MC-102 — Aula 03
Escrita, Leitura, Operações Aritméticas e Conversão de Tipos

Instituto de Computação – Unicamp

2016

Roteiro

Escrevendo na tela: **print**

- Para imprimir um texto, utilizamos o comando **print**. O texto pode ser uma constante do tipo string.

Exemplo

```
print("Olá Pessoal!")
```

Saída: **Olá Pessoal!**

- No meio da constante string pode-se incluir caracteres de formatação especiais. O símbolo especial `\n` é responsável por pular uma linha na saída.

Exemplo

```
print("Olá Pessoal!  \n Olá Pessoal")
```

Saída: **Olá Pessoal!**
Olá Pessoal

Escrevendo o conteúdo de uma variável na tela

- Podemos imprimir, além de texto puro, o conteúdo de uma variável utilizando o comando **print**.

Exemplo

```
a=10
print(a)
```

Saída: **10**

Escrevendo o conteúdo de uma variável na tela

- Podemos imprimir junto com textos o conteúdo de variáveis.
- Para isso usamos o comando **str**, que transforma o conteúdo de uma variável em string, e o operador + , que é o operador de concatenação de strings.

Exemplo

```
a=10
print('O conteúdo de a e:  '  + str(a))
```

Saída: **O conteúdo de a e: 10**

Escrevendo o conteúdo de uma variável na tela

- Podemos imprimir junto com textos o conteúdo de variáveis.
- Outra alternativa para isto é separar os dados a serem impressos por vírgula.

Exemplo

```
a=10
print('O conteúdo de a e:  ', a)
```

Saída: **O conteúdo de a e: 10**

Formatos ponto flutuante

Podemos especificar o número de casas decimais que deve ser impresso em um número ponto flutuante usando o especificador **%.Nf**, onde N especifica o número de casas decimais.

Exemplo

a = 3.1415
print(a)
Saída: 3.1415

Exemplo

a = 3.1415
print("%.2f "%a)
Saída: **3.14**

Exemplo

a = 3.1415
print("%.3f "%a)
Saída: **3.142**

Exemplo

```
pi = 3.1415
r = 7
area = pi*r*r
print("A área de um circulo de raio %.2f" %r + " é: %.2f" %area)
print("A área de um circulo de raio " + str(r) + " é: " + str(area))
```

A saída será:

```
A área de um circulo de raio 7.00 é: 153.93
A área de um circulo de raio 7 é: 153.9335
```

Exemplo

- A função **print** sempre pula uma linha ao final da impressão.
- Se você não quiser que pule uma linha, inclua o parâmetro **end=' '** no **print**.

```
print("3, ", end="")
print("4, ", end="")
print("5 ", end="")
```

A saída será:

```
3, 4, 5
```

A função **input**

- Realiza a leitura de dados a partir do teclado.
- Aguarda que o usuário digite um valor e atribui o valor digitado à uma variável.
- Todos os dados lidos são do tipo string.

```
print("Digite um número:")
a = input()
print("O número digitado é: " + a)
```

OBS: A função input está disponível em Python3. Em versões anteriores (Python2) pode-se usar raw_input no lugar.

A função **input**

- Podemos converter uma string lida do teclado em um número inteiro usando a função **int()**.

```
print("Digite um numero:")
a = int(input())
a = a*10
print("O número digitado vezes 10 é: ", a)
```

A função **input**

- Podemos fazer o mesmo para números ponto flutuante usando a função **float()**.

```
print("Digite um numero:")
a = float(input())
a = a*10
print("O número digitado vezes 10 é %.2f: " %a)
```

A função **input**

- Nos dois exemplos anteriores é esperado que o usuário digite um número.
- Se o usuário digitar um texto não numérico o programa encerrará com um erro de execução.

Exemplo

- O programa abaixo lê dois números e imprime a soma destes.
- Perceba que podemos incluir um texto a ser impresso diretamente no comando **input**.

```
a = float(input("Digite um número:"))
b = float(input("Digite um número:"))
print("A soma dos números é: %.2f" %(a+b))
```

Expressões

- Já vimos que constantes e variáveis são expressões.
- Uma expressão também pode ser um conjunto de operações aritméticas, lógicas ou relacionais utilizadas para fazer "cálculos" sobre os valores das variáveis.

Exemplo

a + b

Calcula a soma de **a** e **b**.

Expressões Aritméticas

- Os operadores aritméticos são: +, -, *, /, //, % , **
- *expressão* + *expressão*: Calcula a soma de duas expressões. Exemplo:

```
>>> 56+9
65
```

- *expressão* - *expressão* : Calcula a subtração de duas expressões. Exemplo:

```
>>> 56-9
47
```

- *expressão* * *expressão* : Calcula o produto de duas expressões.

```
>>> 56*9
504
```

Expressões Aritméticas

- *expressão* / *expressão* : Calcula a divisão de duas expressões. O resultado é sempre um número ponto flutuante.

```
>>> 27/9
3.0
```

- *expressão* // *expressão* : Calcula a divisão de duas expressões. Se os operandos forem inteiros a divisão é inteira. Se um deles for ponto flutuante faz uma divisão truncada.

```
>>> 5//2
2
>>> 5//2.0
2.0
```

Expressões Aritméticas

- *expressão* ** *expressão* : Calcula o valor da expressão à esquerda elevado ao valor da expressão à direita.

```
>>> 2**4
16
>>> 2.2**4
23.425600000000006
```

- *expressão* % *expressão* : Calcula o resto da divisão (inteira) de duas expressões.

```
>>> 5%2
1
>>> 9%7
2
>>> 2%5
2
```

Expressões Aritméticas

Mais sobre o operador resto da divisão: %

- Quando computamos "a dividido por b", isto tem como resultado um valor p e um resto $r < b$ que são únicos tais que

$$a = p * b + r$$

- Ou seja a pode ser dividido em p partes inteiras de tamanho b, e sobrará um resto $r < b$.

Exemplos:
5%2 tem como resultado o valor 1.
15%3 tem como resultado o valor 0.
1%5 tem como resultado o valor 1.
19%4 tem como resultado o valor 3.

Expressões

No exemplo abaixo, quais valores serão impressos?

```
print(27%3)
print(19%5)
print(3%15)
```

Expressões

- As expressões aritméticas (e todas as expressões) operam sobre outras expressões.
- É possível compor expressões complexas como por exemplo:
  `a = b *( (2 / c)+(9 + d * 8) );`

Qual o valor da expressão **5 + 10 % 3**?
E da expressão **5 * 10 % 3**?

Precedência

- Precedência é a ordem na qual os operadores serão avaliados quando o programa for executado. Em Python, os operadores são avaliados na seguinte ordem:
  - **
  - *, /, //, na ordem em que aparecerem na expressão.
  - %
  - + e -, na ordem em que aparecerem na expressão.
- Exemplo: 8+10*6 é igual a 68.

Alterando a precedência

- (*expressão*) também é uma expressão, que calcula o resultado da expressão dentro dos parênteses, para só então calcular o resultado das outras expressões.
  - $5 + 10 \% 3$ é igual a 6
  - $(5 + 10) \% 3$ é igual a 0
- Você pode usar quantos parênteses desejar dentro de uma expressão.
- Use sempre parênteses em expressões para deixar claro em qual ordem a expressão é avaliada!

Conversão de Tipos

- Já vimos o uso das funções **int()**, **float()** e **str()** que servem para converter dados de um tipo no outro especificado pela função.
- A conversão só ocorre se o dado estiver bem formado. Por exemplo **int("aaa")** resulta em um erro.
- Ao convertermos um número **float** para **int** ocorre um truncamento, ou seja, toda parte fracionária é desconsiderada.

```
>>> a = "ola"
>>> int(a)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: invalid literal for int() with base 10: 'ola'
>>> int(2.99)
2
>>> int(-2.99)
-2
>>> float("3.1415")
3.1415
>>>
```

Exercício

- Crie um programa que:
  - Lê uma string, pula uma linha e imprime a string lida.
  - Lê um inteiro, pula uma linha e imprime o inteiro lido.
  - Lê um número ponto flutuante, pula uma linha e imprime o número lido.

Exercício

- Crie um programa que lê dois números reais e que computa e imprime a soma, a diferença, a multiplicação e divisão dos dois números.

Outras Informações: comentários

- O código fonte pode conter comentários direcionados unicamente ao programador. Um comentário em uma linha começa com o caracter # e a linha é ignorada pelo compilador.

Exemplo

```
#Este programa imprime uma mensagem
print("Olá Brasil!")
```

- Comentários são úteis para descrever o algoritmo usado e para explicitar suposições não óbvias sobre a implementação.