



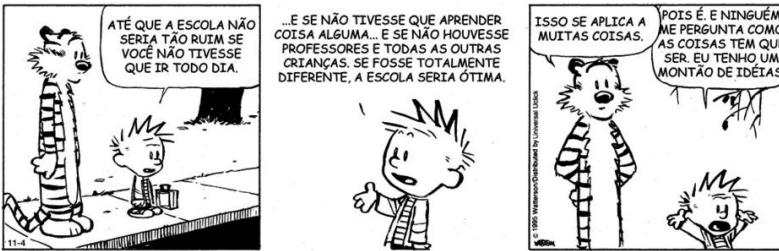
Descrição

[Visualizar envios](#)

EP02


Disponível a partir de: terça, 11 mai 2021, 12:00**Data de entrega:** sábado, 15 mai 2021, 23:59**Arquivos requeridos:** tipos.py ([Baixar](#))**Tipo de trabalho:** Trabalho individual**Redução por avaliação automática:** 0.5 **Avaliações livres:** 10

EP02 - Valores, expressões e seus tipos

Fonte: [Cultura, Estadão](#)O **limite de submissões livres** deste EP é 10.O **prazo de entrega** deste EP é 23h 59m do dia 10/05/2021. O sistema fecha e reabre a partir do dia 11/05 as 12:00 h para envio de EPs com atraso, por mais 5 dias, recebendo desconto de 2 pontos por dia.

Objetivos

Neste EP você utilizará novamente as funções `input()` e `print()`. Desta vez estaremos chamando a sua atenção para o fato de que o resultado de uma expressão depende do operador, `+`, `-`, `*`, `/`, `//` e `%`, e do **tipo** de cada um dos valores dos operandos que compõe a expressão. Neste ponto de MAC0110 você deve se familiarizar com os tipos `str`, `int` e `float`.

As [Gravações, rabiscos, exercícios e programas das reuniões anteriores](#) e leitura de [Pensamentos lógico e aritmético](#) podem ser úteis.Se você ainda não fez isso, instale Python 3 no seu computador seguindo as instruções em [Como instalar Python em seu computador](#).Quando estiver no IME-USP, em algum momento no futuro , você pode também utilizar os computadores do Centro de Ensino de Computação (CEC) ou da rede Linux.

Variáveis, tipos e expressões

Neste EP você exercitará alguns tópicos que temos explorado nas primeiras aulas, a saber, variáveis, expressões, valores e, principalmente, seus tipos. O resultado de operações dependem dos tipos dos seus operandos. Por exemplo, em `3 + 5.0` o operador é `+` e os operandos são o `3` do tipo `int` e `5.0` do tipo `float` e o resultado é `8.0` do tipo `float`.

Você utilizará, além função de entrada `input()`, de saída `print()`, a função:

- `int()` que *converte* um valor para o tipo `int`;
- `float()` que *converte* um valor para o tipo `float`.

Sugerimos que como argumento da função `print()` você utilize *f-strings*.

```

In [1]: a = 6 # a é apelido para 6; depois de '#' é comentário
In [2]: s = f"a = {a}" # isto é uma f-string, {a} é substituído pelo valor de a
In [3]: a
Out[3]: 6
In [4]: s
Out[4]: 'a = 6'
In [5]: print(s) # note que com print() não aparece Out [5]
a = 6
In [6]: nome = 'Dr Who'
In [7]: nome
Out[7]: 'Dr Who'
In [8]: print(f"nome = '{nome}'") # note que com print() não aparece Out [8]
nome = 'Dr Who'
In [9]: nome = 'graúna'
In [10]: print(f"nome = {nome}") # note que com print() ...
nome = graúna
In [11]: type(a) # função que retorna o tipo do argumento
Out[11]: int
In [12]: type(nome)
Out[12]: str
In [13]: type(s)
Out[13]: str
In [14]: print(f"{type(s)}") # note que com ...
<class 'str'>
In [15]: print(f"{type(a)}") # note que ...
<class 'int'>

```

O que você deve fazer

Faça um programa que *leia*:

- uma *string* *s*,
- um número *inteiro* *i* e
- um número *real* *x*

e *imprima* as mensagens e os resultados das expressões como mostrado nos exemplos a seguir. De maneira semelhante ao EP01, as mensagens devem ser **exatamente iguais** às dos exemplos. Observe que

- "*s+s = '1010'*" é **diferente** de "*s+s = 1010*"
- "*s+s = '1010'*" é **diferente** de "*s + s = '1010'*"
- "*i+i = 20*" é **diferente** de "*i+i =20*"
- "*x+x = 20.0*" é **diferente** de "*X+X = 20.0*"

Na dúvida, você pode sempre pergunta o que o interpretador da linguagem Python pensa a respeito:

```

In [16]: "s+s = '1010'" == "s+s = 1010"
Out[16]: False
In [17]: "s+s = '1010'" == "s + s = '1010'"
Out[17]: False
In [18]: "i+i = 20" == "i+i =20"
Out[18]: False
In [19]: "x+x = 20.0" == "x+x = 20.0"
Out[19]: True
In [20]: "x+x = 20.0" == "X+X = 20.0"
Out[20]: False

```

Exemplos

A seguir estão exemplos de **3 execução do programa** como apresentados no *Console do Spyder*.

O que aparece em **vermelho** foi digitado pelo usuário. Observe que nesses exemplos o que aparece dentro do `runfile()` depois do **prompt** (como **In [1]:**) depende de onde você salvou o seu arquivo e da forma que você usou para executá-lo. Essa parte não será idêntica aos exemplos, se for, recomendamos que você aposte na loteria (☹_☹).

Observe também que parte dos espaçamentos é devida ao ambiente do Spyder. Por exemplo, a função `input("mensagem")` pula uma linha antes de exibir a mensagem, o que pode causar alguma pequena confusão inicial.

Exemplo 1

```
In [1]: runfile('/Users/aluno/mac0110-2021/ep02/ep02.py', wdir='/Users/aluno/mac0110-2021/ep02/')

Digite uma string s: 10

Digite um inteiro i: 10

Digite um float x: 10
s = '10' (<class 'str'>)
i = 10 (<class 'int'>)
x = 10.0 (<class 'float'>)
s + s = '10' + '10' = '1010' (<class 'str'>)
i + i = 10 + 10 = 20 (<class 'int'>)
x + x = 10.0 + 10.0 = 20.0 (<class 'float'>)
i * s = 10 * '10' = '1010101010101010' (<class 'str'>)
i * i = 10 * 10 = 100 (<class 'int'>)
i * x = 10 * 10.0 = 100.0 (<class 'float'>)
x / i = 10.0 / 10 = 1.0 (<class 'float'>)
i / i = 10 / 10 = 1.0 (<class 'float'>)
i // i = 10 // 10 = 1 (<class 'int'>)
i / 2 * 3 = 10 / 2 * 3 = 15.0 (<class 'float'>)
i // 2 * 3 = 10 // 2 * 3 = 15 (<class 'int'>)
i % 3 = 10 % 3 = 1 (<class 'int'>)

In [2]:
```

Exemplo 2

```
In [2]: runfile('/Users/aluno/mac0110-2021/ep02/ep02.py', wdir='/Users/aluno/mac0110-2021/ep02/')

Digite uma string s: 2 * 5 + 7

Digite um inteiro i: 5

Digite um float x: 2.5
s = '2 * 5 + 7' (<class 'str'>)
i = 5 (<class 'int'>)
x = 2.5 (<class 'float'>)
s + s = '2 * 5 + 7' + '2 * 5 + 7' = '2 * 5 + 72 * 5 + 7' (<class 'str'>)
i + i = 5 + 5 = 10 (<class 'int'>)
x + x = 2.5 + 2.5 = 5.0 (<class 'float'>)
i * s = 5 * '2 * 5 + 7' = '2 * 5 + 72 * 5 + 72 * 5 + 7' (<class 'str'>)
i * i = 5 * 5 = 25 (<class 'int'>)
i * x = 5 * 2.5 = 12.5 (<class 'float'>)
x / i = 2.5 / 5 = 0.5 (<class 'float'>)
i / i = 5 / 5 = 1.0 (<class 'float'>)
i // i = 5 // 5 = 1 (<class 'int'>)
i / 2 * 3 = 5 / 2 * 3 = 7.5 (<class 'float'>)
i // 2 * 3 = 5 // 2 * 3 = 6 (<class 'int'>)
i % 3 = 5 % 3 = 2 (<class 'int'>)

In [3]:
```

Exemplo 3

```
In [3]: runfile('/Users/aluno/mac0110-2021/ep02/ep02.py', wdir='/Users/aluno/mac0110-2021/ep02/')

Digite uma string s: Olá Mundo!

Digite um inteiro i: 5

Digite um float x: 3.6
s = 'Olá Mundo!' (<class 'str'>)
i = 5 (<class 'int'>)
x = 3.6 (<class 'float'>)
s + s = 'Olá Mundo!' + 'Olá Mundo!' = 'Olá Mundo!Olá Mundo!' (<class 'str'>)
i + i = 5 + 5 = 10 (<class 'int'>)
x + x = 3.6 + 3.6 = 7.2 (<class 'float'>)
i * s = 5 * 'Olá Mundo!' = 'Olá Mundo!Olá Mundo!Olá Mundo!Olá Mundo!Olá Mundo!' (<class 'str'>)
i * i = 5 * 5 = 25 (<class 'int'>)
i * x = 5 * 3.6 = 18.0 (<class 'float'>)
x / i = 3.6 / 5 = 0.72 (<class 'float'>)
i / i = 5 / 5 = 1.0 (<class 'float'>)
i // i = 5 // 5 = 1 (<class 'int'>)
i / 2 * 3 = 5 / 2 * 3 = 7.5 (<class 'float'>)
i // 2 * 3 = 5 // 2 * 3 = 6 (<class 'int'>)
i % 3 = 5 % 3 = 2 (<class 'int'>)

In [4]:
```

Roteiro

- **Baixe** o arquivo `tipos.py` para uma pasta no computador que estiver usando. Este é o único arquivo que deverá ser depositado nesta página.
- **Leia** o cabeçalho com atenção e **preencha** o seu nome e número USP.
- **Implemente e teste** o seu programa. Procure entender e prever o resultado e o tipo `int`, `float` ou `str` do resultante das expressões. Nos seus testes, escreva outras expressões.
- Antes de depositar o seu EP, **verifique** se ele está produzindo exatamente a mesma saída mostrada nos três exemplos anteriores.
- Depois de testar o seu EP, **deposite-o** na página do EP.

O EP02 pode ser enviado 10 vezes sem prejuízo de nota. A partir da 11ª submissão, haverá um desconto de 0,5, meio ponto, por submissão. Verifique as mensagens do programa **avaliador** e, caso necessário, faça eventuais correções e resubmeta o seu EP. Evite submissões desnecessárias, para não ultrapassar o limite de 10 submissões. Procure depurar o seu programa no seu computador, usando o Spyder.

Submissões feitas após às 23h 59m do dia 10/05/2021 receberão desconto de 2 pontos por dia de atraso. O EP não poderá ser enviado após 15/05/2021.

Honestidade Acadêmica

Esse é um exercício individual, não em grupo. Isso não significa que você não pode receber ajuda de outras pessoas, inclusive de seus colegas. De uma forma geral, gostaríamos de incentivar as discussões de ideias, conceitos e alternativas de solução. Nossa maior recomendação é evitar olhar o código fonte de uma solução antes de escrever o seu programa. Em caso de dúvida, consulte nossa [política de colaboração](#).

De forma sucinta, evite as seguintes ações que caracterizam desonestidade acadêmica na realização dos trabalhos individuais desta disciplina:

- buscar e obter uma solução, parcial ou completa, correta ou não, de um EP na internet ou qualquer outro meio físico ou virtual, durante o período de submissão do referido EP;
- solicitar ou obter uma cópia, parcial ou completa, correta ou não, da solução de um EP durante o seu período de submissão;
- permitir que um colega acesse uma cópia, parcial ou completa, correta ou não, do seu EP, durante o período de submissão;
- ainda mais grave é o plágio, que se configura pela utilização de qualquer material não visto em aula ou não descrito no enunciado, que não seja de sua autoria, em parte ou ao todo, e entregar, com ou sem edição, como se fosse seu trabalho, para ser avaliado.

Arquivos requeridos

tipos.py

```
1  # -*- coding: utf-8 -*-
2  #-----
3  # LEIA E PREENCHA O CABEÇALHO
4  # NÃO ALTERE OS NOMES DAS FUNÇÕES, MÉTODOS E ATRIBUTOS
5  # NÃO APAGUE OS DOCSTRINGS
6  #-----
7
8  '''
9      Nome:
10     NUSP:
11
12     Ao preencher esse cabeçalho com o meu nome e o meu número USP,
13     declaro que todas as partes originais desse exercício programa (EP)
14     foram desenvolvidas e implementadas por mim e que portanto não
15     constituem desonestidade acadêmica ou plágio.
16     Declaro também que sou responsável por todas as cópias desse
17     programa e que não distribuí ou facilitei a sua distribuição.
18     Estou ciente que os casos de plágio e desonestidade acadêmica
19     serão tratados segundo os critérios divulgados na página da
20     disciplina.
21     Entendo que EPs sem assinatura devem receber nota zero e, ainda
22     assim, poderão ser punidos por desonestidade acadêmica.
23
24     Abaixo descreva qualquer ajuda que você recebeu para fazer este
25     EP. Inclua qualquer ajuda recebida por pessoas (inclusive
26     monitores e colegas). Com exceção de material de MAC0110, caso
27     você tenha utilizado alguma informação, trecho de código,...
28     indique esse fato abaixo para que o seu programa não seja
29     considerado plágio ou irregular.
30
31     Exemplo:
32
33         A monitora me explicou que eu devia utilizar a função int() quando
34         fazemos leitura de números inteiros.
35
36         A minha função quicksort() foi baseada na descrição encontrada na
37         página https://www.ime.usp.br/~pf/algoritmos/aulas/quick.html.
38
39     Descrição de ajuda ou indicação de fonte:
40
41     '''
42     # escreva seu programa a seguir
43
44
```

Você acessou como Joao Pedro Apolonio de Sousa Matos (Sair)
MAC0110-2021

Disciplinas »

- 2021
- 2020
- 2019
- 2018
- 2017
- 2016
- 2015
- 2014
- 2013
- 2012
- AACCs/FFLCH
- Pró-Reitoria de Pós-Graduação
- Outros

Suporte »

- Documentação
- HelpDesk e Contato
- Guia de uso
- Sobre

Português - Brasil (pt_br)

- Deutsch (de)
- English (en)
- Español - Internacional (es)
- Français (fr)
- Português - Brasil (pt_br)