Révisions types de données Correction exercices

Christophe Viroulaud

Terminale - NSI

Rév 01

Exercice 1

Exercice 1 :

```python
tab = [1 for i in range(5)]
```

Exercice 2

Exercice 2 :

```python
tab = [i for i in range(5)]
```

Exercice 3

Exercice 3 :

```python
tup = tuple(i for i in range(4, -1, -1))
```

Exerice 4

Exercice 4 :

```python
tup = tuple(i for i in range(0, 9, 2))

# seconde méthode
tup = tuple(i for i in range(9) if i%2 == 0)
```

Exercice 5

Exercice 5 :

```python
dico = {i: 1 for i in range(5)}
```

Exercice 6

Exercice 6 :

```python
dico = {chr(65+i): i for i in range(5)}
```

Exercice 7

Exercice 7 :

```python
from random import randint

tab = [randint(0, 100) for i in range(10)]
```

Exercice 8

Exercice 8 :

```python
from random import randint

tab = [randint(0, 100) for i in range(10)]

def maxi(tab: list) -> int:
    maximum = 0
    for element in tab:
        if element > maximum:
            maximum = element
    return maximum

print(tab)
print(maxi(tab))
```

Exercice 9

Exercice 9 :

```python
from random import randint

tup = tuple(randint(0, 100) for i in range(10))

def somme(tup: tuple) -> int:
    resultat = 0
    for element in tup:
        resultat += element
    return resultat

print(tup)
print(somme(tup))
```

Exercice 10

Exercice 10 :

```python
tab = ["qui", "que", "quoi", "dont", "où", "comment"]

i1 = int(input("indice 1: "))
i2 = int(input("indice 2: "))

temp = tab[i1]
tab[i1] = tab[i2]
tab[i2] = temp

print(tab)
```

Exercice 11

Exercice 11 :

```python
bibliotheque = [
    {"titre": "Il était deux fois",
        "auteur": "Franck Thilliez",
        "editeur": "Poche",
        "prix": 8.70},
    {"titre": "Fahrenheit 451",
        "auteur": "Ray Bradbury",
        "editeur": "Folio",
        "prix": 6.30},
    {"titre": "Le guide du voyageur galactique",
        "auteur": "Douglas Adams",
        "editeur": "Folio",
        "prix": 8.10}
]

for livre in bibliotheque:
    print(livre["auteur"])
```

Exercice 12

Exercice 12 :

```python
def lettres(mot: str)->dict:
    """
    compte le nombre d'occurrences
    de chaque lettre du mot
    """
    occurrences = {}
    for l in mot:
        # si la lettre est déjà référencée
        if l in occurrences:
            occurrences[l] += 1
        else:
            occurrences[l] = 1
    return occurrences

print(lettres("bonjour"))
```