# Introduction to Data Science Programming Live Session

Week 2

Taylor Martin

Section 8

Remind me to start recording! ☺

& sync your forks

Berkeley
SCHOOL OF INFORMATION
MIDS • DATA SCIENCE PROGRAM

# Today

- Big Ideas Presentation
- Project Planning
- Week 2 activity & discussion
- Loops exercise & discussion
- Tiptoeing into OOP
- Interview Question Practice
- Survey

# Big Ideas Presentation

- Take it away…

# Project planning & TDD lite/OOP touch

- Define the Project Goals
  - What problem are you solving?
  - Requirements: Main features or functionalities the project must have.
- Break Down the Project
  - Object decomposition (aka "facet analysis"):
    - Outline the main objects and the interactions between them.
  - Process decomposition:
    - How to do the job, step-by-step
    - Define exactly what your objects will need, can do, must do, and when they perform their tasks.

# Project planning & TDD lite Part 2

- Start Coding
- Test Driven Development
  - Unit Tests:
    - Ensure each chunk of code works as expected
    - e.g., have your for loop do nothing but print each element to start.
  - Integration Tests:
    - how parts of your project work together,
    - e.g., does the for loop pick up the input object correctly?

# Breakout

- Complete week_02_activity

# Activity Follow up

- Text handling results
- Variables & Strings…

# Variables

- A variable is just a symbol name to contain a value, e.g., x = 5;
- The left-side is the declaration (x =).
- The right is the instantiation with a value (5).
- Variable names can “hold” many different types of data (the data types from this week)
- It's easy to cast a variable from one type to another.
- In x = ‘5’, 5 is not a number.
- Can convert x from string to int using the function int():
  - x_int = int(x)
- Confirm the changes in the data type using type()

# Strings

- String: a contiguous block of data;
- declared by instantiating a variable using single- or double-quotes, e.g.,
  - x = '5',
  - subject_name = "Smith".
- Strings are immutable.
- Characters need to be represented in a format that computers can store and manipulate.
- UTF-8 is the standard
  - Can represent characters that aren't on a standard keyboard, such as Cyrillic or Mandarin Chinese characters.
  - Possible because each character has a unique code point—a number that represents it.
  - e.g., 名 is U+660E and 娃 is U+54C7

# String Slicing

- Splicing: Since a string is contiguous, we can identify each letter by its position and slice up the string using indices
  - e.g., word = "cat”,
  - cat[0] = ?
- Slicing allows us to pick a start : stop : step technique
  - print(word[0:3:1])
  - print(word[0:3])
  - print(word[0:])

# Week 2 Activity Follow up

- We'll follow up on the calculator part after we do some more focused work on while and for loops.
- Loops.ipynb

# Control of Flow

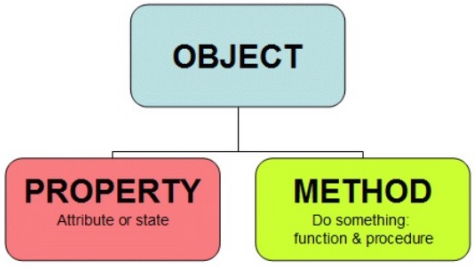
- The interpreter or compiler reads and executes a script starting at the first line straight down to the final line.
- Control-of-flow commands stop the running (flow) of the script to direct it based on our needs.
- There are several basic commands to control how a program acts:
  - ie, change when and why parts of the code are executed
- What are the control-of-flow statements we've encountered?

# Loops

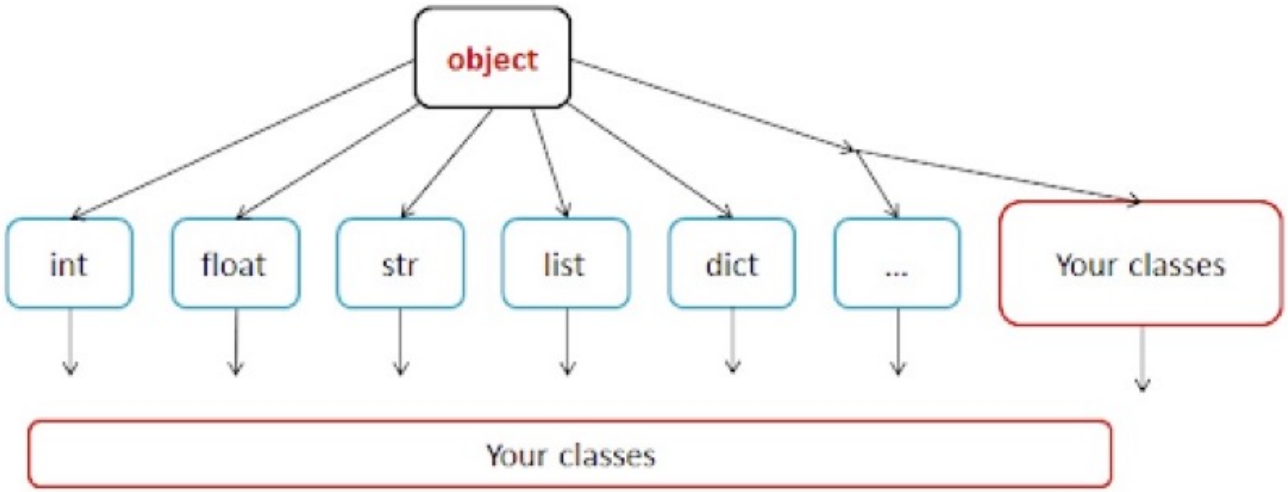
- Why did you use each?
- Use a **for** loop instead of a **while** loop:
  - When you know the exact number of iterations needed:
    - `for t in range(0, 10):`
  - Need to loop through a collection:
    - Lists, tuples, dictionaries, sets, even strings
- Use a **while** loop instead of a **for** loop
  - If you need to loop until a condition is met but don't know in advance how many iterations it will take:
    - while not correct_guess:
    - guess = input("Enter your guess: ")
  - Situations where you need more dynamic behavior
    - Loop termination depends on complex logic that isn't easily tied to a fixed range or collection.

# Tiptoeing into OOP

- An object is just something from the real-world that we want to create in a computer.
- Objects have properties and behaviors(or methods)
- A pen
  - Properties
    - length,
    - ball-point, felt-tip, or ink
  - Behaviors
    - Write, draw, throw ☺




- In computing we built up such objects from a set of pre-defined primitive data type objects (the data types you learned about this week)
- They have their own properties and behaviors


object
int
float
str
list
dict
...
Your classes
Your classes

# OOP big picture

- In object-oriented coding,
- There's a base Object that is the grandparent of them all and encapsulates all the properties and behaviors of that language
- Python's object contains all the primitives we use: int, float, string, and so on ...
- To build more complex "things-in-the-world."

# Data Science Interview Questions: Data Types, Control of Flow, & Strings

- General Knowledge
  - 1. What are the different data types available in Python?
  - 2. How do you differentiate between mutable and immutable data types in Python?
  - 3. Explain how control flow works in Python with examples.
- Coding/Scripting
  - 1. How would you convert a list of integers into a comma-separated string in Python?
  - 2. Write a Python function that checks if a given string is a palindrome.
- See the file week-2-interview-qs.ipynb

# W200.8 Martin Week n Survey

- https://forms.gle/K9Ao1Nz4BNeFxJtH6