

AULA — 06

TIPOS PRIMITIVOS E SAÍDA DE DADOS

Data: 17/07/2023

Profº Guanabara

Teoria

Tipos primitivos básicos:

int: 7 -4 0 9875

float: 4.5 0.076 -15.223 0.7

bool: True False

str: 'Olá' '7.5' ' '

Forma alternativa de utilizar o print

1 versão

```
Print('A soma vale', s)
```

2 versão

```
print('A soma vale { }'.format(s))
```

Prática

Ex:

```
n1= input(digite um valor)
print(type(n1))
```

```
n1= int(input(digite um valor))
n2= int(input(digite um valor))
s = n1+n2
print('A soma vale', s)
```

```
-----  
n1= int(input('digite um valor'))  
n2= int(input('digite um valor'))  
s = n1+n2  
print('A soma entre { } e { } vale { }'.format(n1, n2, s))  
-----
```

```
n1= input('Digite algo')  
print(n.is)
```

Desafios

Desafio 003:

Crie um programa que leia dois números e mostre a soma entre eles.

Desafio 004:

Faça um programa que leia algo pelo teclado e mostre na tela o seu tipo primitivo e todas as informações possíveis sobre ele

AULA — 07

OPERADORES ARITMÉTICOS

Data: 18/07/2023

Prof° Guanabara

Teoria

Operadores Aritméticos:

+ Adição	** Potência
- Subtração	// Divisão inteira
* Multiplicação	% Resto da divisão
/ Divisão	

Nota:

= (Recebe / Atribuição)

== (Igual normal)

Ordem de Precedência:

1 ()

2 **

3 * / // %

4 + -

Prática

Notas:

- 4^{**3} pode ser feito da forma: `pow(4, 3)`

- Raiz quadrada, exemplo: $81^{**(1/2)}$

Calcular raiz quadrada de um número é a mesma coisa que criar a potência dele por meio.

- Raiz cubica: $127^{**(1/3)}$

Notas:

prints que podem ser uteis:

1) Printar ===== (Sinal de igual vezes vinte)

- print('=' * 20)

2) Centraliza a mensagem

- print('prazer em te conhecer {:^20}'.format(nome))

3)

- print('prazer em te conhecer {::=^20}'.format(nome))

Para quebrar uma linha você usa: \n

Para não quebrar no final , end=' '

ex: print ('A soma é {}, \n o Produto é {} e a divisão é {:.3f}'.format(s, m, d), end=' ')

Desafios

Desafio 005:

Faça um programa que leia um número inteiro e mostre na tela o seu sucessor e seu antecessor.

Desafio 006:

Crie um algoritmo que leia um número e mostre o seu dobro, triplo e raiz quadrada.

Desafio 007:

Desenvolva um programa que leia as duas notas de um aluno, calcule e mostre a sua média.

Desafio 008:

Escreva um programa que leia um valor em metros e o exiba convertidos em centímetros e milímetros.

Desafio 009:

Faça um programa que leia um número inteiro qualquer e mostre na tela a sua tabuada.

Desafio 0010:

Crie um programa que leia quanto dinheiro uma pessoa tem na carteira e mostre quantos dólares ela pode comprar.

Considere

US\$ 1,00 = 3.27

Desafio 011:

Faça um programa que leia a largura e a altura de uma parede em metros, calcule a sua área e a quantidade de tinta necessária para pintá-la, sabendo que cada litro de tinta, pinta uma área de 2m².

Desafio 012:

Faça um algoritmo que leia o preço de um produto e mostre seu novo preço, com 5% de desconto.

Desafio 013:

Faça um algoritmo que leia o salário de um funcionário e mostre seu novo salário, com 15% de aumento.

AULA — 08

UTILIZANDO MODULOS

Data: 25/07/2023

Prof° Guanabara

Teoria

Fazer importações:

Ex.:

- import bebida

- import doce

E se eu não quiser importar tudo?

Caso eu queira importar algo específico da biblioteca doce eu faria o seguinte:

- from doce import pudim

Biblioteca padrão, mas tem que importar: (math)

A biblioteca math traz algumas funcionalidades extras da matemática

Funcionalidades de math:

ceil: arredonda para cima

floor: arredonda para baixo

trunc: elimina da virgula para frente sem fazer arredondamento nenhum

pow: potência (funciona de forma similar aos dois asteriscos)

sqrt: calcular raiz quadrada

factorial: cálculo de fatorial

Import math

(importa tudo)

```
from math import sqrt
(importa apenas a funcionalidade para raiz quadrada)
```

e se eu quiser importar 2 coisas ao mesmo tempo?

```
from math import sqrt, ceil
(adicione a virgula)
```

Prática

Import Math

Ex.: 1a

```
Import math
num = int(input("Digite um número: "))
raiz = math.sqrt(num)

print("A raiz de {} é igual a {}".format(num, raiz))
```

Ex.: 1b (mesmo exemplo, porém adicionando o arredondamento direto no print)

```
Import math
num = int(input("Digite um número: "))
raiz = math.sqrt(num)

print("A raiz de {} é igual a {}".format(num, math.ceil(raiz)))
```

Nota

Como ver o que posso importar?

Vc vai em python.org

Vai em docs

Escolha a versão do python

Clique em library Reference

Import Random

Ex.: 1a) importando números aleatórios

Import Random

```
num = random.random ()  
print(num)
```

nota: ele vai gerar números aleatoriamente

Ex.: 1b) importando números aleatórios **inteiros**

Import Random

```
num = random.randint ()  
print(num)
```

nota: ele vai gerar números inteiros aleatoriamente

Ex.: 1b) importando números aleatórios **inteiros entre 1 e 10**

Import Random

```
num = random.randint (1, 10)  
print(num)
```

nota: adicionando (1, 10) pois assim ele vai gerar números aleatórios entre 1 e 10

Nota:

se você escrever import e segurar o control e apertar espaço vai aparecer uma lista das coisas que você pode importar por padrão

Nota: Você pode importar coisas de forma separada que não estão no python por padrão, para isso você deve:

Ir em python.org

Pypy

Assim terá varias coisas para ser importadas separadamente

Desafios

Desafio 016:

Crie um programa que leia um número Real qualquer pelo teclado e mostre na tela a sua porção inteira.

Ex.: Digite um número: 6.127

O número 6.127 tem a parte inteira 6.

Desafio 017:

.Faça um programa que leia o comprimento do cateto oposto e do cateto adjacente de um triângulo retângulo, calcule e mostre o comprimento da hipotenusa

Desafio 018:

Faça um programa que leia um ângulo qualquer e mostre na tela o valor do seno, cosseno e tangente desse ângulo.

Desafio 019:

Um professor quer sortear um dos seus quatro alunos para apagar o quadro. Faça um programa que ajude ele, lendo o nome deles e escrevendo o nome do escolhido.

Desafio 020:

O mesmo professor do desafio anterior quer sortear a ordem de apresentação de trabalhos dos alunos. Faça um programa que leia o nome dos quatro alunos e mostre a ordem sorteada.

Desafio 0021:

Faça um programa em python que abra e reproduza o áudio de um arquivo MP3.

AULA — 09

MANIPULANDO TEXTO

Data: 04/08/2023

Prof° Gustavo Guanabara

Teoria

Cadeia de texto

Para o Python, toda cadeia de texto está entre aspas simples e aspas duplas

Forma de atribuir uma string dentro de uma variável;

frase = 'Curso em vídeo'

Quando esse tipo de atribuição é feito, o python coloca esses dados na memória do computador, mas essa frase não vai inteira, o que ele vai fazer é criar mini espaços dentro da memória dentro do computador e dentro de cada mini espaço ele vai colocar cada uma das letras;

Ex.:

[C][u][r][s][o][][e][m][][v][í][d][e][o][][p][y][t][h][o][n]
[0][1][2][3][4][5][6][7][8][9][10][11][12][13][14][15][16][17][18][19][20]

Perceba que entre "curso" e "em" tem um espaço vazio e esse espaço ocupa esse mini espaço, e cada um desses mini espaços vai receber um índice que é um número sequencial começando de zero e indo até o número de letras que for necessário,

Fatiamento

Fatiar uma string é conseguir pegar pedaços dela

Exemplo de fatiamento;

Frase[9]

o símbolo de colchete é o identificador de uma estrutura de dados do python chamada lista, então se eu mandá-lo escrever print frase, ele vai mandar escrever a frase completa que seria

‘Curso em vídeo Python’, mas se eu mandar escrever frase[9] ele vai conseguir identificar dentro da cadeia de caracteres somente o caractere 9 que no caso é o décimo caractere já que toda string começa pelo zero, no caso a letra 9 seria o [V].

Outra forma de fatiar é a seguinte;

frase[9:13]

de forma simples o 9 é o [V] e o 13 é [o]

dessa forma ele vai começar no 9 e vai até o 13, porém o 13 será excluído imprimindo ‘Vide’ se quiser imprimir até a letra o você terá que colocar até o 14, é sempre 1 a menos no final

Outra forma de fatiamento seria;

Frase[9:21]

Você pode achar que vai dar erro pelo fato da frase conter apenas 20 caracteres, mas muito pelo contrário se colocar até o 21 assim como no exemplo anterior o último caractere será ignorado

Outra forma;

Frase[9:21:2]

Dessa forma ele vai seguir começando pelo 9 e finalizando no 21, porém ele irá seguir pulando de 2 em 2

A frase provavelmente ficaria dessa forma;

[V][d][o][P][t][o]

Outra forma;

Frase[:5]

Sabendo que antes dos dois pontos : é onde ele vai começar e depois é onde vai terminar, quando não se coloca nada antes ele naturalmente começa pelo caractere zero

Outra forma;

Frase[15:]

A frase começa no 15 mas como não foi indicado o final o Python vai levar a frase até o final da string

Outra forma;

Frase[9::3]

Levando em conta os exemplos anteriores dessa mesma forma ele vai começar a frase pelo caractere 9 e vai até o final já que não foi especificado depois dos dois pontos, mas ele fara pulando de 3 em 3.

Análise

Analisar uma string é obter informações sobre ela como, qual o tamanho dela, qual letra ela começa ou termina, qual a primeira palavra inteira.

`len(frase)`

a primeira coisa que vamos fazer é utilizar a função “len” len vem de lenf que significa comprimento, então quando usamos essa função ela vai nos retornar o comprimento da frase, ou seja quantos espaços ou micro espaços ela tem que no caso seria 21 caracteres

`frase.count('o')`

essa função vai basicamente pedir para que o programa conte quantas vezes aparece a letra o na variável frase, lembrando que ele está pedindo apenas a letra o minúscula, sendo assim as letras O maiúsculas não serão contadas já que o python diferencia as minúsculas das maiúsculas

uma outra forma de usa esse comando seria;

`frase.count('o'.0,13)`

isso vai basicamente fazer a contagem junto do fatiamento, o programa vai considerar do 0 até o 13 todos os o, vale lembrar que o caractere o no espaço 13 não será considerado já que o último espaço é desconsiderado quando fatiado

Outra funcionalidade de analize sria a seguinte;

`frase.find('deo')`

isso vai me dizer quanta vezes ele encontrou na frase 'deo', nesse caso ele vai mostrar em que momento começa 'deo' que seria na posição 11

`frase.find('android')`

Dentro da string não tem string android, se você coloca dentro do find uma string que simplesmente não existe, ele te retorna o valor -1

Operador;

`'Curso' in frase`

Isso basicamente está dizendo 'Existe a palavra Curso em frase', como existe essa string na variável frase ele vai te retornar True que é verdadeiro.

Transformação

Por via de regra, uma lista de string é imutável, não sendo possível mexer nela, mas é possível mudar ela através dos métodos

`frase.replace('Python', Android)`

replace é trocar ou reposicionar no caso o programa vai procurar por 'Python' e substituir por 'Android'

Método

`frase.upper()`

isso vai fazer tudo que não estiver em maiúsculo ficar em maiúsculo

Método

`frase.lower()`

esse de forma contraria ao Upper vai fazer tudo ficar em minúsculo

outra funcionalidade;

`frase.capitalize()`

Isso vai jogar todos os caracteres para minúsculo e só o primeiro caractere vai ficar em maiúsculo

Outra funcionalidade;

`frase.title()`

De forma parecida com o Capitalize o tilde vai analisar quantas palavras tem a string, ele faz isso pela posição dos espaços, então onde tem espaços ele faz uma quebra de palavra e vai fazer o capitalize palavra por palavra.

Outra funcionalidade;

frase.strip()

O strip vai remover todos os espaços inúteis no início e no final da string, tome como exemplo alguém leigo em um site e essa pessoa antes de digitar seu nome aperte espaço ou aperte no final do nome o espaço, ficando esse espaço excedente sem motivo algum, essa funcionalidade vai excluir esse espaço, porém somente o espaço do começo e do final da string.

frase.rstrip()

de forma similar ao strip o r na frente que é de right que é direita, dessa forma ele vai remover apenas os últimos espaços que no caso fica à direita.

frase.lstrip()

de forma similar ao rstrip o l na frente que é de left que é esquerda, dessa forma ele vai remover apenas os espaços do início que no caso fica à esquerda.

Divisão

frase.split()

basicamente vai ocorrer uma divisão em sua string considerando os espaços, ele vai pegar onde tem espaço e vai criar uma divisão

Ficaria assim de certa forma;

```
[C][u][r][s][o] [e][m] [V][i][d][e][o] [P][y][t][h][o][n]  
[0][1][2][3][4] [0][1] [0][1][2][3][4] [0][1][2][3][4][5]
```

Como pode ver a numeração sofreu alteração depois da divisão fazendo cada palavra receber indexação nova e cada uma dessas palavras é colocada dentro de uma outra lista, então o split basicamente cria uma lista com todas as palavras de uma cadeia de caracteres

Ele vai separar essas palavras que foram separadas e essas palavras vão ter numerações

```
[Curso] [em] [Vídeo] [Python]  
[ 0 ][1][ 2 ][ 3 ]
```

Junção

se eu tenho nomes separados em listas eu posso utilizar o;

```
'-'.join(frase)
```

Que serve para juntar uma coisa na outra, percebe-se também que tem '-' na frente do Join, isso significa que vai juntar todos os elementos de frase e vai usar esse separador '-' que ia gerar uma string única com a seguinte configuração;

Curso-em-Vídeo-Python

Caso queira que tenha um espaço no lugar do traço é só colocar o espaço entre o '' na frente do join

Prática

Forma de imprimir um texto grande

Exemplo

Olá, seja bem-vindo a minha humilde casa, entre e sente-se no sofá, vou preparar algo para que possamos comer, enquanto isso por favor, tome uma xicara de chá

```
print("""Olá, seja bem-vindo a minha humilde casa, entre e sente-se no  
sofá, vou preparar algo para que possamos comer, enquanto isso por favor,  
tome uma xicara de chá""")
```

Desafios

Desafio 022:

Crie um programa que leia o nome completo de uma pessoa e mostre:

- O nome com todas as letras Maiúsculas.
- O nome com todas minúsculas.
- Quantas letras ao todo (sem considerar espaços).
- Quantas letras tem o primeiro nome.

```
Nome = str(input("Digite seu nome completo: "))
print("Seu nome em maiúsculas é {}".format(nome.upper()))
print("Seu nome em minúsculas é {}".format(nome.lower()))
print("Seu nome tem ao todo {} letras".format(len(nome) - nome.count(' ')))
print("Seu primeiro nome tem {} letras".format(nome.find(' ')))
```

Desafio 023:

Faça um programa que leia um número de 0 a 9999 e mostre na tela cada um dos dígitos separados.

Ex: Digite um número: 1834

unidade: 4
dezena: 3
centena: 8
milhar: 1

```
num = int(input('Informe um número: '))
u = num // 1 % 10
d = num // 10 % 10
c = num // 100 % 10
m = num // 1000 % 10
```

```
print("Analisando o número {}".format(num))
print("Unidade: {}".format(u))
print("Dezena: {}".format(d))
print("Centena: {}".format(c))
print("Milhar: {}".format(m))
```

Desafio 024:

Crie um programa que leia o nome de uma cidade e diga se ela começa ou não com o nome "Santo".

```
cid = str(input("Em que cidade você nasceu? ")).strip()
print(cid[:5].upper() == 'Santo')
```

Desafio 025:

Crie um programa que leia o nome de uma pessoa e diga se ela tem "Silva no nome".

```
nome = str(input("Qual é seu nome completo? ")).strip()
print("Seu nome tem Silva? {}".format('silva' in nome.lower()))
```

Desafio 026:

Faça um programa que leia uma frase pelo teclado e mostre:

- Quantas vezes aparece a letra "A"
- Em que posição ela aparece a primeira vez.
- Em que posição ela aparece a última vez.

```
-----  
  
Frase = str(input("Digite uma frase: ")).upper().strip()  
print('A letra A aparece {} vezes na frase. '.format(frase.count('A')))  
print('A primeira letra A apareceu na posição {}'.format(frase.find('A') + 1))  
print('A ultima letra A apareceu na posição {}'.format(frase.rfind('A') + 1))  
  
-----  
  
-----
```

Desafio 027:

Faça um programa que leia o nome completo de uma pessoa, mostrando em seguida o primeiro e o último nome separadamente.

Ex: Ana Maria de Souza
primeiro = Ana
último = Souza

```
-----  
  
n = str(input('Digite seu nome completo: ')).strip()  
nome = n.split()  
print('Muito prazer em te conhecer! ')  
print('Seu primeiro nome é {}'.format(nome[0]))  
print('Seu último nome é {}'.format(nome[len(nome) - 1]))  
  
-----  
  
-----
```

AULA — 10

CONDIÇÃO

Data: 09/08/2023

Profº Gustavo Guanabara

Teoria

Indentação:

é possível usar a indentação usando a tecla TAB do teclado

Estrutura condicional:

If `carro.esquerda()`:

Bloco `True`

else:

Bloco `False`

Ex.:

```
tempo = int(input("Quantos anos tem seu carro? "))
```

```
if tempo <=3:
```

```
    print("Carro novo")
```

```
else:
```

```
    print("Carro velho")
```

```
print("—FIM—")
```

exemplo de uma condição simplificada:

```
tempo = int(input("Quantos anos tem seu carro? "))
```

```
print("Carro novo" if tempo <=3 else "carro velho")
```

```
print("—FIM—")
```

Prática

```
nome = str(input("Qual é o seu nome"))
if nome == "Gustavo":
    print("Que nome lindo você tem!")
print("Bom dia, {}!".format(nome))
```

```
else:
    print("Seu nome é tão normal!")

print("Bom dia, {}!".format(nome))
```

nota: quando tem o else nós dizemos que é uma estrutura condicional composta, já quando não o tem nós dizemos que é uma estrutura condicional simples.

Desafio

Desafio 028:

Escreva um programa que faça o computador "Pensar" em um número inteiro entre 0 e 5 e peça para o usuário tentar descobrir qual foi o número escolhido pelo computador.

O programa deverá escrever na tela se o usuário venceu ou perdeu.

Desafio 029:

Escreva um programa que leia a velocidade de um carro.

Se ele ultrapassar 80km/h, mostre uma mensagem dizendo que ele foi multado.

A multa vai custar R\$ 7,00 por cada Km acima do limite.

Desafio 030:

Crie um programa que leia um número inteiro e mostre na tela se ele é PAR ou IMPAR.

Desafio 031:

Desenvolva um programa que pergunte a distância de uma viagem em Km. Calcule o preço da passagem, cobrando R\$ 0,50 por Km para viagens de até 200Km e R\$ 0,45 para viagens mais longas

Desafio 032:

Faça um programa que leia um ano qualquer e mostre se ele é bissexto

Desafio 033:

Faça um programa que leia três números e mostre qual é o maior e qual é o menor.

Desafio 034:

Escreva um programa que pergunte o salário de um funcionário e calcule o valor do seu aumento.

Para Salários superiores a R\$ 1.250,00 Calcule um aumento de 10%

Para os inferiores ou iguais, o aumento é de 15%

Desafio 035:

Desenvolva um programa que leia o comprimento de três retas e diga ao usuário de elas podem ou não formar um triângulo.

AULA – 11

CORES NO TERMINAL

Data: 19/08/2023

Profº Gustavo Guanabara

Teoria

Padrão ANSI - escape sequence

- Tudo dentro de ANSI começa com Contrabarra

Sempre que quiser representar uma cor em python você vai começar com `\033[m` Entre esse colchete e esse m você vai por o código da cor.

Você vai preencher com coisas, e essas coisas podem ser nenhum código, 1 código, 2 códigos ou 3 códigos

Ex.: 3 códigos

- o primeiro código é o do comportamento ou seja o estilo
- o segundo código é o do texto
- o terceiro e ultimo é o do background que é a cor de fundo

`\033[0;33;44m`

Códigos para:

Style

- 0 – none (sem estilo nenhum)
- 1 – Bold (Negrito)
- 4 – Underline (Sublinhado)
- 7 – Negative (Inverte as configurações)

Text

30 – Branco

31 - Vermelho

32 - Verde

33 - Amarelo

34 - Azul

35 - Magenta

36 - Ciano

37 - Cinza

Back

40 – Branco

41 - Vermelho

42 - Verde

43 - Amarelo

44 - Azul

45 - Magenta

46 - Ciano

47 - Cinza

\033[m (Padrão do terminal)

Prática

Exemplo:

```
print("\033[1;31;43mOlá Mundo! \033[m")
```

Para a barra de fundo não chegar até o final

```
print("\033[1;31;43mOlá Mundo!")
```

Exemplo para deixar apenas uma parte do código colorido

a = 3

b = 5

```
print("Os valores são \033[32m{ } e \033[31m{ }")
```

Exemplo para deixar formatado

nome = 'Guanabara'

```
print("Olá! Muito prazer em te conhecer, {} {} {}".format('\033[4;34m' , nome , '\033[m'))
```


Exemplo mais avançado

```
nome = 'Guanabara'
```

```
cores = {'limpa': '\033[m', 'Azul': '\033[34m', 'Amarelo': '\033[33m', 'Preto e branco':  
\033[7;30m'}
```

```
print("Olá! Muito prazer em te conhecer, {} {} {}".format('\033[4;34m', nome, '\033[m'))
```

Desafio

Desafio 000:

Desafio 000:
