

Exercitar Python

1. Insira a parte que falta do código abaixo para a saída "Hello World".

_____("Hello World")

2. Comentários em Python são escritos com um caractere especial, qual?

_____ Isto é um comentário

3. Use uma string de várias linhas para fazer um comentário de várias linhas:

Isto é um comentário
escrito em
mais do que uma linha!

4. Crie uma variável com o nome **car_nome** e atribua o valor **Volvo** .

_____ = "_____"

5. Crie um variável x e atribua o valor 50.

_____ = _____

6. Exiba a soma de 5 + 10, usando duas variáveis: x e y.

_____ = _____
y = 10
print(x _____ y)

7. Indique o(s) caractere(s) que não está corretos(s) no nome da variável:

2my-first_name = "John"

8. Insira a sintaxe correta para atribuir o mesmo valor a todas as três variáveis em uma linha de código.

X _____ y _____ z _____ "banana"

9. Insira a palavra-chave correta para que a variável x pertença ao escopo global.

```
def myfunc():
```

```
_____ x
```

```
X = "Fantástico"
```

10. O exemplo de código a seguir imprime o tipo de dados de **x**, que tipo de dados seria?

```
X = 5
```

```
print(type())
```

```
_____
```

11. O exemplo de código a seguir imprime o tipo de dados de **x**, que tipo de dados seria?

```
X = ("Hello World")
```

```
print(type())
```

```
_____
```

12. O exemplo de código a seguir imprime o tipo de dados de x, que tipo de dados seria?

```
X = 20.5
```

```
print(type())
```

```
_____
```

13. O exemplo de código a seguir imprimiria o tipo de dados de x, que tipo de dados seria?

```
x = ["maçã", "banana", "cereja"]
```

```
print(type())
```

```
_____
```

14. O exemplo de código a seguir imprime o tipo de dados de x, que tipo de dados seria?

```
x = ("apple", "banana", "cherry")
```

```
print(type(x))
```

```
_____
```

15. O exemplo de código a seguir imprime o tipo de dados de x, que tipo de dados seria?

```
x = {"nome" : "João", "idade" : 36}
```

```
print(type(x))
```

16. O exemplo de código a seguir imprimiria o tipo de dados de x, que tipo de dados seria?

```
x = True
```

```
print(type())
```

17. Insira a sintaxe correta para converter x num número flutuante.

```
x = 5
```

```
x = _____ (x)
```

18. Insira a sintaxe correta para converter x em um inteiro.

```
x = 5.5
```

```
x = _____ (x)
```

19. Insira a sintaxe correta para converter x em um número complexo.

```
x = 5
```

```
x = _____ (x)
```

20. Use o método **len** para imprimir o comprimento da string.

```
x = "Hello World"
```

```
print(_____)
```

21. Use o método **len** para imprimir o comprimento da string.

```
x = "Hello World"
```

```
print(_____)
```

22. Obtenha o primeiro caractere da string txt.

```
txt = "Hello World"
```

```
x = _____
```

23. - Obtenha os caracteres do índice 2 ao índice 4 (llo).

```
txt = "Hello World"
```

```
x = _____
```

24. `print('Hello, world!')`

25. `print(5 + 10)`

26.

```
a = 5
```

```
b = 7
```

```
print(a + b)
```

27. Agora é sua vez. Escreva um programa que leia três números e imprima sua soma;

28. `print(3 * 7, (17 - 2) * 8);`

29. `print(2 ** 16)` # dois asteriscos são usadas para exponenciação (2 elevado a 16);

30. `print(37 / 3)` # barra simples é uma divisão;

31. `print(37 // 3)` # barra dupla é uma divisão inteira;
Devolve/retorna apenas o quociente da divisão (ou seja, sem resto)

32. `print(37 % 3)` # sinal de percentagem é um operador de módulo;
(Devolve o resto da divisão)

33. Faça um programa que receba dados do utilizador(inteiro) e verifique se é par ou não; (Um número é ímpar quando, na divisão por dois, o resto é diferente de zero.)

34. Faça um programa que receba dados do utilizador(inteiro) e verifique se o número é cheio ou redondo. por exemplo: 100, 200, 500 e etc. (Para isso, dividirmos o mesmo por 100 e compararmos o módulo com 0)

35. Desta vez, o programa lê dois números. Execute este exemplo:

```
a = int(input())
```

```
b = int(input())
```

```
print(a + b)
```

36. Pedir 2 valores ao utilizador e exibe o resultado da divisão e do resto;

Listas

37. Imprima o segundo item da lista das frutas.

```
frutas = ["maçã", "banana", "cereja"]
```

```
print(_____)
```

38. Altere o valor de "maçã" para "kiwi", na lista das frutas.

```
frutas = ["maçã", "banana", "cereja"]
```

```
_____ = _____
```

39. Use o método **append** para adicionar "**laranja**" à lista das frutas.

```
frutas = ["maçã", "banana", "cereja"]
```

```
_____
```

40. Use o método **insert** para adicionar "**limão**" como o segundo item da lista das frutas.

41. Use o método **remove** para remover "banana" da lista das frutas.

```
frutas = ["maçã", "banana", "cereja"]
```

```
_____
```

42. Faça um programa que receba a idade de um utilizador e exiba se a idade introduzida é de uma criança, adulto ou idoso (criança – 12 anos, adolescente – 13, adulto – 18 e idoso).

Tuples

43. Use a sintaxe correta para imprimir o primeiro item do tuple frutas.

```
frutas = ["maçã", "banana", "cereja"]  
  
print(_____)
```

44. Use a sintaxe correta para imprimir o último item do tuple frutas.

```
frutas = ["maçã", "banana", "cereja"]  
  
print(_____)
```

45. Use a sintaxe correta para imprimir o número de itens do tuple frutas.

```
frutas = ["maçã", "banana", "cereja"]  
  
print(_____)
```

SET - Conjuntos

46. Verifique se "maçã" se encontra presente no conjunto de frutas.

```
frutas = {"maçã", "banana", "cereja"}  
  
if "maçã" _____ frutas:  
    print("Sim, maçã é uma fruta!")
```

47. Use o método **add** para adicionar "laranja" ao conjunto das frutas.

```
frutas = {"maçã", "banana", "cereja"}  
  
_____
```

48. Use o método **remove** para remover "banana" do conjunto frutas.

```
frutas = {"maçã", "banana", "cereja"}  
  
_____
```

Dicionários

49. Use o método **get** para imprimir o valor da chave "modelo" o dicionário **carro**.

```
carro = {  
    "marca": "Volvo",  
    "modelo": "X60",  
    "ano": 2022  
}  
  
print()
```

50. Altere o valor "ano" de 1664 para 2020.

```
carro = {  
    "marca": "Volvo",  
    "modelo": "X60",  
    "ano": 2022  
}  
_____ = _____
```

51. Adicione o par chave/valor "cor" : "preto" ao dicionário **carro**.

```
carro = {  
    "marca": "Volvo",  
    "modelo": "X60",  
    "ano": 2022  
}  
_____
```

52. Use o método **clear** para esvaziar o dicionário **carro**.

```
carro = {  
    "marca": "Volvo",  
    "modelo": "X60",  
    "ano": 2022  
}  
_____
```

If ... Else

53. Imprima "Hello World" se **a** for maior que **b**.

```
a = 50  
b = 10  
_____ a _____ b _____--  
    print("Hello World")
```

54. Imprima "Hello World" se **a** não for igual a **b**.

```
a = 50
b = 10
_____ a _____ b _____
    print("Hello World")
```

55. Imprima "Sim" se **a** for igual a **b**, caso contrário, imprima "Não".

```
a = 50
b = 10
_____ a _____ b _____
    print("Sim!")
_____
    print("Não!")
```

56. Imprima "1" se **a** for igual a **b**, imprima "2" se **a** for maior que **b**, caso contrário imprima "3".

```
a = 50
b = 10
_____ a _____ b _____
    print("1")
_____
    print("2")
_____
    print("3")
```

57. Imprima "Olá" se **a** for igual a **b**, e **c** for igual a **d**.

```
if a == b _____ c == d:
    print("Olá")
```

58. Imprima "Olá" se **a** for igual a **b**, ou se **c** for igual a **d**.

```
if a == b _____ c == d:
    print("Olá")
```

Ciclo While

59. Imprima **i** enquanto **i** for menor que 6.

```
i = 1
_____ i < 6 _____
    print(i)
    i += 1
```


60. Pare o loop se i for 3

```
i = 1
while i < 6:
    if i == 3:
        _____
        i += 1
```

61. No **loop**, quando i é 3, salta diretamente para a próxima iteração.

```
i = 0
while i < 6:
    i += 1
    if i == 3:
        _____
    print(i)
```

62. Imprima uma mensagem quando a condição for falsa.

```
i = 1
while i < 6:
    print(i)
    i += 1
    _____

print("i é menor que 6")
```

Ciclo For

63. Percorra os itens da lista frutas.

```
frutas = ["maçã", "banana", "cereja"]
_____ x _____ frutas _____
print(x)
```

64. Use a função **range** para percorrer um conjunto de códigos 6 vezes.

```
for x in _____ :
    print(x)
```

65. Saia do **loop** quando **x** for "banana".

```
frutas = ["maçã", "banana", "cereja"]
for x in frutas:
    if x == "banana":
        _____
    print(x)
```

Funções

66. Crie uma função chamada `my_func`.

```
_____  
:  
print("Olá, através de uma função")
```

67. Execute uma função chamada `my_func`.

```
def my_func():  
    print("Olá, através de uma função ")  
_____
```

68. Dentro de uma função com dois parâmetros, imprima o primeiro parâmetro.

```
def my_func(nome, apeli):  
    print(_____)
```

69. Faça que a função devolva o parâmetro **x + 5**.

```
def my_func(x):  
    _____
```

70. Quando não sabemos o número de argumentos que serão passados para a função, existe um prefixo que podemos adicionar na definição da função, qual é esse prefixo?

```
def my_func( _____ crianca):  
  
    print("A criança mais nova é " + crianca[2])
```

71. Se não sabemos o número de argumentos de *palavras* -chave que serão passados na função, existe um prefixo que podemos adicionar à função, qual é esse prefixo?

```
def my_func( _____ crianca):  
  
    print("O apelido é " + crianca["apeli"])
```