



PYTHON

- HIGH LEVEL
- INTERPRETED
- OBJECT ORIENTED
- DYNAMICALLY
TYPED

script.py IPython Shell

```
1 # Python program to check if the input number is prime or not
2
3 num = 407
4
5 # take input from the user
6 # num = int(input("Enter a number: "))
7
8 # prime numbers are greater than 1
9 if num > 1:
10     # check for factors
11     for i in range(2,num):
12         if (num % i) == 0:
13             print(num,"is not a prime number")
14             print(i,"times",num//i,"is",num)
15             break
16     else:
17         print(num,"is a prime number")
18
19 # if input number is less than
20 # or equal to 1, it is not prime
21 else:
22     print(num,"is not a prime number")
```

Run

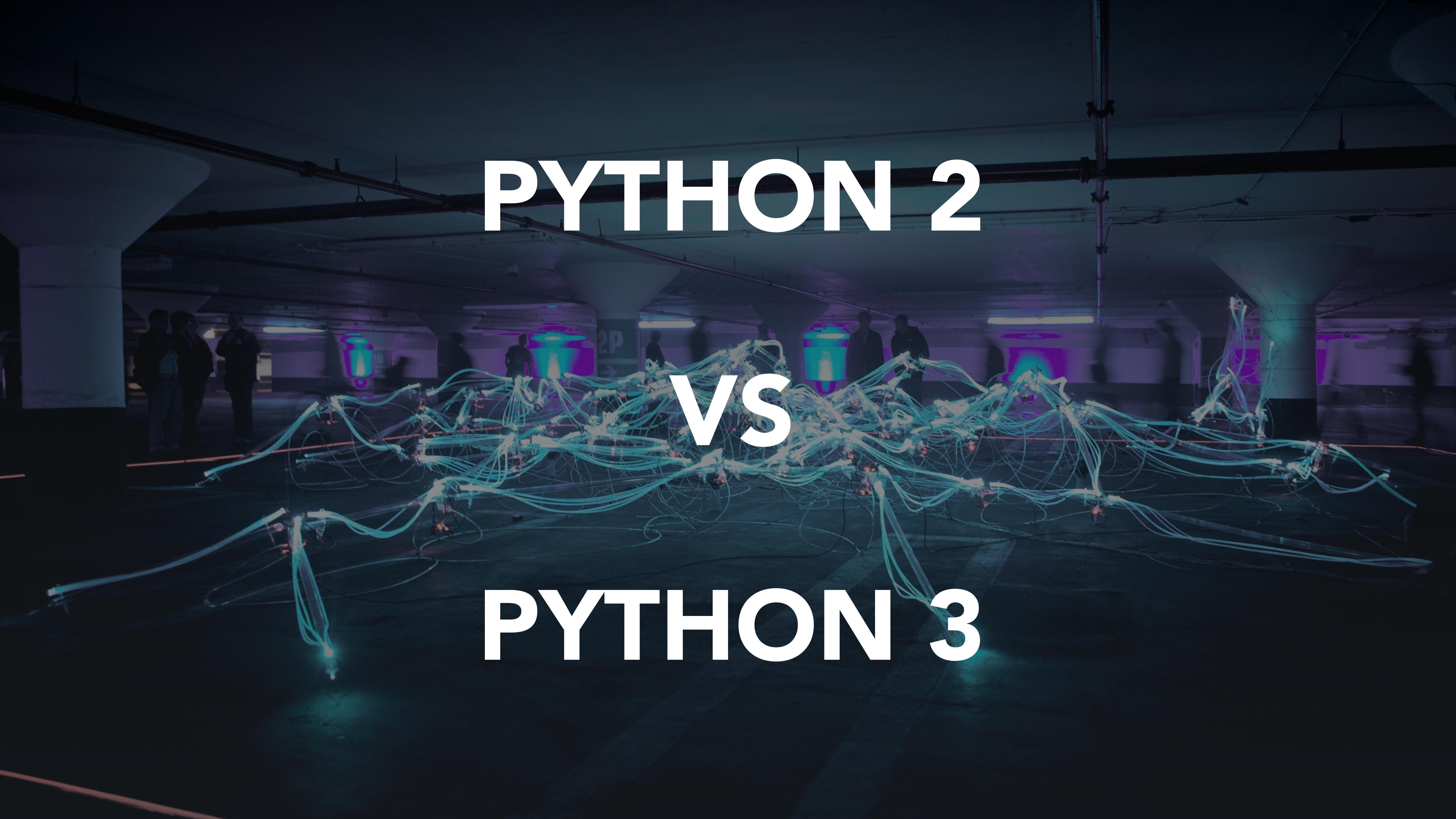




PRINT "HELLO"

VS

PRINT("HELLO")



PYTHON 2
VS
PYTHON 3



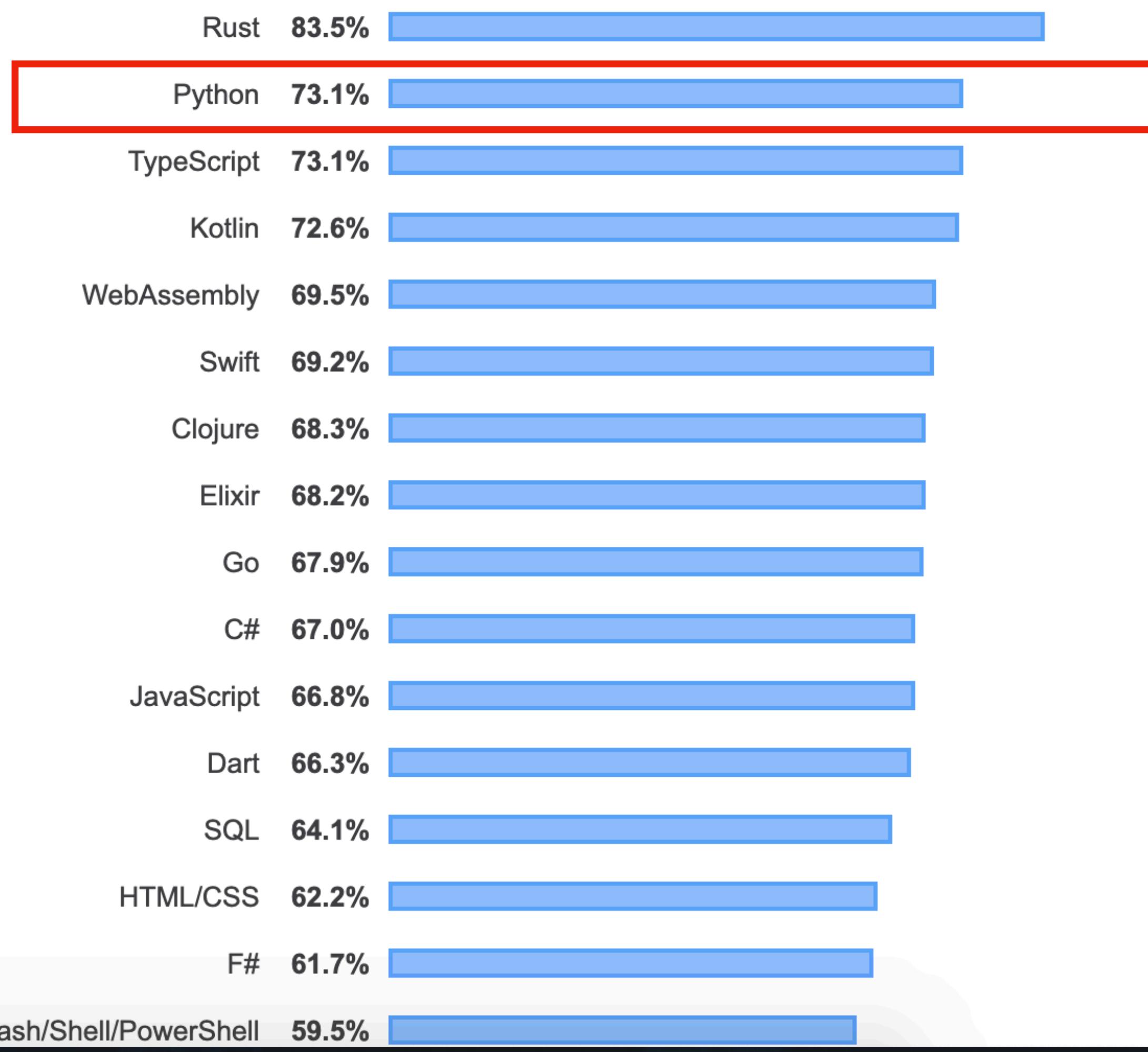
Most Loved, Dreaded, and Wanted

Most Loved, Dreaded, and Wanted Languages

Loved

Dreaded

Wanted

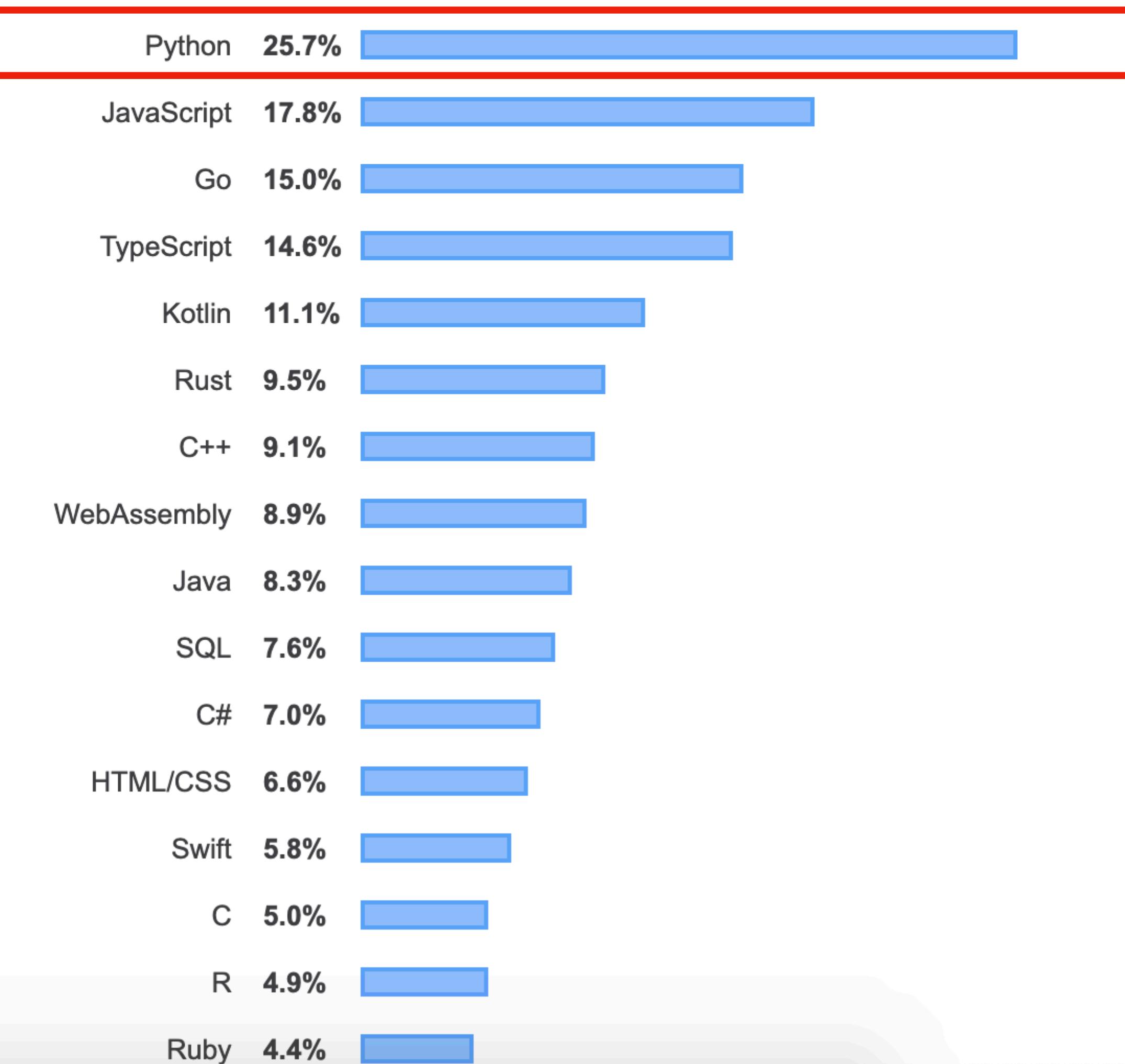




Most Loved, Dreaded, and Wanted

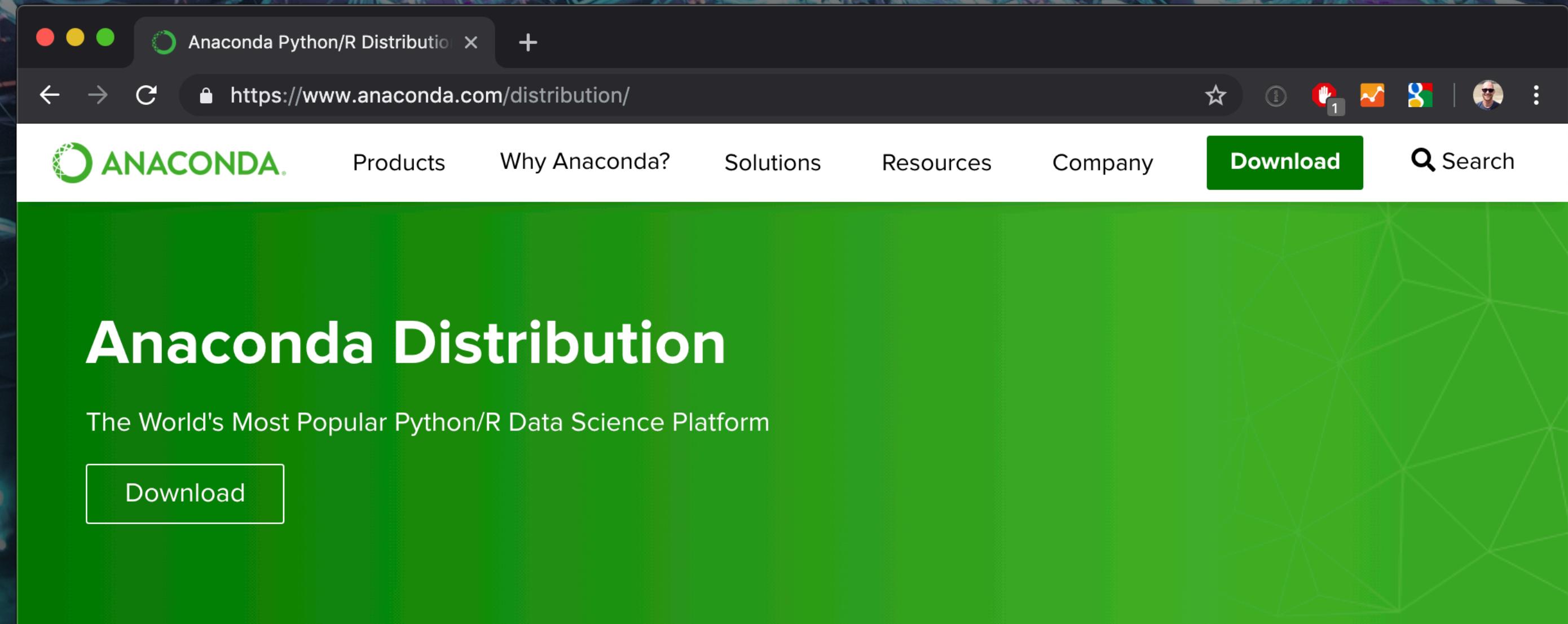
Most Loved, Dreaded, and Wanted Languages

Loved Dreaded **Wanted**



SETUP

USE ANACONDA



The screenshot shows a web browser displaying the Anaconda Python/R Distribution website at <https://www.anaconda.com/distribution/>. The page features a large green header with the text "Anaconda Distribution" and "The World's Most Popular Python/R Data Science Platform". Below the header is a "Download" button. The main content area contains a paragraph about the open-source Anaconda Distribution and logos for various scientific computing libraries: Jupyter, Spyder, NumPy, SciPy, and Numba.

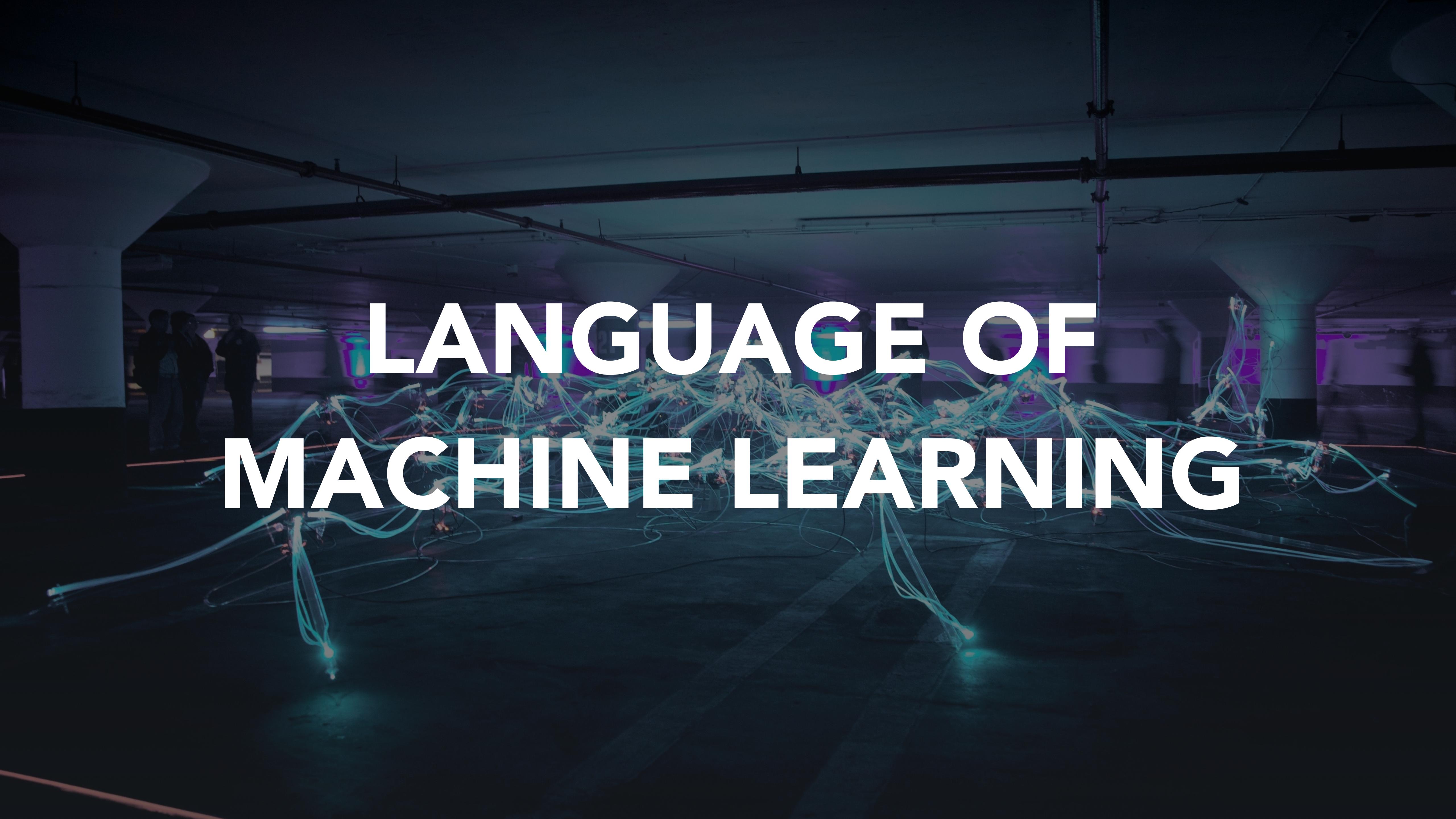
Anaconda Python/R Distribution

The World's Most Popular Python/R Data Science Platform

Download

The open-source **Anaconda Distribution** is the easiest way to perform Python/R data science and machine learning on Linux, Windows, and Mac OS X. With over 11 million users worldwide, it is the industry standard for data science and machine learning.

jupyter spyder NumPy SciPy Numba



LANGUAGE OF MACHINE LEARNING

PACKAGES

- NUMPY
- PANDAS
- TENSORFLOW, PYTORCH,

...

JUPYTER NOTEBOOKS

IP[y]: Notebook spectrogram Last Checkpoint: a few seconds ago (autosaved) IPython (Python 3)

File Edit View Insert Cell Kernel Help

Cell Toolbar: None

Simple spectral analysis

An illustration of the [Discrete Fourier Transform](#) using windowing, to reveal the frequency content of a sound signal.

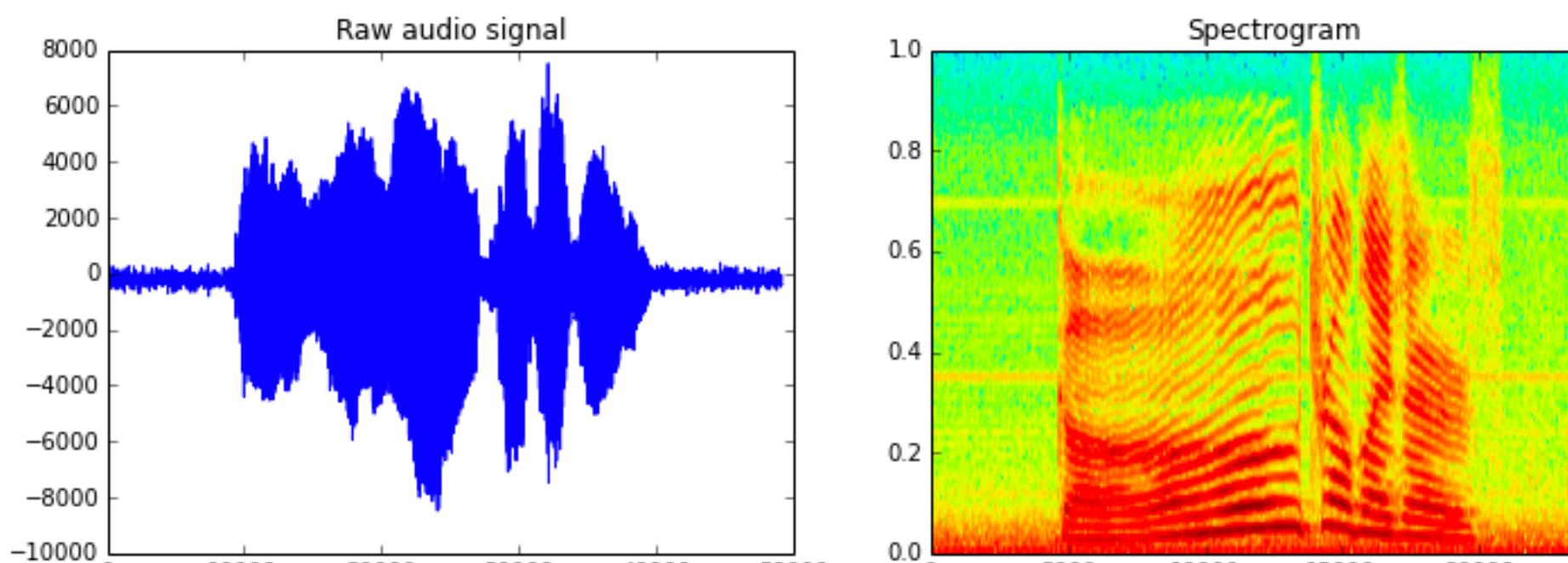
$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi}{N} kn} \quad k = 0, \dots, N - 1$$

We begin by loading a datafile using SciPy's audio file support:

```
In [1]: from scipy.io import wavfile  
rate, x = wavfile.read('test_mono.wav')
```

And we can easily view its spectral structure using matplotlib's builtin specgram routine:

```
In [2]: %matplotlib inline  
from matplotlib import pyplot as plt  
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))  
ax1.plot(x); ax1.set_title('Raw audio signal')  
ax2.specgram(x); ax2.set_title('Spectrogram');
```



ADVANTAGES

- "REAL" PROGRAMMING LANGUAGE
- EASY TO LEARN AND USE
- UNIVERSALLY APPLICABLE
- PACKAGES FOR EVERYTHING
- MOST COMMON LANGUAGE FOR MACHINE LEARNING (EXAMPLES, PACKAGES,...)

DISADVANTAGES

- NOT BEST IN PERFORMANCE
(INTERPRETED)
- (THERE ARE BETTER LANGUAGES FOR
WEB-APPLICATIONS)



WANNA DO DATA
SCIENCE OR ML?
USE PYTHON!