

Operadores



Aula de hoje...

- Operadores
 - Aritméticos (usados em contas)
 - Relacionais (usados em comparações numéricas)
 - Lógicos (usados em comparações lógicas)
 - De atribuição (armazenamento de valores em variáveis)

Operadores aritméticos

Operador	Exemplo	Prioridade
(x)	$(1 + 2) * 3 \rightarrow 9$	1
**	$2 ** 3 \rightarrow 8$	2
+x	+15	3
-x	$-(5+3) \rightarrow -8$	3
*	$5 * 3 \rightarrow 15$	4
/	$5 / 3 \rightarrow 1.66$	4
//	$5 // 3 \rightarrow 1$	4
%	$5 \% 3 \rightarrow 2$	4
+	$5 + 3 \rightarrow 8$	5
-	$5 - 3 \rightarrow 2$	5

- ▶ Sempre têm **números como operandos**
- ▶ Sempre têm **resultado numérico**

Operadores aritméticos

- Operadores com a mesma prioridade (precedência)
 - Analisados da esquerda para a direita
- Divisão de inteiros (//)
 - Resultado é somente a parte inteira da divisão
- Divisão (/)
 - Resultado fracionário

Exemplo

■ Considerando

$$x = 512$$

$$y = 9.2 - (x // 10 - 14 / 5) + 14 * 0.1$$

■ Resolução de y

$$y = 9.2 - (512 // 10 - 14 / 5) + 14 * 0.1$$

$$y = 9.2 - (51 - 14 / 5) + 14 * 0.1$$

$$y = 9.2 - (51 - 2.8) + 14 * 0.1$$

$$y = 9.2 - 48.2 + 14 * 0.1$$

$$y = 9.2 - 48.2 + 1.4$$

$$y = -39 + 1.4$$

$$y = -37.6$$

Conversão de Tipos

- Em algumas situações o programador deseja transformar o tipo de uma expressão
 - Para isso, basta envolver a expressão a ser transformada por “tipo(expressão)”

- Exemplo: transformar um real em um inteiro

```
a = 5.1  
x = int(a)    # x vale 5
```

- Exemplo: transformar um inteiro em um real

```
b = 5  
y = float(b)  # y vale 5.0
```

Exemplo

$x = \text{int}(3.3 / (5/2) - 5)$

$y = \text{int}(3.3) / (5/2) - 5$

■ Resolução de x

$x = \text{int}(3.3 / (5/2) - 5)$

$x = \text{int}(3.3 / 2.5 - 5)$

$x = \text{int}(1.32 - 5)$

$x = \text{int}(-3.68)$

$x = -3$

■ Resolução de y

$y = \text{int}(3.3) / (5/2) - 5$

$y = \text{int}(3.3) / 2.5 - 5$

$y = 3 / 2.5 - 5$

$y = 1.2 - 5$

$y = -3.8$

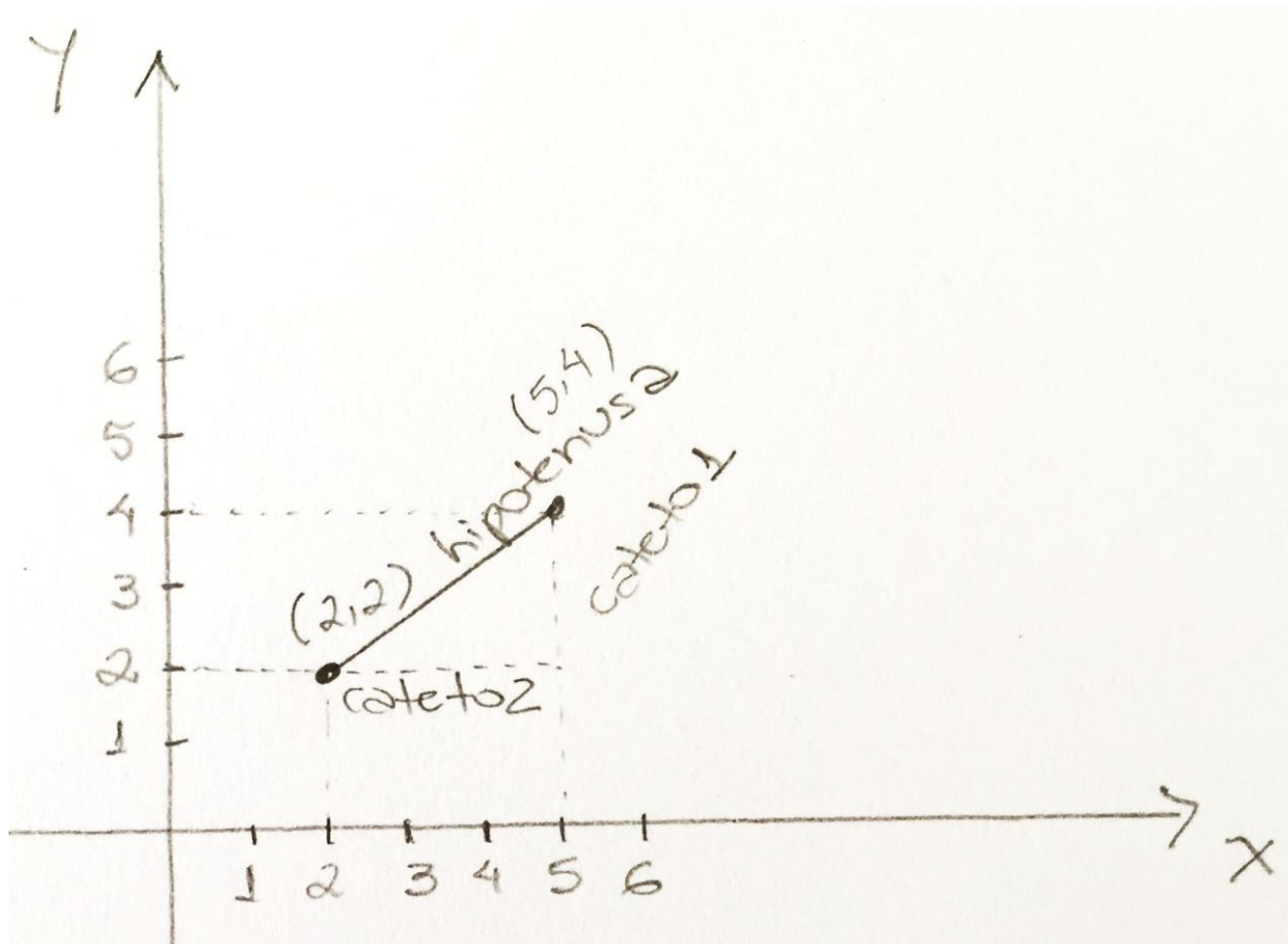
Funções matemáticas

- Para usar algumas das funções matemáticas ou constantes no Python, colocar **import math** no início do programa
- Constantes
 - `math.pi` → 3.1415...
 - `math.e` → 2.7182...

Funções matemáticas

função	Descrição	Exemplo
<code>abs(x)</code>	Valor absoluto	<code>abs(-5.3) → 5.3</code>
<code>round(x, y)</code>	Arredonda x em y dígitos	<code>round(2.677, 2) → 2.68</code>
<code>math.ceil(x)</code>	Arredonda para cima	<code>math.ceil(5.3) → 6</code>
<code>math.floor(expr)</code>	Arredonda para baixo	<code>math.floor(5.3) → 5</code>
<code>max(n1, n2, ...)</code>	Maior dentre vários números	<code>max(3, 4, 5) → 5</code>
<code>min(n1, n2, ...)</code>	Menor dentre vários números	<code>min(3, 4, 5) → 3</code>
<code>math.sqrt(x)</code>	Raiz quadrada	<code>math.sqrt(16) → 4.0</code>
<code>math.log(x, y)</code>	Logaritmo de x na base y	<code>math.log(2, 10) → 0.3010299...</code>
<code>math.log(x)</code>	Logaritmo natural (base e)	<code>math.log(2) → 0.69314718...</code>
<code>math.exp(x)</code>	$e^{**}x$	<code>math.exp(2) → 7.38905609...</code>

Exemplo – Distância entre dois pontos



Exemplo – Distância entre dois pontos

```
import math
```

```
x1 = float(input('Entre com a coordenada x do 1o. ponto:'))  
y1 = float(input('Entre com a coordenada y do 1o. ponto:'))  
x2 = float(input('Entre com a coordenada x do 2o. ponto:'))  
y2 = float(input('Entre com a coordenada y do 2o. ponto:'))
```

```
cateto1 = y2-y1  
cateto2 = x2-x1  
hipotenusa = math.sqrt(cateto1 ** 2 + cateto2 ** 2)  
print('A distancia entre os dois pontos é', hipotenusa)
```

Funções matemáticas: trigonometria

Função	Descrição	Exemplo
<code>math.sin(x)</code>	Seno	<code>math.sin(0) → 0.0</code>
<code>math.asin(x)</code>	Arco seno	<code>math.asin(1) → 1.5707963267948966</code>
<code>math.cos(x)</code>	Cosseno	<code>math.cos(0) → 1.0</code>
<code>math.acos(x)</code>	Arco cosseno	<code>math.acos(-1) → 3.141592653589793</code>
<code>math.tan(x)</code>	Tangente	<code>math.tan(1) → 1.5574077246549023</code>
<code>math.atan(x)</code>	Arco tangente	<code>math.atan(1) → 0.7853981633974483</code>
<code>math.degrees(x)</code>	Converte radianos para graus	<code>math.degrees(math.pi) → 180.0</code>
<code>math.radians(x)</code>	Converte graus para radianos	<code>math.radians(180) → 3.141592653589793</code>

- ▶ Funções trigonométricas trabalham com radiano
- ▶ Ver mais em <https://docs.python.org/3/library/math.html>

Números aleatórios

- Algumas aplicações necessitam que o computador sorteie um número
 - Função `random.random()`
 - Gera número pseudo aleatório no intervalo $[0, 1)$
- A partir desse número, é possível gerar números em outros intervalos
 - $\text{inicio} + (\text{fim} - \text{inicio}) * \text{random.random}()$
- Para usar, colocar **`import random`** no início do programa

Números aleatórios (exemplos)

- Número entre 0 e 1

```
print(random.random())
```

- Número entre 5 e 6

```
print(5 + random.random())
```

- Número entre 0 e 10

```
print(random.random() * 10)
```

- Número entre 50 e 70

```
print(50 + random.random() * 20)
```

Números aleatórios

- É possível gerar números aleatórios inteiros

```
import random  
x = random.randint(3, 9)  
# x conterá um número inteiro sorteado  
# entre 3 e 9, inclusive
```

Operadores relacionais

Operador	Exemplo	Prioridade
$x < y$	$5 < 3 \rightarrow \text{False}$	6
$x \leq y$	$5 \leq 3 \rightarrow \text{False}$	6
$x > y$	$5 > 3 \rightarrow \text{True}$	6
$x \geq y$	$5 \geq 3 \rightarrow \text{True}$	6
$x == y$	$5 == 3 \rightarrow \text{False}$	6
$x != y$	$5 != 3 \rightarrow \text{True}$	6

- ▶ Prioridade sempre inferior aos operadores aritméticos
- ▶ Sempre têm **números ou strings como operandos**
- ▶ Sempre têm **resultado booleano**

Operadores lógicos

Operador	Exemplo	Prioridade
not x	not True \rightarrow False	7
x and y	True and False \rightarrow False	8
x or y	True or False \rightarrow True	9

- ▶ Prioridade sempre **inferior** aos operadores relacionais
- ▶ Sempre têm **booleanos como operandos**
- ▶ Sempre têm **resultado booleano**

Tabelas Verdade

NOT, AND, OR

a	not a
True	False
False	True

a	b	a and b
True	True	True
True	False	False
False	True	False
False	False	False

a	b	a or b
True	True	True
True	False	True
False	True	True
False	False	False

Operadores de atribuição

- Variável do lado esquerdo, valor ou expressão do lado direito

```
x = 0
```

- Pode-se atribuir valor a várias variáveis ao mesmo tempo

```
x = y = z = 0 # x, y e z terão valor 0
```

- Pode-se também atribuir valores diferentes para variáveis diferentes ao mesmo tempo

```
x, y = 1, 2 # x terá o valor 1, e y terá  
o valor 2
```

Operadores de atribuição

Operador	Exemplo
<code>var = expr</code>	<code>x = 10 + 5</code>
<code>var += expr</code>	<code>x += 5 → x = x + 5</code>
<code>var -= expr</code>	<code>x -= 5 → x = x - 5</code>
<code>var *= expr</code>	<code>x *= 5 → x = x * 5</code>
<code>var /= expr</code>	<code>x /= 5 → x = x / 5</code>
<code>var //= expr</code>	<code>x //= 5 → x = x // 5</code>
<code>var %= expr</code>	<code>x %= 5 → x = x % 5</code>
<code>var **= expr</code>	<code>x **= 5 → x = x ** 5</code>

Exemplo

```
x = 10
```

```
y = -2
```

```
z = 5
```

```
w = x * y < z / x or x / y > z * x and z * y < x
```

- Como o valor de w seria avaliado pelo interpretador Python?

Resolução de w

```

w = x * y < z / x or x / y > z * x and z * y < x
w = 10 * -2 < 5 / 10 or 10 / -2 > 5 * 10 and 5 * -2 < 10
w = -20 < 5 / 10 or 10 / -2 > 5 * 10 and 5 * -2 < 10
w = -20 < 0.5 or 10 / -2 > 5 * 10 and 5 * -2 < 10
w = -20 < 0.5 or -5 > 5 * 10 and 5 * -2 < 10
w = -20 < 0.5 or -5 > 50 and 5 * -2 < 10
w = -20 < 0.5 or -5 > 50 and -10 < 10
w = True or -5 > 50 and -10 < 10
w = True or False and -10 < 10
w = True or False and True
w = True or False
w = True

```

Referências

- Tutorial do Python 3
 - <https://docs.python.org/3/reference/expressions.html#operator-precedence>
- Slides feitos em conjunto com Aline Paes e Vanessa Braganholo

Operadores

