CIS 192
Python programming
Instructor: Jorge Mendez
http://www.seas.upenn.edu/~cis192/jorge/
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
# Hello world!

<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>{ public static void main(String args[])</code></td>
<td><code>using namespace std;</code></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><code>{</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>cout &lt;&lt; &quot;Hello world!&quot;;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>return 0;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>}</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

Huge job market

Tons of libraries and frameworks
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

Great salaries
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
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

<table>
<thead>
<tr>
<th>Python specifics</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Basic syntax</td>
</tr>
<tr>
<td>• Data types, data structures, functions, iterators, generators</td>
</tr>
<tr>
<td>• Function arguments, file I/O, functional programming, classes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Pip, Conda environments</td>
</tr>
<tr>
<td>• Jupyter notebooks, Spyder, PyCharm, pydb</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Packages</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NumPy, Pandas, Matplotlib — data analysis</td>
</tr>
<tr>
<td>• scikit-learn, Pytorch — machine learning</td>
</tr>
<tr>
<td>• Pillow — image processing</td>
</tr>
<tr>
<td>• NLTK — natural language processing</td>
</tr>
<tr>
<td>• Flask, Django — web development</td>
</tr>
</tbody>
</table>
Some logistics

• Shared 19X lecture at Towne 100
  • 21 Jan: Linux command-line
  • 28 Jan: git & github
  • 4 Feb: internet protocols (like HTTP, DNS)
Some logistics

• We will use Python 3.7
  • 3.8 was released as the newest stable release in October 2019
  • Lab machines have Python 3.6.9

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

• No required development environment for the course
  • For early testing, you may use Jupyter notebooks
  • For turning in homework, you can use your favorite text editor
  • We will look into different IDEs later in the course
Some logistics

• The syllabus will be continuously updated at https://www.seas.upenn.edu/~cis192/jorge/
  • Slides and examples will be available after each class
  • Homeworks will be uploaded according to the syllabus

• We will be using Piazza for discussions and announcements https://piazza.com/upenn/spring2020/cis192202/home
  • Please post publicly whenever possible
  • Please join if you’re not already added
Some logistics

• Grading will be based on 8 homework assignments and a final project
  • Details 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_%28programming_language%29
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 int x)
  • Assign reference of object 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
Live example
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
Live example
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
Live example
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