



INSTITUTO FEDERAL

Paraná

Campus Paranaguá

Seminário - Python

Aluno: *Jéssica Cristina Gonçalves*

Turma: TADS-18

Sumário

Histórico da Linguagem	2
Como a linguagem aborda os principais conceitos de OO (Classes, herança, poliformismo, encapsulamento);	3
Sintaxe da linguagem;	4
Sequências Imutáveis e Mutáveis	5
Ambiente de desenvolvimento (IDE, compiladores, como instalar);	6
Referências Bibliográficas	7

Histórico da Linguagem

Python é uma linguagem de computador, que surgiu em 1989, foi criada pelo Guido van Rossum no Centro de Matemática e Tecnologia da Informação (CWI, Centrum Wiskunde e Informatica), na Holanda, como sucessor da linguagem de programação ABC, capaz de lidar com exceções e interagir com o sistema operacional Amoeba, que era o sistema que a equipe de Rossum trabalhava na época.

Python é uma linguagem open source, ou seja, de gratuito acesso, além de ter sido criada para facilitar a melhor legibilidade do código.

Atualmente a linguagem é gerenciada pela **Python Software Foundation** (também conhecida por PSF) que é uma organização sem fins lucrativos que se dedica à linguagem de programação Python. A PSF foi fundada em 6 de Março de 2001.

A PSF tem como missão, promover, proteger e ajudar no desenvolvimento da linguagem, além de facilitar e ajudar o crescimento da comunidade de desenvolvedores. Em Novembro de 2018, the PSF criou a Python in Education Board Committee (Comitê de Python na Educação) com o objetivo de tornar a linguagem mais presente na educação.

Depois de pesquisas feitas foram determinadas 3 áreas de foco, Recursos (estudos, projetos, currículo), localização(Tradução primária) e Mobile.

Boa parte dos sistemas operacionais IOS e Desktop são suportados por python atualmente, porém o Android ainda precisa de trabalho, já que o android está em maior número no mercado a PSF acha importante esse estudo.

Como a linguagem aborda os principais conceitos de OO (Classes, herança, poliformismo, encapsulamento);

Quando criado Python tinha o objetivo de ser usado em diferentes plataformas.No começo o objetivo era usar o Python na Amoeba, o sistema operacional que estava sendo desenvolvido, e no UNIX, o sistema operacional que estava sendo usado em desktops. Também era ideia de Rossum oferecer suporte ao Windows e Macintosh.

Decidiu então, que o Python teria dois tipos diferentes de módulos: possibilitando você escrever módulos em Python, mas também pode escrever um

módulo inteiramente em código C. E esse módulo pode disponibilizar novos tipos e novos objetos.

Classes em python, ela é composta de atributos que representam este objeto. Quando a classe é atribuída a outro objeto, este objeto é chamado de instância da classe. Os métodos, que também são definidos para programação orientada a objetos, são funções dentro de uma classe.

Todos os métodos ou atributos da classe são obrigados a passar por parâmetro ou argumento a palavra self antes de tudo, isso é convencionado pelo próprio desenvolvedor, por padrão se chama self, o self foi convencionado nos parâmetros dos métodos e atributos devido a filosofia da linguagem de estar tudo explícito, especificar de onde o valor vem.

Deitel, em "Programando em Java", diz que "quando utilizamos herança para criar uma classe a partir da classe existente, a nova classe herda os atributos e comportamentos dessa classe existente, podendo adicionar atributos e comportamentos para solucionar nossas necessidades".

O polimorfismo condiz com a reutilização de código: é fato que ocorrem funções semelhantes em várias partes do software; então definimos estas funções em uma biblioteca, e todas as outras funções que dela precisarem poderão chamá-la sem a necessidade de reescrevê-la.

Sintaxe da linguagem;

Python tem sua indentação definida em blocos, não tendo símbolos “abre” e “fecha”. Em Python tudo é objeto. Isso quer dizer que um objeto do tipo string, por exemplo, tem seus próprios métodos.

O conceito de variável é uma associação entre um nome e um valor, mas não é necessário declarar o tipo da variável, portanto, o tipo relacionado a variável pode variar durante a execução do programa isto implica em muitos aspectos no uso da linguagem.

Este conceito é chamado em programação de "duck typing" (tipagem do pato). Em Python existem 4 tipos numéricos:

- inteiro (int)
- ponto flutuante (float)
- booleano (bool)
- complexo (complex)

Em todas as linguagens de programação existem controles de fluxos de dados, em Python também. Os identificadores pass, break e continue, são instruções muito usadas em controle de fluxos.

Em Python existem dois tipos de estruturas de dados:

- Sequências: são objetos ordenados e finitos;
- Dicionários é um conjunto de elementos de mapeamentos indexados por chaves.

Manipulação de string, lower() = minúsculo, upper = maiúsculo, strip() = tira espaços e caracteres, split() = transforma em lista, find() = procurando substrings, replace() = substitui uma substring.

Em Python as funções são definidas pela palavra reservada def, exemplo:

def NOME(PARÂMETROS)

COMANDOS

E são chamadas:

NOME(ARGUMENTOS)

Sequências Imutáveis e Mutáveis

As sequências imutáveis são as strings e as tuplas, dois ou mais elementos de qualquer tipo dentro de parênteses e separados por vírgula. As sequências mutáveis são as listas. Listas são elementos entre colchetes e separados por vírgula. Em outras linguagens chamadas de vetor.

A mais simples é a função range, que gera uma lista de zero a n-1, para gerar uma lista que comece em um ponto arbitrário, deve - se passar este ponto como o início da lista. E pilhas, que nada mais são do que listas em que a inserção e a remoção de elementos acontece na mesma extremidade.

Identificadores reservados da linguagem:

Identificadores	Função
and	e
assert	forçar um tipo para a variável
break	parar
class	definição da classe

continue	continuar
def	definição de função
del	deleção de objeto
elif	não-se
else	senão
except	exceção
exec	executar função
finally	finalizar função
for	definição de laço para
from	importar uma variável dentro de um módulo
global	definição de variável global
if	definição de laço se
import	importação de módulos externos
in	em
is	e
lambda	funções recursivas
not	não
or	ou
pass	passar para outra função sem executá-la
print	imprimir na tela
raise	laço try
return	retornar objeto
try	definição de laço tente
while	definição de laço enquanto

Ambiente de desenvolvimento (IDE, compiladores, como instalar);

Que IDE usar?

Depende muito da sua preferência. Você pode usar qualquer editor de texto padrão, como o Notepad++, Gedit ou Sublime até o VI puro. Não existe padrão. Para quem vem do MATLAB ou R, o Spyder pode ser muito útil. O Pycharm é outro IDE muito utilizado na comunidade.

Aonde eu encontro os módulos para utilizar no meu projeto?

Alguns módulos já vem por padrão no Python puro, por exemplo o módulo matemático. Outros, devem ser baixados de um repositório, como é o caso do Django ou Numpy. Hoje, mais de 107 mil projetos estão cadastrados no repositório oficial.

Referências Bibliográficas

Entrevista com Guido van Rossum - <https://www.youtube.com/watch?v=J0Ag44Pze-w>

Python Software Foundation - <https://pyfound.blogspot.com/>

Venners, Bil - **The Making of Python** <https://www.artima.com/intv/python.html>

Vaz, Welton - **Python como surgiu** <https://www.eusoudev.com.br/python-como-surgiu/>

WikiLivros - **Python** <https://pt.wikibooks.org/wiki/Python>