About me

Born and raised in Caracas, Venezuela

Electrical Engineer by background

4th year CIS PhD student with Eric Eaton at Lifelong Machine Learning group
My path with Python

- **2014**: Senior year in college, Image processing project
- **2016**: At Penn, Day-to-day research
- **2017**: Summer, Data Science internship at Capital One
- **2019**: Summer, Research internship at Facebook
Why learn Python?

It is simple, yet powerful.
<table>
<thead>
<tr>
<th>Python</th>
<th>Java</th>
<th>C++</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>print('Hello world!')</code></td>
<td><code>class HelloWorld</code></td>
<td><code>#include&lt;iostream&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>{</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>public static void main(String args[])</code></td>
<td><code>using namespace std;</code></td>
</tr>
<tr>
<td></td>
<td><code>{</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>System.out.println(&quot;Hello world!&quot;);</code></td>
<td><code>int main()</code></td>
</tr>
<tr>
<td></td>
<td><code>}</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>{</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>}</code></td>
<td><code>cout &lt;&lt; &quot;Hello world!&quot;;</code></td>
</tr>
<tr>
<td></td>
<td><code>}</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>}</code></td>
<td><code>return 0;</code></td>
</tr>
</tbody>
</table>
Why learn Python?

- It is simple, yet powerful
- Data science and machine learning
- Web development
- Tons of libraries and frameworks
- Huge job market
- Great salaries
Data analysis and visualization

• NumPy — scientific computing
• Pandas — data analysis with table-like structures
• Matplotlib — 2D plotting library
Machine learning

• From classical machine learning...
  • scikit-learn
• ... to modern deep learning
  • TensorFlow
  • PyTorch
  • Keras
Machine learning applications in Python

- Image-to-image translation
- StarGAN
- OpenAI Gym
- Wit.AI
Why learn Python?

- It is simple, yet powerful
- Data science and machine learning
- Web development

Great salaries

Tons of libraries and frameworks

Huge job market
Web development with Python

• Great for server-side applications
• Web frameworks like Flask, Django, Pyramid...
• Focused on rapid development
Popular websites that use Python
Why learn Python?

- It is simple, yet powerful
- Data science and machine learning
- Web development
- Tons of libraries and frameworks
Wide range of packages

• Nearly 200,000 projects in the Python Package Index (PyPI)
  • https://pypi.org/
  • Install with `pip install ProjectName`
• We already saw examples for data analysis, machine learning, and web development
• Plus web scraping, GUI, image processing, games, natural language processing...
Why learn Python?

- It is simple, yet powerful
- Data science and machine learning
- Web development
- Tons of libraries and frameworks
- Huge job market
- Great salaries
<table>
<thead>
<tr>
<th>Language</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Python</td>
<td>25.7%</td>
</tr>
<tr>
<td>JavaScript</td>
<td>17.8%</td>
</tr>
<tr>
<td>Go</td>
<td>15.0%</td>
</tr>
<tr>
<td>TypeScript</td>
<td>14.6%</td>
</tr>
<tr>
<td>Kotlin</td>
<td>11.1%</td>
</tr>
<tr>
<td>Rust</td>
<td>9.5%</td>
</tr>
<tr>
<td>C++</td>
<td>9.1%</td>
</tr>
<tr>
<td>WebAssembly</td>
<td>8.9%</td>
</tr>
<tr>
<td>Java</td>
<td>8.3%</td>
</tr>
<tr>
<td>SQL</td>
<td>7.6%</td>
</tr>
<tr>
<td>C#</td>
<td>7.0%</td>
</tr>
<tr>
<td>HTML/CSS</td>
<td>6.6%</td>
</tr>
<tr>
<td>Swift</td>
<td>5.8%</td>
</tr>
<tr>
<td>C</td>
<td>5.0%</td>
</tr>
<tr>
<td>R</td>
<td>4.9%</td>
</tr>
<tr>
<td>Ruby</td>
<td>4.4%</td>
</tr>
<tr>
<td>Scala</td>
<td>4.3%</td>
</tr>
<tr>
<td>Bash/Shell/PowerShell</td>
<td>4.0%</td>
</tr>
<tr>
<td>Dart</td>
<td>3.8%</td>
</tr>
<tr>
<td>PHP</td>
<td>3.5%</td>
</tr>
<tr>
<td>Elixir</td>
<td>3.4%</td>
</tr>
<tr>
<td>F#</td>
<td>3.3%</td>
</tr>
<tr>
<td>Assembly</td>
<td>3.1%</td>
</tr>
<tr>
<td>Clojure</td>
<td>2.2%</td>
</tr>
<tr>
<td>Erlang</td>
<td>2.0%</td>
</tr>
</tbody>
</table>

StackOverflow 2019 survey

- Top language current developers want to learn
- Second language current users want to continue to use

https://insights.stackoverflow.com/survey/2019
Why learn Python?

- It is simple, yet powerful
- Data science and machine learning
- Web development
- Tons of libraries and frameworks
- Huge job market
- Great salaries
StackOverflow 2019 survey

- Languages above the line earn better than average given years of experience
- Size of the circles represents number of users

https://insights.stackoverflow.com/survey/2019
What we will we cover in this course

**Python specifics**
- Basic syntax
- Data types, data structures, functions, iterators, generators
- Function arguments, file I/O, functional programming, classes

**Environment**
- Pip, Conda environments
- Jupyter notebooks, Spyder, PyCharm, pydb

**Packages**
- NumPy, Pandas, Matplotlib — data analysis
- scikit-learn, Pytorch — machine learning
- Pillow — image processing
- NLTK — natural language processing
- Flask, Django — web development
Some logistics

• The syllabus will be continuously updated at https://www.seas.upenn.edu/~cis192/jorge/

• We will be using Piazza for discussions and announcements https://piazza.com/upenn/fall2019/cis192202/
  • Please post publicly whenever possible

• We will use Python 3.7 (latest stable release)

• Your main resource will be Python’s documentation https://docs.python.org/3/
Some logistics

• Grading will be based on 10 homework assignments and a final project
  • Details will be posted on the website
• Be sure to read the collaboration policy
  • Absolutely no code sharing outside of the team project!
Python
Python (from Wikipedia)

• Interpreted: not compiled
• High-level: abstracted away from hardware
• General-purpose: widest variety of application domain
• Readable: ease of comprehending purpose, control flow, and operation

https://en.wikipedia.org/wiki/Python_(programming_language)
Python (from Wikipedia)

- Dynamically typed: types are assigned at runtime
- Garbage-collected: unused objects’ memory is freed

https://en.wikipedia.org/wiki/Python_(programming_language)
Python (from Wikipedia)

• Procedural, object-oriented, and functional
• Often used for scripting

https://en.wikipedia.org/wiki/Python_(programming_language)
Indentation

- Whitespace indentation to delimit code blocks
  - Compare to Java’s use of curly braces
- May be achieved with tabs or spaces
  - Typically 4 spaces per block
- Must be consistent within a block
More on readability

• Semicolons are optional
• Comments start with #
• Comment blocks delimited with ’’’ or ’’’’’
Assignments

• Assignment operator: =
  • Example: \( x = 2 \)
  • No previous variable declaration (i.e., no \( \text{int } x \))
  • Assign reference of object \( \text{int} \) with value 2 to name \( x \)
  • Compare to Java: copy value 2 to storage memory assigned to \( x \) upon declaration
  • \( x \) can be assigned a differently typed object later

• Chained assignment is allowed
  • Example: \( a = b = 2 \)
  • Careful! Chained assignments point to the same object, so mutable variables will change for all references
Control flow

• **if** — for conditional execution
  • Along with **else** and **elif** (contraction of else if)
• **for** — for repeated execution over iterable objects
• **while** — for repeated execution until stopping condition
• All these statements must be separated from their corresponding code block by a colon (`:`)
• **break, continue** — for disrupting the flow of iterations
The **range()** function

- Creates an iterable over a range of integer values
  - Useful for for loops
- **range(stop)** — from 0 to stop-1
- **range(start, stop)** — from start to stop-1
- **range(start, stop, step)** — from start to stop-1 on steps of step
Expressions

• +, −, * — standard addition, substraction, multiplication
• ** — exponentiation
• / — floating point division
• // — integer division
Expressions

• `==, <, >, <=, >=` — compare elements by value
  • `==` compares objects by value too (like Java’s `equals()`)
  • `a <= b <= c` is equivalent to `a <= b` and `b <= c`
• `is` — compare by reference (identity)
  • Java’s `==` equivalent for objects
• `and, or, not` — Boolean operators
• `x if cond else y` — conditional expressions
  • Compare to Java’s `cond ? x : y`
Examples

```python
x = 2
if x == 1:
    print('x is 1 :(')
elif x == 2:
    print('x is 2 :)')
else:
    print('Oh-oh...')
```
Examples

```python
i = 0
while i < 10:
    if i == 5:
        print('Skip this...')
        i += 1
        continue
    print('{}-th iteration'.format(i))
    i += 1
```

```python
for i in range(10):
    if i == 5:
        print('Skip this...')
        continue
    print('{}-th iteration'.format(i))
```
Examples

```python
this compares x and y

x = 2 ** 3  # 8
y = 2 * 3    # 6
if y < 7 and 7 < x:
    print('Success!')
else:
    print('Oops...')
print(x, y)
```

This compares x and y

```python
x = 2 ** 3  # 8
y = 2 * 3    # 6
if y < 7 < x:
    print('Success!')
else:
    print('Oops...')
print(x, y)
```
Examples

```python
x = 10
y = 7
if x > y:
    z = x
else:
    z = y
print(z)
```

```python
x = 10
y = 7
z = x if x > y else y
print(z)
```
Takeaways

• Python is simple and powerful
• Available packages for wide array of problems: data science, machine learning, web, image processing, language processing
• Great job market
• A few initial details on Python syntax