

Examen Blanc Langage Evolué | Nom : ..... Prénom : .....Groupe : .....

## EXERCICE 1 : QCM (10 points)

**Q1** : Que va renvoyer la commande suivante :

```
>>> type ( [1,2] )
```

- <class 'list'>
- <class 'tuple'>
- <class 'dict'>
- <class 'str'>

**Q2** : Que va renvoyer la commande suivante :

```
>>> type ( { 1:2 , 4:8 } )
```

- <class 'list'>
- <class 'tuple'>
- <class 'dict'>
- <class 'str'>

**Q3** : Soit :

```
>>> a = [ 1 , 2 , 3 , { 1:2 } , [ 1 , 2 ] ]
>>> len(a)
```

Que va afficher le code suivant : **len(a)**

**Q4** : Que va afficher le code suivant :

```
x = True
y = False
z = False
if x or (y and z):
    print ("OUI")
else:
    print ("NON")
```

**Q5** : Indiquez la liste des valeurs qui seront affichées par le code suivant :

```
for x in range(2):
    print(x)
for x in range(4,6):
    print(x)
```

- 2, 4, 6
- 0,1, 4, 5
- 0, 1, 2, 4, 5, 6
- 0, 1, 4, 5, 6, 7, 8, 9
- 1, 2, 4, 5, 6

**Q6** : Indiquez la liste des valeurs qui seront affichées par le code suivant :

```
for x in range(5,16,3):
    print(x)
```

**Q7** : Indiquez ce que va afficher le code suivant :

```
try:
    print (2+"Math-STID")
except:
    print ("erreur")
else:
    print ("OK")
```

**Q8** : Indiquez ce que va afficher le code suivant si l'utilisateur tape 14 en réponse à l'instruction « input » :

```
try:
    a = input("Donne une valeur")
except:
    print ("erreur")
else:
    print ("OK")
```

**Q9** : Soit une liste initialisée comme suit :

```
>>> modules = ['math', 'physique', 'chimie', 'algo']
```

Quelle est la commande python permettant de créer une chaîne « S » composée des éléments de la liste « modules » séparés par le caractère « : ».

- S = join(modules , ":" )
- S = ":".join(modules)
- S = modules + ":"
- S = modules.split(":")

**Q10** : Soit les commandes suivantes :

```
>>> capitale = {}
>>> capitale['Algérie']='alger'
>>> capitale['algérie']='Alger'
>>> capitale['Tunisie']='Tunis'
>>> print(len(capitale))
```

Que va afficher la dernière commande ?

**Q11** : Soit les commandes suivantes :

```
>>> eleve = ['amal', 'said', 'mohand', 'amira']
>>> eleve = []
>>> print (len(eleve))
```

Que va afficher la dernière commande ?

**Q12** : Soit les commandes suivantes :

```
>>> eleve = ['amal', 'said', 'mohand', 'amira']
>>> print ( eleve[-1] )
```

Que va afficher la dernière commande ?

**Q13** : Soit les commandes suivantes :

```
>>> eleve = ['amal', 'said', 'mohand', 'amira']
>>> print ( eleve[-1][-1] )
```

Que va afficher la dernière commande ?

**Q14** : Soit les commandes suivantes :

```
>>> eleve = ['amal', 'said', 'mohand', 'amira']
>>> print ( eleve[0][0 :3] )
```

Que va afficher la dernière commande ?

**Q15** : Que va afficher le programme suivant :

```
>>> s = "x y z"
>>> r = s.split()
>>> r.append(10)
>>> print(r)
```

**Q16** : Que va afficher le programme suivant :

```
>>> v = 10
>>> def f(x):
    v = 14
    return x*2
>>> print(v)
```

**Q17** : Que va afficher le programme suivant :

```
>>> def f(x, y=10):
    return x+y
>>> print(f(5,6))
```

**Q18** : Que va afficher le programme suivant :

```
>>> def f(x, y=10):
    return x+y

>>> print(f(5))
```

**Q19** : Que va afficher le programme suivant :

```
>>> def f(x, y=10):
    return x+y

>>> print(f("Physique", "Math"))
```

**Q20** : Que va afficher le programme suivant :

```
>>> notes = [ 12, 14, 15 ]
>>> eleves = [ "amal", "said", "nawel" ]
>>> d = { }
>>> for i in range(len(notes)):
    d[eleves[i]] = notes[i]
>>> print(d)
```



