```
s1 = " ".join(mots)
```

```
s1 contient la chaîne 'Einstein Albert UPV'
```

```
s2 = ";".join(mots)
```

```
s2 contient la chaîne 'Einstein;Albert;UPV'
```

Accéder aux lettres

```
nom = 'Einstein'
```

```
print(nom[3])           's'  
print(nom[3:6])        'ste'  
print(nom[3:])          'stein'  
print(nom[:3])         'Ein'
```