

Contents

References



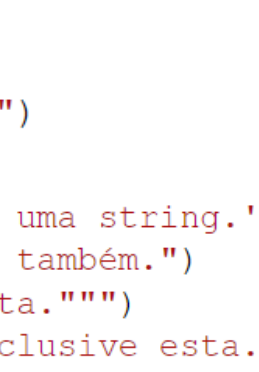
- ▶ Python é um exemplo de uma linguagem de alto nível; outras linguagens de alto nível que você pode ouvir falar são C++, PHP, Java, C#, entre outras.
- ▶ linguagens de baixo nível, geralmente denominadas linguagens de máquina ou assembly. Linguagem de máquina é a codificação de instruções em binário (010010001...)
- ▶ Vantagens x Desvantagens

Converter linguagens de alto nível em linguagens de baixo nível



- ▶ Muitas linguagens modernas usam ambos os processos. O programa é primeiro compilado em uma linguagem de baixo nível denominada byte code, que é então interpretada por um programa chamado de máquina virtual.

Depuração ou debugging



Print

```
1 print("2"+"3")
2 print(2+3)
1 print('Isto é uma string.')
2 print("E esta também.")
3 print("""e esta.""")
4 print('""e inclusive esta...""')
1 print('""Oh não", ela exclamou, "A bicicleta
6 print('""Esta mensagem se estende
7 por diversas linhas
8 de texto.""')
```

Variáveis

- ▶ O operador de atribuição, =, não deve ser confundido com igualdade (operador ==).
- ▶ Nomes de variáveis podem ser longos.
- ▶ Podem conter letras e dígitos, porém devem começar com uma letra ou sublinhado (\_).
- ▶ Apesar de ser possível usar letras maiúsculas, em geral não fazemos isso. Mas, lembre-se que Dia e dia são variáveis diferentes.
- ▶ Nomes de variáveis nunca podem conter espaços em branco.
- ▶ Palavras reservadas

```
76 trombones = "grande parada"
mais$ = 1000000
class = "Computação I"
```

Palavras Reservadas

Python tem aproximadamente 30 palavras reservadas (e esse número pode variar, à medida em que a linguagem evolui):

|         |       |        |          |        |          |
|---------|-------|--------|----------|--------|----------|
| and     | as    | assert | break    | class  | continue |
| def     | del   | elif   | else     | except | exec     |
| finally | for   | from   | global   | if     | import   |
| in      | is    | lambda | nonlocal | not    | or       |
| pass    | raise | return | try      | while  | with     |
| yield   | True  | False  | None     |        |          |

Biblioteca keyword.py

Comandos e Expressões

```
1 print(1 + 1)
2 print(len("olá"))
20 + 32
hora - 1
hora * 60 + minuto
5 ** 2
(5 + 9) * (15 - 7)
1 print(7 / 4)
2 print(7 // 4)
3 minutos = 645
4 horas = minutos // 60
5 print(horas)
```

Operadores e Operandos

```
1 quociente = 7 // 3
2 print(quociente)
3 resto = 7 % 3
4 print(resto)
```

Operador resto

Precedência de operações

- ▶ **Parênteses** têm a mais alta precedência e podem ser usados para forçar que uma expressão seja calculada na ordem desejada.
- ▶ **Exponenciação** tem a segunda precedência mais alta
- ▶ **Multiplicação e divisões** têm a mesma precedência, que são mais altas que adição e subtração, que também têm a mesma precedência.
- ▶ Operadores com a mesma precedência são executados da esquerda para a direita.

Funções para conversão de tipo

```
1 print(3.14, int(3.14))
2 print(3.9999, int(3.9999))
3 print(3.0, int(3.0))
4 print(-3.999, int(-3.999))
5
6 print("2345", int("2345"))
7 print(17, int(17))
8 print(int("23garrafas"))
int
1 print("123.45")
2 print(float("123.45"))
float
1 print(str(17))
2 print(str(123.45))
str
```

Entrada

```
1 n = input("Por favor digite seu nome: ")
2 print("Oi", n)
1 totalMins = int(input("Tempo em minutos:"))
2 horas = totalMins // 60
3 minutos = totalMins % 60
4 print(totalMins, "=", horas, "h e", minutos, "mins")
```

References

Thank you !

robsonms@uneb.br