Exercices fonctions correction

Christophe Viroulaud

Première - NSI

Lang 06

Sommaire

Exercice 1

```python
def est_pair(x: int) -> bool:
    """
    vérifie si x est pair

    Args:
        x (int): entier

    Returns:
        bool: True si x est pair
    """
    if x % 2 == 0:
        return True
    else:
        return False

# appel de la fonction
print(est_pair(5))
```

Sommaire

Exercice 2

```python
def valeur_absolue(x: int) -> int:
    """
    renvoie la valeur absolue de x
    """
    if x < 0:
        return -x
    else:
        return x


print(valeur_absolue(-4))
```

Remarque

La `docstring` est moins détaillée mais toujours présente.

Sommaire

Exercice 3

```python
def surface(r: int) -> float:
    """
    renvoie la surface du disque de rayon r
    """
    return 3.14*r**2

print(surface(5))
```

```python
from math import pi

def surface2(r: int) -> float:
    """
    renvoie la surface du disque de rayon r
    utilise la bibliothèque math
    """
    return pi*r**2

print(surface2(5))
```

Sommaire

Exercice 4

```python
def est_majeur(age: int) -> bool:
    """
    vérifie si la personne est majeur

    Args:
        age (int): âge de la personne

    Returns:
        bool: True si majeur
    """
    if age >= 18:
        return True
    else:
        return False
```

Exercice 4 - version 2

```python
def est_majeur2(age: int) -> bool:
    """
    vérifie si la personne est majeur

    Parameters
    ----------
    age : int
        âge de la personne.
    Returns
    -------
    boolean

    """
    return age >= 18
```

Sommaire

Exercice 5

```python
def puissance(x: int, n: int) -> int:
    """
    élève x à la puissance n

    Parameters
    ----------
    x : int
        entier.
    n : int
        exposant.

    Returns
    -------
    res : int
    """
    res = 1
    for i in range(n):
        res *= x
    return res
```

Sommaire

Exercice 6

```python
from random import randint


def lancer_des() -> int:
    """
    renvoie la somme de deux dés
    """
    de1 = randint(1, 6)
    de2 = randint(1, 6)
    return de1+de2
```

Sommaire

Exercice 7

```python
def pythagore(a: int, b: int, c: int) -> bool:
    """
    vérifie si le triangle a, b, c est rectangle
    """
    cotes = a**2+b**2
    hyp = c**2
    if cotes == hyp:
        return True
    else:
        return False
```

Exercice 7 - version 2

```python
def pythagore2(a: int, b: int, c: int) -> bool:
    """
    vérifie si le triangle a, b, c est rectangle

    Parameters
    ----------
    a, b, c: int
        mesures des côtés.

    Returns
    -------
    Boolean.

    """
    return a**2 + b**2 == c**2
```

Sommaire

Exercice 8

```
1  def somme(n: int) -> int:
2      """
3      renvoie la somme des entiers de 1 à n
4      """
5      somme = 0
6      for i in range(n+1):
7          somme = somme + i
8      return somme
```

Sommaire

Exercice 9

```python
def est_premier(x: int) -> bool:
    """
    renvoie True si x est un nombre premier
    """
    diviseur = 2
    # si le reste est nul, c'est que nous avons un diviseur
    while diviseur < x and not(x % diviseur == 0):
        diviseur += 1
    # On a divisé par tous les nombres < x
    if diviseur == x:
        return True
    else:  # on s'est arrêté avant
        return False
```

Sommaire

Exercice 10

```
1  def valeur_proche(a: int) -> int:
2      """
3      renvoie l'entier inférieur le plus proche
4      de racine(a)
5      """
6      entier = 1
7      while entier**2 <= a:
8          entier += 1
9      return entier-1
```

Exercice 10

```
1  def racine(a: int) -> float:
2      """
3      calcule une approximation de
4      racine carrée de a
5      """
6      x = valeur_proche(a)
7      for i in range(20):
8          x = 0.5*(x+a/x)
9      return x
```

Exercice 11

```
1  def triangle(c: int) -> None:
2      """
3      Trace un triangle noir
4      """
5      t.begin_fill()
6      for _ in range(3):
7          t.forward(c)
8          t.left(120)
9      t.end_fill()
```

Exercice 11

```
1  # programme principal
2  t.up()
3  for _ in range(3):
4      t.left(90)
5      t.forward(50)
6      t.right(90)
7      triangle(100)
8  t.done()
```