



Estrutura de Dados

Noções de Python

Tipos de dados

- **int**
 - valores inteiros, sem casas decimais.
- **float**
 - valores com casas decimais;
 - o python utiliza "." como separador decimal;
- **str (string)**
 - tipo composto por um conjunto de caracteres;
 - é possível acessar partes específicas através da indicação de uma posição específica ou conjunto de posições específicas;
 - uma string é analisada como um vetor de caracteres, onde cada caractere ocupa uma posição;
 - A numeração das posições ocupadas pela string começa pelo valor 0 (zero);
 - Possibilitam realizar comparações entre valores string (==, >, <, !=);
 - Não é possível mudar um caracter específico da string;

Tipos de dados

- Algumas funções:

<code>len(<string>)</code>	Retorna o tamanho (quantidade de caracteres) que compõem a string.
<code><string>[i1:i2]</code>	Retorna um trecho da String cujo índice começa em <i>i1</i> e termina em <i>i2 - 1</i> .
<code><string>[:i2]</code>	Retorna um trecho da String cujo índice começa em 0 e termina em <i>i2 - 1</i> .
<code><string>[i1:]</code>	Retorna um trecho da String cujo índice começa em <i>i1</i> e termina no final da string.
<code><string>[<posição>]</code>	Retorna o caracter que está na posição informada.
<code>find(<string>,<caracter>)</code>	Retorna a posição do caractere procurado, na string. Caso não seja encontrado é retornado o valor '-1'.
<code>lower()</code>	Retorna uma cópia da string em letras minúsculas.
<code>upper()</code>	Retorna uma cópia da string em letras maiúsculas.

Conversão de tipos

- É possível converter valores para outros tipos caso esse valor seja compatível com o novo tipo;
- Caso a conversão não seja possível, um erro é gerado;

int(<valor>)	Converte o valor para inteiro.
float(<valor>)	Converte o valor para real (float).
string(<valor>)	Converte o valor para string.

Tipos de dados

```
# SLIDE - EXEMPLO 01
anoAtual = 2021
print("obtendo um texto qualquer",end="\n")
print()
nome = input("Qual o seu nome completo? ")
print()
anoNascimento = int(input("Qual o ano do seu nascimento (YYYY)? "))
print()

print("O nome tem ",len(nome)," caracteres.",end="\n")
print("O primeiro caracter é ",nome[0],end="\n")
print("O último caracter é ",(nome[len(nome) - 1]),end="\n")
print("O nome que você digitou foi ",nome.upper(),end="\n")
print("A sua idade é ",anoAtual-anoNascimento," anos.",end="\n")
texto = input("Qual texto deseja procurar neste nome? ")
if nome.find(texto) != -1:
    print("Esse texto existe no nome. Está na posição ",nome.find(texto))
else:
    print("Esse texto não existe no nome.")
print()
```

Operadores matemáticos

Operador	Função	Aplicação
+	Soma	$c = a + b$
-	Subtração	$x = y - z$
*	Multiplicação	$p = n * 5$
/	Divisão	$a = b / c$
**	Potenciação	$q = a ** 2$
//	Divisão inteira	$d = a // 2$
%	Resto da divisão	$r = a \% b$

Expressões booleanas

Operador	Função	Aplicação
==	Igual	if a == b:
!=	Diferente	if a != b:
>	Maior que	if a > b:
<	Menor que	if a < b:
>=	Maior igual	if a >= b:
<=	Menor igual	if a <= b:

Operadores lógicos

Operador	Função	Aplicação
and	E	if a == b and b == c:
or	Ou	if a == b or a == c::
not	Não	if not (a > b):

Entrada de dados

- Recebe um valor do teclado e atribui a uma variável;
- O valor lido é sempre uma string;
- O valor lido deve ser convertido para o formato necessário usado pelo programa;

Sintaxe:

```
...  
<variável> = input(<texto>)  
...
```

Exemplo:

```
...  
Nome = input('Qual o seu nome? ')  
Idade = int(input('Informe a sua idade: '))  
...
```

Entrada de dados

Exemplo:

```
print()
print("Cálculo da média de um aluno")
print()
nome = input("Nome do aluno: ")
nota1 = float(input("nota 1o. bimestre: "))
nota2 = float(input("nota 2o. bimestre: "))
nota3 = float(input("nota 3o. bimestre: "))
nota4 = float(input("nota 4o. bimestre: "))
media = (nota1 + nota2 + nota3 + nota4)/4
print()
print("Desempenho do aluno")
print("=====")
print("Nome.....: ", nome.upper(), end="\n")
print("Notas: \n")
print("1o. bimestre: ", nota1, end="\n")
print("2o. bimestre: ", nota2, end="\n")
print("3o. bimestre: ", nota3, end="\n")
print("4o. bimestre: ", nota4, end="\n")
print("Média final.: ", media, end="\n")
```

Cálculo da média de um aluno

```
Nome do aluno: Leonel Carlos de Alcântara Abreu
nota 1o. bimestre: 8.9
nota 2o. bimestre: 10
nota 3o. bimestre: 9.2
nota 4o. bimestre: 9.3
```

Desempenho do aluno

=====

```
Nome.....: LEONEL CARLOS DE ALCÂNTARA ABREU
Notas:
```

```
1o. bimestre: 8.9
2o. bimestre: 10.0
3o. bimestre: 9.2
4o. bimestre: 9.3
Média final.: 9.35
```

Saída de dados

- A função *print* exibe o texto passado como parâmetro;
- O texto pode ser a concatenação de vários valores;
- Os textos a serem concatenados devem ser separados por “,”(vírgula), ou “+” (mais);
- Quando nenhum valor é passado como parâmetro uma linha em branco é exibida;

Exemplo:

```
...
nome = input('Qual o seu nome? ')
idade = input('Informe a sua idade: ')
print()
print(nome, 'tem', idade, 'anos.')
...
```

Entrada de dados

Exemplo:

```
print()
print("Cálculo da média de um aluno")
print()
nome = input("Nome do aluno: ")
nota1 = float(input("nota 1o. bimestre: "))
nota2 = float(input("nota 2o. bimestre: "))
nota3 = float(input("nota 3o. bimestre: "))
nota4 = float(input("nota 4o. bimestre: "))
media = (nota1 + nota2 + nota3 + nota4)/4
print()
print("Desempenho do aluno")
print("=====")
print("Nome.....: ", nome.upper(), end="\n")
print("Notas: \n")
print("1o. bimestre: ", nota1, end="\n")
print("2o. bimestre: ", nota2, end="\n")
print("3o. bimestre: ", nota3, end="\n")
print("4o. bimestre: ", nota4, end="\n")
print("Média final.: ", media, end="\n")
```

Cálculo da média de um aluno

```
Nome do aluno: Leonel Carlos de Alcântara Abreu
nota 1o. bimestre: 8.9
nota 2o. bimestre: 10
nota 3o. bimestre: 9.2
nota 4o. bimestre: 9.3
```

Desempenho do aluno

=====

```
Nome.....: LEONEL CARLOS DE ALCÂNTARA ABREU
Notas:
```

```
1o. bimestre: 8.9
2o. bimestre: 10.0
3o. bimestre: 9.2
4o. bimestre: 9.3
Média final.: 9.35
```

Estrutura de decisão - if

- Sintaxe 1:

```
...  
if <condição>:           # SE A CONDIÇÃO FOR VERDADEIRA OS COMANDOS SERÃO EXECUTADOS  
    comando 1  
    comando 2  
    ...  
    comando n  
...
```

- Sintaxe 2:

```
...  
if <condição>:           # SE A CONDIÇÃO FOR VERDADEIRA ESSES COMANDOS SERÃO EXECUTADOS  
    comando 1  
    ...  
    comando n  
else:                   # SE A CONDIÇÃO FOR FALSA ESSES COMANDOS SERÃO EXECUTADOS  
    comando 1  
    ...  
    comando n
```

Estrutura de decisão - if

- Sintaxe 3:

```
...  
if <condição>:      # SE A CONDIÇÃO FOR VERDADEIRA OS COMANDOS SERÃO EXECUTADOS  
    comandos  
elif <condição>:    # SENÃO, SE ESSA CONDIÇÃO FOR VERDADEIRA OS COMANDOS SERÃO EXECUTADOS  
    comandos  
else:               # SE NENHUMA DAS CONDIÇÕES FOREM VERDADEIRAS, ESSES SÃO EXECUTADOS  
    comandos
```


Estrutura de decisão - if

- Exemplo::

```
print()
print("Cálculo da média de um aluno")
print()
nome = input("Nome do aluno: ")
nota1 = float(input("nota 1o. bimestre: "))
nota2 = float(input("nota 2o. bimestre: "))
nota3 = float(input("nota 3o. bimestre: "))
nota4 = float(input("nota 4o. bimestre: "))
media = (nota1 + nota2 + nota3 + nota4)/4
print()
print("Desempenho do aluno")
print("=====")
print("Nome.....: ",nome.upper(),end="\n")
print("Notas:")
print("1o. bimestre: ",nota1,end="\n")
print("2o. bimestre: ",nota2,end="\n")
print("3o. bimestre: ",nota3,end="\n")
print("4o. bimestre: ",nota4,end="\n")
print("Média final.: ",media,end="\n")
situacao = "Reprovado"
if media >= 6:
    situacao = "Aprovado"
elif (media >= 4):
    situacao = "Recuperação"
else:
    situacao = "Reprovado"
print("Situação.....: ",situacao.upper(),end="\n")
if media >= 6:
    mensagem = "Parabéns pela sua aprovação!"
else:
    if (media >= 4 and media < 6):
        mensagem = "Se dedique mais aos estudos."
    else:
        mensagem = "Estude mais no ano que vem."
print("Mensagem: ",mensagem)
```

Estrutura de decisão - if

- Exemplo::

Cálculo da média de um aluno

```
Nome do aluno: Ludimila Matins Andrade
nota 1o. bimestre: 10
nota 2o. bimestre: 10
nota 3o. bimestre: 9.8
nota 4o. bimestre: 9.7
```

Desempenho do aluno

=====

```
Nome.....: LUDIMILA MATINS ANDRADE
Notas:
```

```
1o. bimestre: 10.0
2o. bimestre: 10.0
3o. bimestre: 9.8
4o. bimestre: 9.7
```

```
Média final.: 9.875
```

```
Situação.....: APROVADO
```

```
Mensagem: Parabéns pela sua aprovação
```

Cálculo da média de um aluno

```
Nome do aluno: Nathaniel Martis Vieira
nota 1o. bimestre: 5.5
nota 2o. bimestre: 7
nota 3o. bimestre: 3.2
nota 4o. bimestre: 4.2
```

Desempenho do aluno

=====

```
Nome.....: NATHANIEL MARTIS VIEIRA
Notas:
```

```
1o. bimestre: 5.5
2o. bimestre: 7.0
3o. bimestre: 3.2
4o. bimestre: 4.2
```

```
Média final.: 4.975
```

```
Situação.....: RECUPERAÇÃO
```

```
Mensagem: Se dedique mais aos estudos.
```

Cálculo da média de um aluno

```
Nome do aluno: Altamira Miranda de Carvalho
nota 1o. bimestre: 2.5
nota 2o. bimestre: 0
nota 3o. bimestre: 1.5
nota 4o. bimestre: 3.5
```

Desempenho do aluno

=====

```
Nome.....: ALTAMIRA MIRANDA DE CARVALHO
Notas:
```

```
1o. bimestre: 2.5
2o. bimestre: 0.0
3o. bimestre: 1.5
4o. bimestre: 3.5
```

```
Média final.: 1.875
```

```
Situação.....: REPROVADO
```

```
Mensagem: Estude mais no ano que vem.
```


Estrutura de Repetição - for

- Quando se sabe quantas vezes o bloco de comandos será executado;

Sintaxe 1:

```
...  
for <contador> in (<intervalo>):  
    <comandos>  
...
```

Sintaxe 2:

```
...  
for <contador> in (1,2,3,4,5):  
    <comandos>  
...
```

Sintaxe 3:

```
...  
for <contador> in ['João', 'Antônia', Augusto]:  
    <comandos>  
...
```

Sintaxe 4:

```
...  
// contagem de 0 a 50 com incremento de 2.  
for <contador> in (0,50,2):  
    <comandos>  
...
```

Sintaxe 5:

```
...  
// contagem de 10 a 0 com decremento de 1.  
for <contador> in (10,0,1):  
    <comandos>  
...
```

Sintaxe 6:

```
...  
// contagem de 1 a 50 com incremento de 1.  
for <contador> in (1,50):  
    <comandos>  
...
```

Sintaxe 7:

```
...  
for <contador> in (100):          // contagem de  
0 a 100 com incremento de 1.  
    <comandos>  
...
```

Estrutura de Repetição - while

- Quando não se sabe quantas vezes o bloco de comandos será executado (estrutura *while*);
- Enquanto a condição for verdadeira o bloco de comandos será executado;

Sintaxe:

```
...  
while <condição>:  
    <comandos>  
...
```

Comandos: break, continue, pass

- Agregam funcionalidades às estruturas de repetição;

break

- Interrompe a execução da repetição;
- O fluxo de execução é direcionado para a primeira linha de comando após o bloco de comandos;

continue

- Retorna ao início da estrutura sem que os demais comandos dessa estrutura sejam executados, iniciando uma nova iteração;

pass

- É utilizado quando a estrutura é implementada mas não é desejada a execução das suas linhas de comando;

Comandos: break, continue, pass

Exemplo:

```
...  
while <condição>:  
    <comandos>  
    if <condição>:  
        break  
    else:  
        continue  
    <comandos>  
...
```

```
...  
while <condição>:  
    pass  
...
```



Funções

- As funções são subprogramas (módulos) que podem ser executados, sempre que necessário;
- Evita a repetição de código nos programas;
- Podem ser importadas para ser utilizadas por vários programas;
- Podem retornar valores, usando a palavra reservada **return**;
- O comando **return** encerra a execução da função;

Sintaxe:

```
def <nome>([<parâmetros>]):  
    <comandos>
```

Exemplo 1:

```
def pularLinha(quantidade):  
    for contador in (1,quantidade):  
        print()
```

Funções

Exemplo 2:

```
def media(x1,x2):  
    mediaCalculada = (x1 + x2) / 2  
    return mediaCalculada  
...  
mediaAluno = media(nota1,nota2)  
...
```

Exemplo 3:

```
if __name__ == '__main__':  
    <comandos>
```

Funções

Programa Exemplo:

```
def maior(a,b):
    maiorValor = a
    if (maiorValor) < b:
        maiorValor = b
    return maiorValor

def par(numero):
    numeroPar = False
    if (numero % 2 == 0):
        numeroPar = True
    return numeroPar

#if __name__ == '__main__':
print("Verificando 2 números informados")
print()
valor1 = int(input("Informe um número: "))
valor2 = int(input("Informe outro número: "))
print()
print("Informações sobre os números informados",end="\n")
print("Primeiro número: ",valor1)
parImpar = 'Ímpar'
if par(valor1):
    parImpar = 'Par'
print('O número ',valor1,'é: ',parImpar)
print("Segundo número: ",valor2)
parImpar = 'Ímpar'
if par(valor2):
    parImpar = 'Par'
print('O número ',valor2,'é: ',parImpar)
maiorNumero = maior(valor1,valor2)
print('O maior número entre ',valor1,' e ',valor2,' é: ',maiorNumero)
```

Módulos

- Permite criar subprogramas que podem ser usados por outros programas;
- Permite a reutilização de código;
- Funções são colocadas em arquivos exclusivos (módulos) e podem ser usadas por quaisquer outros programas;
- A utilização desse módulo se dá através da sua importação pelos programas ou por outros módulos que dele necessitar;

Importando

- A importação do módulo é feita através da palavra reservada *from...import*;

Sintaxe:

```
import <nome do módulo>
```

Sintaxe:

```
from <nome do módulo> import <lista das funções desejadas>
```

Sintaxe:

```
from <nome do módulo> import *
```


Módulos

Exemplo:

```
def maior(a,b):  
    maiorValor = a  
    if (maiorValor) < b:  
        maiorValor = b  
    return maiorValor  
  
def par(numero):  
    numeroPar = False  
    if (numero % 2 == 0):  
        numeroPar = True  
    return numeroPar  
  
def tipoInteiro(numero):  
    tipoInteiroNumero = 'NULO'  
    if numero < 0:  
        tipoInteiroNumero = 'NEGATIVO'  
    elif numero > 0:  
        tipoInteiroNumero = 'POSITIVO'  
    return tipoInteiroNumero  
  
def tracejar(qtd):  
    for numero in range(qtd):  
        print('-',end='')  
    return
```

Módulos

Exemplo:

```
#from exemploModulo02 import par,maior,tipoInteiro,tracejar
from exemploModulo01 import par,maior
from exemploModulo02 import tipoInteiro,tracejar

valor1 = 0      # PRIMEIRO VALOR INFORMADO
valor2 = 0      # SEGUNDO VALOR INFORMADO

tracejar(50)
print()
print("Verificando 2 números informados",end='\n')
tracejar(50)
print()
valor1 = int(input("Informe um número: "))
valor2 = int(input("Informe outro número: "))
tracejar(50)
print()
print()
print("Informações sobre os números informados",end="\n")
tracejar(50)
print()
print("Primeiro número: ",valor1)
parImpar = 'Ímpar'
if par(valor1):
    parImpar = 'Par'
print('O número ',valor1,'é: ',parImpar)
print('O número ',valor1,' é: ',tipoInteiro(valor1))
tracejar(50)
print()
print("Segundo número: ",valor2)
parImpar = 'Ímpar'
if par(valor2):
    parImpar = 'Par'
print('O número ',valor2,'é: ',parImpar)
print('O número ',valor2,' é: ',tipoInteiro(valor2))
maiorNumero = maior(valor1,valor2)
tracejar(50)
print()
print('O maior número entre ',valor1,' e ',valor2,' é: ',maiorNumero)
tracejar(50)
print()
```

Classes

- A classe define o objeto;
- As declarações podem aparecer em qualquer parte do código, mas recomenda-se deixá-las próximas ao começo do programa (após o **import**);
- A definição da classe se dá com a palavra reservada **class**;

```
class <nome da classe>:  
    <corpo da classe>
```

- Um objeto é uma estrutura composta no Python, instanciado no programa;
- Para instanciar o objeto chamamos a função com o mesmo nome da classe;

```
<objeto> = <nome da classe>([<parâmetros>])
```

Classes

- Quando um objeto é criado o construtor (**`__init__([<parâmetros>])`**) dessa classe é executado;
- A retirada desses objetos da memória é feita por um coletor de lixo, chamado de **Garbage Collector**;
- Ao apagar um objeto, o método especial **`__done__()`** é executado;

Classes

```
class retangulo:
    lado1 = 1.0
    lado2 = 1.0

    def __init__(self,lado1=1,lado2=1):
        self.lado1 = lado1
        self.lado2 = lado2

    def setLado1(self,lado1):
        self.lado1 = self.validarMedidaLado(lado1)
        return

    def getLado1(self):
        return self.lado1

    def setLado2(self, lado2):
        self.lado2 = self.validarMedidaLado(lado2)
        return

    def getLado2(self):
        return self.lado2

    def validarMedidaLado(self,medida):
        if isinstance(medida,float) and medida >0:
            return medida
        else:
            return 1

    def perimetro(self):
        return ((self.getLado1() * 2) + (self.getLado2() * 2))

    def area(self):
        return ((self.getLado1() * self.getLado2()))

    def informacoes(self):
        print("Dados do Retângulo:",end="\n")
        print("Lado 1: {:.2f}".format(self.getLado1()))
        print("Lado 2: {:.2f}".format(self.getLado2()))
        print("Perímetro: {:.2f}".format(self.perimetro()))
        print(f"Área: {:.2f}".format(self.area()))
        return

if __name__ == "__main__":
    meuRetangulo = retangulo(3,5)
    print()
    meuRetangulo.informacoes()

print()
print("Trabalhando com um novo retângulo",end="\n")
meuRetangulo.setLado1(float(input("Informe a medida do lado 1: ")))
meuRetangulo.setLado2(float(input("Informe a medida do lado 2: ")))
print()
meuRetangulo.informacoes()
```


Classes

```
class retangulo:
    lado1 = 1.0
    lado2 = 1.0

    def __init__(self, lado1=1, lado2=1):
        self.lado1 = lado1
        self.lado2 = lado2

    def setLado1(self, lado1):
        self.lado1 = self.validarMedidaLado(lado1)
        return

    def getLado1(self):
        return self.lado1

    def setLado2(self, lado2):
        self.lado2 = self.validarMedidaLado(lado2)
        return

    def getLado2(self):
        return self.lado2

    def validarMedidaLado(self, medida):
        if isinstance(medida, float) and medida > 0:
            return medida
        else:
            return 1

    def perimetro(self):
        return ((self.getLado1() * 2) + (self.getLado2() * 2))

    def area(self):
        return ((self.getLado1() * self.getLado2()))

    def informacoes(self):
        print("Dados do Retângulo:", end="\n")
        print("Lado 1: {0:.2f}".format(self.getLado1()))
        print("Lado 2: {0:.2f}".format(self.getLado2()))
        print("Perímetro: {0:.2f}".format(self.perimetro()))
        print("Area: {0:.2f}".format(self.area()))
        return
```

Classes

```
from classRetangulo2 import retangulo

meuRetangulo = retangulo(3,5)
if __name__ == "__main__":
    meuRetangulo = retangulo(3,5)
    print()
    meuRetangulo.informacoes()

print()
print("Trabalhando com um novo retângulo",end="\n")
meuRetangulo.setLado1(float(input("Informe a medida do lado 1: ")))
meuRetangulo.setLado2(float(input("Informe a medida do lado 2: ")))
print()
meuRetangulo.informacoes()
```