Welcome to 1902: Python!

Jordan Schwartz

Some slides inspired by Tony Liu and David Cao

Week 1

Introductions
Why Python?
Logistics
Setting up
Getting started!

The special nature of a minicourse

Since we're a relatively small course, we have the advantage of getting to know and work with each other pretty closely.

Take advantage of that! Work together when allowed.

Let's introduce ourselves!

Jordan Schwartz (she/they)

Year: 1st Yr Masters

Major: CIS

Hometown: San Carlos, CA (SF bay area)

Undergrad: UC Berkeley, Cog Sci and CS

Email: jjschwa@seas.upenn.edu

Office Hours: TBD

Extracurriculars: Partner dancing (Swing, Salsa, Bachata, Contra, Blues, Fusion), cooking, crochet & knitting

Introducing your TA:

Introducing your TA:

Introduce yourself

- Name
- Pronouns (if you're comfortable sharing)
- Year
- Major
- How many pigeons could you hold (if they were cooperative) and what's your strategy?

Why Python?

- Simple, but powerful
  - Most commonly built off of C
    - High performance
    - Can use existing C libraries (like NumPy)

lecture.java

```java
class HelloWorld {
    public static void main(String args[]) {
        System.out.println("Hello world!");
    }
}
```

Why Python?

- Simple, but powerful
  - Most commonly built off of C
    - High performance
    - Can use existing C libraries (like NumPy)

lecture.cpp

```cpp
#include <iostream>

using namespace std;

int main() {
    cout << "Hello world!";
    return 0;
}
```

Why Python?

- Simple, but powerful
  - Most commonly built off of C
    - High performance
    - Can use existing C libraries (like NumPy)

lecture.py

```
1   if __name__ == '__main__':
2       print("Hello world!")
3
```

Why Python?

✅ Simple but powerful

→ Data Science

Data Science with Python

- Libraries/packages commonly used for data exploration and visualization:
  - NumPy - arrays and computation
  - Pandas - tables and data analysis
  - Matplotlib - plotting
  - Jupyter/Colab - notebook to write Python code in

Data Visualization/Interaction

- Gapminder visualization (https://demo.bokeh.org/gapminder)
- Streamlit image processing (https://bgremoval.streamlit.app/)

Machine Learning with Python

- Traditional ML – **Scikit-learn**
  - Supervised and Unsupervised learning, Classification, Regression
- Deep Learning – **Keras, Pytorch, Tensorflow**
  - Neural Networks (NNs), Convolutional Neural Networks (CNNs)
- Reinforcement Learning – **Gymnasium**
  - Value-Based, Policy-Based, Actor-Critic, Model-Based RL

Machine Learning with Python: Research

Example Projects

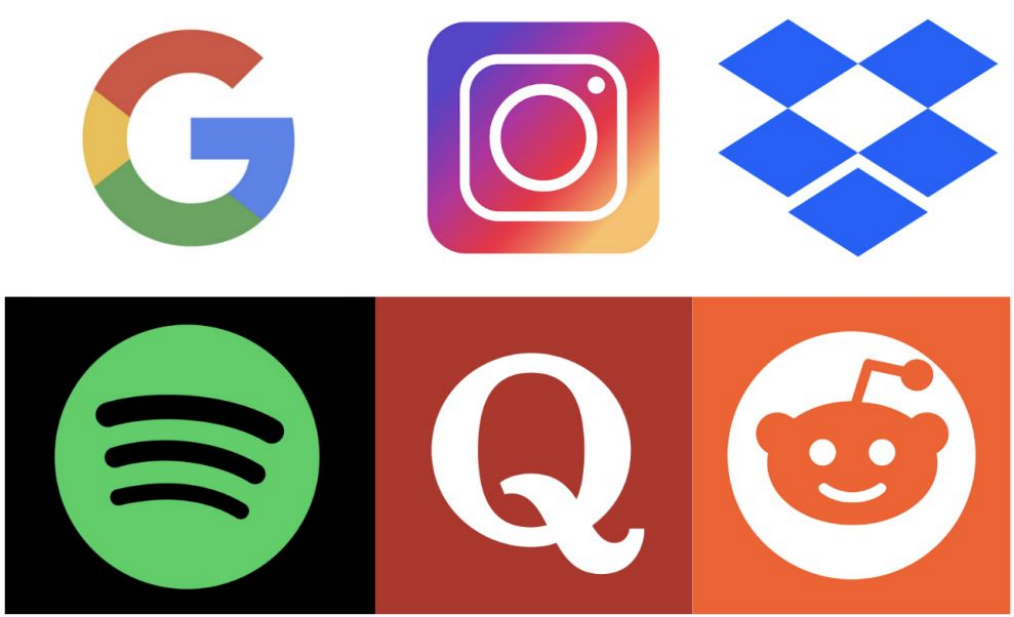
- Stable Diffusion: Prompt to image generation
  - source code (https://github.com/CompVis/stable-diffusion)
  - demo (https://colab.research.google.com/github/huggingface/notebooks/blob/main/diffusers/stable_diffusion.ipynb)
- DeepMind AI learns 57 Atari games (https://deepmind.com/blog/article/Agent57-Outperforming-the-human-Atari-benchmark)

Why Python?

✅ Simple but powerful
✅ Data Science

→ Web Development

Popular websites that use Python



- Rapid Development & Prototyping
- Readability & Maintainability
- Extensive Libraries & Frameworks
- Scalability
- Data Science and Machine Learning Integration
- Strong Community Support

Why Python?

✅ Simple but powerful
✅ Data Science
✅ Web Development

→ Widely used and has many packages

Python Packages

Python Package Index (PyPI)
https://pypi.org/ → 719,547 projects

+ Many other packages for web scraping, image processing, games, natural language processing...

What we will cover in this course

Module 1: Pythonic Programming

- Python environment and setup
- REPL
- Variables
- Primitives
- Control Flow
- Functions
- Data structures
- Recursion
- Magic functions
- Classes and objects
- Iterators and generators
- Higher order functions
- File I/O and context management
- Modules
- Testing
- Scripting
- Exceptions

Module 2: Graphics and Game Dev

- Turtle Graphics
  - Fractals
- Pygame

Module 3: Machine Learning

- Jupyter and Colab
- Numpy
- Pandas
- Matplotlib
- EDA
- Machine learning with sci-kit learn
- Deep learning

Module 4: Web Development

- Web scraping and BeautifulSoup
- Flask
- Django
- Docker and containerization

Module 5: Lightning Topics

- Python security, pickling, serialization
- ???
  - If you have a request, please let us know!

Final Project Presentations

More to come….

But we will dedicate our last lecture to final project group presentations

Logistics

Lecture and Office Hours

- Lectures will be Wednesdays 10:15-11:45 AM in AGH 214 (here!)

- Jordan's OH:
- Anushka's OH:
- Suhani's OH:

Course Structure

Weekly lecture (in class worksheet)

1-2 Homeworks per module

Final (group) project (including 1 checkpoint, presentation, and code deliverable)

Resources

Main website:
EdStem
Gradescope

Grading

7 Homeworks: 55%
Final Project: 25% →
    Checkpoint: 5%
    Final Presentation: 10%
    Final Coding Deliverable: 10%
Attendance and Participation: 10%

One unexcused absence will be granted no questions asked.
Excused absences will be given for unavoidable conflicts, e.g. a job interview, illness.
There will be opportunities for extra credit throughout the course!

Attendance and Participation

- Attendance is **required** for credit (and worth 10% of your grade!)
- Fill out a worksheet based on in class exercises
- Submit worksheet at the end of the class
- Attendance points based on completion NOT CORRECTNESS → it's in your benefit to guess an answer and pay attention when we go over the solution

Assignments

- HWs will be released along with the corresponding lectures
- Roughly 1-2 HWs per module.
- Some will have a quicker turnaround than others due to the nature of the length of the modules
- HWs are graded on correctness, some will have an autograder, some will be manually graded

Assignment Deadlines and Late Submissions

- All assignment deadlines are **30 mins prior to the start of class** → 9:45 on Wednesdays
- Grace period window: 24 hours where you can submit assignments late with **no penalty**
- 5 late days - can apply 1 per assignment - stacks on top of 24 hour grace period
  - The latest any assignment may be submitted for full credit is 48 hours late
- After 48 hours, 10% penalty per day it is late
- **Exception**: no late days nor grace period will apply to the final project because of the final project in class presentations.

Final Project

More details to come later in the course, but the gist is:

- 2-3 person groups
- Using skills and techniques you learned in this class
- 1 checkpoint
- Final presentation on the last day of class
- Final coding deliverable

Let’s jump into it!

Python Basics

- Python is interpreted (https://docs.python.org/3/tutorial/interpreter.html)
  - No compilation, unlike C, C++, Rust, Go, etc.
- Python interpreters instead provide a REPL: Read, Eval, Print Loop

[Coding Demo]

Comments

lecture.py

```
1    # this is a comment
2    print("hello world!")
3    """
4    this
5    is a
6    multiline
7    comment
8    print("bananas")
9    """
10   print("hello world...again!")
```

```
jordanschwartz@Jordans-2024-MacBook-Pro 1902 % python3 lecture.py
hello world!
hello world...again!
```

Variables

[Coding Demo]

🔑: Python is dynamically typed (variables can be reassigned to different types)

Check in (and how participation works)

What would the following code display?

Box _:

```
>>> x = 5
>>> y = 5
>>> x + y
```

_____

Primitive Types

- int
- float
- bool
- str
- None

[code demo]

🔑: Be wary of weird behavior, especially with booleans

Check in

Box _:

```
>>> False or (False or ((3 and True) or 1 / 0))
```

A) True
B) False
C) 3
D) Error

Conditionals

[coding demo]

🔑: Use == for comparison and = for assignment
🔑: if, elif, else
🔑: Python is an indentation based language

Iteration

[code demo]

🔑: while loops are useful for working with numbers
🔑: for loops are useful for sequences

Check in

Box _:
Write a block of code that will utilize some variable x, check if it's even and print "It's even!" if it is and otherwise print "It's odd!"

Check in

Box _:
Write a block of code that will utilize some variable x, check if it's even and print "It's even!" if it is and otherwise print "It's odd!"

```
if x % 2 == 0:
   print("It's even!")
else:
   print("It's odd")
```

Functions

[code demo]

🔑: defined with def keyword
🔑: no return type or argument type specification needed
🔑: default return type is None

Set Up Time!

Installations

- Python, version 3 or higher → https://www.python.org/downloads/
- Coding environment: VSCode

Playing around

- Open a terminal
- Start an interpreter
  - `python3` or `python`
- Run a print statement
- Run `import this`

Don’t hesitate to reach out with any questions or concerns, either now or throughout the semester!

Thank you for coming!

Thanks!

Do you have any questions?

youremail@freepik.com
+34 654 321 432
yourwebsite.com



Instructions for use

If you have a free account, in order to use this template, you must credit **Slidesgo** by keeping the **Thanks** slide. Please refer to the next slide to read the instructions for premium users.

As a Free user, you are allowed to:

- Modify this template.
- Use it for both personal and commercial projects.

You are not allowed to:

- Sublicense, sell or rent any of Slidesgo Content (or a modified version of Slidesgo Content).
- Distribute Slidesgo Content unless it has been expressly authorized by Slidesgo.
- Include Slidesgo Content in an online or offline database or file.
- Offer Slidesgo templates (or modified versions of Slidesgo templates) for download.
- Acquire the copyright of Slidesgo Content.

For more information about editing slides, please read our FAQs or visit our blog:
https://slidesgo.com/faqs and **https://slidesgo.com/slidesgo-school**


slidesgo