

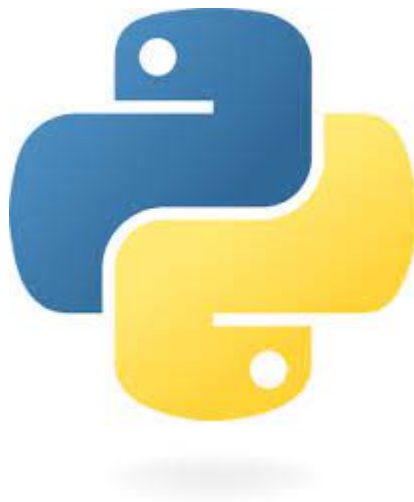
1. Introdução à linguagem: Python3

Autor: Fabrício Domingos da Silva

Este é um material básico e introdutório, voltado para as aulas de Algoritmos de Programação e Programação 1, dos respectivos cursos de Engenharia de Computação e Telemática, do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, campus Campina Grande. Os conteúdos abordam sintaxes básicas, utilização de estruturas da linguagem e outros artifícios mais básicos, somente para que os leitores possam se familiarizar com a linguagem, sem aprofundar muito no assunto. São abordados principalmente os recursos mais utilizados nas disciplinas.

Sobre linguagens de programação: Python ¶

Python é uma linguagem que foi criada no final de 1989, por Guido van Rossum no Instituto de Pesquisa Nacional para Matemática e Ciência da Computação, nos Países Baixos e se trata de linguagem de programação de **alto nível**, **INTERPRETADA**, **Orientada à objetos**, com **tipagem de dados forte e dinâmica**.



Linguagens de programação de alto nível, são linguagens que se aproximam da linguagem normal humana. Elas foram criadas com o intuito de facilitar, tanto a escrita dos códigos e compreensão do fluxo de execução, quanto a detecção de erros e correção de bugs nos códigos. Algumas linguagens de alto nível comuns são Python, Java, etc., já quando se fala em linguagens de baixo nível, a mais famosa é a linguagem Assembly, onde o programador realiza chamadas, que são instruções diretas ao sistema operacional. Existem também, as linguagens de médio nível, como C, no caso da linguagem C, ela é considerada uma linguagem de médio nível, pois permite que o programador tenha o controle sob endereços de memória e outros artifícios que se

aproximam mais das linguagens de baixo nível, mas ao mesmo tempo, é uma linguagem em que se obtém uma boa interpretação e abstração do que ocorre em cada linha, pois se aproxima bastante da linguagem humana, como as linguagens de alto nível.

Quando falamos de execução de códigos em linguagens de programação, temos dois tipos de linguagens de programação, as linguagens interpretadas e as compiladas. As linguagens de programação que são interpretadas realizam a execução de código linha a linha no código, ou seja, todas as vezes em que se executa um código em uma linguagem interpretada, as linhas do código, em ordem, vão ser transformadas em instruções a serem passadas ao sistema operacional e, em seguida, serem passadas ao processador. Já as linguagens que são compiladas, são lidas pelo compilador e traduzidas em instruções, que vão ser transformadas em arquivo executável, já com todo o código traduzido para o nível de instrução. A compilação de um código permite que seja eliminado um gargalo das linguagens interpretadas, que é a necessidade de sempre realizar a tradução do código em instruções, todas as vezes em que o código é executado.

As linguagens de programação, como a linguagem humana, possuem 3 termos importantes a serem memorizados, são eles: Léxico, Sintaxe e Semântica. A parte Léxica de uma linguagem de programação se trata da gramática que ela possui, as palavras reservadas da linguagem, que são funções e termos já embutidas nas linguagens, como `print()`, `int`, `if`, `else`, `while` ou `for`, em python. A parte sintática, diz respeito, por exemplo, à como essas palavras reservadas são utilizadas no código. Já a semântica de uma linguagem de programação se refere ao significado dos programas escritos nessa linguagem. Isso inclui o que as instruções fazem, como as variáveis são manipuladas e como os programas interagem com o ambiente em que estão sendo executados.

Em resumo, o léxico define as palavras que podem ser usadas em uma linguagem de programação, a sintaxe define as regras para combinar essas palavras em programas válidos e a semântica define o significado dos programas escritos nessa linguagem.

In [1]:

```
#Código exemplo 1: (Comentários em códigos são sempre importantes,  
#caso o programador queira realizar demarcações ou quiser visitar o código  
#Comentários em python são definidos pelo caractere # antes do comentário  
# escritas com objetivo de comentário sem a demarcação, acarretam erro no  
print("Hello World!")
```

Hello World!

No código acima, imprimimos na tela com a função `print()`, que está embutida da linguagem python, a frase "Hello World!". É importante observar que a frase na função está entre aspas, pois as aspas são o artifício da linguagem que permite a identificação de strings, que é um dos tipos de dados presente na linguagem e a falta das aspas nesse caso causaria um erro de sintaxe no código que seria informado com uma mensagem de erro pela linguagem. Exemplo:

In [2]:

```
#Código exemplo 2:  
print(Hello World!)
```

```
File "C:\Users\domin\AppData\Local\Temp\ipykernel_4120\316  
7491202.py", line 2  
    print(Hello World!)  
          ^  
SyntaxError: invalid syntax
```

O interpretador python detecta erros de sintaxe na fase anterior à execução. O interpretador verifica, linha a linha, o código em busca de erros de sintaxe e quando encontrados a execução do código é interrompida e a mensagem de erro é exibida. Contudo, existem erros nos códigos que não são necessariamente sintáticos, são eles os erros semânticos, que se tratam de erros que ocorrem durante o tempo de execução do código. Eles ocorrem geralmente quando há algum engano na lógica do programador, o que acarreta erro e falha futuros, gerando uma interrupção, com mensagem de erro, do código.

Exemplo:

In [3]:

```
#Código exemplo 3  
lista_cod3 = [1, 2, 3]  
for i in range(4):  
    print(lista_cod3[i])
```

```
1  
2  
3
```

```
-----  
-----  
IndexError                                Traceback (most re  
cent call last)  
~\AppData\Local\Temp\ipykernel_4120\86569942.py in <module>  
      2 lista_cod3 = [1, 2, 3]  
      3 for i in range(4):  
----> 4     print(lista_cod3[i])
```

IndexError: list index out of range

No código exemplo 3, é definida uma lista (tipos de dados vão ser abordados mais à frente neste material) com 3 elementos, porém, no escopo da estrutura de repetição for, tenta-se acessar um quarto elemento na lista, como o elemento não existe, ocorre um erro em tempo de execução.

Palavras reservadas da linguagem Python

Como citado anteriormente, as linguagens de programação geralmente possuem palavras reservadas. Essas palavras reservadas são o que definem cada função e artifícios embutidas nas linguagens, não podendo ser utilizadas pelo programador em outra função além daquela definida pela linguagem, mas é sempre importante lembrar, que o uso de bibliotecas externas de funções e outras utilidades podem aumentar, quando utilizadas, o número de palavras reservadas nas linguagens.

Em python, existem diversas palavras reservadas que definem tais funções. As palavras reservadas de python são:

Palavras	reservadas	da	linguagem	Python
and	continue	finally	is	raise
as	def	for	lambda	return
assert	del	from	None	True
async	elif	global	nonlocal	try
await	else	if	not	while
break	except	import	or	with
class	False	in	pass	yield

Exemplo:

In [4]:

```
#Código exemplo 4
if = 12
```

File "C:\Users\domin\AppData\Local\Temp\ipykernel_4120\566182613.py", line 2

```
    if = 12
      ^
```

SyntaxError: invalid syntax

No código de exemplo 4 exibido acima, é realizada a tentativa de se atribuir um valor à uma variável com o nome "if", porém, como if é uma das palavras reservadas pela linguagem, a atribuição acarreta um erro de sintaxe.

Outro ponto a ser abordado na linguagem, é que Python é uma linguagem *case sensitive*, ou seja, ela diferencia caracteres maiúsculos de minúsculos, seja em suas variáveis ou strings.