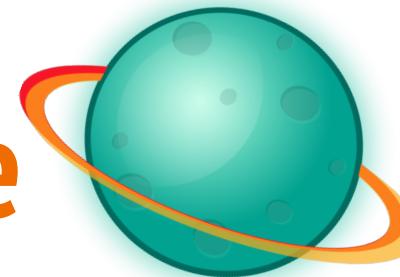
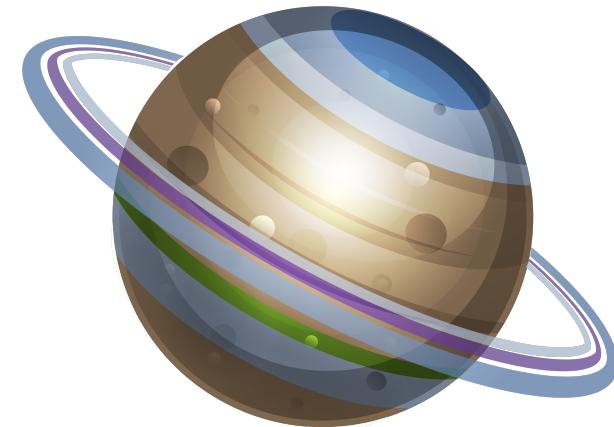


# CS106A Code in Place



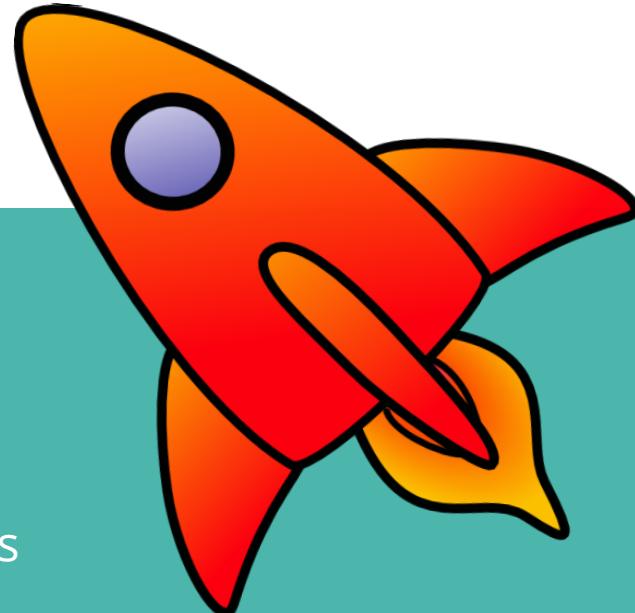
Section 3: Variables & Your Weight  
on Mars and other planets

Week 3: May 12<sup>th</sup>, 2023



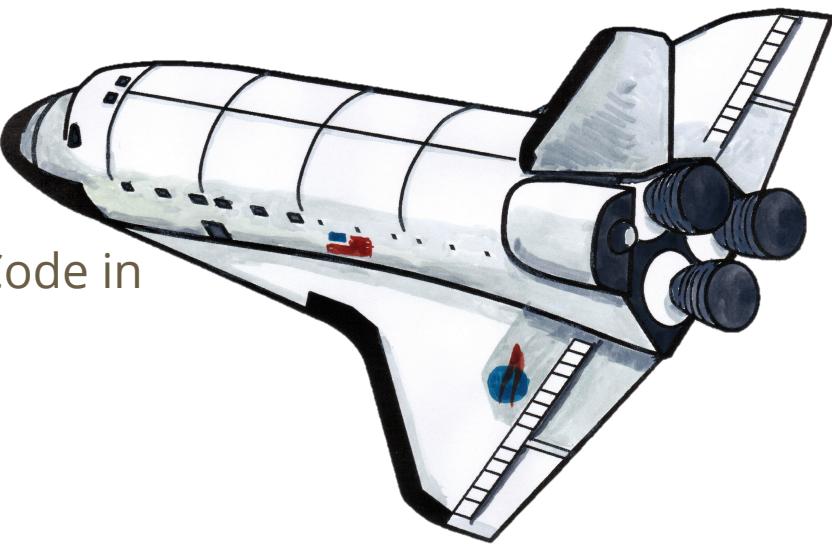
# Section Plan

- Catching up
- A few tips on the process of coding
- Review concepts: variables
- Write a program to calculate our weight on Mars
- *Extra: Write a program to calculate our weight on any planet*



# Catching-up

- I hope you all are doing well! Hopefully Code in Place hasn't been stressful.
- Let's do a quick check-in question!  
*(Feel free to answer in the chat!)*
- Choose 1 of the following questions:
  - If you could teach Karel **ONE** more command, what would it be?
  - What is your favourite food?
  - What is a hobby of yours that you do in your free time?



# Tips on the Process of Coding: writing code

- Outside of the code in place interface: use **autocomplete!**
- Python forces you to be neat. **Whitespace matters.**
- **Toggle indentation** by:
  - highlighting a section and
  - pressing tab (tabs in)
    - or shift+tab (tabs out).



# Tips on the Process of Coding : coding is the easy part

- You will spend most of your time **debugging**.
- The code does **exactly** what you tell it to do.
- Think of it like a **detective game**.



# Tips on the Process of Coding : debugging

- When you **design** your code, think like an **architect**.
- When you **test** it, think like a **vandal**.
- There are two kinds of bugs (errors).



# 1) Syntax Errors

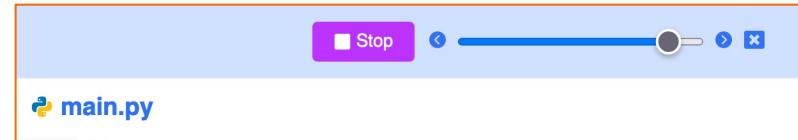
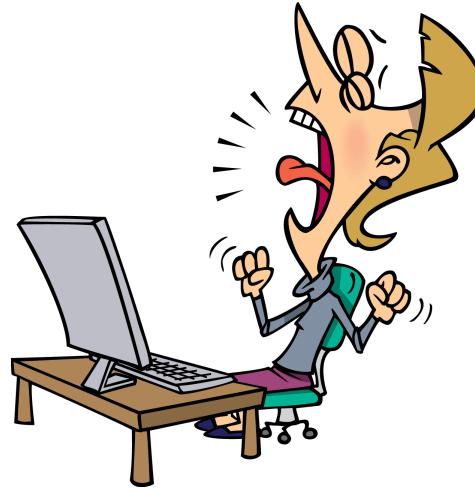
- “Typos”
- Your code won’t run!
- **Common examples:**
  - Misspellings
  - Indentation errors
  - Missing or extra () or :



CoolClips.com

## 2) Logic Errors

- Less obvious than syntax errors.
- Code may or may not run...
- But doesn't do the right thing.
- Ideas:
  - Teddy Bear Approach (Explain your problem out loud to someone else).
  - Walk through code with IDE Debugger (or by stepping through code in Ed)



# Tips: Coding in the real world

## Study vs the working world



0

Social

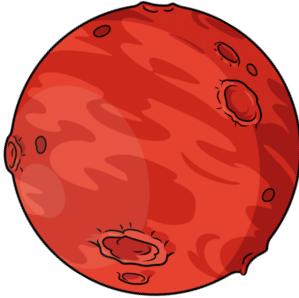
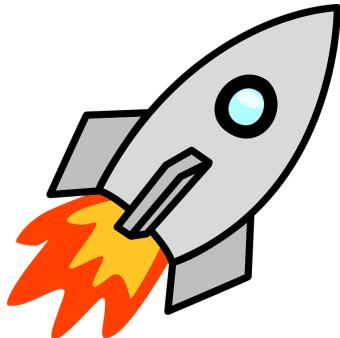


Faatimah 10 May at 17:45

While teaching in higher education, students often complained about the disparity between what was being taught at university, and the skills that were required to be successful in the workplace in a particular field.

I'm curious to know, how would you compare what's learnt in a computer science degree to what's actually done on a day to day basis working in this field?

# Concept Review



- Variables

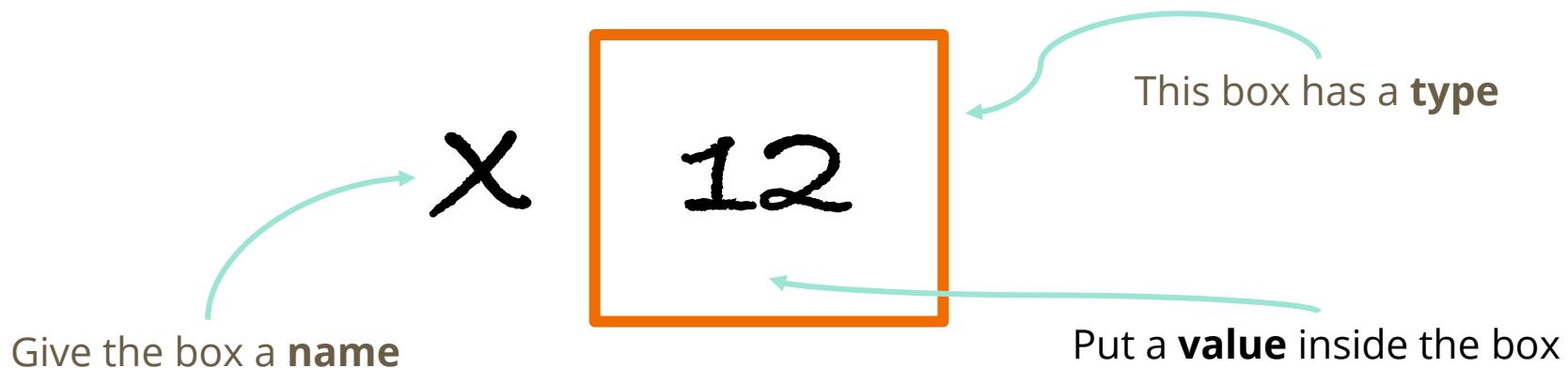
- What is a variable?
- Defining VS Using a variable
- Types of variables
- Casting & rounding

- Conditional statements

- The if statement
  - Conditional operators
  - Difference between = and ==
-

# Concept Review – Variables: Intro

- What is a variable?
  - A variable is a **place to store information** in a program. Think of it like a **box**:



# Concept Review – Variables: defining v. using

## *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 “box” to use what's inside.

*print(var\_name)*

*sum = num1 + num2*

# Concept Review – Variables: types

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

**int**

integer value  
(numbers with no decimals)

**x = 5**

**float**

any number that is not an int  
**x = 5.0**

**bool**

boolean logical values  
(True/False)  
**x = True**

**string**

text characters/words/sentences  
(single or double quotes)  
**x = "hello"**

**Note:** The string “5” is not the same as integer 5!

# Concept Review – Variables: casting & rounding

## Casting

- Sometimes, we can ***change from one variable type to another*** via ***typecasting***.
- *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!*

## Rounding

- While we didn't cover this in lecture yet, it may be relevant in the future!
- There is a *round(float, num\_decimals)* function which lets us round a floating point number to a certain amount of decimals!
- *For example: 19.724555555 -> 19.725 by using round(19.724555555, 3).*

# Concept Review – The If Statement in Python

Operates a lot like the if statement in Karel

```
if (condition):  
    #run this code
```

```
10 if front_is_clear:  
11     move()
```

But what does the “condition” part look like?

# Concept Review – Conditionals

The first conditional to get started with is the “equals” condition: **x == y**

```
if (x == y):  
    print("x and y have the same value!")
```

The print statement will run if the **value** of x is **equal to** the **value** of y

# Concept Review – Other Conditionals

You will learn more about conditionals in tomorrow's lecture.

For now, we will focus on the “`==`” conditional and one other:

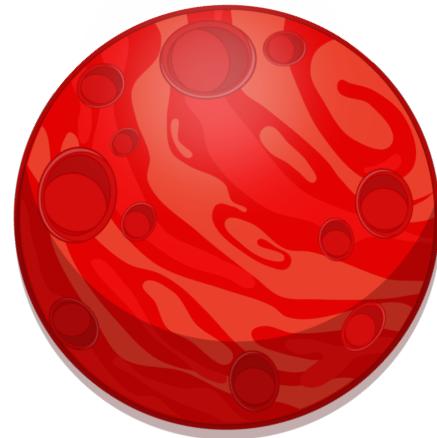
The “`!=`” conditional means ‘not equals’, it is the logical opposite of “`==`”

```
if (x != y):  
    print("x and y have different values; they are not equal!")
```

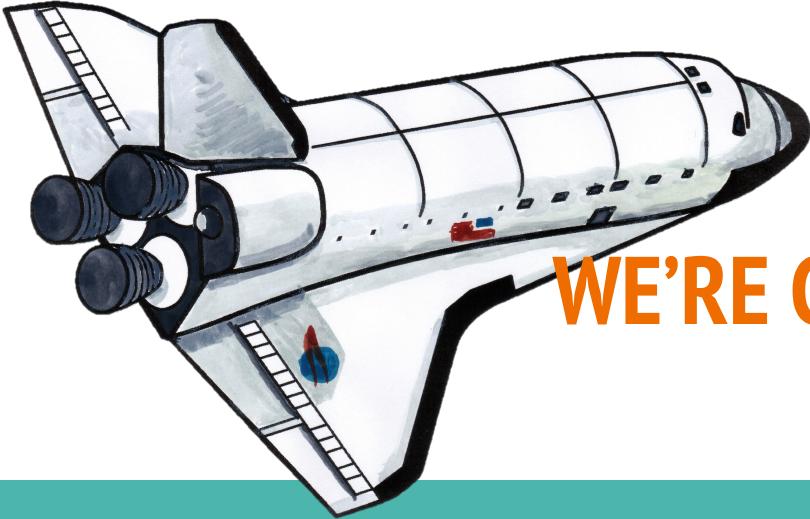
# Concept Review – Difference between = and ==

= (equals sign)	<b>Assign</b> to a variable <code>x = 3</code>
== (double equals sign)	<b>Compare</b> the left side to the right side <code>x == 3</code>

# Concept Review



*Any questions?*



# WE'RE GOING TO MARS!



## Problem:

- Calculate how much we would weigh on Mars!
- An Earthling's weight on Mars is 37.8% of their weight on Earth

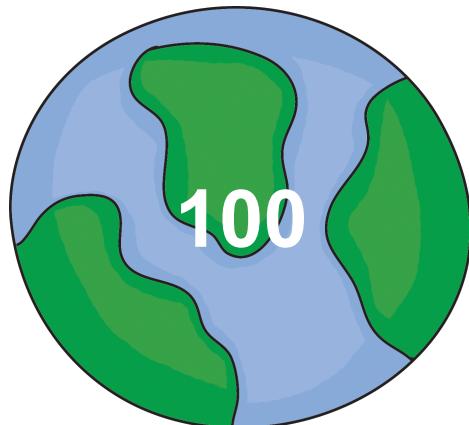
# Going to Mars: End goal

Sample Input

*Enter a weight on Earth:*



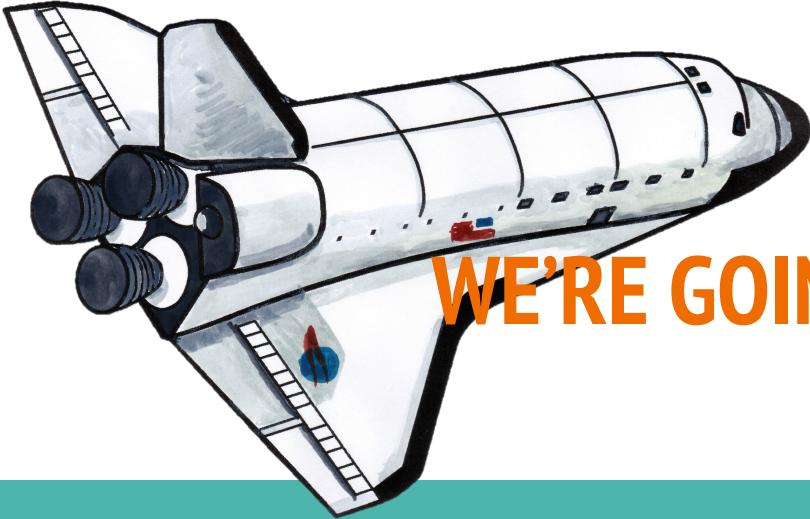
*Enter a weight on Earth: **100***



Sample Input

*The equivalent weight  
on Mars: 37.8*





# WE'RE GOING BEYOND MARDS!



## Problem:

- Calculate how much we would weigh on other planets!
- The program should prompt 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. 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%

# Going beyond Mars: End goal

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 Input

*The equivalent weight  
on Mars: 45.36*



---

---

# See you next week!

And on the forum

Section 3: Variables  
Next week: Functions

---

---