

Uma posição de memória (variável) pode conter, *entre* outros, valores dos seguintes **tipos**:

- Um número *inteiro*
 - 7, 3, 2, 0, -3, 4, etc
- Um número *real* (ou *número de ponto-flutuante (float)*)
 - 7.23 , 3.1415, 2.0, -4.023, etc.

Operadores que podem ser usados em expressões em Python:

- + : Soma
- - : Subtração
- * : Multiplicação
- / : Divisão
 - **Atenção!** O resultado de divisão tem o mesmo tipo de seus operandos!
 - $7 / 2$ resulta em 3, pois 7 é inteiro e 2 é inteiro. Logo, o resultado é inteiro.
 - Neste caso, / resulta no **quociente** (ou **parte inteira**) do resultado
- Relembrando:

$$\begin{array}{r} 7 \quad | \quad \underline{2} \\ -6 \quad 3 \leftarrow \text{quociente} \end{array}$$

$$\overline{1} \leftarrow \text{resto}$$

- Para que o resultado seja um número real, *um de seus* operadores deve ser real
 - $7.5 / 2 = 3.75$
 - $7 / 2.0 = 3.5$
- Utilize **float(valor)** para converter um valor para real
 - $\text{float}(7) / 2 = 7.0 / 2 = 3.5$
- Pode ser usado também com variáveis
 - $\text{float}(var) / 2$, se $var=7$, resulta em 3.5
- Para obter a parte inteira da divisão mesmo se os operandos forem reais, use //
- $7.5 // 2 = 3.0$
- Para obter o **resto da divisão** (entre números inteiros), utilize o operador **%** (*mod*)
 - $7 \% 2$ resulta em 1, pois o resto da divisão de 7 por 2 é 1 (vide acima)
 - Pode ser utilizado para, por exemplo, separar o último dígito de um número inteiro
 - Se $n = 1024$, então
 - $n / 10 = 102$
 - $n \% 10 = 4$
 - Também pode ser usado para contar unidades maiores
 - Se 1 metro = 100 cm, então
 - 12394 centímetros são
 - $12394 / 100 = 123$ metros, e
 - $12394 \% 100 = 94$ centímetros
 - Se $a < b$, então $a / b = 0$ e $a \% b = a$. Isto implica que
 - 34 centímetros são
 - $34 / 100 = 0$ metros, e
 - $34 \% 100 = 34$ centímetros,
 - ... como o esperado.

- Python fornece “programas prontos” (*funções*) que realizam cálculos matemáticos comumente usados
 - Raiz quadrada, Exponenciação, etc.
- Para usá-los, inclua, *no começo do seu programa*, a seguinte linha:
 - `import math`
- Isto permite utilizar as seguintes expressões:
 - `math.pi` : Pi
 - `math.sqrt(var)` : Raiz quadrada de var
 - `math.pow(var,n)`: var elevado a n

Como exemplo, este programa

```
import math
# Leitura
x = input()
n = input()

# Imprime pi (3,14...)
print math.pi

# Imprime a raiz quadrada de x
print math.sqrt(x)

# Imprime x elevado a n
print math.pow(x,n)
```

Se executado com essa entrada

```
2
3
```

irá produzir a saída

```
3.14159265359
1.41421356237
8.0
```

Que corresponde ao valor de Pi, à raiz quadrada de 2, e a 2³.