



UniEVANGÉLICA
UNIVERSIDADE EVANGÉLICA DE GOIÁS

Engenharia de Software

Algoritmos de Programação

Aula 2: Aspectos Fundamentais de Programação

Professor: Dr. Henrique Valle de Lima
henrique.lima@unievangelica.edu.br





JITTD P

Jesus Is The Top Da Parada!

DEUS USA NOSSAS EXPERIÊNCIAS
MAIS DIFÍCEIS PARA NOS LEVAR PARA
MAIS PERTO DO SEU AMOR.

➤ Escreva em **narração descritiva**, **fluxograma** e **pseudocódigo** um programa que **solicite** que o usuário **digite** as quatro notas que um aluno obteve. Em seguida, o programa deve **calcular a média** das notas e **apresentar** a média final

Vamos praticar mais um pouco?

- ▶ **Exercício 1:** Escreva um pseudocódigo que calcule e exiba a soma de dois números inteiros.
- ▶ **Exercício 2:** Desenvolva um pseudocódigo para calcular a média de três números reais e apresentá-la na tela.
- ▶ **Exercício 3:** Escreva um pseudocódigo para converter uma temperatura de graus Celsius para Fahrenheit, tendo o valor em Celsius fornecido pelo usuário.

$$C = \frac{F - 32}{1,8}$$

ou

$$F = 1,8C + 32$$

Vamos praticar mais um pouco?

▶ **Exercício 1:** Escreva um pseudocódigo que calcule e exiba a soma de dois números inteiros.



Início do Algoritmo

Escreva "Digite o primeiro número: "

Leia N1

Escreva "Digite o segundo número: "

Leia N2

$R = N1 + N2$

Escreva "A soma dos dois números é: ", R

Fim do Algoritmo

Vamos praticar mais um pouco?

▶ **Exercício 2:** Desenvolva um pseudocódigo para calcular a média de três números reais e apresentá-la na tela.

Início do Algoritmo

Escreva "Digite o primeiro número: "

Leia N1

Escreva "Digite o segundo número: "

Leia N2

Escreva "Digite o terceiro número: "

Leia N3

$R = (N1 + N2 + N3)/3$

Escreva "A média dos 3 números é: ", R

Fim do Algoritmo

Vamos praticar mais um pouco?

➤ **Exercício 3:** Escreva um pseudocódigo para converter uma temperatura de graus Celsius para Fahrenheit, tendo o valor em Celsius fornecido pelo usuário.

$$C = \frac{F - 32}{1,8}$$

ou

$$F = 1,8C + 32$$

Início do Algoritmo

Escreva "Digite a temperatura em graus Celsius: "

Leia C

$$F = (1,8 * C) + 32$$

Escreva "A temperatura em Fahrenheit é: ", F

Fim do Algoritmo

Tipos de Dados



TIPOS PRIMITIVOS DE DADOS

▶ Cada linguagem de programação tem seus próprios tipos primitivos. Em Python temos:

- ▶ **int** – utilizado para números inteiros
- ▶ **str** – utilizado para conjuntos de caracteres
- ▶ **bool** – armazena True ou False
- ▶ **list** – agrupa um conjunto de elementos
- ▶ **tupla** – semelhante ao tipo list, porém imutável
- ▶ **dic** – agrupa grupos de elementos que serão recuperados por uma chave

▶ Cada primitiva tem uma função no sistema, e o nome é reservado.

PALAVRAS RESERVADAS

- ▶ Algumas palavras em Python são reservadas
- ▶ Isso significa que não podemos utilizar essas palavras em qualquer parte do código, a não ser para cumprir sua função já definida
- ▶ Antes de programar, você precisa saber quais são elas

PALAVRAS RESERVADAS

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

OBS: O Python é case sensitive

A Linguagem Python





abc
LANGUAGES

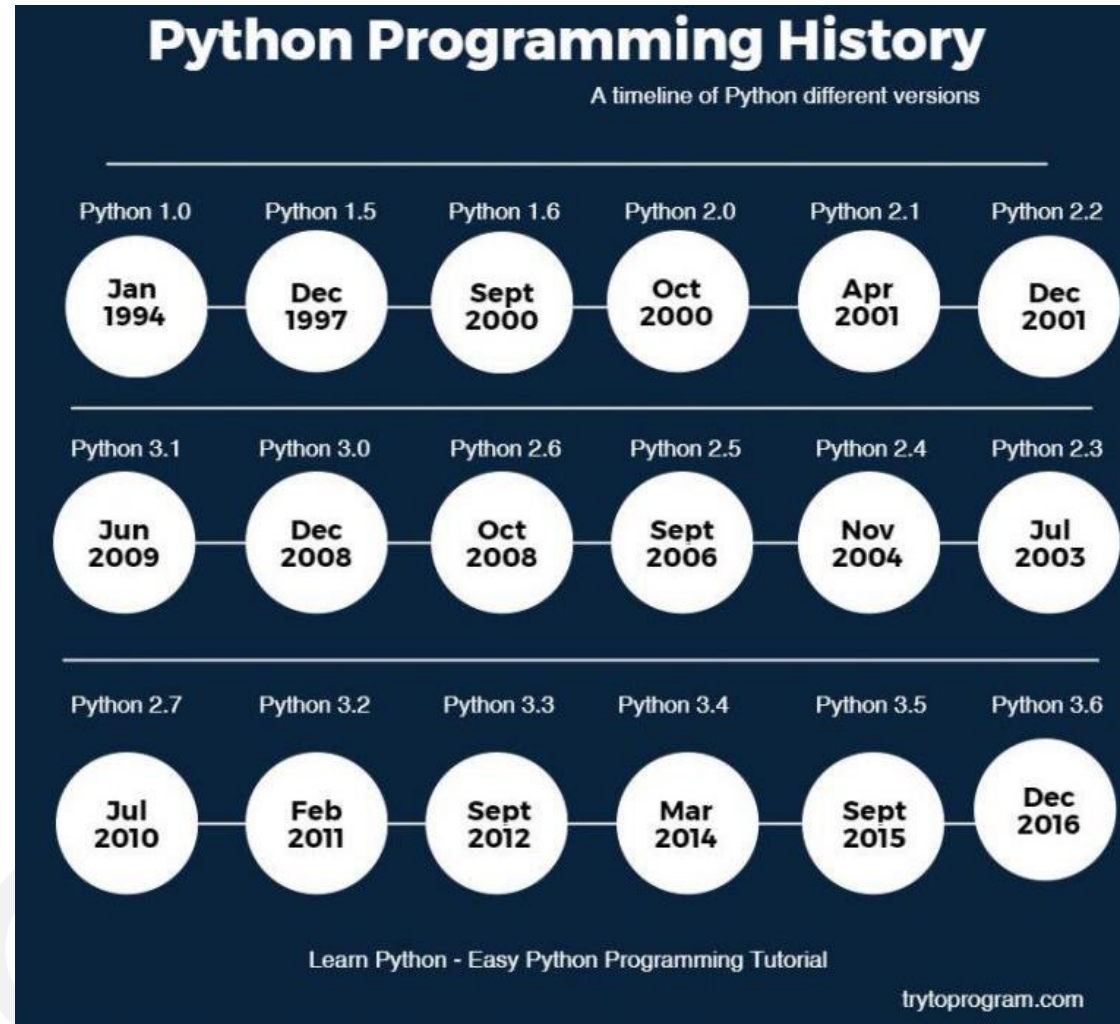
Década de 1980, Van Rossum trabalhava em uma linguagem chamada “ABC”, no S.O. “AMOEBA”. Rossum começou a pensar em uma nova linguagem que possuísse uma sintaxe semelhante à da “ABC” e com o acesso semelhante ao do “AMOEBA”.

Em 1991, foi lançada a primeira versão da linguagem “Python” 1.0.

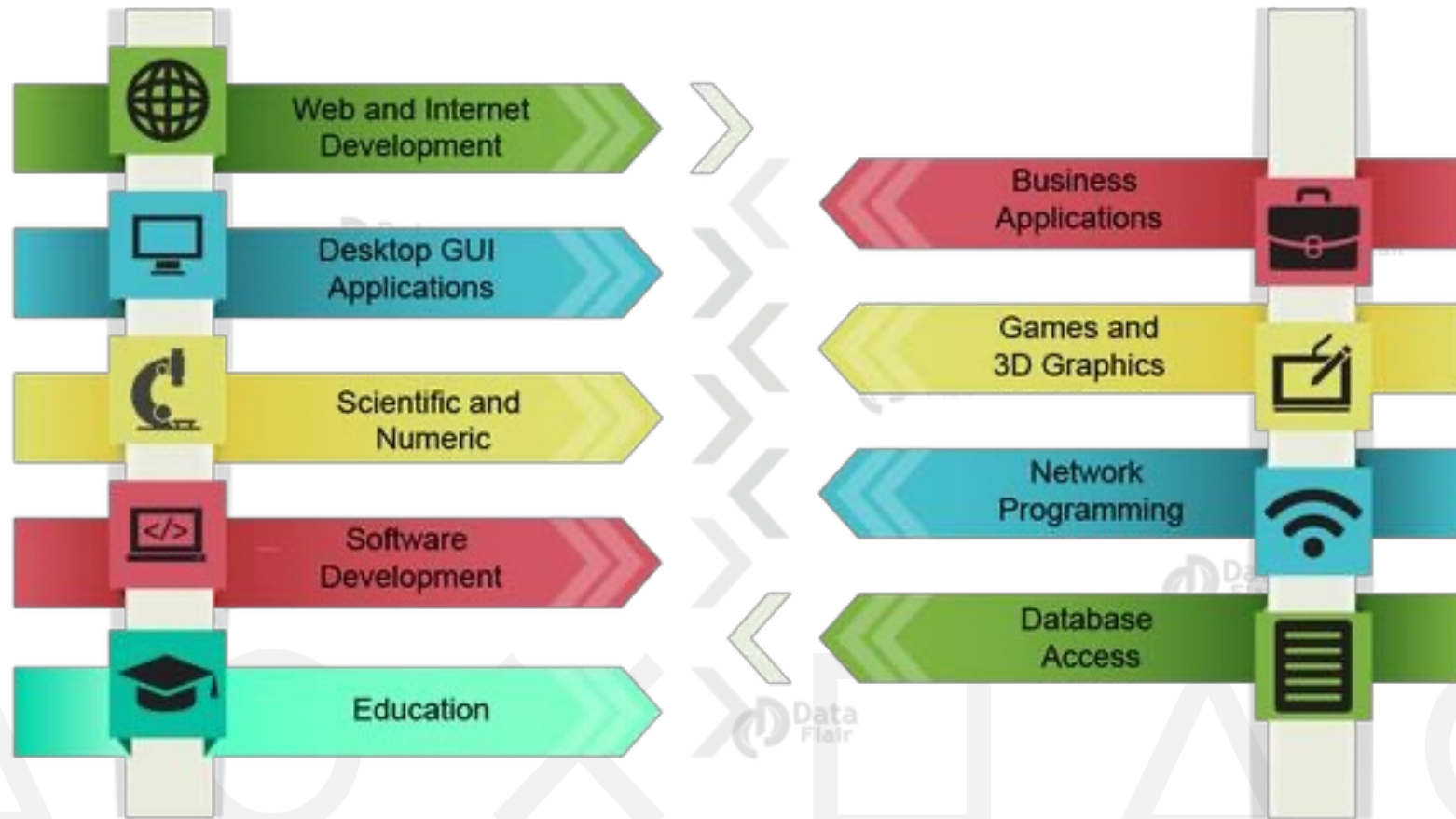
CURIOSIDADE

Segundo o site “Trytoprogram”, na década de 70 existia um famoso programa de comédia na BBC, chamado “Monty Python’s Flying Circus”, do qual Rossum era fã. Então, quando o projeto estava concluído, esse foi nomeado “Python”.





Python Applications



Desenvolvimento Web e Internet

O Python oferece muitas opções para [desenvolvimento na web](#) :

- Frameworks como [Django](#) e [Pyramid](#) .
- Micro-estruturas como [Flask](#) e [Bottle](#) .
- Sistemas avançados de gerenciamento de conteúdo, como o [Plone](#) e o [django CMS](#) .

A biblioteca padrão do Python suporta muitos protocolos da Internet:

- [HTML e XML](#)
- [JSON](#)
- [E-mail processing](#).
- Suporte para [FTP](#) , [IMAP](#) e outros [protocolos da Internet](#) .
- [Interface de soquete](#) fácil de usar .

E o Índice de Pacotes ainda tem mais bibliotecas:

- [Solicitações](#) , uma poderosa biblioteca cliente HTTP.
- [BeautifulSoup](#) , um analisador de HTML que pode lidar com todos os tipos de HTML estranho.
- [Feedparser](#) para analisar feeds RSS / Atom.
- [Paramiko](#) , implementando o protocolo SSH2.
- [Twisted Python](#) , um framework para programação de rede assíncrona.

Educação

O Python é uma linguagem excelente para o ensino de programação, tanto no nível introdutório quanto em cursos mais avançados.

- Livros como [Como Pensar como um Cientista da Computação](#) , [Programação em Python: Uma Introdução à Ciência da Computação](#) e [Programação Prática](#) .
- O [Grupo de Interesse Especial em Educação](#) é um bom lugar para discutir questões de ensino.

Científico e Numérico

Python é amplamente utilizado em computação [científica e numérica](#) :

- [SciPy](#) é uma coleção de pacotes para matemática, ciências e engenharia.
- [Pandas](#) é uma biblioteca de análise e modelagem de dados.
- [O IPython](#) é um poderoso shell interativo que apresenta fácil edição e gravação de uma sessão de trabalho e suporta visualizações e computação paralela.
- O [Curso de Carpintaria de Software](#) ensina habilidades básicas para computação científica, executando bootcamps e fornecendo materiais de ensino de acesso aberto.

GUIs de desktop

A biblioteca [Tk](#) GUI está incluída na maioria das distribuições binárias do Python.

Alguns kits de ferramentas que podem ser usados em várias plataformas estão disponíveis separadamente:

- [wxWidgets](#)
- [Kivy](#) , para escrever aplicativos multitouch.
- Qt via [pyqt](#) ou [pyside](#)

Toolkits específicos da plataforma também estão disponíveis:

- [GTK +](#)
- Microsoft Foundation Classes através das [extensões do win32](#)

Desenvolvimento de software

O Python é frequentemente usado como uma linguagem de suporte para desenvolvedores de software, para controle e gerenciamento de construções, testes e de muitas outras maneiras.

- [SCons](#) para controle de compilação.
- [Buildbot](#) e [Apache Gump](#) para compilação e testes contínuos automatizados.
- [Roundup](#) ou [Trac](#) para rastreamento de bugs e gerenciamento de projetos.

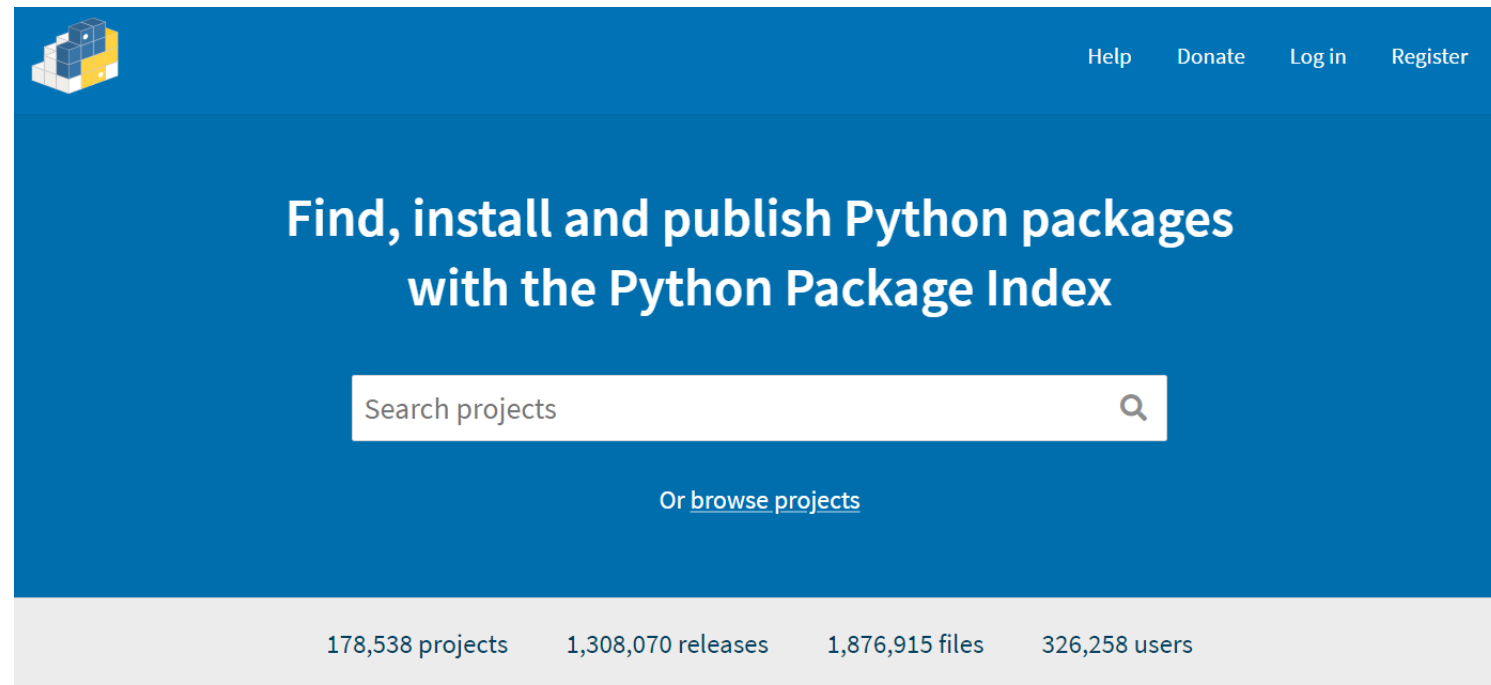
Aplicativos de negócios

O Python também é usado para construir sistemas de ERP e e-commerce:

- [Odoo](#) é um software de gerenciamento completo que oferece uma variedade de aplicativos de negócios que formam um conjunto completo de aplicativos de gerenciamento corporativo.
- [O Tryton](#) é uma plataforma de aplicativos de uso geral de alto nível e três níveis.

Python é usado em muitos
“**applications domains**”.

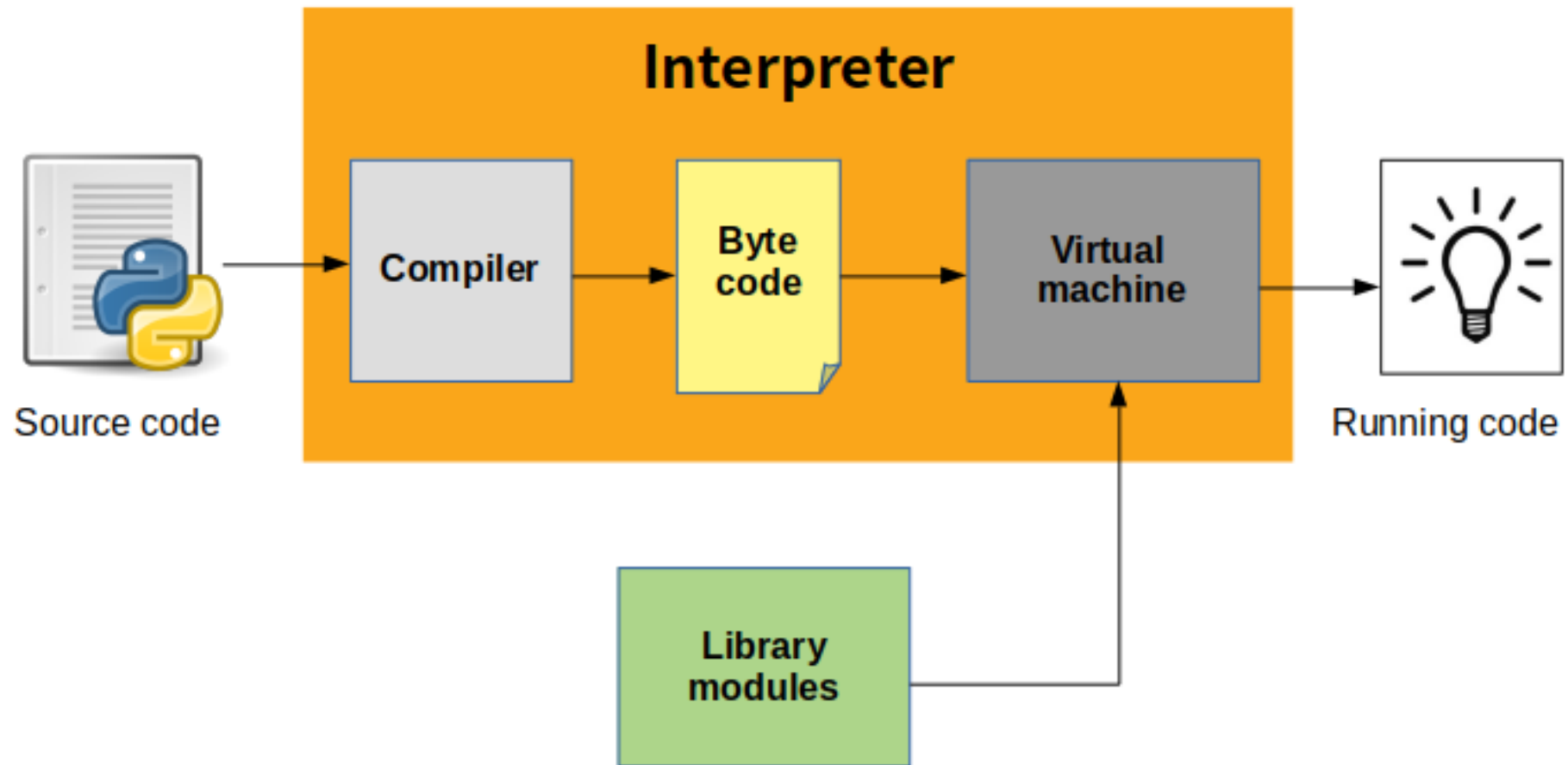
The Python Package Index
lists thousands of third
party modules for Python.



The Python Package Index (PyPI) is a repository of software for the Python programming language.

PyPI helps you find and install software developed and shared by the Python community. [Learn about installing packages.](#)

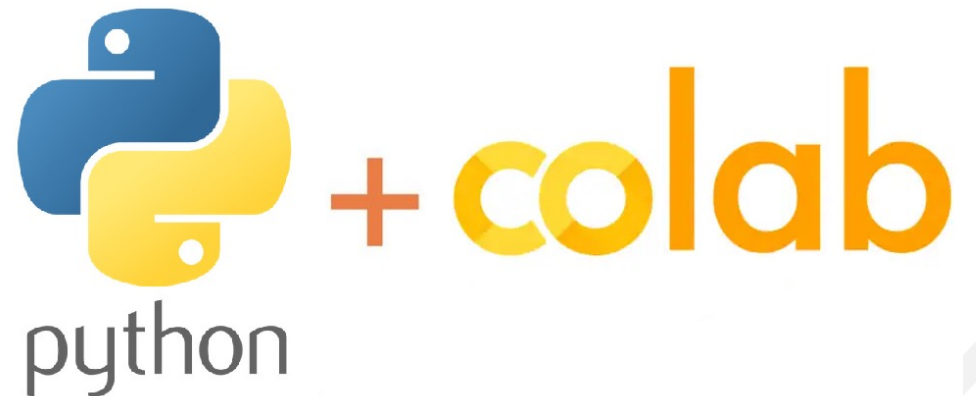
Package authors use PyPI to distribute their software. [Learn how to package your Python code for PyPI.](#)





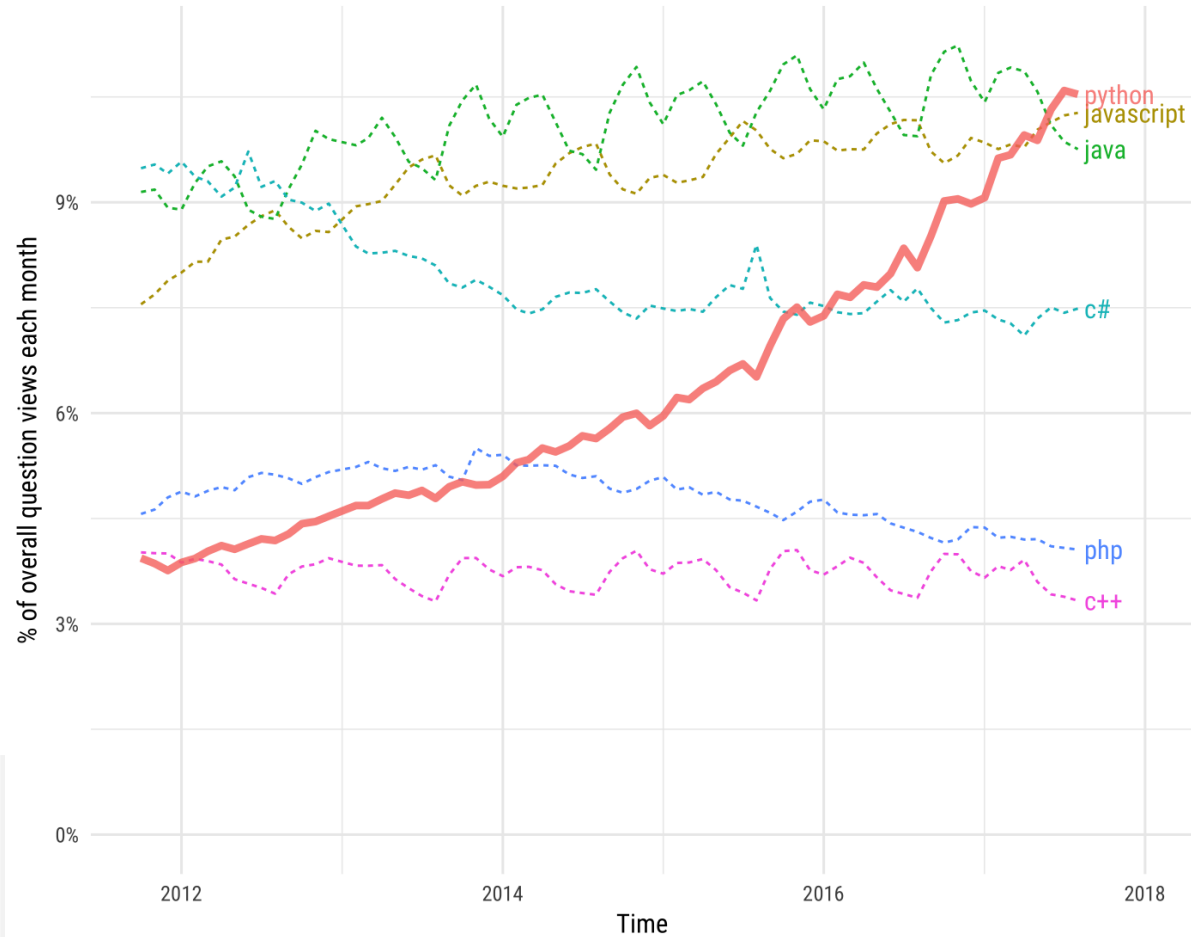
PyScripter

WING PYTHON
IDE
THE INTELLIGENT DEVELOPMENT ENVIRONMENT FOR PYTHON



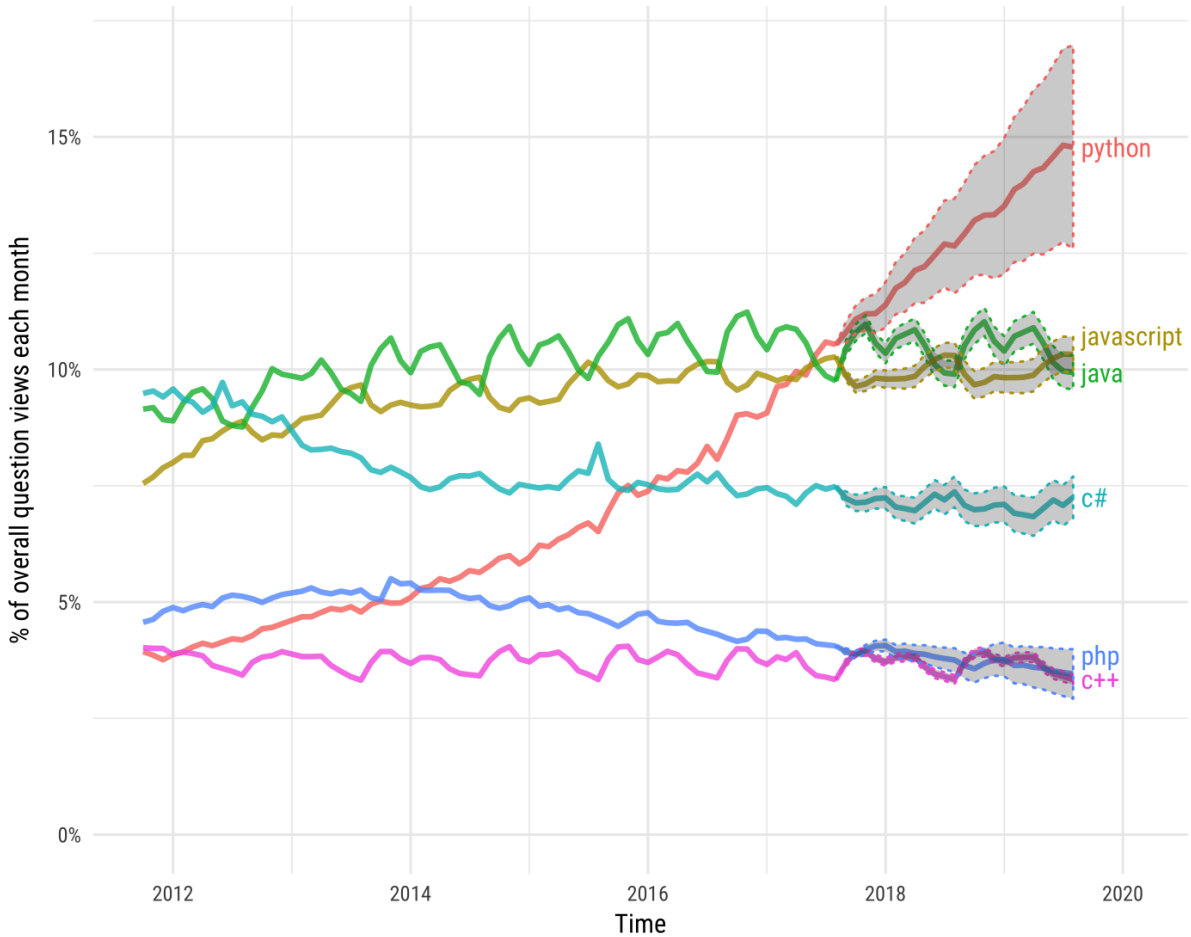
Growth of major programming languages

Based on Stack Overflow question views in World Bank high-income countries



Projections of future traffic for major programming languages

Future traffic is predicted with an STL model, along with an 80% prediction interval.



The popular YouTube video sharing system is largely written in Python



Google makes extensive use of Python in it's web search system



Dropbox storage service codes both its server and client software primarily in Python



The Raspberry Pi single-board computer promotes Python as its educational language



COMPANIES USING PYTHON



BitTorrent peer-to-peer file sharing system began its life as a Python Program



NASA uses Python for specific Programming Task



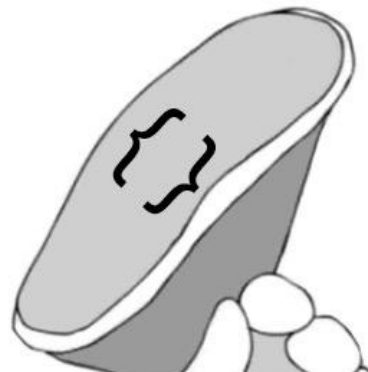
The NSA uses Python for cryptography and intelligence analysis

NETFLIX

Netflix and Yelp have both documented the role of Python in their software infrastructures



Ew, I stepped in shit.



```
1  from Class_Aeroporto import Aeroporto
2
3  sair = False
4  a = Aeroporto()
5  while not sair:
6      _1 = input('Adicionar um novo avião à lista? [s ou S / n ou N] ')
7      if _1 != 'n' and _1 != 'N':
8          print('_1 = ' + _1)
9          a.aviaoParaFila()
10         _2 = input('Autorizar decolagem do primeiro veículo da fila? ')
11         if _2 == 'S' or _2 == 's':
12             print('_2 = ' + _2)
13             a.autorizaDecolagem()
14         else:
15             pass
16         _3 = input('Deseja listar todos os aviões que estão na lista de espera? ')
17         if _3 == 'S' or _3 == 's':
18             print('_3 = ' + _3)
19             a.mostraFilaDeEspera()
20         else:
21             pass
22         _4 = input('Deseja listar as características do primeiro avião da fila? ')
23         if _4 == 'S' or _4 == 's':
24             print('_4 = ' + _4)
25             print('Primeiro veículo da fila: ', a.caracteristicas_PrimeiroDaFila())
26         else:
27             pass
28
29     e = input('\n\nEXIT? [OK] ')
30     if e == 'OK':
31         sair = True
32     else:
33         sair = False
34
```


Entrada e Saída de Dados



O QUE É E/S?

► Comandos de **entrada e saída (e/s)** são aqueles que permitem uma **iteração** com o usuário através de dispositivos periféricos, por exemplo, o **monitor**, **teclado...**

➤ Os sistemas são construídos para **interagirem** com o ser humano. Na maioria dos casos, o sistema **aguarda informações e comandos, processam** os dados e **exibem** os resultados



Saída → **print()**

▶ Aspas simples ou normal

▶ Vamos testar?

```
print ("Eu sou uma mensagem exibida na tela")
```

▶ A concatenação é uma forma de escrita dos dados que mescla tipos diferentes de informações

▶ Se **String**

usar (+);

senão

usar (,);

▶ Pode-se concatenar Strings de várias formas. É como uma soma.

```
print ("Eu sou um texto " + str(323) + " já o cara anterior, é um número")
```

ENTRADA DE DADOS

▶ Entrada → input()

▶ Obs.: Sempre String.

```
variavel = input()
```


- ▶ Uma vez dentro do sistema, podemos manipular, alterar e até imprimir o valor na tela

```
print (variavel)
```

- ▶ Note que agora não utilizamos "" no comando print

```
print (variavel) é diferente de print ("variavel")
```

NÍMR



Expectativa é igual paçoca.

Do nada, esfarela tudo!

Ana Maria Braga

Operadores Matemáticos

OPERADORES MATEMÁTICOS

▶ Assim como na matemática, na computação utilizamos operadores matemáticos

$+, -, *, /$ → normais

$//$ → inteiros

$**$ → potência

▶ Expressões aritméticas

```
>>> 11 + 7
```

```
18
```

```
>>> 18 - 11
```

```
7
```

```
>>> 7 * 11
```

```
77
```

```
>>> 77 / 11
```

```
7.0
```

```
>>> 7 % 11
```

```
7
```

```
>>> 7 ** 11
```

```
1977326743
```

```
>>> 1977326743 // 11
```

```
179756976
```

```
>>> 1977326743 / 11
```

```
179756976.636363
```

OPERADORES LÓGICOS

Operador	Nome do Operador	Exemplo	Significado
>	Maior que	$X > Y$	X é maior que Y?
>=	Maior ou igual	$X \geq Y$	X é maior ou igual a Y?
<	Menor que	$X < Y$	X é menor que Y?
<=	Menor ou igual	$X \leq Y$	X é menor ou igual a Y?
==	Igualdade	$X == Y$	X é igual a Y?
!=	Diferente de	$X != Y$	X é diferente de Y?
and	E	X and Y	Será verdadeiro se X e Y forem verdadeiros
or	OU	X or Y	Será verdadeiro se X ou Y forem verdadeiros
not	Negação	not X	Retorna verdadeiro se receber uma expressão falsa, e vice-versa
is	Igualdade	X is Y	Retorna verdadeiro se X for igual a Y

OPERADORES LÓGICOS

$x = 5$

$y = 7$

$x > y \gg \text{Falso}$

$x \geq y \gg \text{Falso}$

$x < y \gg \text{Verdadeiro}$

$x \leq y \gg \text{Verdadeiro}$

$x == y \gg \text{Falso}$

$x \neq y \gg \text{Verdadeiro}$

OPERADORES LÓGICOS

Tabela Verdade

A	B	$\neg A$	$\neg B$	$A \wedge B$	$A \vee B$
V	V				
V	F				
F	V				
F	F				

OPERADORES LÓGICOS

Tabela Verdade

A	B	$\neg A$	$\neg B$	$A \wedge B$	$A \vee B$
V	V	F	F	V	V
V	F	F	V	F	V
F	V	V	F	F	V
F	F	V	V	F	F

OPERADORES LÓGICOS

z = Verdadeiro

w = Falso

z E w >>> Falso

z OU w >>> Verdadeiro

NÃO z >>> Falso

NÃO w >>> Verdadeiro

z É w >>> Falso

Na lógica proposicional, uma **tautologia** (do grego ταυτολογία) é uma fórmula proposicional que é verdadeira para todas as possíveis valorações de suas variáveis proposicionais. Por exemplo, a fórmula proposicional $(A) \vee (\neg A)$ ("A ou não-A") é uma tautologia, porque é verdadeira para todas as valorações de A.

Vamos praticar a tautologia



$$(A \wedge B) \vee (\neg A) \vee (\neg B)$$



Constantes



▶ O valor é imutável

▶ Por exemplo, o número 2 nunca terá outro sentido senão o de representar duas unidades

- ▶ Variáveis são como caixas, que podem guardar qualquer objeto
- ▶ Na computação, variáveis guardam números, letras, nomes, resultados, listas...
- ▶ Para guardar um valor, basta nomearmos a variável e usar o operador

=

minha_variavel = 10

sua_variavel = “A”

outra_variavel = “Paulo”



▶ Para mostrar o que existe guardado dentro de uma variável basta usar o comando `print ()`

```
print (minha_variavel)
```

```
print (sua_variavel)
```

```
print (outra_variavel)
```

- ▶ Existem regras para nomearmos uma variável em Python
- ▶ Deve SEMPRE começar com uma letra comum (a-z ou A-Z)
- ▶ `variavel`
- ▶ `lugar`
- ▶ `nome`

▶ Não podem começar com números

▶ 123

▶ 123variável

▶ 1nome



- um23
- um23variavel
- um_nome

▶ Não podem ter acentos

▶ variável

▶ número

▶ amanhã



- variavel
- numero
- amanha

▶ Não podem ter ç

▶ espaço

▶ açaí

▶ paçoca



- espaco
- acai
- pacoca

▶ Não podem ter espaços

▶ minha variável

▶ meu sistema

▶ nosso exemplo



- minha_variavel
- meu_sistema
- nosso_exemplo

▶ Não podem ter caracteres especiais, exceto o _ (mas ele não pode vir no início)

▶ `variavel*variável`

▶ `exemplo+valor`

▶ `_variável`



- `variavel_vezes_variavel`
- `Exemplo_mais_valor`
- `variavel`

▶ Boas práticas

▶ Utilizar letras minúsculas com _ separando as informações

▶ `nomecompleto`



`nome_completo`

▶ Utilizar letras maiúsculas para separas as informações

▶ `nomecompleto`



`nomeCompleto`

- ▶ O uso de uma variável em Python é simplificada, não precisamos preocupar em dizer para o sistema o tipo do dado que será guardado naquele local
- ▶ Contudo, para realizarmos operações, precisamos saber qual o tipo da variável
- ▶ Função `type()` → `int`, `float`, `char`, `str`, etc...

Tipo	Representação
int - inteiro	Números inteiros
float - real	Números com casas decimais
double	Números com uma quantidade maior de casas decimais
char - caractere	Um caractere, que pode ser letra, número ou caractere especial
str	Uma série de “chars”, formando uma frase ou um texto
bool - booleano	Variável lógica que assume somente os valores “True” e “False”
list - lista	Tipagem que agrupa vários tipos de variáveis em uma
tuple - tupla	É uma espécie de lista, mas não permite a alteração do valor após a criação
dict - dicionário	Tipagem que estabelece uma chave associada a um valor. (chave, valor)

► Função `type()` → `int`, `float`, `complex`, `str`, `list`, `dict`, etc...

```
print (type(1))    >>> <class 'int'>
print (type(1.1))  >>> <class 'float'>
print (type(1 + 1.1j)) >>> <class 'complex'>
print (type('Python')) >>> <class 'str'>
print (type([1, 1.1, 1 + 1.1j, 'Python'])) >>> <class 'list'>
```

► Com o conhecimento adquirido até o momento, escreva um Pseudocódigo, e um programa em linguagem de programação python que pergunte o nome de uma pessoa, salve-o em uma variável e imprima a mensagem:

“Olá NomeDaPessoa. Tenha um excelente dia!”

