

---

# GAG009: Alg Prog de Computadores

## Aula 02 – Expressões

Prof. Anilton

Faculdade de Computação

Universidade Federal de Uberlândia





# Atribuição

- ✓ Operador de Atribuição “=”
  - Associa o valor de uma expressão a uma variável, onde esta é criada neste processo;
- ✓ Forma: ***nome\_da\_variável = valor ou expressão***
- ✓ O operador de atribuição “=” armazena o valor ou resultado de uma expressão contida à sua **direita** na variável especificada à sua **esquerda**;
- ✓ Linguagem Python suporta múltiplas atribuições.

```
>>>
>>> x = 1
>>> x
1
>>> y = 2.5
>>> y
2.5
>>> a = b = 3
>>> a
3
>>> b
3
>>>
```

# Operadores Aritméticos



- ✓ Permitem criar expressões aritméticas utilizando números inteiros e fracionários;
- ✓ Seguem a precedência da matemática: **multiplicações e divisões são realizadas antes de soma e subtração.**

Operador	Descrição	Exemplo
+	Soma	$2 + 3$
-	Subtração	$3 - 1$
*	Multiplicação	$2 * 4$
/	Quociente da Divisão	$4.5 / 2$
**	Exponenciação	$2 ** 3$

# Operadores Aritméticos



- Podemos alterar a precedência usando parênteses ();

```
>>>
>>> 2 + 3 * 2
8
>>> (2 + 3) * 2
10
>>>
```

- O operador de subtração “-” também pode ser utilizado para inverter o sinal de um valor.

```
>>>
>>> x = 10
>>> y = -x
>>> x
10
>>> y
-10
>>>
```

# Operadores Aritméticos



- Alguns operadores são definidos apenas para os valores inteiros;

Operador	Descrição	Exemplo
//	Quociente da Divisão Inteira	5 // 2
%	Resto da Divisão Inteira	5 % 2

- Exemplos

```
>>>
>>> x = 10 // 3
>>> x
3
>>>
>>> x = 10 % 3
>>> x
1
>>>
```

# Operadores Aritméticos

---



- ✓ Os operadores aritméticos funcionam com ambos os tipos: **int** e **float**;
- ✓ Devemos apenas estar atentos ao tipo resultante da operação quando combiná-los;
- ✓ Operação
  - $\text{int} + \text{int} \Rightarrow \text{int}$
  - $\text{float} + \text{float} \Rightarrow \text{float}$
  - $\text{int} + \text{float} \Rightarrow \text{float}$
  - $\text{float} + \text{int} \Rightarrow \text{float}$

# Operadores Relacionais



- Comparação entre os valores de diferentes variáveis.

Operador	Descrição	Exemplo
==	Igual	x == 5
!=	Diferente	x != 5
>	Maior do que	x > y
>=	Maior ou igual a	x >= 10
<	Menor do que	y < 100
<=	Menor ou igual a	y <= z

- Operador retorna **True** (verdadeiro) ou **False** (falso).

```
>>>
>>> x = 10
>>> y = 20
>>> x == 11
False
>>> x != y
True
>>> 2*x > y
False
>>> 2*x >= y
True
>>>
```



# Operadores Lógicos

- São operadores que trabalham com valores lógicos e retornam valor lógico verdadeiro (1) ou falso (0);

Operador	Descrição	Exemplo
and	Operador “E”	<code>x == 5 and x &lt; y</code>
or	Operador “OU”	<code>x != 5 or x &lt; 0</code>
not	Operador de negação	<code>not (x &gt; y)</code>

- Exemplos

```
>>> x = 10
>>> y = 20
>>> x > y or x > 0
True
>>> not (x > y)
True
>>> x > y and x > 0
False
>>> not (x > y and x > 0)
True
```

Tabela Verdade

A	B	not A	not B	A and B	A or B
False	False	True	True	False	False
False	True	True	False	False	True
True	False	False	True	False	True
True	True	False	False	True	True



# Atribuição Simplificada



- Python permite simplificar expressões matemáticas.

Operador	Descrição	Exemplo
<code>+=</code>	Soma	<code>c += a</code> equivale a <code>c = c + a</code>
<code>-=</code>	Subtração	<code>c -= a</code> equivale a <code>c = c - a</code>
<code>*=</code>	Multiplicação	<code>c *= a</code> equivale a <code>c = c * a</code>
<code>/=</code>	Quociente da Divisão	<code>c /= a</code> equivale a <code>c = c / a</code>
<code>//=</code>	Quociente da Divisão Inteira	<code>c //= a</code> equivale a <code>c = c // a</code>
<code>%=</code>	Resto da Divisão Inteira	<code>c %= a</code> equivale a <code>c = c % a</code>

# Atribuição Simplificada



- Python permite simplificar expressões matemáticas.

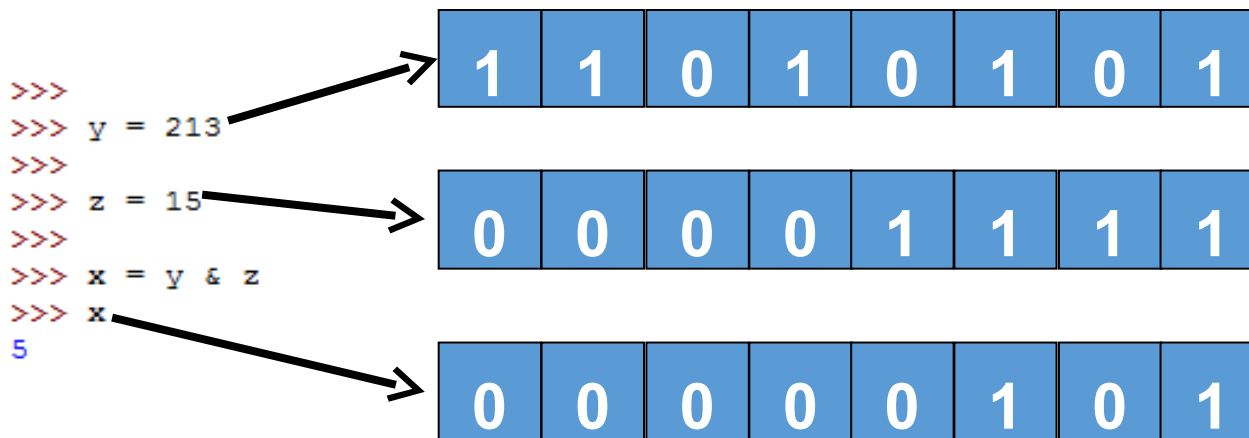
Operador	Descrição	Exemplo
<code>+=</code>	Soma	<code>c += a</code> equivale a <code>c = c + a</code>
<code>-=</code>	Subtração	<code>c -= a</code> equivale a <code>c = c - a</code>
<code>*=</code>	Multiplicação	<code>c *= a</code> equivale a <code>c = c * a</code>
<code>/=</code>	Quociente da Divisão	<code>c /= a</code> equivale a <code>c = c / a</code>
<code>//=</code>	Quociente da Divisão Inteira	<code>c //= a</code> equivale a <code>c = c // a</code>
<code>%=</code>	Resto da Divisão Inteira	<code>c %= a</code> equivale a <code>c = c % a</code>



# Operadores Bit-a-Bit

- Operações bit-a-bit o valor (alto nível) é representado por sua forma binária (baixo nível) e operações são feitas nos bits.

Operador	Descrição	Exemplo
<b>&amp;</b>	<b>E bit-a-bit</b>	<b>x = y &amp; z</b>
<b> </b>	<b>Ou bit-a-bit</b>	<b>x = y   z</b>
<b>~</b>	<b>Complemento bit-a-bit</b>	<b>x = y ~ z</b>



Maior precedência
**
~
* / % //
+ -
>> <<
&
<= < > >=
<> == !=
= %= /= //=- += *= **=
Menor precedência

# Módulo



- Um módulo Python é um arquivo de extensão `.py` contendo código-fonte Python. Este arquivo pode conter variáveis, funções, classes, etc;
- A medida que um programa cresce em tamanho e complexidade, mais módulos Python são usados;
- Comando **import** é a instrução mais básica para trabalhar com módulos. Alguns dos módulos mais comuns são: `math`, `sys`, `os`, `time`, `random`
- Funções do módulo: `nome-módulo.nome-função`

```
>>> import math
>>> math.e
2.718281828459045
>>> math.pi
3.141592653589793
>>> math.sqrt(4)
2.0
>>> |
```



# Alguma dúvida?

Prof. Anilton

Bloco B – Sala 133

[anilton.ufu@outlook.com](mailto:anilton.ufu@outlook.com)

[anilton@ufu.br](mailto:anilton@ufu.br)