

Exercitar Python

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

```
print("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** .

```
car_nome = "Volvo"
```

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

```
X = 50
```

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

```
x = 5
```

```
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():
```

```
global 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())
```

```
int
```

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

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

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

```
str
```

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

```
X = 20.5
```

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

```
float
```

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(x))
```

list

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))
```

tuple

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))
```

dict

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

```
x = True  
print(type(x))
```

bool

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

```
x = 5  
x = float(x)
```

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

```
x = 5.5
```

```
x = int (x)
```

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

```
x = 5
```

```
x = complex (x)
```

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

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

```
print(len(x))
```

21. Obtenha o primeiro caractere da string txt.

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

```
x = txt[0]
```

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

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

```
x = txt[2:5]
```

23. `print('Hello, world!')` → **Hello, world**

24. `print(5 + 10)` → **15**

25.

```
a = 5
```

```
b = 7
```

```
print(a + b) → 12
```

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

```
a = 5  
b = 7  
c = 10  
print(a + b + c) → 22
```

27. `print(3 * 7, (17 - 2) * 8);` → 21, 120

28. `print(2 ** 16)` → 65536 # dois asteriscos são usadas para exponenciação (2 elevado a 16);

29. `print(37 / 3)` → 12.333333333333334 # barra simples é uma divisão;

30. `print(37 // 3)` → 12 # barra dupla é uma divisão inteira;
Devolve/retorna apenas o quociente da divisão (ou seja, sem resto)

31. `print(37 % 3)` → 1 # sinal de porcentagem é um operador de módulo; (Devolve o resto da divisão)

32. 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.)

```
num = int(input("Digite um número: "))  
if(num % 2 == 0):  
    print("Número é par.")  
else:  
    print("Número é ímpar")
```

33. 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)

```
num = int(input("Digite um número: "))  
if(num % 100 == 0):  
    print("Número é redondo.")  
else:  
    print("O número não é redondo")
```

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

```
a = int(input()) → 20
```

```
b = int(input()) → 30
```

```
print(a + b) → 50
```

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

```
num1 = float(input("Digite um número: "))
```

```
num2 = float(input("Digite outro número: "))
```

```
divisao = num1 / num2
```

```
resto = num1 % num2
```

```
print()
```

```
print(num1, "divido por", num2, "é igual a: ", divisao)
```

```
print("O resto da divisão entre", num1, "e", num2, "é igual a:", resto)
```

Listas

36. Imprima o segundo item da lista das frutas.

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

```
print(frutas[1])
```

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

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

```
frutas[0] = "kiwi"
```

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

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

```
fruta.append("laranja")
```

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

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

```
frutas.insert("limão")
```

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

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

```
frutas.remove("banana")
```

41. 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 – 18, adulto – 67 e idoso).

```
idade = eval(input("Indique uma idade: "))  
if idade < 12:  
    print('criança')  
elif idade < 18:  
    print('adolescente')  
elif idade < 67:  
    print('adulto')  
else:  
    print('idoso')
```

Tuples

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

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

```
print(frutas[0])
```

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

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

```
print( frutas[-1])
```

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

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

```
print(len(frutas))
```

SET - Conjuntos

45. Verifique se “maçã” se encontra presente no conjunto de frutas.

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

```
if "maçãs" in frutas:
```

```
    print("Sim, maçã é uma fruta!")
```

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

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

```
frutas.add("laranja")
```

47. Use o método **remove** para remover “ banana” do conjunto frutas.

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

```
frutas.remove("banana")
```

Dicionários

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

```
carro = {
```

```
    "marca": "Volvo",
```

```
    "modelo": "X60",
```

```
    "ano": 2022
```

```
}
```

```
print(carro.get("modelo"))
```


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

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

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

```
carro = {  
    "marca": "Volvo",  
    "modelo": "X60",  
    "ano": 2022  
}  
car["color"] = "preto"
```

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

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

If ... Else

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

```
a = 50  
b = 10  
if a > b :  
    print("Hello World")
```

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

```
a = 50  
b = 10  
if a != b :  
    print("Hello World")
```

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

```
a = 50
b = 10
if a == b:
    print("Sim!")
else:
    print("Não!")
```

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

```
a = 50
b = 10
if a == b:
    print("1")
elif a > b:
    print("2")
else:
    print("3")
```

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

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

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

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

Ciclo While

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

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

59. Pare o loop se i for 3

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

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

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

61. Imprima uma mensagem quando a condição for falsa.

```
i = 1
while i < 6:
    print(i)
    i += 1
else:
    print("i é menor que 6")
```

Ciclo For

62. Percorra os itens da lista frutas.

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

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

```
for x in range(6):
    print(x)
```

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

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

Funções

65.Crie uma função chamada my_func.

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

66.Execute uma função chamada my_func.

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

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

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

68.Faça que a função devolva(**retorne**) o parâmetro **x + 5**.

```
def my_func(x):
    return x + 5
```

69.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])
```

70.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"])
```