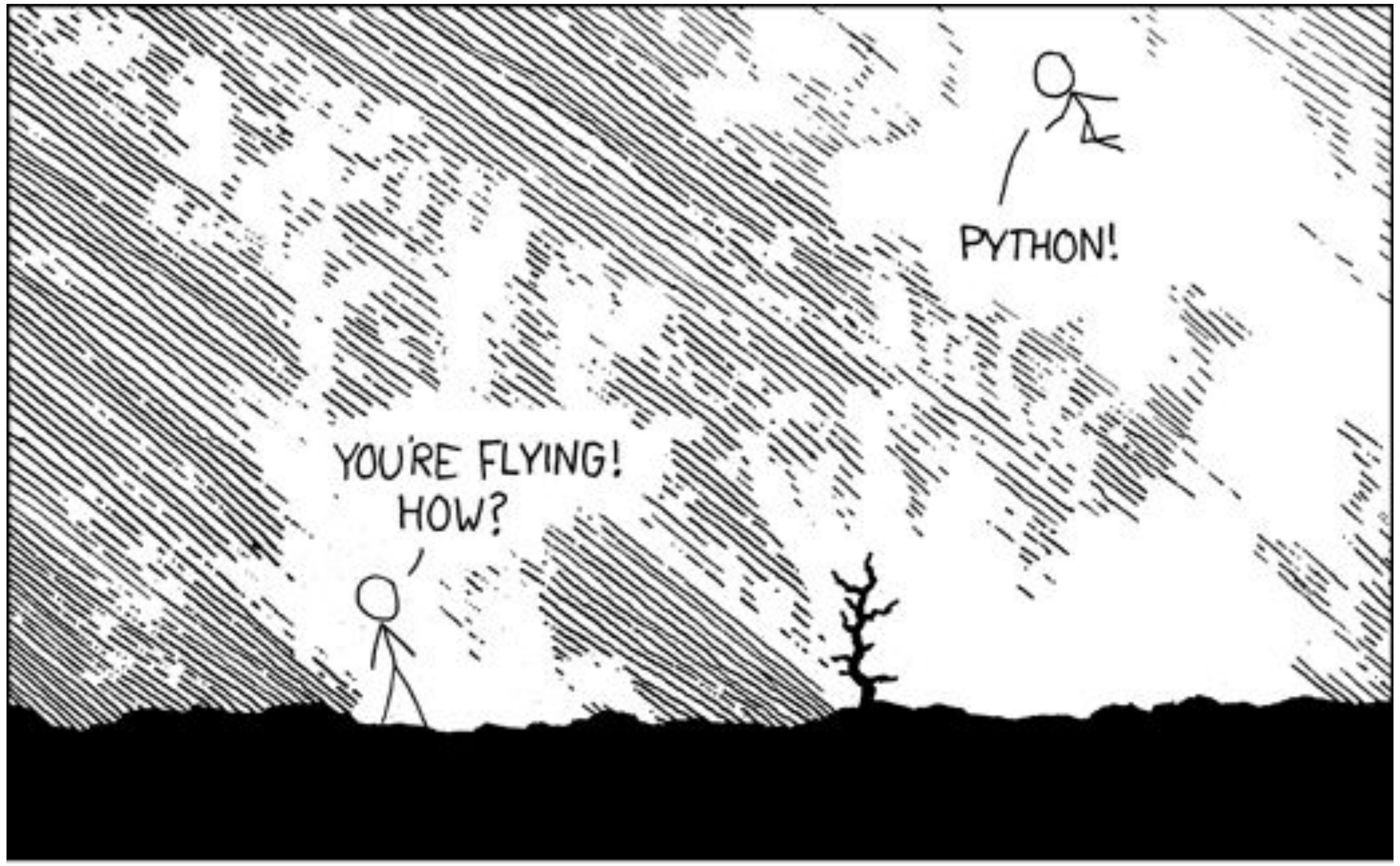
Python Scripting - Part 1

Spring 2023
PCfB Class 4
February 10, 2023

```
IMPORT OS
CUR_PATH = OS.GETCWD()
IGNORE_SET = SET(["__INIT__.PY",
"COUNT_SOURCELINES.PY"])
```

webs
YEAH, HE'S A PARSER-TONGUE.
I DIDN'T KNOW HARRY SPOKE PYTHON.



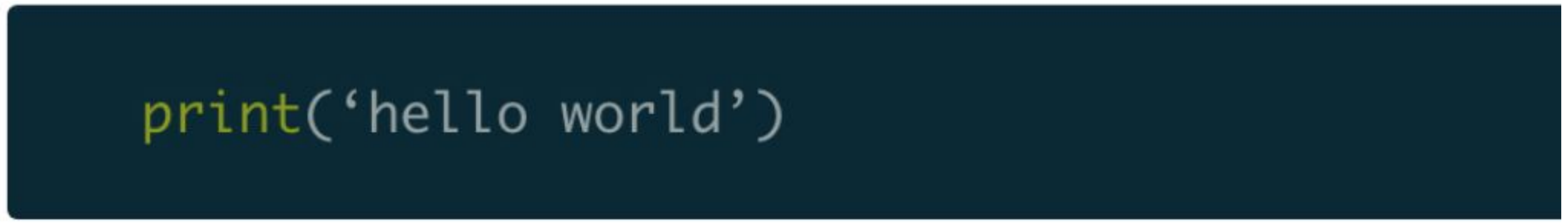
Outline

- Why Python?
- Data types
- Variables
- Methods

Why Python?

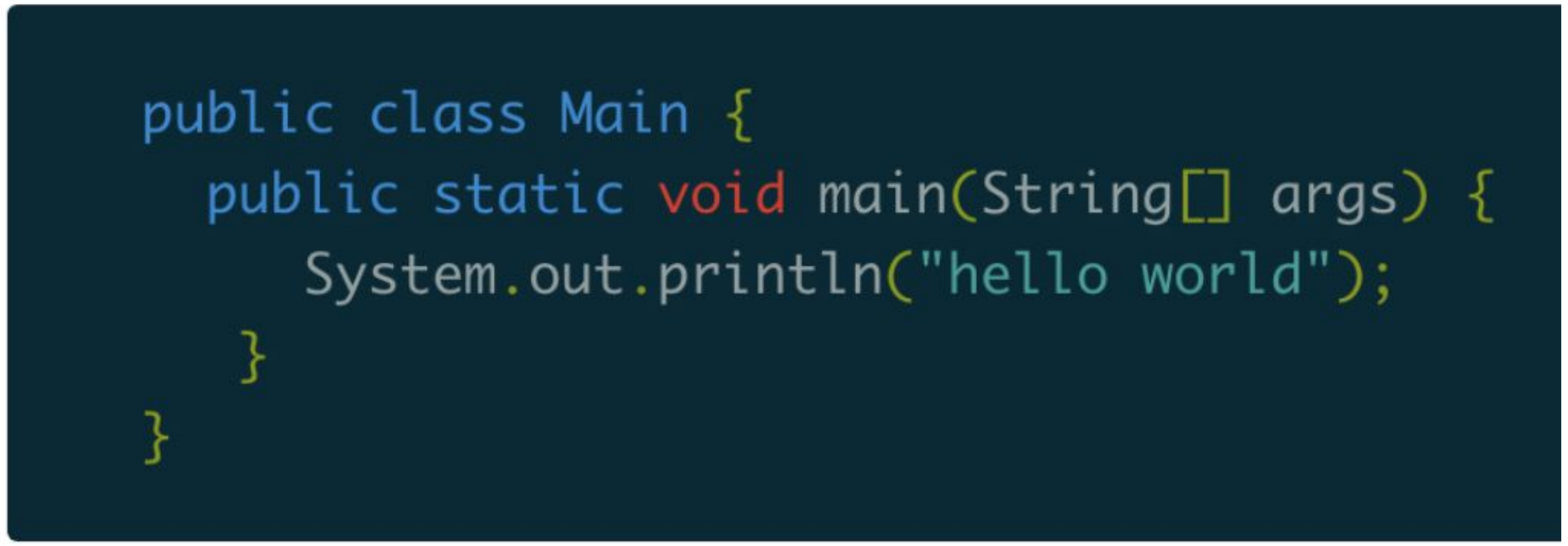
Enhanced readability

PYTHON

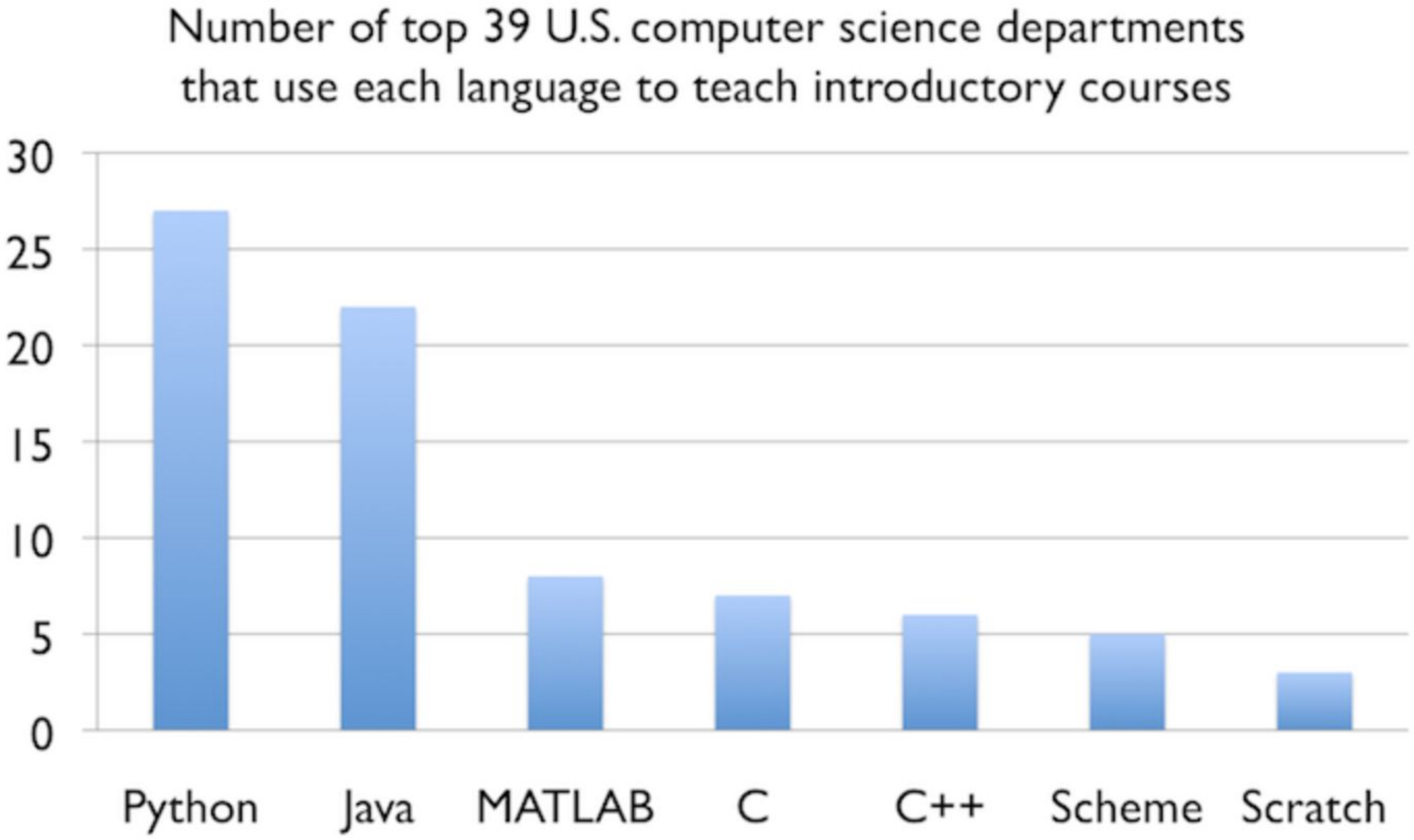


```
print('hello world')
```

JAVA



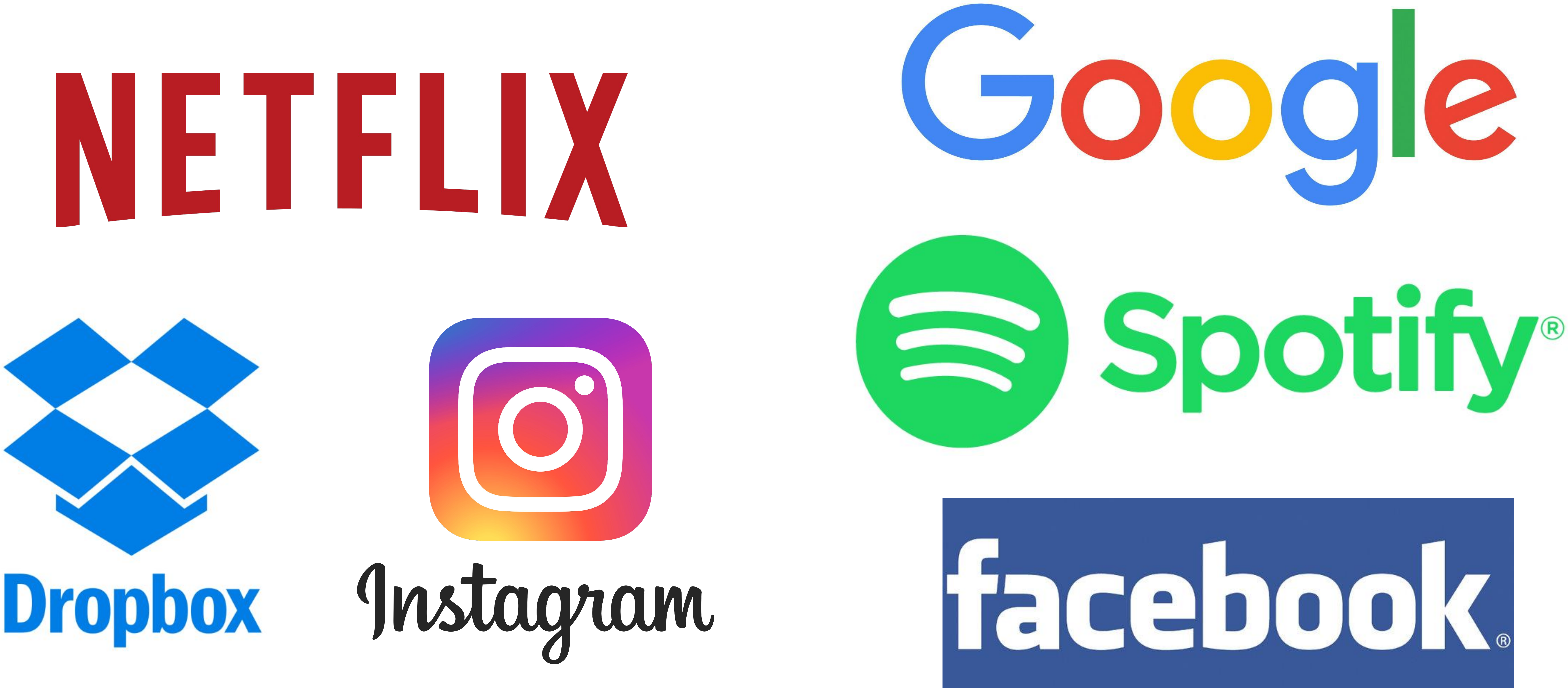
```
public class Main {
  public static void main(String[] args) {
     System.out.println("hello world");
   }
}
```



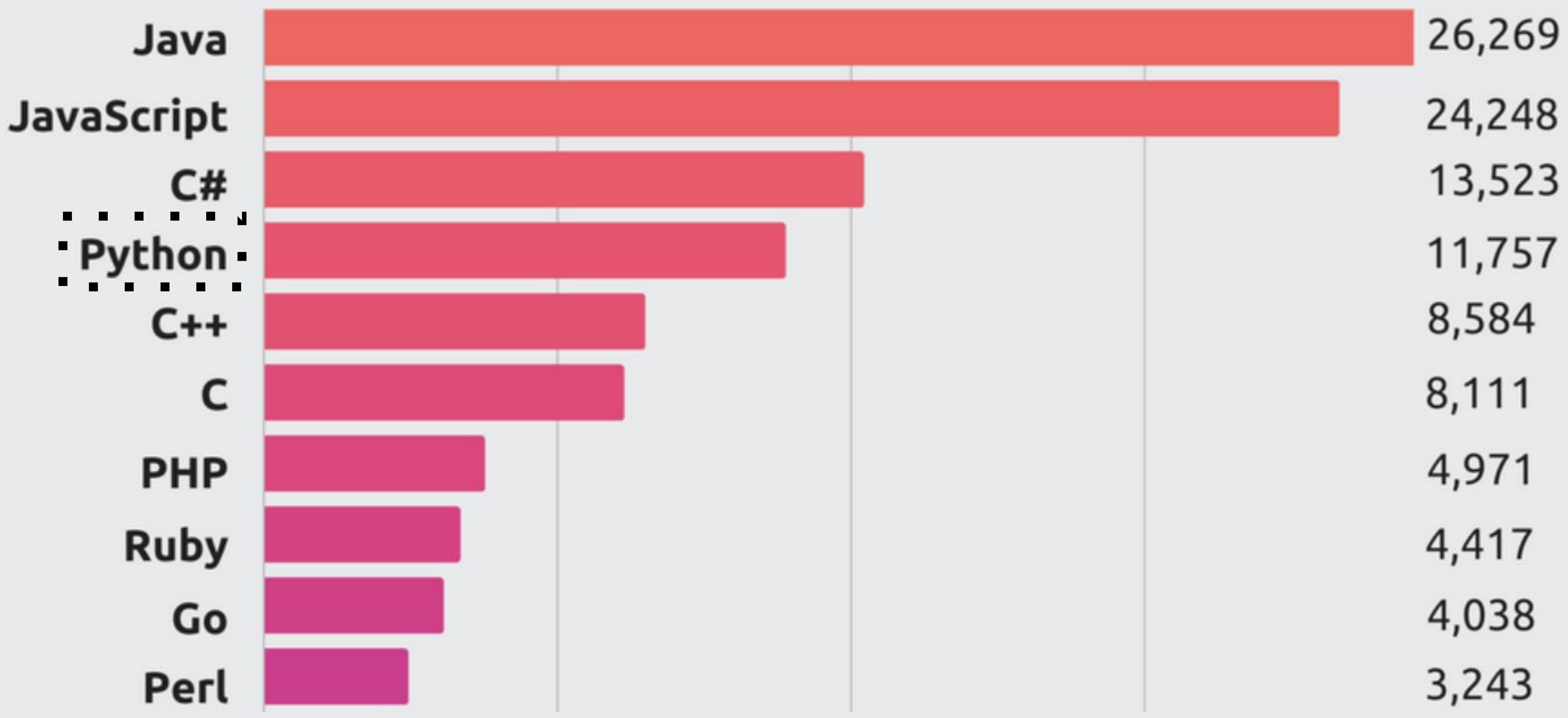
Number of top 39 U.S. computer science departments that use each language to teach introductory courses

Analysis done by Philip Guo (www.pgbovine.net) in July 2014, last updated 2014-07-29

Still very powerful

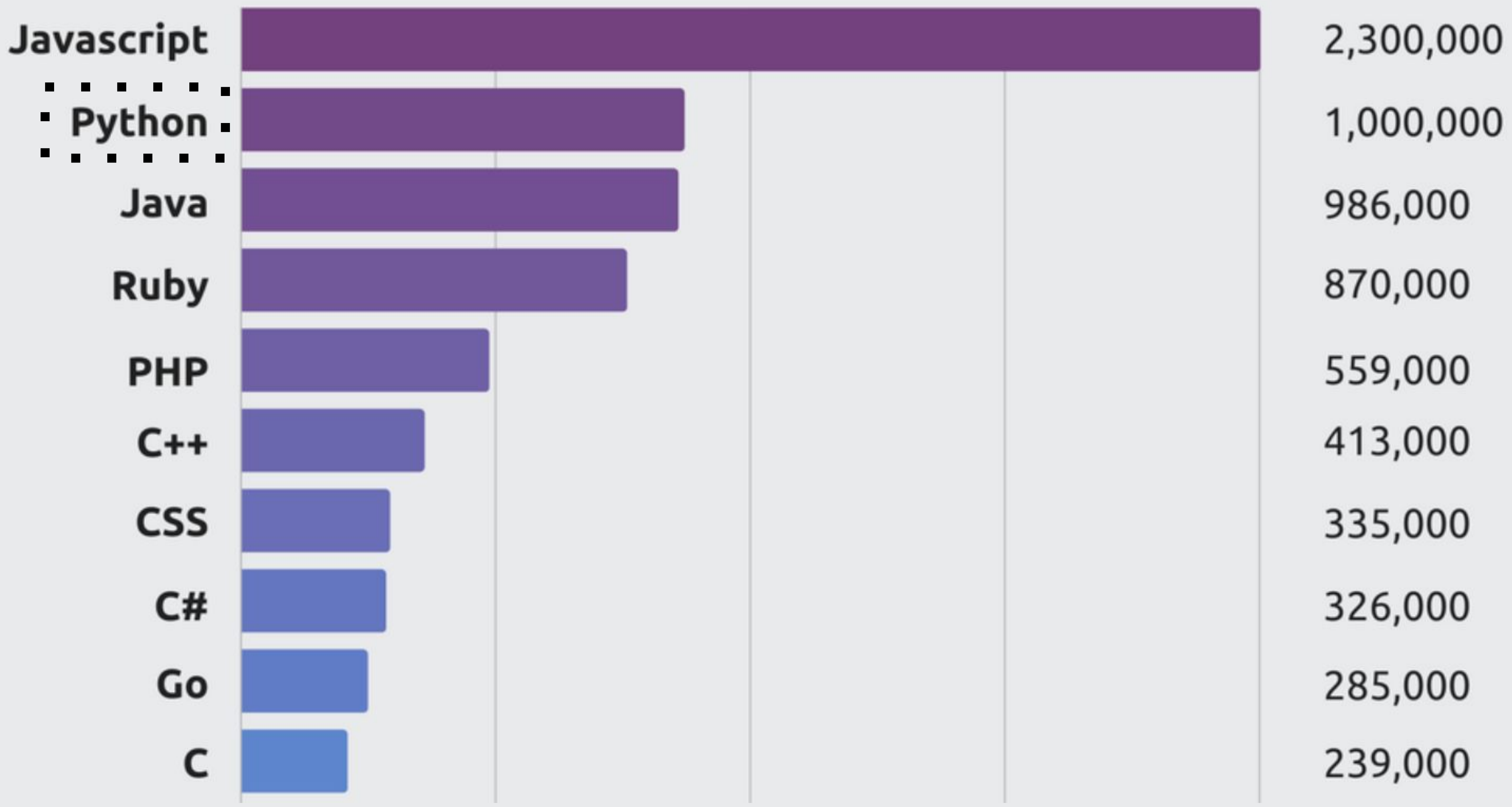


Very popular



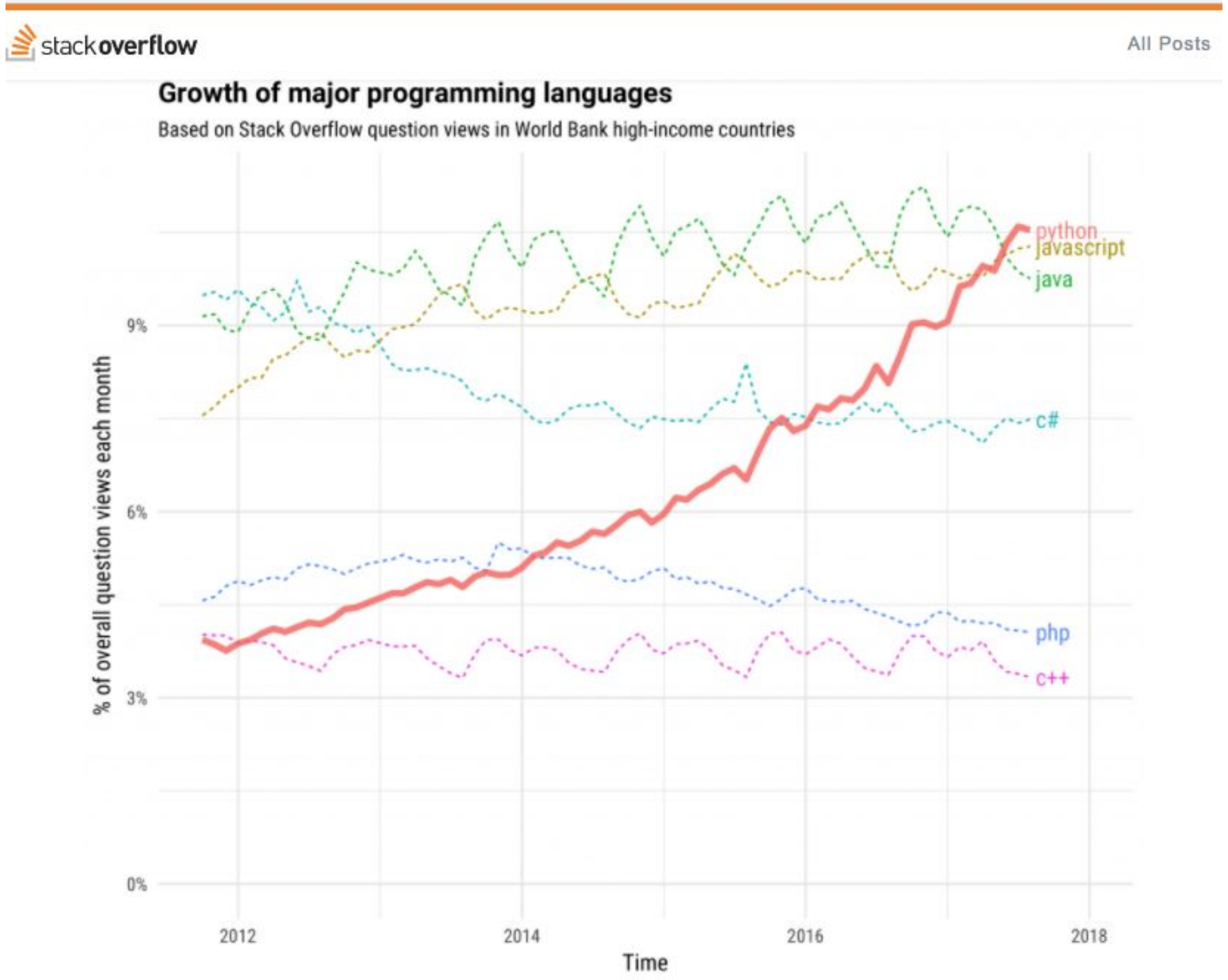
Most In-Demand Languages

Indeed Job Openings - Dec. 2017



Most Pull Requests 2017

GitHub

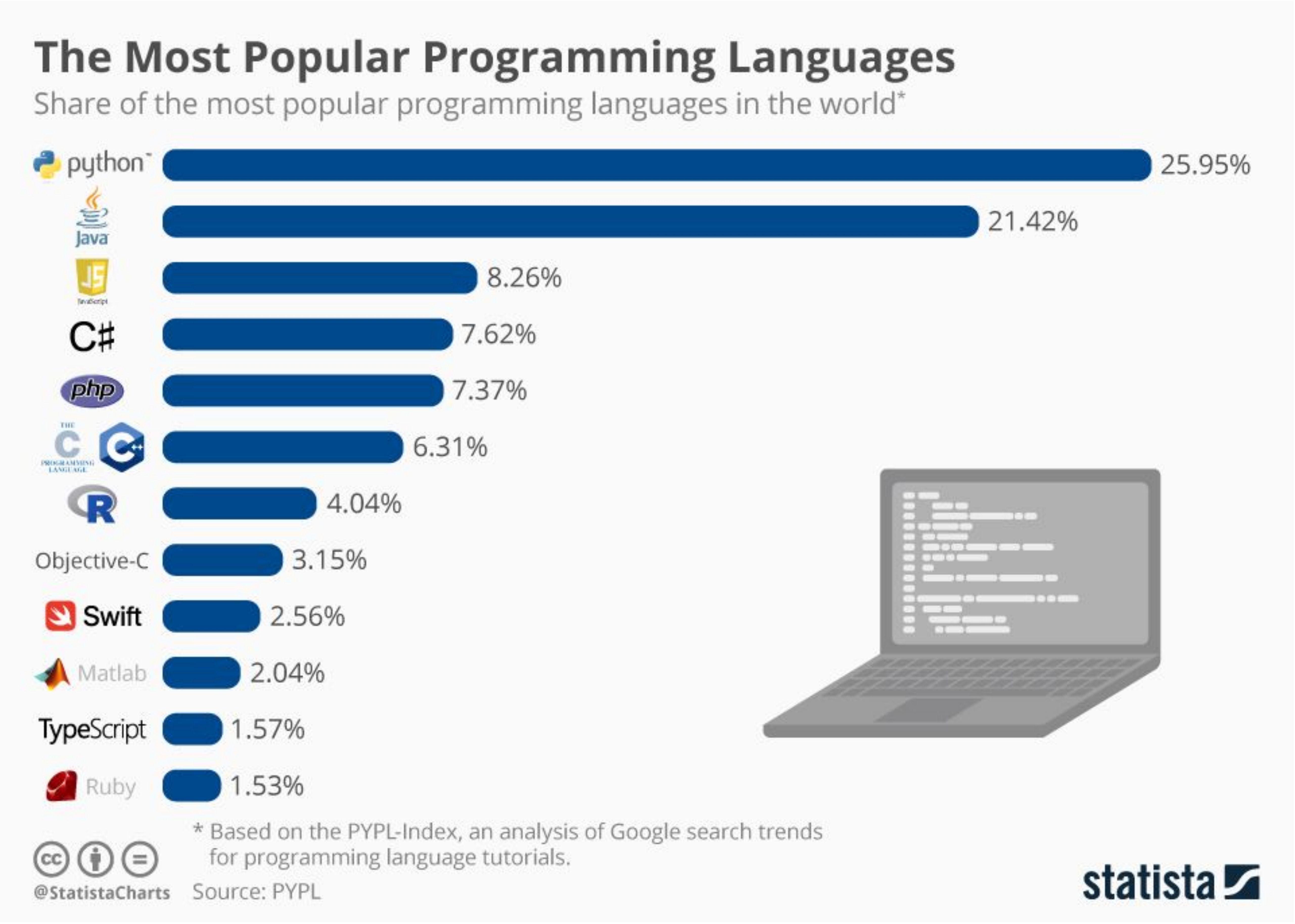


stackoverflow

All Posts

Growth of major programming languages

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



The Most Popular Programming Languages

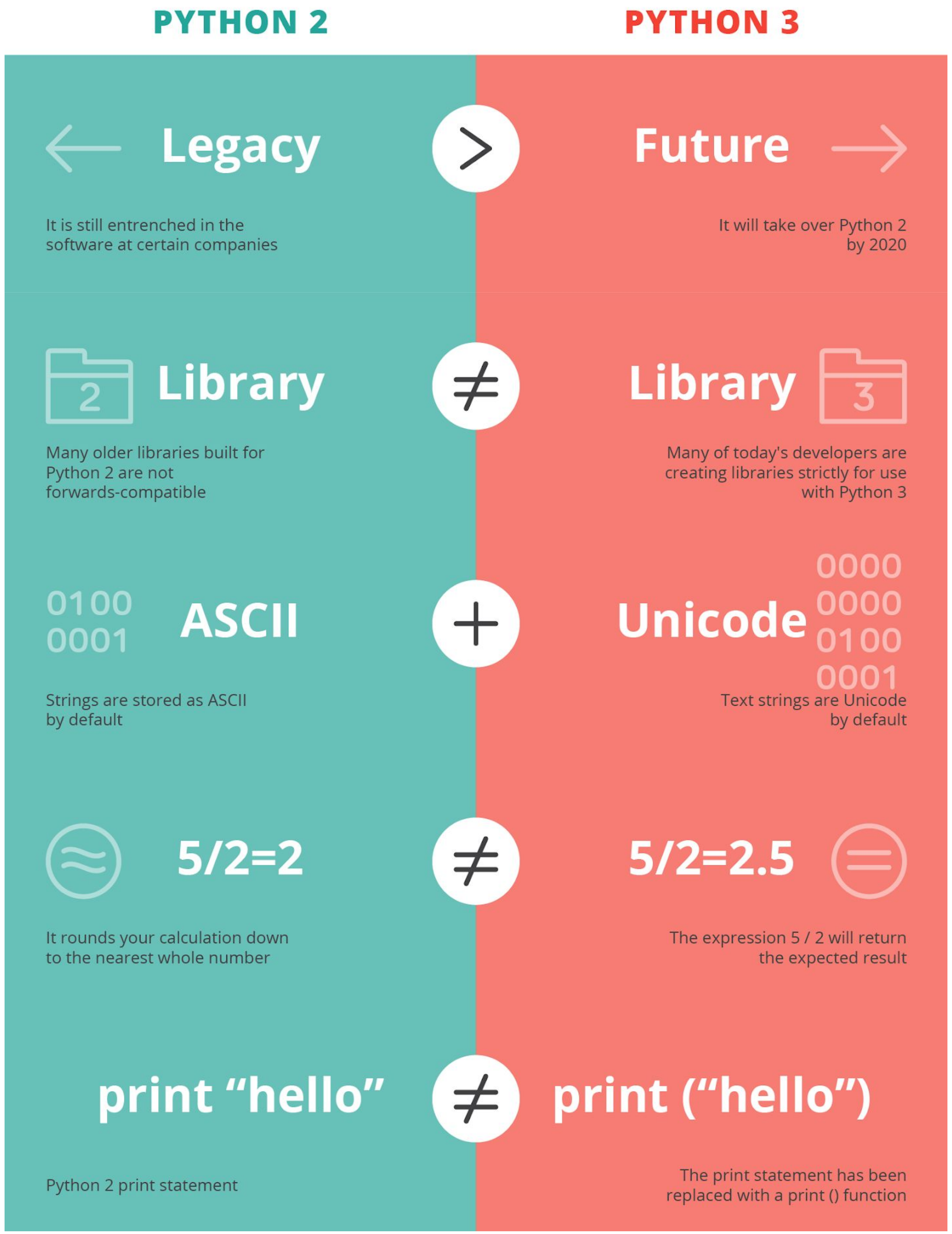
Share of the most popular programming languages in the world*

* Based on the PYPL-Index, an analysis of Google search trends for programming language tutorials.

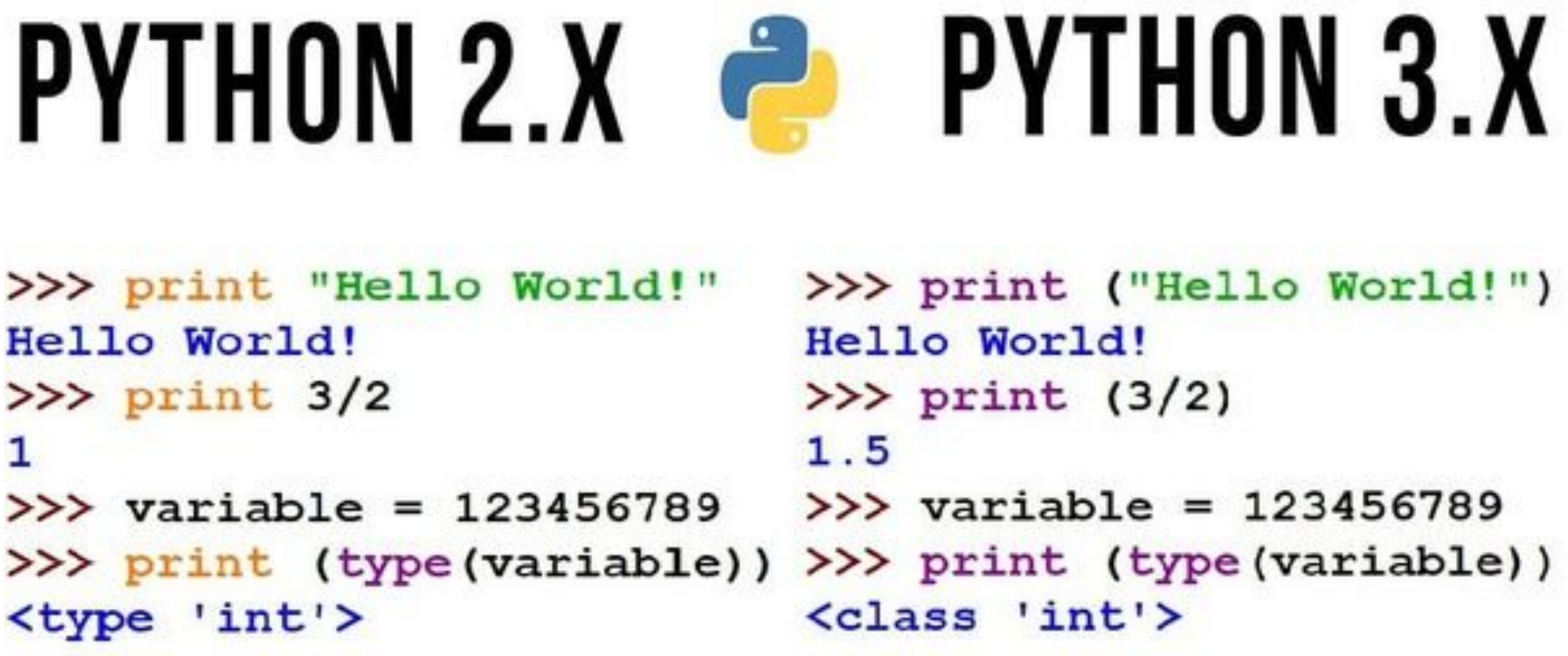
@StatistaCharts Source: PYPL

statista

python™ vs. R



PYTHON 2		PYTHON 3
Legacy It is still entrenched in the software at certain companies	>	**Future** It will take over Python 2 by 2020
Library Many older libraries built for Python 2 are not forwards-compatible	≠	**Library** Many of today's developers are creating libraries strictly for use with Python 3
0100 0001 **ASCII** Strings are stored as ASCII by default	+	**Unicode** 0000 0000 0100 0001 Text strings are Unicode by default
5/2=2 It rounds your calculation down to the nearest whole number	≠	**5/2=2.5** The expression 5 / 2 will return the expected result
print "hello" Python 2 print statement	≠	**print ("hello")** The print statement has been replaced with a print () function



PYTHON 2.X

```
>>> print "Hello World!"
Hello World!
>>> print 3/2
1
>>> variable = 123456789
>>> print (type(variable))
<type 'int'>
```

PYTHON 3.X

```
>>> print ("Hello World!")
Hello World!
>>> print (3/2)
1.5
>>> variable = 123456789
>>> print (type(variable))
<class 'int'>
```

Ways to use Python

1. Stand-alone scripts

- Code saved in text file, executed on command line
- As described in PCfB book

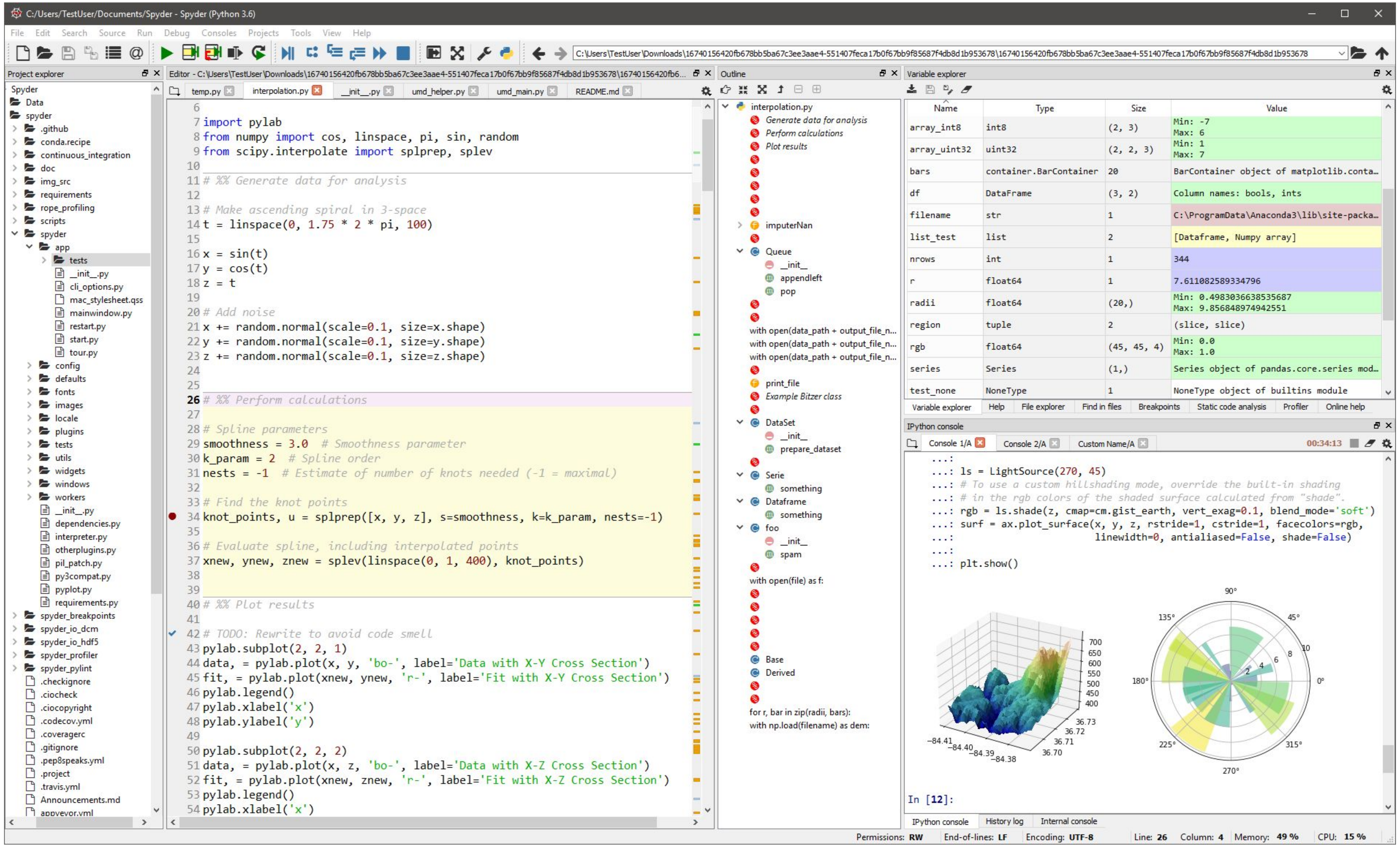
2. Interactive mode via command line

- Enter commands 1-by-1 on command line
- Good for testing

3. Jupyter notebook

- Rich, web-based interface; results presented inline
- Good for teaching purposes and sharing code

Interactive development environments (IDEs)



```python

import pylab
from numpy import cos, linspace, pi, sin, random
from scipy.interpolate import splprep, splev

# %% Generate data for analysis

# Make ascending spiral in 3-space
t = linspace(0, 1.75 * 2 * pi, 100)

x = sin(t)
y = cos(t)
z = t

# Add noise
x += random.normal(scale=0.1, size=x.shape)
y += random.normal(scale=0.1, size=y.shape)
z += random.normal(scale=0.1, size=z.shape)


# %% Perform calculations

# Spline parameters
smoothness = 3.0  # Smoothness parameter
k_param = 2  # Spline order
nests = -1  # Estimate of number of knots needed (-1 = maximal)

# Find the knot points
knot_points, u = splprep([x, y, z], s=smoothness, k=k_param, nests=-1)

# Evaluate spline, including interpolated points
xnew, ynew, znew = splev(linspace(0, 1, 400), knot_points)


# %% Plot results

# TODO: Rewrite to avoid code smell
pylab.subplot(2, 2, 1)
data, = pylab.plot(x, y, 'bo-', label='Data with X-Y Cross Section')
fit, = pylab.plot(xnew, ynew, 'r-', label='Fit with X-Y Cross Section')
pylab.legend()
pylab.xlabel('x')
pylab.ylabel('y')

pylab.subplot(2, 2, 2)
data, = pylab.plot(x, z, 'bo-', label='Data with X-Z Cross Section')
fit, = pylab.plot(xnew, znew, 'r-', label='Fit with X-Z Cross Section')
pylab.legend()
pylab.xlabel('x')
```

Name	Type	Size	Value
array_int8	int8	(2, 3)	Min: -7 Max: 6
array_uint32	uint32	(2, 2, 3)	Min: 1 Max: 7
bars	container.BarContainer	20	BarContainer object of matplotlib.conta…
df	DataFrame	(3, 2)	Column names: bools, ints
filename	str	1	C:\ProgramData\Anaconda3\lib\site-packa…
list_test	list	2	[Dataframe, Numpy array]
nrows	int	1	344
r	float64	1	7.611082589334796
radii	float64	(20,)	Min: 0.4983036638535687 Max: 9.856848974942551
region	tuple	2	(slice, slice)
rgb	float64	(45, 45, 4)	Min: 0.0 Max: 1.0
series	Series	(1,)	Series object of pandas.core.series mod…
test_none	NoneType	1	NoneType object of builtins module

```
   ...:
   ...: ls = LightSource(270, 45)
   ...: # To use a custom hillshading mode, override the built-in shading
   ...: # in the rgb colors of the shaded surface calculated from "shade".
   ...: rgb = ls.shade(z, cmap=cm.gist_earth, vert_exag=0.1, blend_mode='soft')
   ...: surf = ax.plot_surface(x, y, z, rstride=1, cstride=1, facecolors=rgb,
   ...:                        linewidth=0, antialiased=False, shade=False)
   ...:
   ...: plt.show()

In [12]:
```

SPYDER

PC

Data types

Data types

String

Integer

Floating point

Boolean

Converting between types

String

Integer

Floating point

Boolean

Data containers

List

```
[1, ‘1’, ‘one’, [1,2]]
```

Dictionary

```
{1:‘one’, 2:‘two’, 3:’three’}
```

Variables

Methods

Dot notation

dir()

```
['__add__', '__class__', '__contains__', '__delattr__', '__delitem__', '__dir__',
'__doc__', '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__', '__gt__',
'__hash__', '__iadd__', '__imul__', '__init__', '__init_subclass__', '__iter__', '__le__',
'__len__', '__lt__', '__mul__', '__ne__', '__new__', '__reduce__', '__reduce_ex__',
'__repr__', '__reversed__', '__rmul__', '__setattr__', '__setitem__', '__sizeof__',
'__str__', '__subclasshook__', 'append', 'clear', 'copy', 'count', 'extend', 'index', 'insert',
'pop', 'remove', 'reverse', 'sort']
```

#Comment, #comment, #comment

- Used to:
  - Guide others through your script
  - Indicate assumptions being made
  - Document changes made across versions
- You really can’t have too many comments!
- Most will probably be more useful to YOU than others

Demo