

Aula 2 - O básico da programação

June 10, 2021

1 Estrutura de um programa

É comum na programação iniciarmos nosso código pelo método/função main. No nosso curso utilizaremos a estrutura abaixo:

```
[3]: def main():  
    # comandos  
    pass  
  
    # -----  
    # A linha a seguir executa o programa  
    main()
```

1.1 Entendendo a estrutura

O python funciona por 'identação'. Executa primeiro o mais a esquerda possível de cima para baixo

```
[4]: # Escopo pai  
    # Escopo filho  
        # Escopo neto  
            # Quarto bisneto
```

def -> Define uma função

() -> Chama um função

: -> Declara o início de um novo nível de escopo

2 Variáveis

Uma variável é um nome que se refere a um valor.

2.1 Atribuição

Comando de atribuição cria uma nova variável e lhe dá um valor. Set ->Setando -> Atribuindo um valor

2.2 Motivação de uso

São usadas para guardarmos valores que serão usados mais tarde no programa

varivel = valor variavel = novo_valor variavel = novo_novo_valor
variavel

Exemplo:

```
[15]: primeiro_numero = 15  
      segundo_numero = 10  
      soma = primeiro_numero + segundo_numero  
      soma
```

[15]: 25

```
[23]: primeiro_numero = 20
```

```
[24]: segundo_numero = 25
```

```
[16]: soma
```

[16]: 25

```
[27]: soma = soma + ( primeiro_numero + segundo_numero )
```

```
[28]: soma
```

[28]: 190

```
[35]: soma = soma - 10  
      soma
```

[35]: 140

2.3 Exercício 2.1

Dado dois valores 'a' e 'b' calcule a soma

```
[41]: # soma = a + b  
      # onde a, b e soma são variáveis  
      a = 1  
      b = 4  
      soma = a + b  
      soma
```

[41]: 5

2.4 Exercício 2.2

Dado dois valores 'a' e 'b' calcule a multiplicação

```
[45]: a = 12
      b = 47
      multiplicacao = a + b
      multiplicacao
```

[45]: 59

Lembrete: Não colocar acentos nem caracteres especiais em nomes de variáveis
você # errado
voce # certo

2.5 Comandos get (recuperando o valor de uma variável)

Os comandos **get** chamam uma variável e retornam (recuperam) o seu valor
Ex:
variavel1 (set) = variavel2 (get)

```
[66]: multiplicacao
      soma
```

[66]: 5

Por padrão o **Jupyter** mostra o valor da última variável usada sem atribuição (comando get)

2.6 Comandos set (atribuindo o valor de uma variável)

Os comandos **set** atribuem um valor em uma variável
Ex:
variavel1 (set) = variavel2 (get)

```
[64]: multiplicacao = 59
      soma = 5
```

3 Métodos de saída de dados (output)

3.0.1 Usamos o método Print() para mostrar dados ao usuário

Nós enviamos um texto (string) por parâmetro para a função print()

3.1 Entendendo o vocabulário

3.1.1 String

É um tipo de variável que representa texto.
Todo texto deve estar entre aspas (duplas ou simples).
Ex:
'Texto' "Texto"

3.2 Parâmetro

São valores que vão vir a ser utilizados pela função que chamamos. Colocamos no meio dos parênteses da função

Ex:

```
* main()
* main("Texto",21)
* print("Texto que eu quero que mostre ao usuário")
```

```
[55]: def main():
        # comandos
        a = 1
        b = 4
        soma = a + b
        # Print = saída de dados
        print("A soma é: soma")

        # -----
        # A linha a seguir executa o programa
        main()
```

A soma é: soma

```
[56]: def main():
        # comandos
        a = 1
        b = 4
        soma = a + b
        # Print = saída de dados
        print("A soma é:",soma)

        # -----
        # A linha a seguir executa o programa
        main()
```

A soma é: 5

```
[60]: def main():
        # comandos
        a = 1
        b = 4
        soma = a + b
        # Print = saída de dados
        print("A soma de", a, "com", b, "é", soma)

        # -----
        # A linha a seguir executa o programa
        main()
```

A soma de 1 com 4 é 5

4 Entrada de dados (input)

4.1 Utilizamos o método input para receber dados do usuário

Atribuímos a uma variável o valor de um input.

O input aceita um texto como parâmetro para a descrição do que vai recebido.

```
[61]: # Soma qualquer valor
def main():
    # comandos
    a = input("Digite o seu primeiro número:")
    b = input("Digite o seu segundo número:")
    soma = a + b
    # Print = saída de dados
    print("A soma de", a, "com", b, "é", soma)

# -----
# A linha a seguir executa o programa
main()
```

Digite o seu primeiro número:10

Digite o seu segundo número:25

A soma de 10 com 25 é 1025

```
[63]: # Soma qualquer valor
def main():
    # comandos
    nome = input("Digite o seu primeiro nome:")
    sobrenome = input("Digite o seu sobrenome:")
    # Print = saída de dados
    print("Seu nome completo é:", nome, sobrenome)

# -----
# A linha a seguir executa o programa
main()
```

Digite o seu primeiro nome:Felipe

Digite o seu sobrenome:Cabrera

Seu nome completo é: Felipe Cabrera

4.1.1 Lembrete:

Diferente do get, onde o Jupyter mostra apenas o último, o print é nativo do Python e pode ser usado quantas vezes quiser!

```
[67]: print("a")
```

a

```
[68]: print("b")
```

b

```
[69]: print('a')
      print('b')
      print("C")
      print(2)
```

a

b

C

2

5 Tipos de variáveis

Tipo define o comportamento e o entendimento daquela variável

Ex: Podemos entender como sendo número ou como sendo um texto

- 23 é um texto ou número? - “23” é certamente um texto - ‘23’ também certamente é um texto - 23 sem nada é um numero - 23.0 também é um numero

```
[71]: "23" + "10"
```

```
[71]: '2310'
```

```
[72]: 23 + 10
```

```
[72]: 33
```

```
[73]: 23 + '10'
```

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-73-59d822eef1e9> in <module>
----> 1 23 + '10'
```

```
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

int: Inteiro (número inteiro)

str: String (texto)

float: Ponto flutuante (número com vírgula)

```
[74]: type(23)
```

```
[74]: int
```

```
[75]: type("23")
```

```
[75]: str
```

```
[76]: # Soma qualquer valor
def main():
    # comandos
    a = input("Digite o seu primeiro número:")
    b = input("Digite o seu segundo número:")
    soma = a + b
    # Print = saída de dados
    print("O tipo de a é", type(a))
    print("O tipo de b é", type(b))
    print("A soma de", a, "com", b, "é", soma)

# -----
# A linha a seguir executa o programa
main()
```

```
Digite o seu primeiro número:1
Digite o seu segundo número:4
O tipo de a é <class 'str'>
O tipo de b é <class 'str'>
A soma de 1 com 4 é 14
```

5.1 Convertendo tipos

Na programação chamamos essa conversão de *cast*

```
[77]: int("23")
```

```
[77]: 23
```

```
[78]: str(23)
```

```
[78]: '23'
```

```
[79]: # Soma qualquer valor
def main():
    # comandos
    a = int(input("Digite o seu primeiro número:"))
    b = int(input("Digite o seu segundo número:"))
    soma = a + b
    # Print = saída de dados
    print("O tipo de a é", type(a))
    print("O tipo de b é", type(b))
    print("A soma de", a, "com", b, "é", soma)

# -----
# A linha a seguir executa o programa
main()
```

```
Digite o seu primeiro número:10
Digite o seu segundo número:20
O tipo de a é <class 'int'>
O tipo de b é <class 'int'>
A soma de 10 com 20 é 30
```

5.2 Tentativa com número ‘quebrado’ (com vírgula)

```
[80]: # Soma qualquer valor
def main():
    # comandos
    a = int(input("Digite o seu primeiro número:"))
    b = int(input("Digite o seu segundo número:"))
    soma = a + b
    # Print = saída de dados
    print("O tipo de a é",type(a))
    print("O tipo de b é",type(b))
    print("A soma de", a, "com", b, "é", soma)

# -----
# A linha a seguir executa o programa
main()
```

```
Digite o seu primeiro número:23.5
```

```
-----
ValueError                                Traceback (most recent call last)
<ipython-input-80-79de51bbfb3f> in <module>
    12 # -----
    13 # A linha a seguir executa o programa
----> 14 main()

<ipython-input-80-79de51bbfb3f> in main()
     2 def main():
     3     # comandos
----> 4     a = int(input("Digite o seu primeiro número:"))
     5     b = int(input("Digite o seu segundo número:"))
     6     soma = a + b

ValueError: invalid literal for int() with base 10: '23.5'
```

```
[81]: type(23.5)
```

```
[81]: float
```

```
[84]: # Soma qualquer valor
def main():
```



```

# comandos
a = float(input("Digite o seu primeiro número:"))
b = float(input("Digite o seu segundo número:"))
soma = a + b
# Print = saída de dados
print("O tipo de a é",type(a))
print("O tipo de b é",type(b))
print("A soma de", a, "com", b, "é", soma)

# -----
# A linha a seguir executa o programa
main()

```

```

Digite o seu primeiro número:12.34
Digite o seu segundo número:56.78
O tipo de a é <class 'float'>
O tipo de b é <class 'float'>
A soma de 12.34 com 56.78 é 69.12

```

5.3 Exercício 2.3

Fazer uma função que multiplique dois números entrados pelo usuário

```

[87]: def main():
    primeiro_numero = float(input("Digite seu primeiro número:"))
    segundo_numero = float(input("Digite seu segundo número:"))
    multiplicacao = primeiro_numero * segundo_numero
    print(f"A multiplicação entre {primeiro_numero} e {segundo_numero} =
    ↳{multiplicacao}")
# -----
# A linha a seguir executa o programa
main()

```

```

Digite seu primeiro número:10
Digite seu segundo número5
A multiplicação entre 10.0 e 5.0 = 50.0

```

5.4 Exercício 2.4

Fazer uma função que divida dois números entrados pelo usuário

```

[88]: def main():
    primeiro_numero = float(input("Digite seu primeiro número:"))
    segundo_numero = float(input("Digite seu segundo número:"))
    divisao = primeiro_numero / segundo_numero
    print("A divisão entre {0} e {1} = {2}".
    ↳format(primeiro_numero,segundo_numero,divisao))
# -----
# A linha a seguir executa o programa

```

```
main()
```

```
Digite seu primeiro número:50
Digite seu segundo número10
A divisão entre 50.0 e 10.0 = 5.0
```

5.5 Exercício 2.5

Fazer uma função que subtraia quatro números entrados pelo usuário

```
[91]: def main():
    primeiro_numero = float(input("Digite seu primeiro número:"))
    segundo_numero = float(input("Digite seu segundo número"))
    terceiro_numero = float(input("Digite seu terceiro número"))
    quarto_numero = float(input("Digite seu quarto número"))
    subtração = primeiro_numero - segundo_numero - terceiro_numero -
    ↪quarto_numero
    print("A subtração de {} com {} com {} com {} é igual a {}".
    ↪format(primeiro_numero,segundo_numero,
    ↪terceiro_numero,quarto_numero,subtração))
# -----
# A linha a seguir executa o programa
main()
```

```
Digite seu primeiro número:10
Digite seu segundo número5
Digite seu terceiro número2
Digite seu quarto número-7
A subtração de 10.0 com 5.0 com 2.0 com -7.0 é igual a 10.0
```

6 Formatação de strings

6.1 Podemos formatar com o format

“string original com {} para indicar as posições do format onde vamos trocar na string”.format(variavel_para_formatar)

```
[109]: numero = 10
      letra = 'a'
      palavra = 'ByLearn'
```

```
[110]: "meu número vale {}, minha letra vale {}, minha palavra vale {}".
      ↪format(numero,letra,palavra)
```

```
[110]: 'meu número vale 10, minha letra vale a, minha palavra vale ByLearn'
```

6.2 Podemos formatar com o f-string

São strings começadas com f antes das aspas.

Nelas colocamos o valores das variaveis dentro das {}

```
[112]: f'meu número vale {numero}, minha letra vale {letra}, minha palavra vale_
↪{palavra}'
```

```
[112]: 'meu número vale 10, minha letra vale a, minha palavra vale ByLearn'
```

6.3 Formatando números

Podemos modificar a saídas dos números sem modificar os valores reais

Ex: Número 2 => Valor é 2. Saída => 2.00 => Valor ainda é 2

Número 2.5 => Valor é 2.5 Saída => 2 => Valor ainda é 2.5

```
[114]: numero = 2.57
print(numero)
```

2.57

Para inteiros uso o 'd' e para floats uso o 'f'

```
[124]: print('{:.2f}'.format(numero))
print('{:.1f}'.format(numero))
print('{:.0f}'.format(numero))
print(f'{numero:.2f}')
print(f'{numero:.1f}')
print(f'{numero:.0f}')
```

2.57

2.6

3

2.57

2.6

3

```
[118]: numero
```

```
[118]: 2.57
```

```
[119]: 17 / 5
```

```
[119]: 3.4
```

```
[120]: 10 / 3
```

```
[120]: 3.3333333333333335
```

```
[121]: print('{:.3f}'.format(10/3))
```

3.333

[]: