

Prenom : \_\_\_\_\_

Nom : \_\_\_\_\_

## Initiation à Python : questionnaire

### Question 1

Un programme contient cette instruction :

```
age = input("Quel est ton âge ?")
```

Pendant une execution du programme, un utilisateur répond 28 à la question.  
Que contient la variable `age` ?

- A. 28
- B. "28"
- C. None

### Question 2

Pour concaténer un entier `n` à une chaîne de caractère, j'écris :

- A. "une chaîne" + `str(n)`
- B. "une chaîne" + `int(n)`
- C. "une chaîne" + `"n"`

### Question 3

Pour vérifier qu'une variable `n` est un entier, j'utilise :

- A. `isinstance(n, int)`
- B. `n == "int"`
- C. `int(n) == True`

### Question 4

À la fin de l'exécution de ce programme, que contiendra la variable `x` ?

```
def dire_bonjour():  
    print("Bonjour !")
```

```
x = dire_bonjour()
```

- A. "Bonjour !"
- B. ""
- C. None

### Question 5

Dans le programme de la question précédente, pour que la fonction `dire_bonjour()` renvoie "Bonjour !", j'aurais dû remplacer la deuxième ligne par :

- A. `return "Bonjour !"`
- B. `return print("Bonjour !")`
- C. Rien du tout, c'était déjà bon !

### Question 6

Pour savoir si une variable `n` contient l'entier 20, j'écris :

- A. `n = 20`
- B. `n = "20"`
- C. `n == 20`
- D. `n == "20"`

### Question 7

Qu'affiche le programme suivant ?

```
for i in range(0,10):
    if i % 2 == 0:
        continue
    print("La variable i vaut " + str(i))
```

- A. Un message pour chaque entier impair entre 0 et 9 compris
- B. Rien du tout
- C. Il y a une erreur de syntaxe

### Question 8

Qu'affiche le programme suivant ?

```
for i in range(0,10):
    if i % 2 == 0:
        continue
    print("La variable i vaut " + str(i))
```

- A. Un message pour chaque entier impair entre 0 et 9 compris
- B. Rien du tout
- C. Il y a une erreur de syntaxe

### Question 9

Une liste a été créée de cette manière :

```
ma_liste = [ "Le", "Python", "c'est", "cool", "!" ]
```

Laquelle de ces instructions renvoie "Python" ?

- A. `ma_liste[1]`
- B. `ma_liste[2]`
- C. `ma_liste[3]`

#### Question 10

Pour découper une chaîne de caractère `s` par rapport aux `;` qu'elle contient et obtenir ainsi une liste, j'utilise :

- A. `s.strip(';')`
- B. `s.join(';')`
- C. `s.split(';')`

#### Question 11

Pour importer la librairie permettant de manipuler du json, j'écris au début de mon programme :

- A. `include json`
- B. `import json`
- C. `require json`

#### Question 12

Ecrire une fonction `pairs` qui prend en argument une liste d'entier et renvoie la liste entiers pairs qu'elle contient. Par exemple, `pairs([3, 6, 9, 5, 2])` renverra `[6, 2]`.

```
|  
|  
|  
|  
|  
|  
|
```

#### Question 13

Des méthodes dont les noms sont entourés de deux underscores (**name**) sont:

- A. des méthodes spéciales qui permettent notamment d'implémenter les opérations de base du langage python pour ses propres classes.
- B. des attributs magiques qui sont utiles pour convertir des propriétés en objets et objets dérivés.
- C. les méthodes de la bibliothèque standard de python et téléchargées depuis pypi.org.

#### Question 14

Compléter ce programme pour afficher dans la console le type de Salameche à partir du contenu de `d` :

```
d = [ "pikachu", "salameche" ]
```

```
|  
|  
|
```

#### Question 15

Écrire la première ligne (`for ...`) qui permet d'itérer sur les valeurs d'une liste `l`.

```
|  
|  
|
```

#### Question 16

On veut écrire un programme qui, étant donné une chaîne de caractère `stuff`, essaye de la convertir en entier avec `mon_entier = int(stuff)`. Pour savoir si la conversion a échoué et stocker `-1` dans `mon_entier` à la place, j'utilise de préférence:

- A. `if` et `else`
- B. `isinstance`
- C. `try` et `except`

#### Question 17

Ajouter, avant les deux lignes déjà écrites, quelques lignes permettant de définir une classe `Voiture`, et que `v.color` renvoie `"bleu"`

```
|  
|  
|  
|  
|  
|  
|
```

```
v = Voiture()  
print(v.color)
```

### Question 18

En python, ce que l'on note généralement `self` représente :

- A. Le constructeur de la classe
- B. La conversion en format JSON de l'objet
- C. L'instance de objet en train d'être modifié / étudié

### Question 19

Pour tester et rendre explicite une hypothèse faite par un programme (par exemple tester si un nombre est positif ou si la longueur d'une liste est 3), j'utilise :

- A. `if`
- B. `except`
- C. `assert`

### Question 20

Pour faciliter la compréhension de mon programme par mes collègues et mon futur moi, j'appelle mes fonctions et mes variables :

- A. Par des noms qui décrivent précisément ce qu'elles font / contiennent
- B. Avec une seule lettre comme `a`, `b`, `f`, `x` . . .
- C. Obiwan Kénobi
- D. La réponse D.