
PYTHON

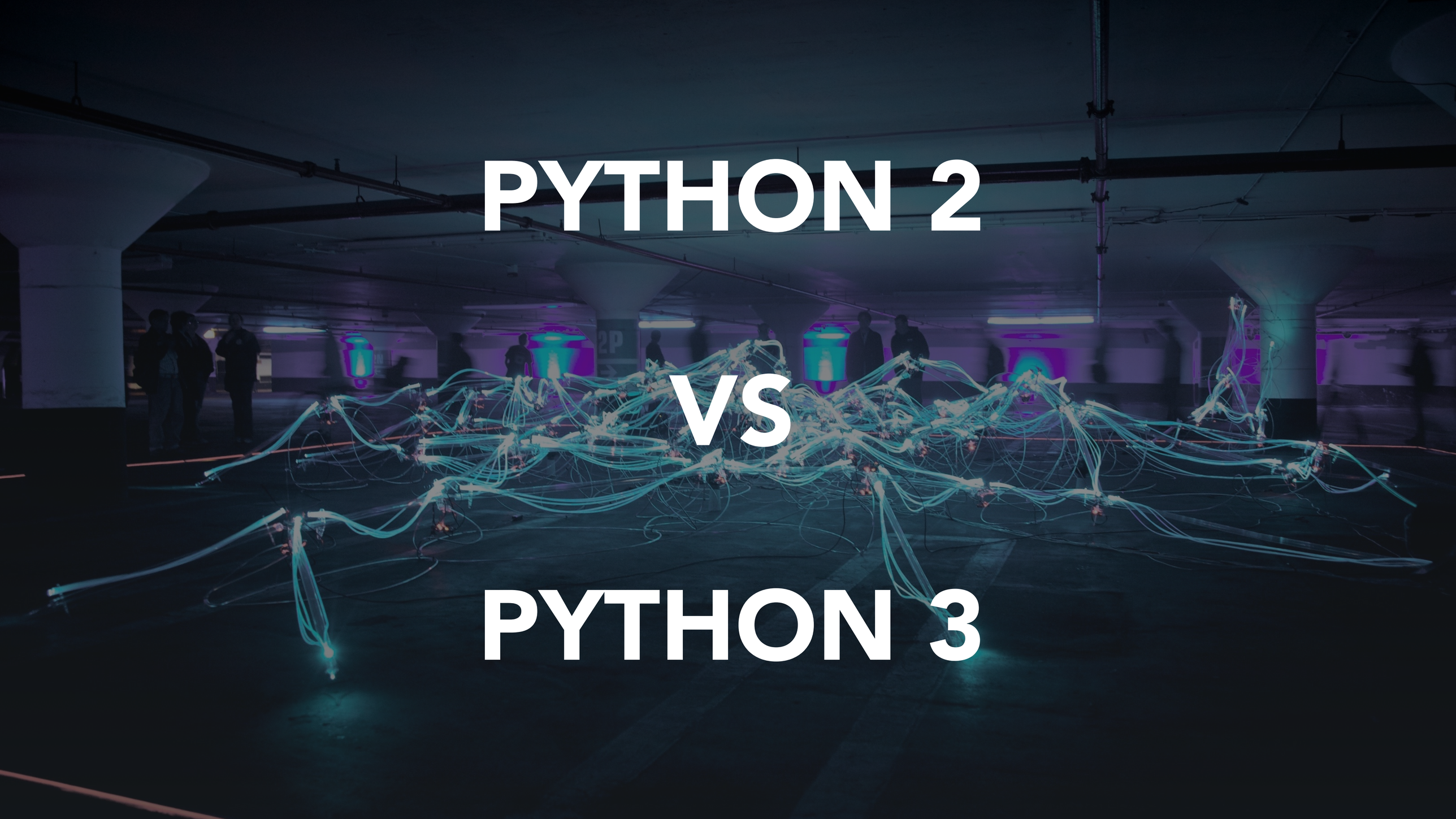

- HIGH LEVEL
- INTERPRETED
- OBJECT ORIENTED
- DYNAMICALLY TYPED

script.py IPython Shell

```python
# Python program to check if the input number is prime or not

num = 407

# take input from the user
# num = int(input("Enter a number: "))

# prime numbers are greater than 1
if num > 1:
    # check for factors
    for i in range(2,num):
        if (num % i) == 0:
            print(num,"is not a prime number")
            print(i,"times",num//i,"is",num)
            break
    else:
        print(num,"is a prime number")

# if input number is less than
# or equal to 1, it is not prime
else:
    print(num,"is not a prime number")
```

Run

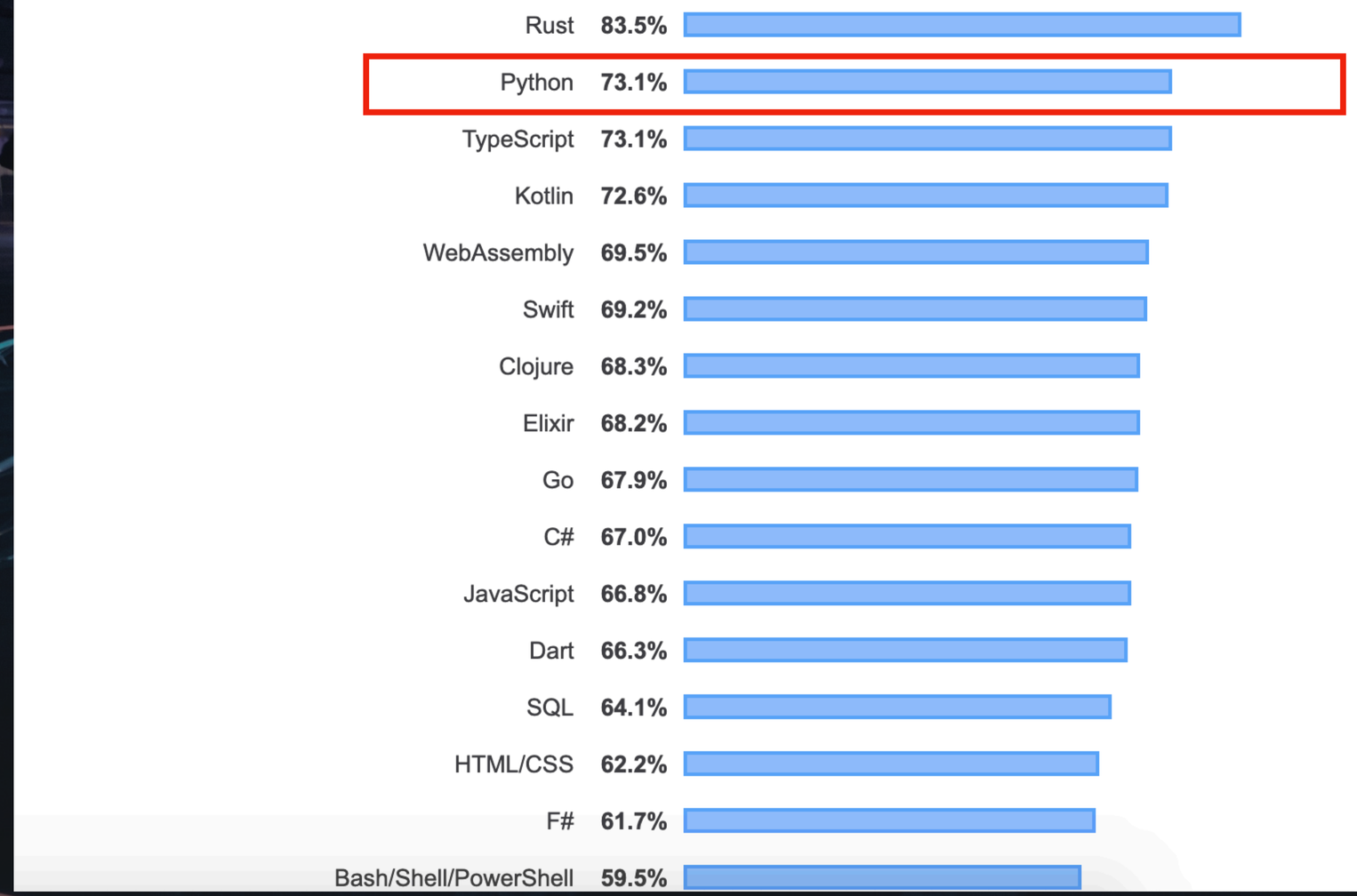

PRINT “HELLO”
VS
PRINT(“HELLO”)


PYTHON 2
VS
PYTHON 3

Most Loved, Dreaded, and Wanted

Most Loved, Dreaded, and Wanted Languages

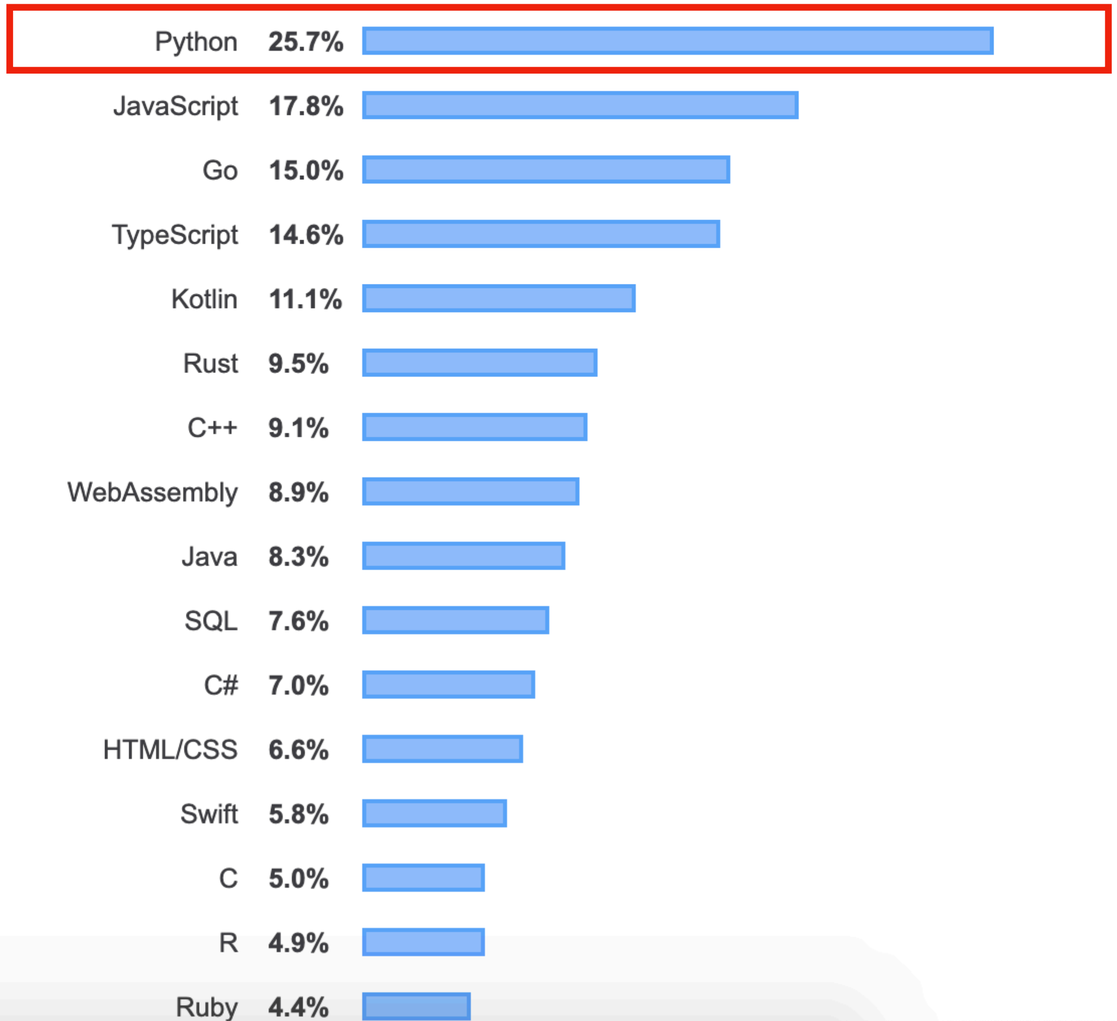
Loved | Dreaded | Wanted



Most Loved, Dreaded, and Wanted

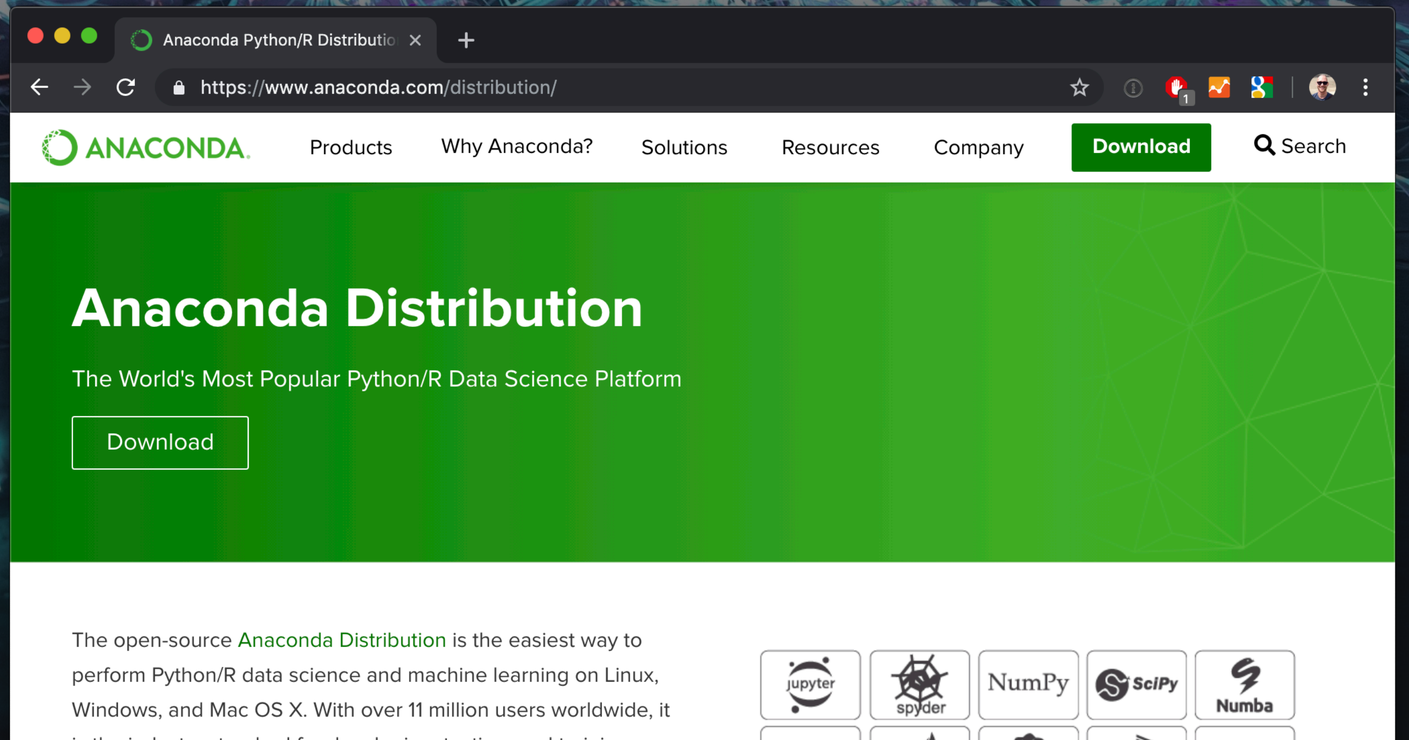
Most Loved, Dreaded, and Wanted Languages

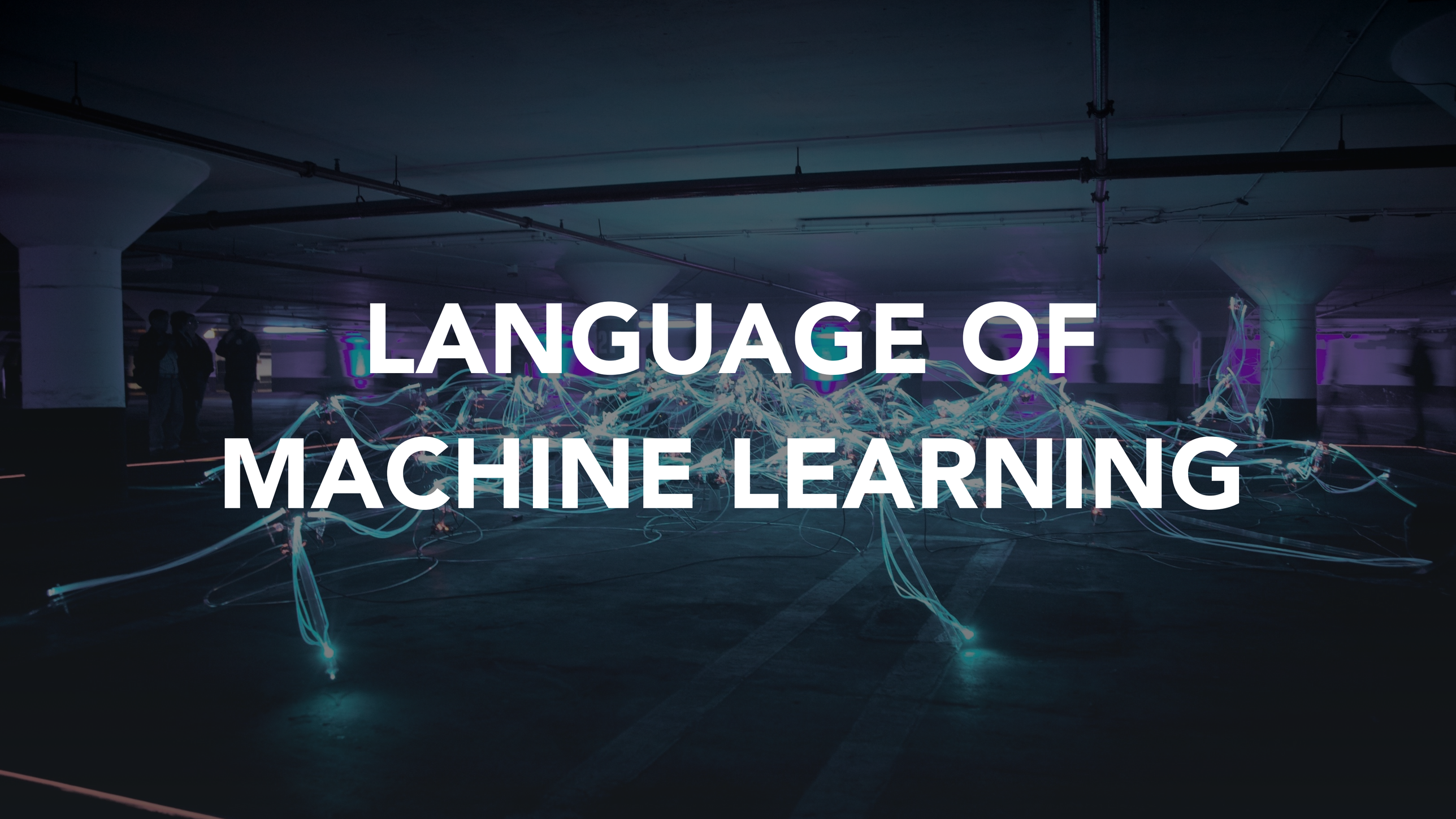
Loved | Dreaded | Wanted

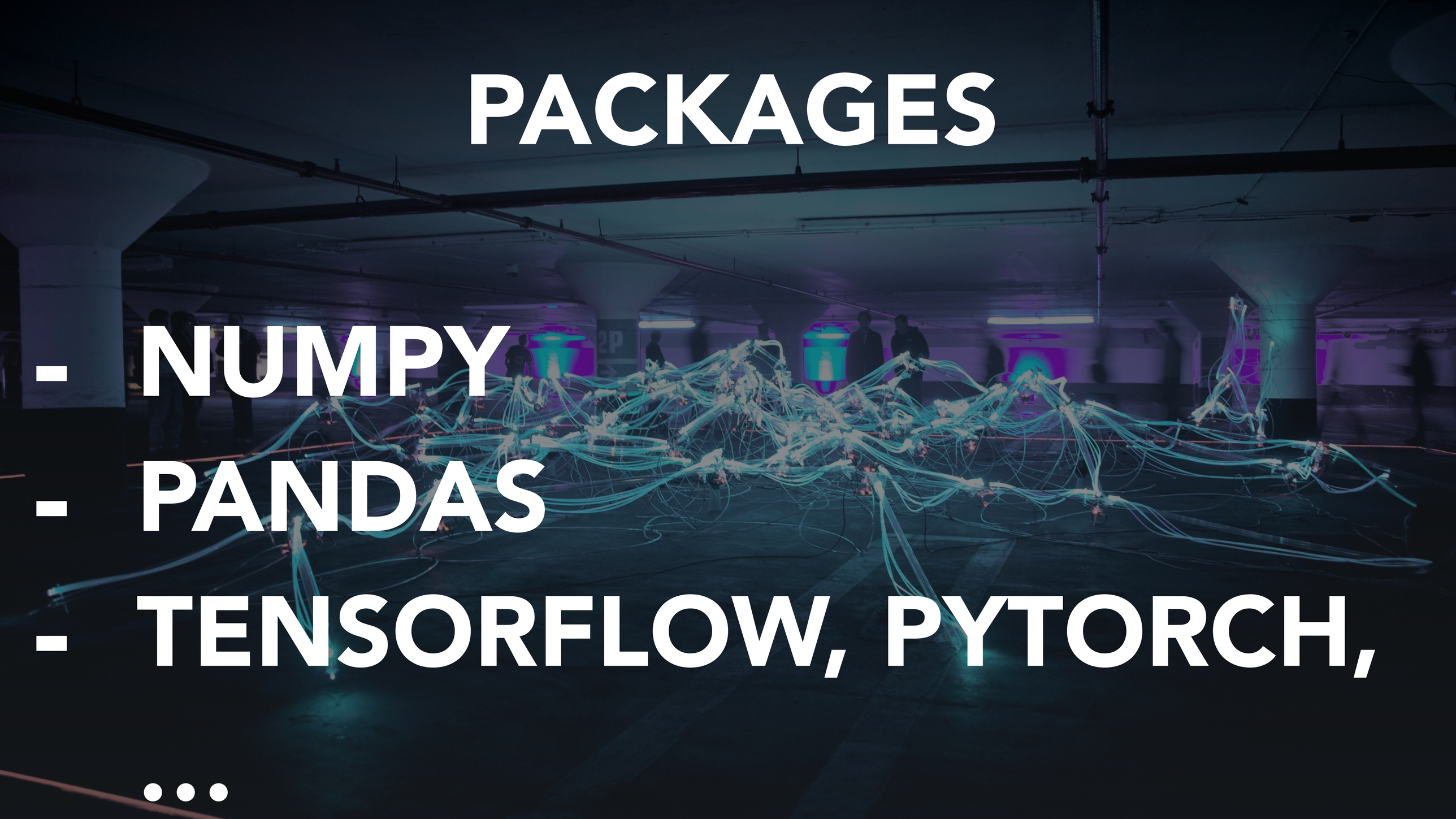


SETUP

USE ANACONDA




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

Code Cell Toolbar: None

Simple spectral analysis

An illustration of the Discrete Fourier Transform using windowing, to reveal the frequency content of a sound signal.

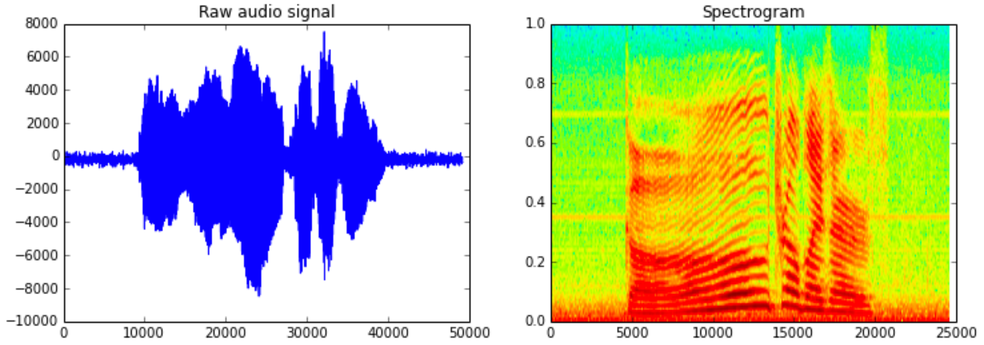
$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \qquad k = 0, \ldots, N-1$$

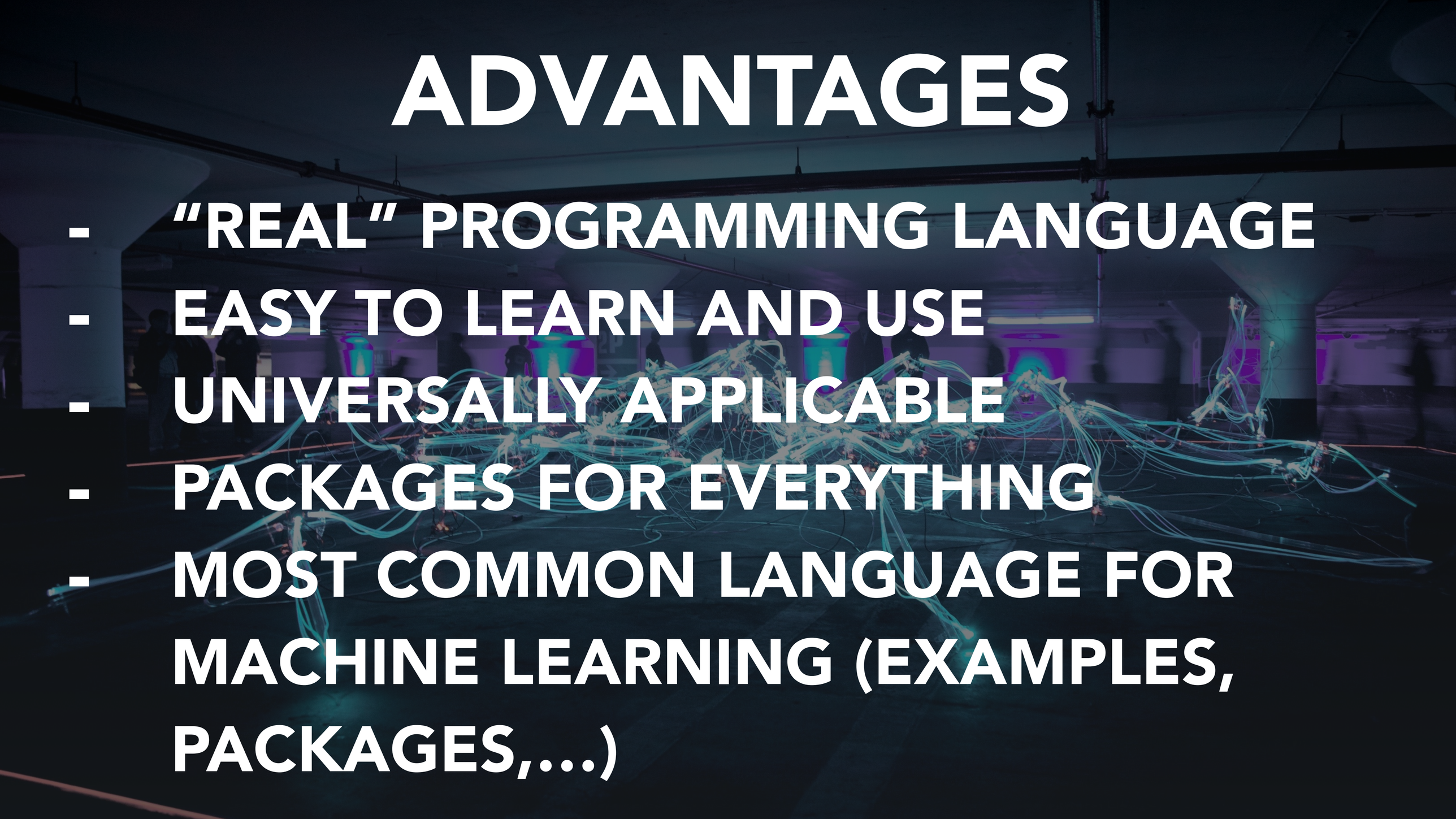
We begin by loading a datafile using SciPy's audio file support:

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!