Week 1: Agenda

- ✔ Syllabus
- ✔ Data, Variables, Expressions, Functions
- ✔ Quiz#1: Participation, Academic Honesty Policy
  - Examples

Announcements

- OH started
  - See the calendar on the course website
- You should be able to access:
  - CS Academy
  - Gradescope
  - Discord
- HW1 is out

Homework 1

	hw1
95	**0**
90s	**0**
80s	**0**
70s	**0**
60s	**0**
50s	**0**
40s	**0**
30s	**0**
20s	**0**
10s	**0**
00s	**0**
not submitted	~50

Homework 1 (Thursday stats)

	hw1
75	**0**
70s	**0**
60s	**3**
50s	**0**
40s	**1**
30s	**2**
20s	**0**
10s	**2**
00s	**3**
not submitted	**32**

Data

Data Types

- Numbers (e.g., int, float): Whole numbers and decimals.
- Text (str): Strings of characters (e.g., "Hello").
- Boolean (bool): True or False.
- Others: Lists, tuples, dictionaries, etc.

How to know the type?

- type()

Variables

What are they?

Named storage for data

Why do we need them?

- Makes code readable and reusable
  - Easier to read and modify later if needed
- Enable Dynamic Input
    - Programs can adapt to different inputs without code changes
- Reusability and Flexibility
  - Update a value once instead of modifying multiple lines
  - It will be obvious later

Operations and Expressions

Expressions: Code that produces a value (e.g., 2 + 3).

Operators: Symbols for calculations or logic.

- Arithmetic: + - * / % ** //
- Comparison: == != > < >= <=
- Logical: and, or, not
- Precedence Rules:

```
result = 2 + 3 * 4
```

Important to know:

- Integer division (//)
- Modulo (%)
- Float and equality

Nice Tricks:

- Concatenation: 'Hello' + ' World' → 'Hello World'
- Repeat Strings: 'Hi' * 3 → 'HiHiHi'
- Playing with digits

Functions

What? Reusable blocks of code that perform tasks.

- Defining vs Using Functions

Why? Avoid repetition, organize code.

Conditionals

- Conditionals allow programs to make decisions.
- Real-world problems are full of conditions (discounts, access, warnings).
- Without them, programs would be boring and rigid.
- What you need to know:
  - Syntax: if, elif, else
  - Conditions must be True or False
  - Use indentation to group code blocks
  - Order matters: Python checks conditions top to bottom

Data and Expressions: Types of problems

- Math Problems
  - Examples: `isRightTriangle`

Data and Expressions: Types of problems

- Math Problems
  - Examples: `isRightTriangle`

isRightTriangle

Write the function `isRightTriangle(x1, y1, x2, y2, x3, y3)` that takes 6 integers representing the coordinates of 3 points in the plane: `(x1, y1), (x2, y2), (x3, y3)`. These are the vertices of a triangle.

Return **True** if the triangle formed by these three points is a right triangle (has one 90-degree angle), and **False** otherwise.

You may assume the points are distinct and not collinear.

How to tackle a programming problem?

A systematic approach

How to approach a programing problem

1. Read the Problem
   - Understand the requirements and constraints.
2. Brainstorm Test Cases
   - Generate examples to clarify expected results and edge cases.
3. Devise an Algorithm
   - Outline a clear step-by-step plan.
4. Validate the Algorithm
   - Check logic against your test cases; refine if needed.
5. Implement the Code
   - Translate the algorithm into a working program (and run tests)

isRightTriangle

Write the function `isRightTriangle(x1, y1, x2, y2, x3, y3)` that takes 6 integers representing the coordinates of 3 points in the plane: `(x1, y1)`, `(x2, y2)`, `(x3, y3)`. These are the vertices of a triangle.

Return **True** if the triangle formed by these three points is a right triangle (has one 90-degree angle), and **False** otherwise.

You may assume the points are distinct and not collinear.

Try to solve this problem on your own.
We will return to it, and other problems , next week.

Data and Expressions: Types of problems

- Math Problems
  - Examples: `isRightTriangle`
- Extracting Digits
  - Examples: `isSmallFair`, `isSmallPal (Quiz 1, Pitt S22)`

Data and Expressions: Types of problems

- Math Problems
  - Examples: `isRightTriangle`
- Extracting Digits
  - Examples: `isSmallFair`, `isSmallPal (Quiz 1, Pitt S22)`
- …
- Mixed

Data and Expressions: Types of problems

- Math Problems
  - Examples: `isRightTriangle`
- Extracting Digits
  - Examples: `isSmallFair`, `isSmallPal (Quiz 1, Pitt S22)`
- …
- Mixed

`isSmallFair(n)` (Pittsburgh F21)

- We will say that a value **n** is *"fair"* if it is an integer and it has the same number of even digits as odd digits (ignoring leading 0's). A *"small fair"* number is a fair number with exactly 4 digits.
- For example, 1083, 1081, and -1092 are each small fair numbers because each have two odds and two evens.
- With this in mind, and without using strings or loops, write the function `isSmallFair(n)` that takes a value **n**, that may or may not be an integer, and returns **True** if **n** is a small fair number, and **False** otherwise. **Do not crash if n is not an integer!**

`timeInterval(t1,t2)` Quiz, Qatar F21

Write the function `timeInterval(t1, t2)` which, given two non-negative integers `t1`, `t2`, that encode two 24-hour times in the format `hhmm`, returns the time interval, in minutes, between those two times. If `t2 < t1`, you should assume that `t2` refers to a next day time. You can assume that `0 < hh < 24` is the hour, and `0 < mm < 60` are the minutes. If `hh > 0`, then mm is always a two-digit number. If `hh == 0`, then mm can be either a one or a two-digit number, depending on its value.

- 1503 is 15 hours, 3 minutes, or 3:03pm.
- 849 is 8 hours, 49 minutes, or 8:49am.
- 0 is 0 hours, 0 minutes, or 12:00am midnight.
- 59 is 0 hours, 59 minutes, or 12:59am.
- 101 is 1 hour, 1 minute, or 1:01am.

For example...

- `timeInterval(1400, 1545)` returns the time interval between 14 o'clock and 15:45 (same day) which is 105 minutes.
- `timeInterval(2359, 31)` returns the time interval between 23:59 and 00:31 (next day), which is 32 minutes.
- `timeInterval(31, 2359)` returns the time interval between 00:31 and 23:59 (same day), which is 1408 minutes.
- `timeInterval(1200, 0)` returns the time interval between noon and midnight (next day), which is 720 minutes.
- Hint: There are 1440 minutes in a day.