Welcome to Section

Week 4

Slides by Maggie Lee and Cameron Mohne

Agenda

Check-in

How is everyone doing?

Recap

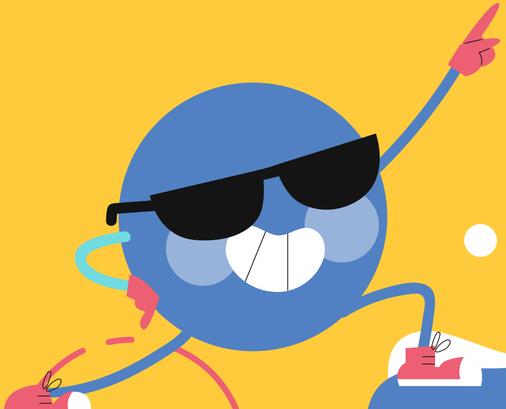
Reviewing logic, logical operators, & comparison operators!

High-Low

Section Problem!

Check In

What's your favorite sport or game to play/watch?



Questions?

Does anyone have any questions before we move on to the recap?

Recap

If/Elif/Else Statements

As a quick reminder:

- If-Statements take a condition and only runs a block of code if the condition evaluates to `True`.
- Elif-Statements take a condition and only tests it if the prior If-Statement (and all prior Elif-Statements) evaluate to `False`.
- Else-Statements take no condition and runs a block of code if the prior If-Statement (and Elif-Statements, if there are any) evaluate to `False`.

Still confused? Don’t worry! We’ll practice on these next slides! :)

If/Elif/Else Statements

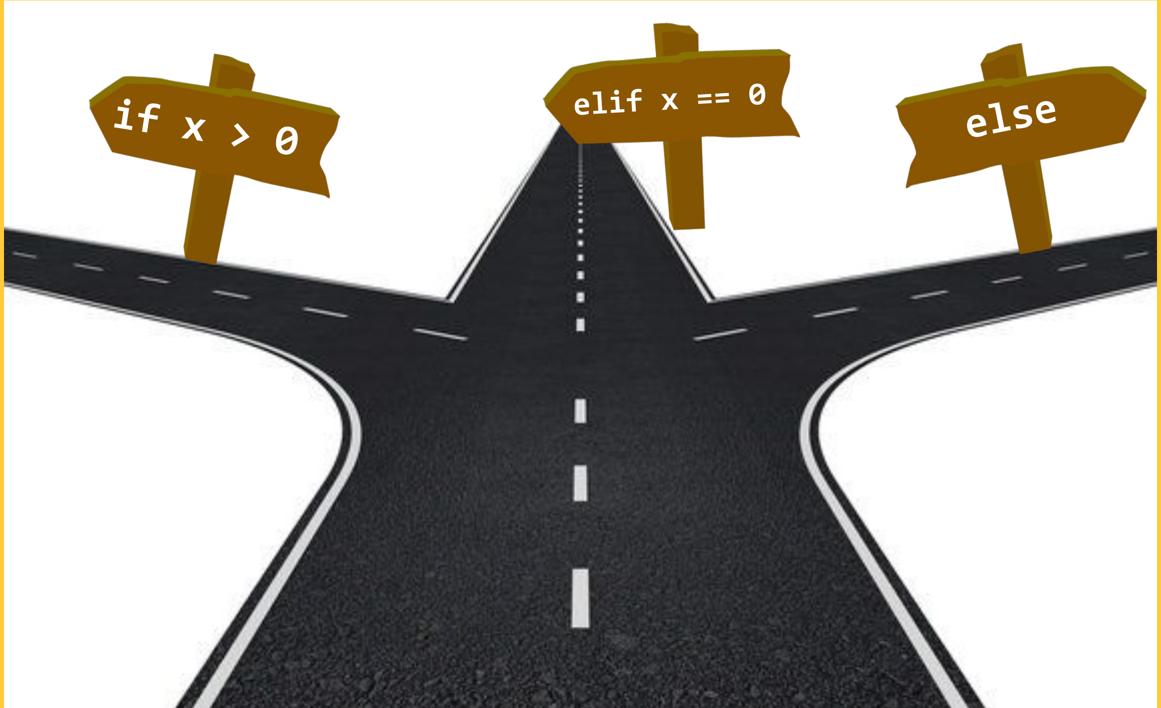
You can think of these as a forked road!
Let's run through some examples.

if x > 0
elif x == 0
else

If/Elif/Else Statements

```
x = 0
```

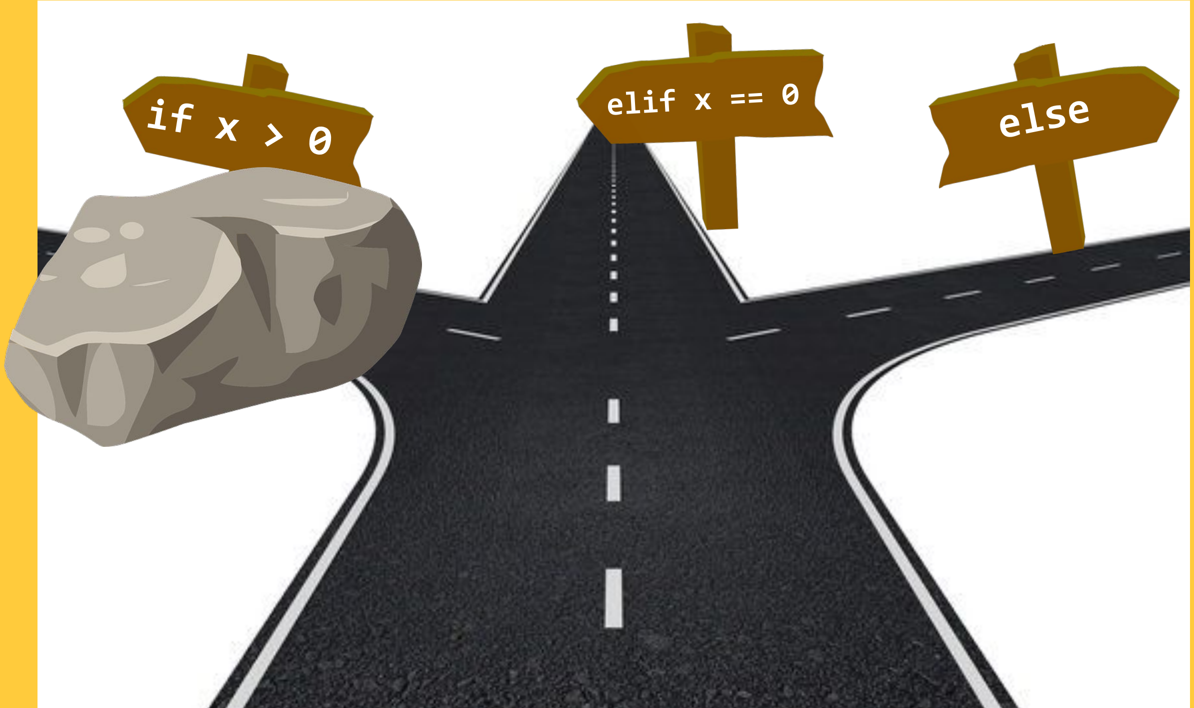
Which path would we take: left, middle, or right?



If/Elif/Else Statements

```
x = 0
```

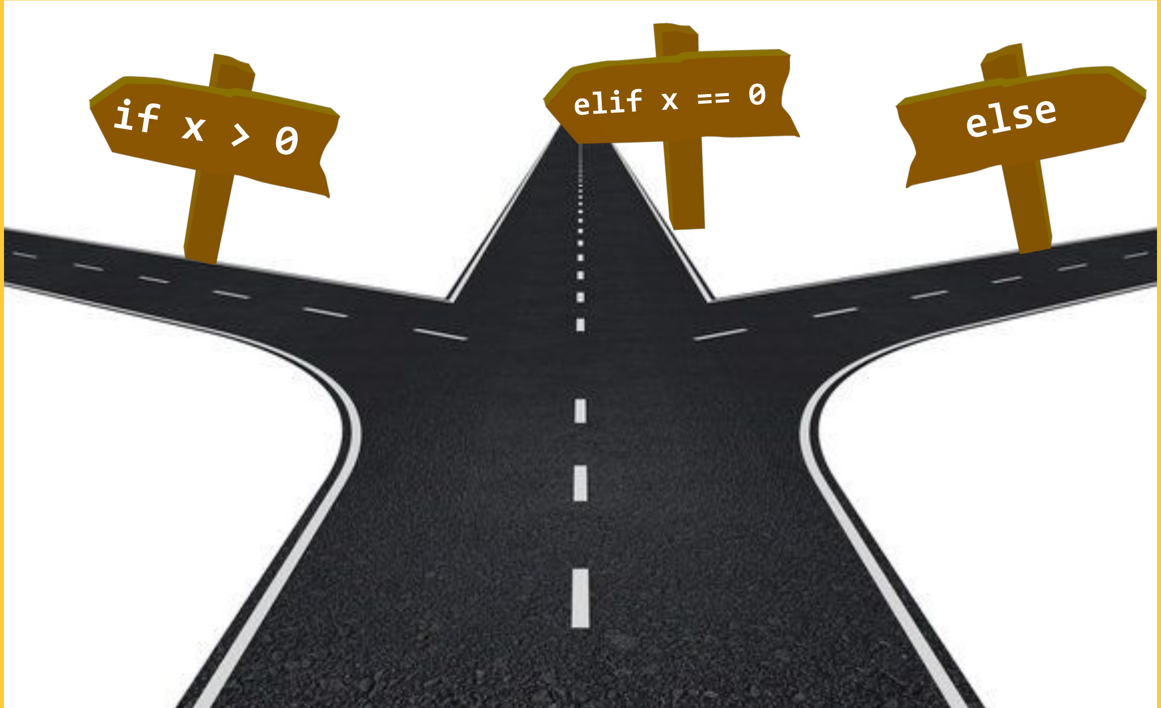
Which path would we take: left, middle, or right?



If/Elif/Else Statements

```
x = -5
```

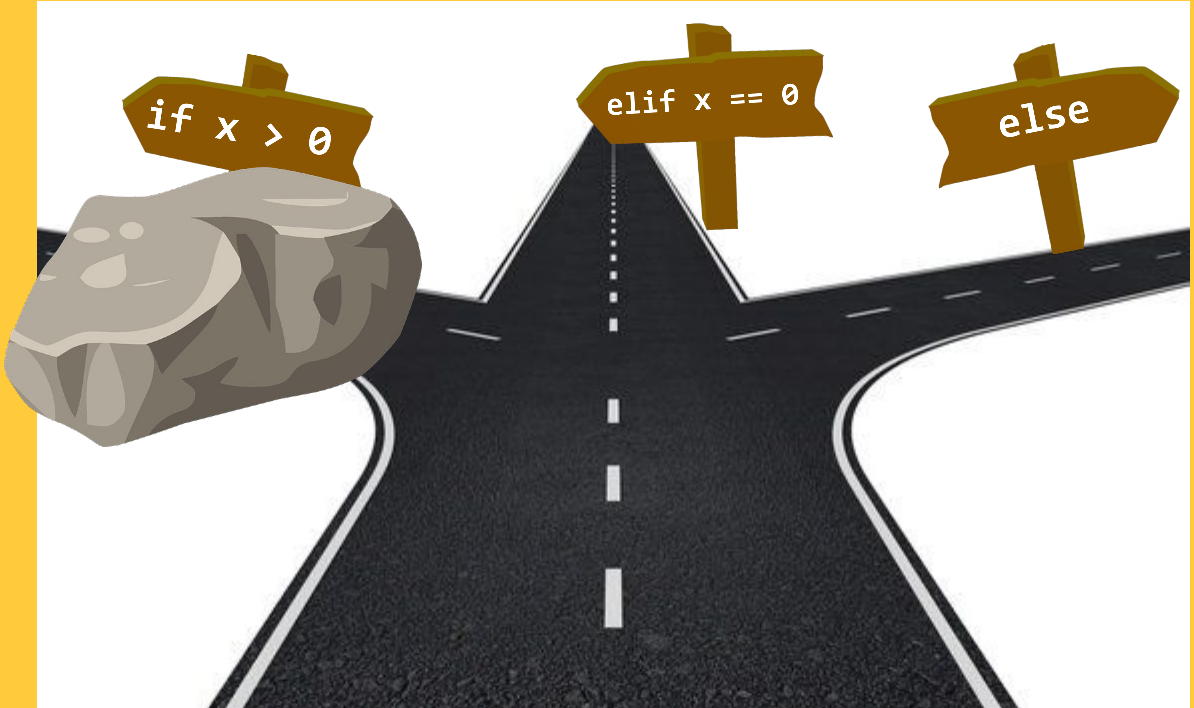
Which path would we take: left, middle, or right?



If/Elif/Else Statements

```
x = -5
```

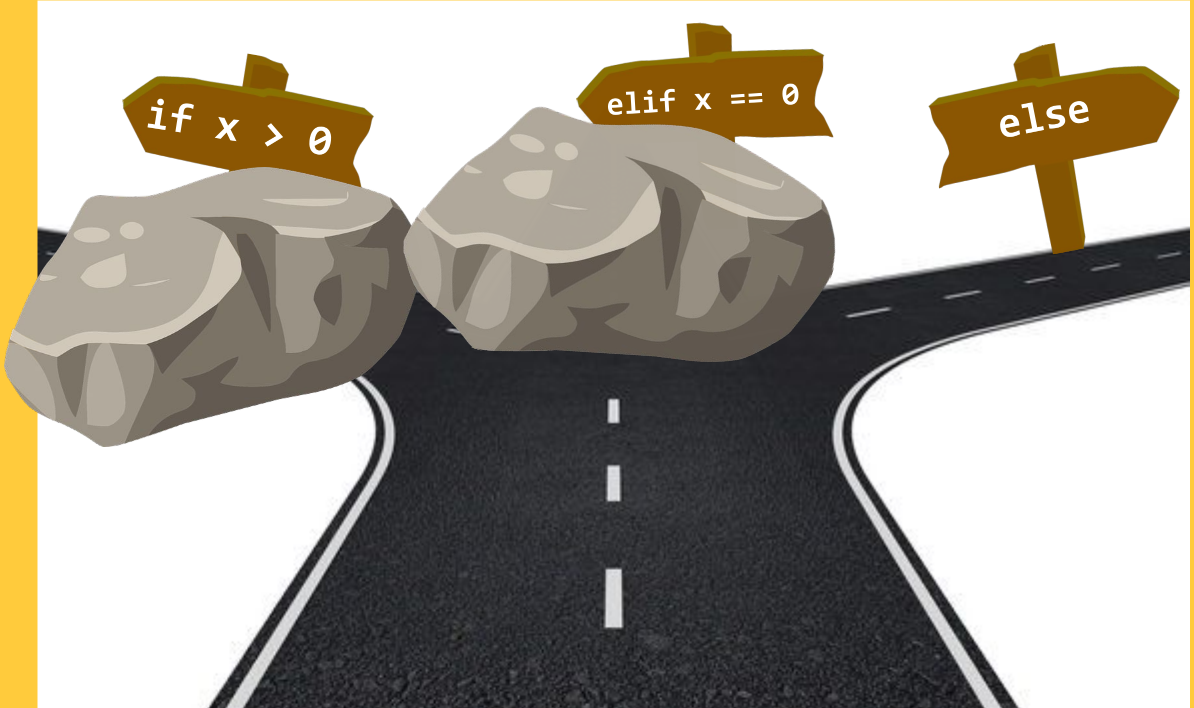
Which path would we take: left, middle, or right?



If/Elif/Else Statements

```
x = -5
```

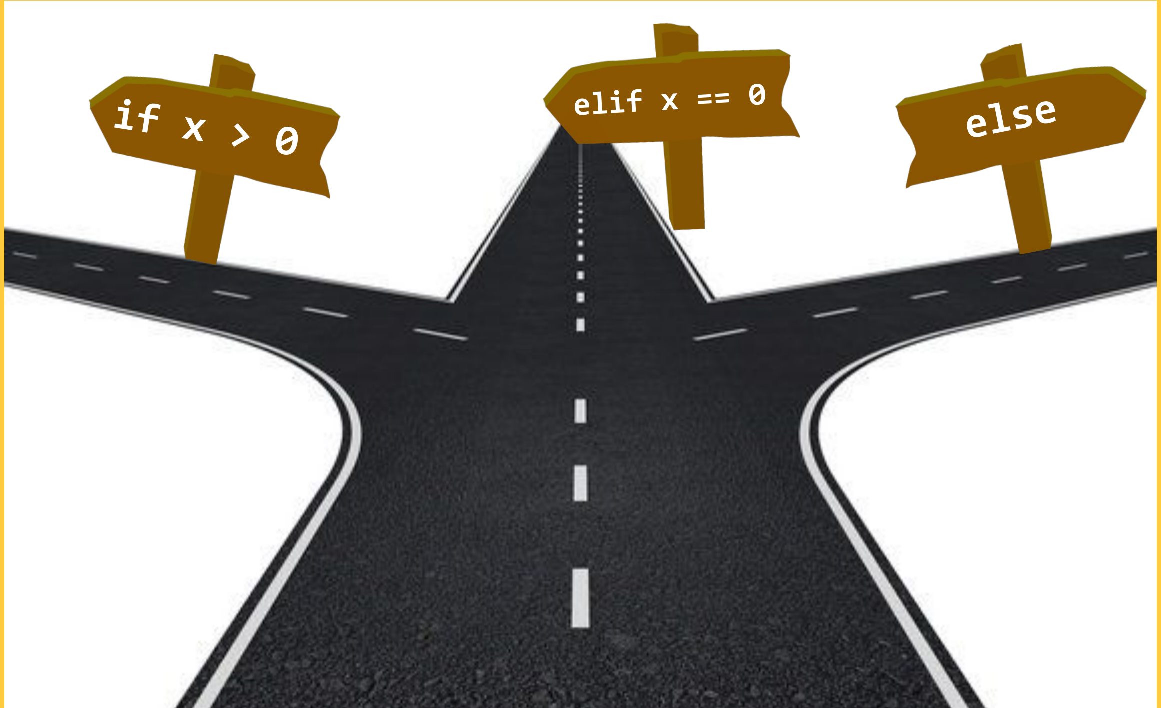
Which path would we take: left, middle, or right?



If/Elif/Else Statements

```
x = 8
```

Which path would we take: left, middle, or right?

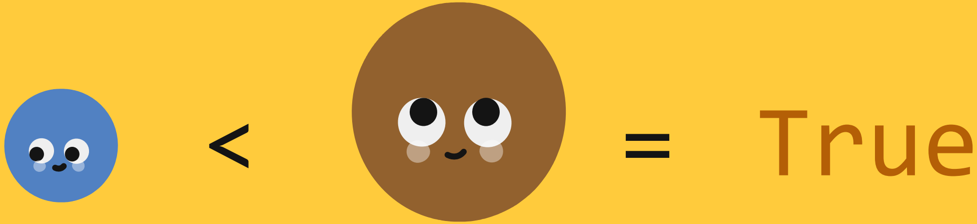


Comparison Operators

When you’re working in Python (especially with numbers) there are many times where you need to compare two different values.

In the context of numbers, the main comparison operators you’ll see are:

- `==` operator; evaluates to `True` if both sides are equivalent
- `>` operator; evaluates to `True` if the left side is larger than the right
- `<` operator; evaluates to `True` if the left side is smaller than the right
- `!=` operator; evaluates to `True` if both sides are NOT equivalent



Logical Operators

Lastly, there are times where we want to write code that requires multiple or more complex conditions.

Going over the logical operators again, we have:

- `and` operator; only runs code if both conditions joined by it evaluate to `True`. If either is `False`, the entire statement is `False`!
- `or` operator; runs code if either of the conditions joined by it evaluate to `True`. The entire statement is only ever `False` if both conditions are `False`!
- `not` operator; only takes in one condition! It then swaps the condition’s evaluation. This means if a condition previously evaluated to `True` it becomes `False` and vice-versa.

Questions?

What questions do you have before we start the section problem?

High Low

Section Problem

Setting Context

You are a game developer tasked with creating the next hit game: **High-Low**.

The game goes like this:

1. Two numbers are generated from 1 to 100 (inclusive on both ends): one for you and one for a computer, who will be your opponent. You don't get to see what number the computer has!
2. You make a guess, saying your number is either higher than or lower than the computer's number
3. If your guess matches the truth (ex. you guess your number is higher, and then your number is actually higher than the computer's), you get a point!

These steps make up one round of the game. The game is over after all rounds have been played. Let's walk through an example of one round of High-Low!

Example Round

Step One: Generate the Numbers

Your Number	Your Choice	Computer's Number
?	?	?

Example Round

Step One: Generate the Numbers

Your Number	Your Choice	Computer's Number
88	?	?

Example Round

Step Two: Make Your Choice (higher or lower)

Your Number	Your Choice	Computer's Number
88	?	?

Example Round

Step Two: Make Your Choice (higher or lower)

Your Number	Your Choice	Computer's Number
88	higher	?

Example Round

Step Three: Check the Results

Your Number	Your Choice	Computer's Number
88	higher	?

Example Round

Step Three: Check the Results

Your Number	Your Choice	Computer’s Number
88	higher	35

Example Round

Step Four: Be Happy You Won!!

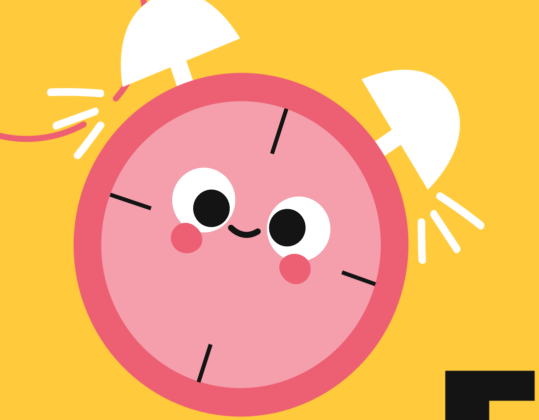
✓

Your Number	Your Choice	Computer’s Number
88	higher	35

Questions?

What questions do you have before we begin coding the problem?

Let’s Code!



Extensions

Additional features to add if time allows.

Take Your Pick!

Safeguard User Input

In an ideal world, people would only enter valid inputs. However, sometimes intentionally or unintentionally, invalid inputs may be entered instead which breaks our code! Can we modify our code in order to ensure that the input can only be “higher” or “lower”?

Conditional Ending Messages

A printed score doesn’t say much. To give more context to the result, we can add conditional messages at the end which gauge players on how they performed! Write different messages for winning all, at least half, or less than half the rounds.