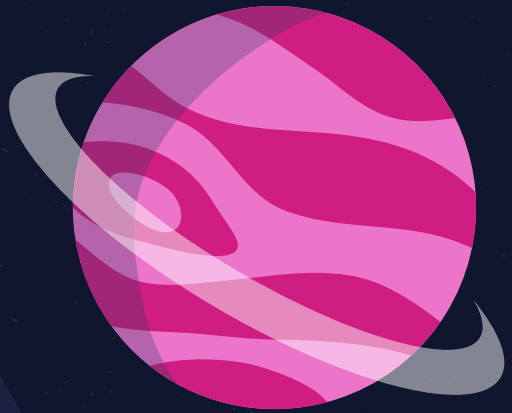
WEEK 3

WELCOME TO SECTION!!!

Section Leader
Eesha Tariq

Agenda



01

Check–in

How are all of you doing?

02

Recap

Revisiting lecture content

03

To Mars!!!

Section Problem #1

04

To Infinity & Beyond

Section Problem #2

Check-in

Welcome back to another week of Code in Place section!

I hope you all are doing well! Hopefully CiP hasn't been stressful.

Let's do a quick check-in question! (Feel free to answer in the chat!)

Feel free to answer *any* of the following:

- ★ If you could teach Karel ***ONE*** more command, what would it be?
- ★ What is your favorite food?
- ★ What is a hobby of yours that you do in your free time?

Announcements

Come attend some events! @ codeinplace.stanford.edu/cip5/events

Previous Events (recordings available!):

- Various Karel Review Problems with Head TA Ty (Stone Mason, Piles and Jigsaw, Fill Karel)
- Karel Review: Checkerboard and Midpoint with Joseph L
- LeetKarel - Studying Algorithms Using Karel with Nick B

Upcoming Events (all in Pacific Time; check events page for local time):

- **Every Monday:** Monday Mornings with Chris (reoccurring)
- ~~**May 9th, 3:30pm PT:** Explaining Fencepost Coding Problems~~ [RESCHEDULED for May 16th]
- **May 9th, 8:00pm PT:** Game Design Workshop 1: Guest Speaker - WattDesigns on YouTube
- **May 10th, 1:30 PM:** Transitioning to Console (N-sided-dice and Sunrise-fill-in-the-blanks) [NEW]
- **May 10th, 3:30 PM:** Programming for accessibility: guest speaker [NEW]
- **May 10th, 5:00pm PT:** FGLI in Stem Panel

Recap

Revisiting variables

Defining and Using Variables

Defining Variables

We define variables by writing the variable's name, followed by an equals sign, and then what we want to store inside.

```
var_name = value
```

Using variables

When we use the variable's name—unless we are changing its value—we essentially open the "suitcase" to use what's inside.

```
print(var_name)
sum = num1 + num2
```

Variable Types

Types

Variables have ***types***. These types are what differentiates letters and numbers to our computers.

Example Types

`Strings`:
Words/Sentences

`Ints`:
Integers; numbers without decimals

`Floats`:
Any other number that isn't an integer

Casting

Sometimes, we can **change from one variable type to another** via ***typecasting***. For example, we may have a number represented as a string, and we need to cast it to be a number so we can do math with it!

Constant

- A **constant** is a **fixed value** that **does not change** during the execution of a program.
- Once assigned, a constant **cannot be modified**.

Example:DAYS_OF_WEEK is a constant: it won't change.

Before We Start...

Are there any questions?

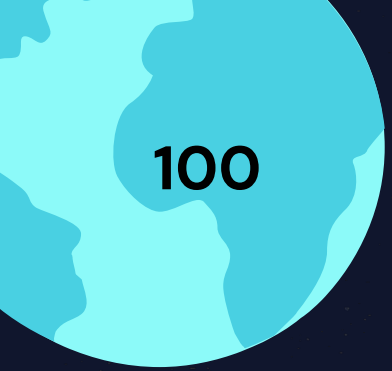
Section Problem #1:
To Mars

Storytime

Section Problem #1: To Mars!!!

One of the things that NASA engineers need to account for is the fact that due to the weaker gravity on Mars, an Earthling's weight on Mars is **37.8%** of their weight on Earth.

Write a Python program that prompts an Earthling to enter their weight on Earth and prints their calculated weight on Mars.



Section Problem #1: End Goal

Sample Input

```
Enter a weight on Earth:
```

↓

```
Enter a weight on Earth: 100
```

Sample Output

```
The equivalent weight on Mars: 37.8
```

* ***User input*** is italicized and bolded for visual clarity *

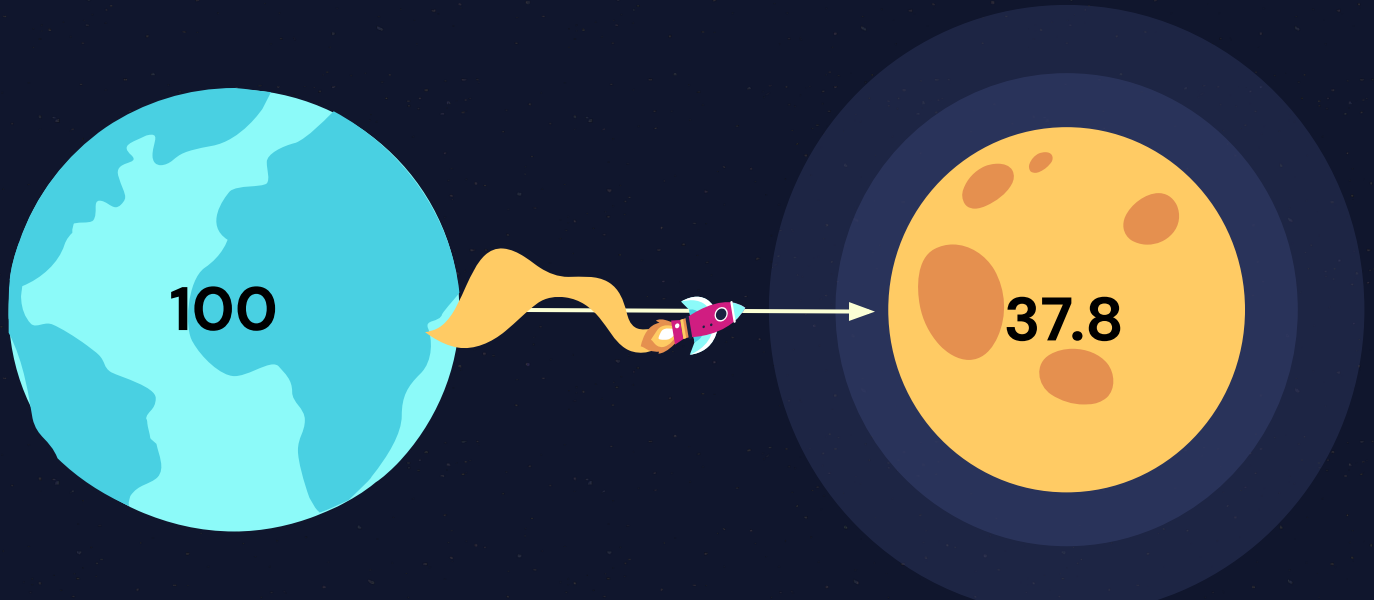
Pre-Code Discussion

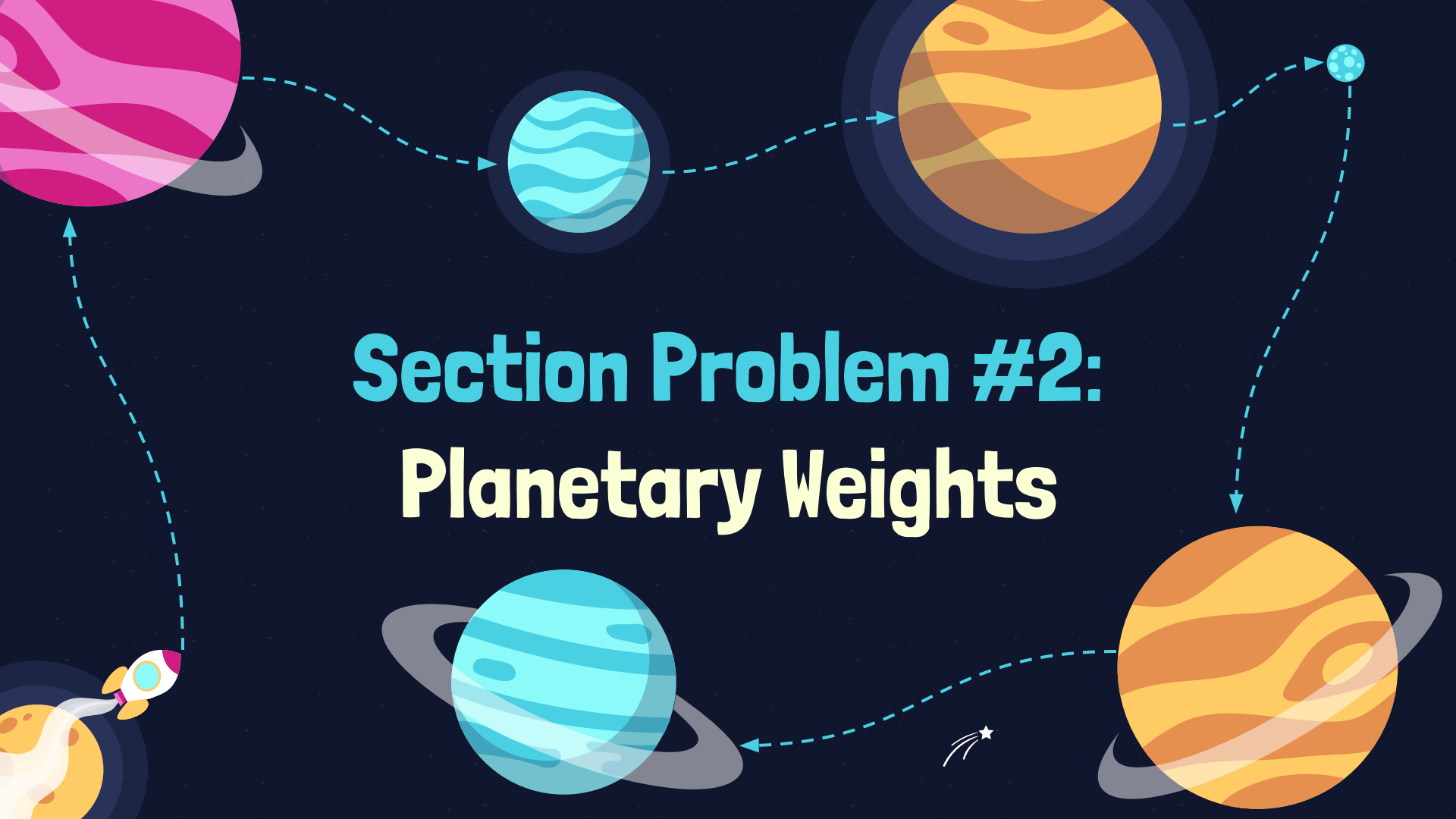
Inputs, Constants, & Types

(Also, questions if you have them!!!)

Let's Code!

For this activity, you'll be working in breakout rooms with each other




Section Problem #2:
Planetary Weights

Section Problem #2: Planetary Weights

Mars is not the only planet in our solar system with its own unique gravity. In fact, **each planet** has a **different** gravitational constant, which affects how much an object would weigh on that planet.

Section Problem #2: Planetary Weights

Write a Python program that prompts an Earthling to enter their weight on Earth and then to enter the name of a planet in our solar system. The program should print the equivalent weight on that planet.

* You can assume that the user will always type in a planet with the first letter capitalized and you do *not* need to worry about the case where they type in something other than one of the above planets. *






Section Problem #2: Planetary Weights

Here's a list of constants for each planet's gravity compared to Earth's:

- ★ Mercury: 37.6%
- ★ Venus: 88.9%
- ★ Mars: 37.8%
- ★ Jupiter: 236.0%
- ★ Saturn: 108.1%
- ★ Uranus: 81.5%
- ★ Neptune: 114.0%

Section Problem #2: End Goals

Sample Input

```
Enter a weight on Earth:
        ↓
(Input 1) Enter a weight on Earth: 120
        ↓
Enter a planet:
        ↓
(Input 2) Enter a planet: Mars
```

Sample Output

```
The equivalent weight on Mars: 45.36
```

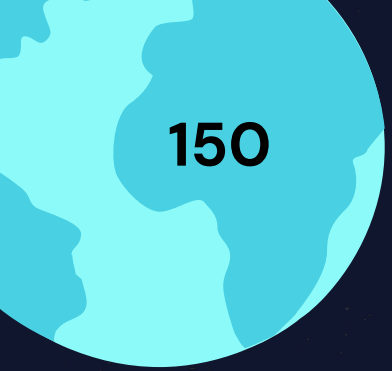
* ***User input*** is italicized and bolded for visual clarity *

Section Problem #2: End Goals

Full Run

```
Enter a weight on Earth: 120
    Enter a planet: Mars
The equivalent weight on Mars: 45.36
```

* ***User input*** is italicized and bolded for visual clarity *



Section Problem #2: End Goals

Sample Input

```
Enter a weight on Earth:
        ↓
(Input 1) Enter a weight on Earth: 150
        ↓
Enter a planet:
        ↓
(Input 2) Enter a planet: Jupiter
```

* ***User input*** is italicized and bolded for visual clarity *

Sample Output

```
The equivalent weight on Jupiter:
354.0
```

Section Problem #2: End Goals

Full Run

```
Enter a weight on Earth: 150
Enter a planet: Jupiter
The equivalent weight on Jupiter: 354.0
```

* ***User input*** is italicized and bolded for visual clarity *

Pre-Code Discussion

What's different?

(Also, questions if you have them!!!)

Let's Code!

For this activity, we'll break out again, and then come back together!