Nafisa's Section

CS106A - Code in Place 2025

Section 3

Week Three Overview

Ice Breaker & Check-in

Quick Concept Review

Problem: Mars Weight Calculator (Understanding & Strategy)

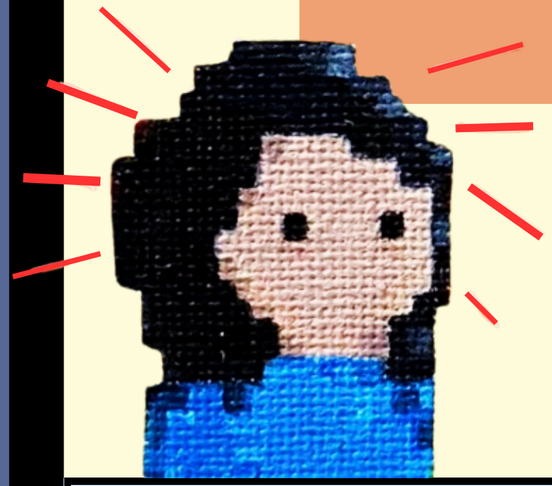
Solve It Together

Testing and Refinement

Wrap-up and Q&A

Let's Connect!

- If moving to another planet were an option, where would you go?



Quick Review: Variables & Input

- Variables:
  - Like labeled boxes that store information.
  - Every variable always has three parts - the name, the type, and the value.
- input() function:
  - Gets text from the user.
  - Question: What type of data does input() always return?

Quick Review: Constants & Typecasting

- Constants:
  - Values that don't change in your program.
  - Naming: ALL_CAPS_WITH_UNDERSCORES (e.g., PI = 3.14)
- Type Casting:

  Changing a variable from one type to another
  - num_str = "10"
  - num_int = int(num_str)
  - num_float = float(num_str)

Mars Weight Calculator

- NASA's Ingenuity helicopter flies on Mars!
- Weight on Mars is 37.8% of Earth weight due to weaker gravity.
- Goal: Write a Python program that:
  - Prompts an Earthling to enter their weight on Earth.
  - Calculates and prints their equivalent weight on Mars.
  - Output should be rounded to two decimal places.

How do we get our output to two decimal places?

The handout mentions a round() function. Let's look at it:

round(value_to_round, number_of_decimal_places)

```
x = 3.1415926
rounded_x = round(x, 2) # Result: 3.14
print(rounded_x)

x = 2.71828
rounded_x = round(x, 4) # Result: 2.7183
print(rounded_x)

x = 3
rounded_x = round(x, 4) # Result: 3 (doesn't add trailing zeros)
print(rounded_x)
```

This is how!

Brainstorming

Before we code, what are the steps?

- Get the user's weight on Earth.
  - What function do we use?
  - What type will this input be?
- Store the Mars gravity factor (37.8%).
  - Should this be a variable or a constant? Why?
- Calculate the weight on Mars.
  - Earth Weight * Mars Factor
  - Watch out for types! Can we multiply a string by a number directly for this?
- Round the Mars weight to 2 decimal places.
- Print the result in a user-friendly way.

Pseudocode, Implementation, and Test

See you at the IDE!

Bonus: Planetary Weight Calculator

Mars isn't the only planet! Each has different gravity.

- Goal:

  Prompt for Earth weight.

  Prompt for a planet name (first letter capitalized, e.g., "Mars", "Jupiter").

  Print the equivalent weight on that planet, rounded to 2 decimal places.

Gravity Factors (vs. Earth):

Mercury: 37.6%

Venus: 88.9%

Mars: 37.8%

Jupiter: 236.0%

Saturn: 108.1%

Uranus: 81.5%

Neptune: 114.0%

How is this different from Mars Weight?

- We need another input: the planet name.
- We need to store many gravity factors (constants!).
- We need to choose which gravity factor to use based on the planet name entered by the user.
    - How can we make a decision in our code?

Making Decisions: if and ==

- We know Python can make decisions using if statements
- To check if two things are equal, we use == (double equals sign).

  x = 10 (This is assignment - setting x to 10)

  if x == 10: (This is comparison - checking if x is equal to 10)

```
planet_choice = input("Enter a planet: ")
if planet_choice == "Mars":
    print("Calculating for Mars!")
if planet_choice == "Jupiter":
    print("Calculating for Jupiter!")
```

What We Practiced Today

- Getting user input with input() (returns a string!).
- Using variables and CONSTANTS.
- The importance of types and type casting (float(), int()).
- Rounding numbers with round(value, places).
- Making decisions with if and comparing with ==.

- PEP 8 Reminders: snake_case for functions/variables, ALL_CAPS for constants.
- Keep practicing! These are foundational skills.

THANK YOU

See you in the next session!