Control Flow

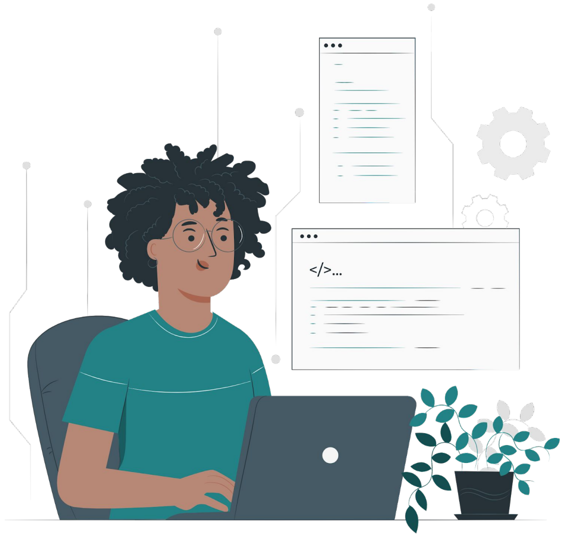
6.0001Lecture 2

LAST WEEK

- Computation
- Memory and variables
- Data types (integer, float, string)
- Input
- Style
- Debugging

The Educational Justice Institute

Today's Agenda

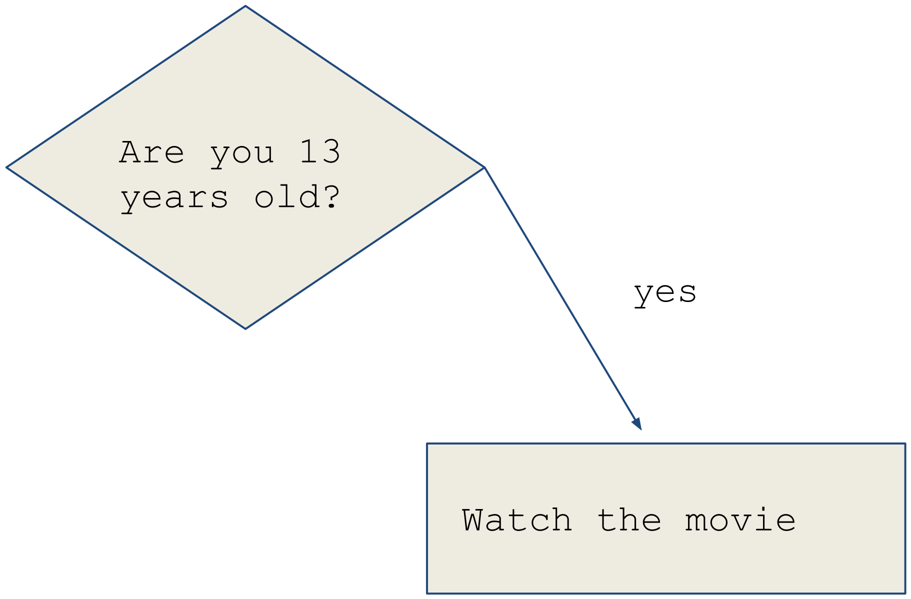


1. Control Flow
2. Breakout Rooms

Can someone watch the PG-13 movie?

- `You are a movie clerk`
- `How would you make this decision?`

Can someone watch the PG-13 movie?



Control Flow

Control flow is what we call the types of constructs in a programming language that allow us to **control** the **flow** of execution in our code.

So far, Python has executed each line 1 by 1, in order.

```python
age = int(input("How old are you?"))
print("Enjoy the movie")
```

Control flow helps our code make decisions and perform things based on those decisions.

Control Flow

In Python, control flow is

- Conditionals (if/elif/else)
- Loops (while, for)

Control Flow

In Python, control flow is

- **Conditionals (if/elif/else)**
- Loops (while, for)

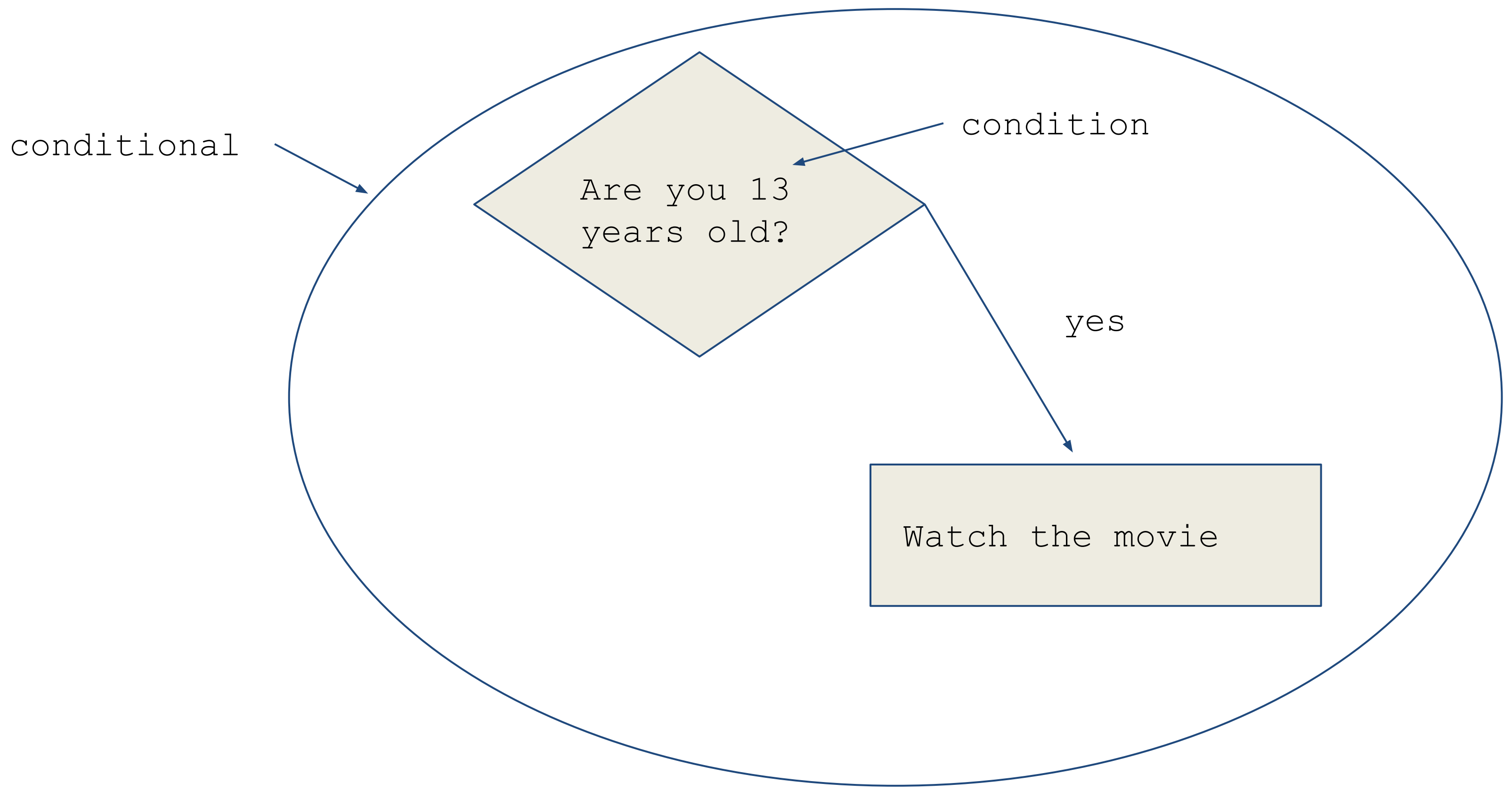
We will learn conditionals today, and loops next week.

Conditionals

condition: expression that evaluates to True or False, used in a conditional statement.

conditional: control structure to run a code block only if a certain condition is met.

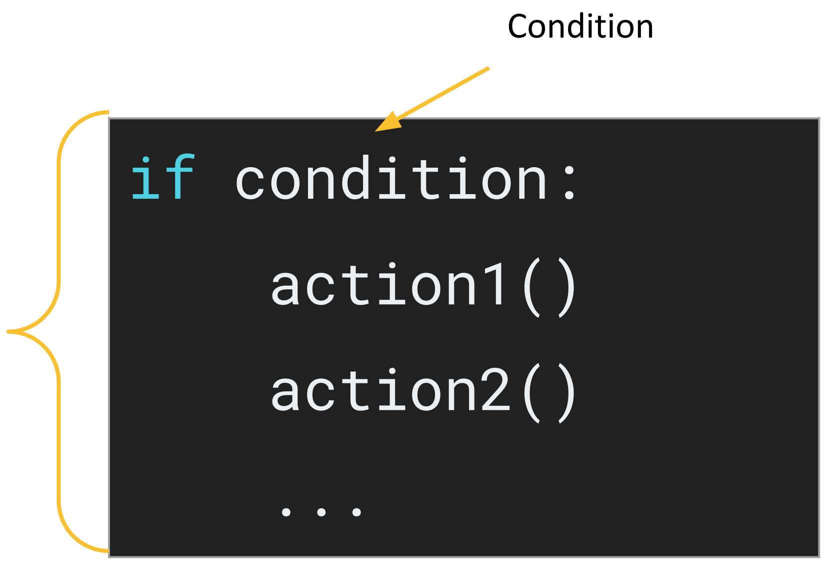
Can someone watch the PG-13 movie?



If Statements

We tell Python what's part of the conditional based on indentation.

All lines that are indented immediately after the condition are part of the conditional, and will only be executed if the condition is true.



```
if condition:
    action1()
    action2()
    ...
```

If `condition` evaluates to True, then all indented lines after the condition get executed.

Otherwise, they get skipped

If Statements

We tell Python what’s part of the conditional based on indentation.

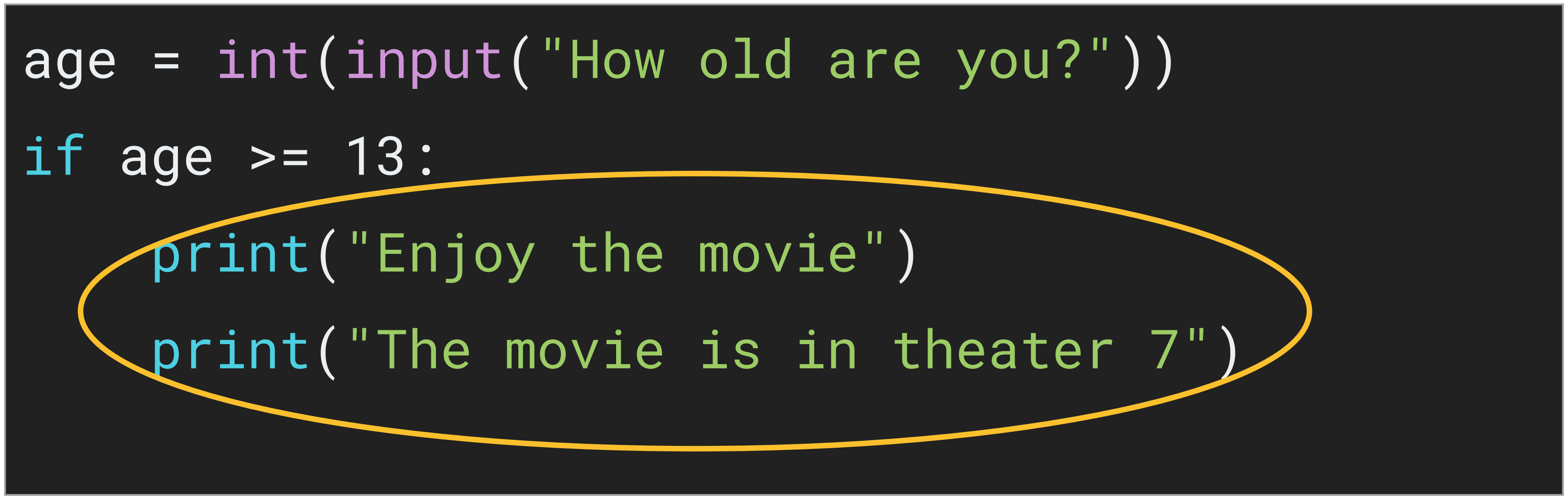
All lines that are indented immediately after the condition are part of the conditional, and will only be executed if the condition is true.

```
age = int(input("How old are you?"))
if age >= 13:
    print("Enjoy the movie")
```

If Statements

We tell Python what’s part of the conditional based on indentation.

All lines that are indented immediately after the condition are part of the conditional, and will only be executed if the condition is true.



```python
age = int(input("How old are you?"))
if age >= 13:
    print("Enjoy the movie")
    print("The movie is in theater 7")
```

If Statements

We tell Python what’s part of the conditional based on indentation.

As soon as the lines go back to the previous level of indentation, they are no longer part of the conditional, and get executed every time.

```
age = int(input("How old are you?"))
if age >= 13:
    print("Enjoy the movie")
    print("The movie is in theater 7")
print("Next customer!")
```

Coding Example

example_1.py

If Statements

Let’s cover some other conditions we can check.

```python
age = int(input("How old are you?"))
if age >= 13:
    print("Enjoy the movie")
    print("The movie is in theater 7")
print("Next customer!")
```

Comparison Operators

It can be handy to be able to compare values. Python to the rescue…
All of the comparison operators evaluate to **True** or **False**

Operator	Name	Example
>	Greater than	a > b
<	Less than	a < b
>=	Greater than or equal to	a >= b
<=	Less than or equal to	a <= b
==	Equal to	a == b
!=	Not equal to	a != b

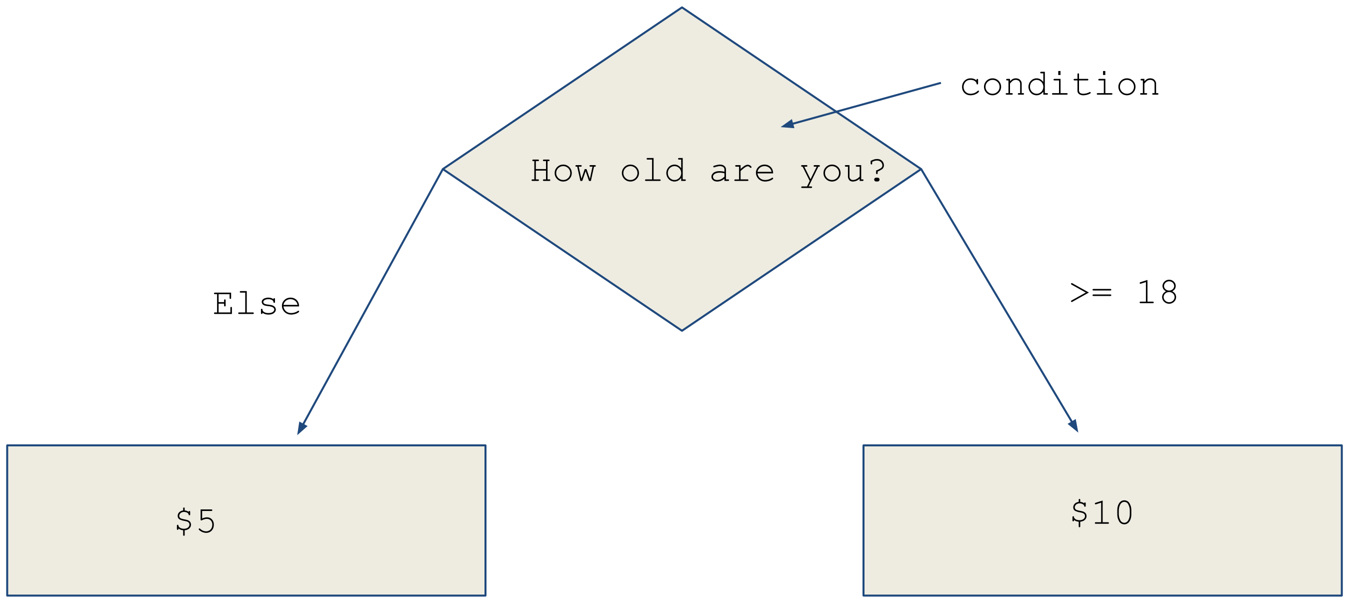
Comparison Operators

It can be handy to be able to compare values. Python to the rescue…
All of the comparison operators evaluate to a boolean, **True** or **False**

Operator	Name	Example
>	Greater than	a > b
<	Less than	a < b
>=	Greater than or equal to	a >= b
<=	Less than or equal to	a <= b
==	Equal to	a == b
!=	Not equal to	a != b

Common mistake is to accidentally use one equals sign, =, which is used for variable assignment, instead of two equals signs, ==, for comparison.

How much to charge for a movie ticket?



Else Statement

What if we also want to give alternative instructions for if the condition *isn’t* met?

Use an `else` statement:

```
if condition:
    action1()
    action2()
    ...
else:
    alt_action1()
```

Else Statement

What if we also want to give alternative instructions for if the condition *isn't* met?

Use an `else` statement:

```python
age = int(input("How old are you?"))
if age >= 18:
    print("The movie ticket will be $10")
else:
    print("The movie ticket will be $5")
```

Else Statement

What if we also want to give alternative instructions for if the condition *isn't* met?

Use an `else` statement:

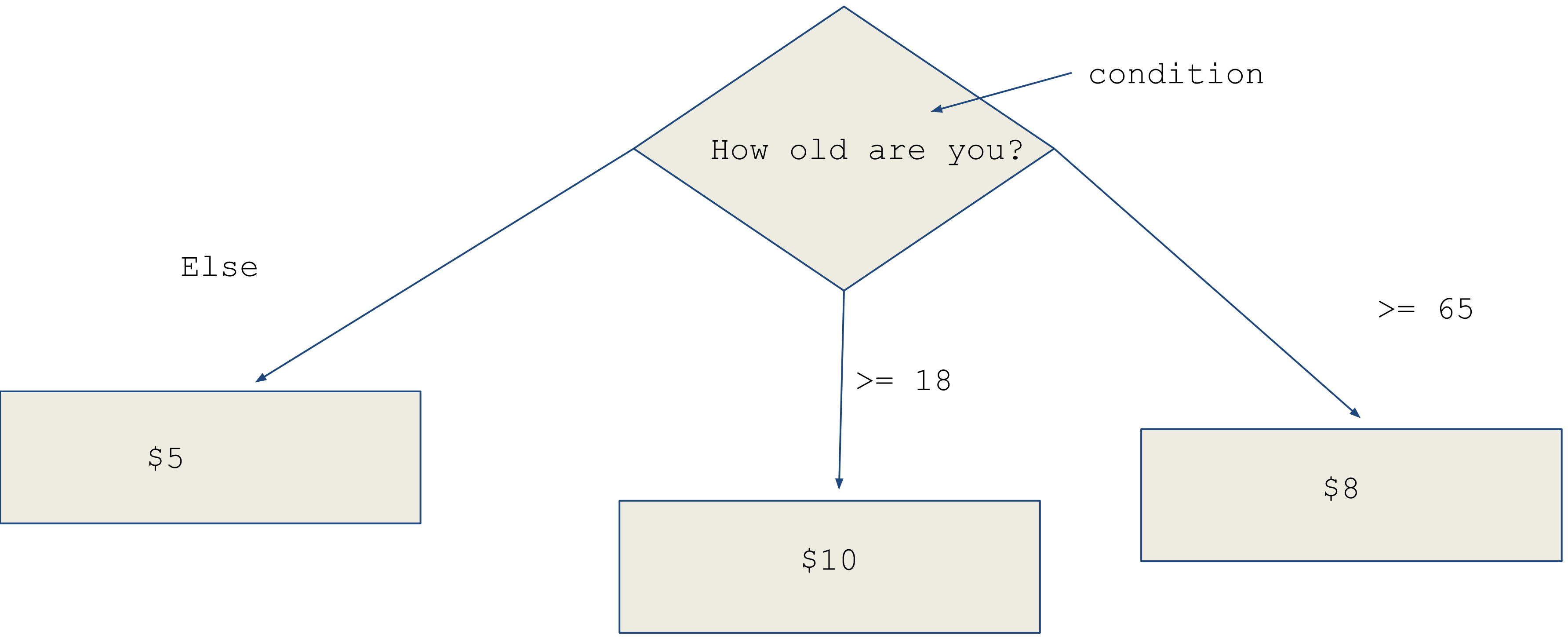
As soon as the lines go back to the previous level of indentation, they are no longer part of the conditional, and get executed every time.

```python
age = int(input("How old are you?"))
if age >= 18:
    print("The movie ticket will be $10")
else:
    print("The movie ticket will be $5")
print("Enjoy the movie")
```

Coding Example

example_2.py

How much to charge for a movie ticket?



Elif Statement

But sometimes there are more than just 2 choices…

Elif Statement

But sometimes there are more than just 2 choices…

Use an `elif` statement:

`elif` is short for "else if"

```
if condition1:
    action1()
    action2()
    ...
elif condition2:
    alt_action1()
    ...
else:
    alt_action2()
```

Elif Statement

We can keep adding as many conditions as we want.

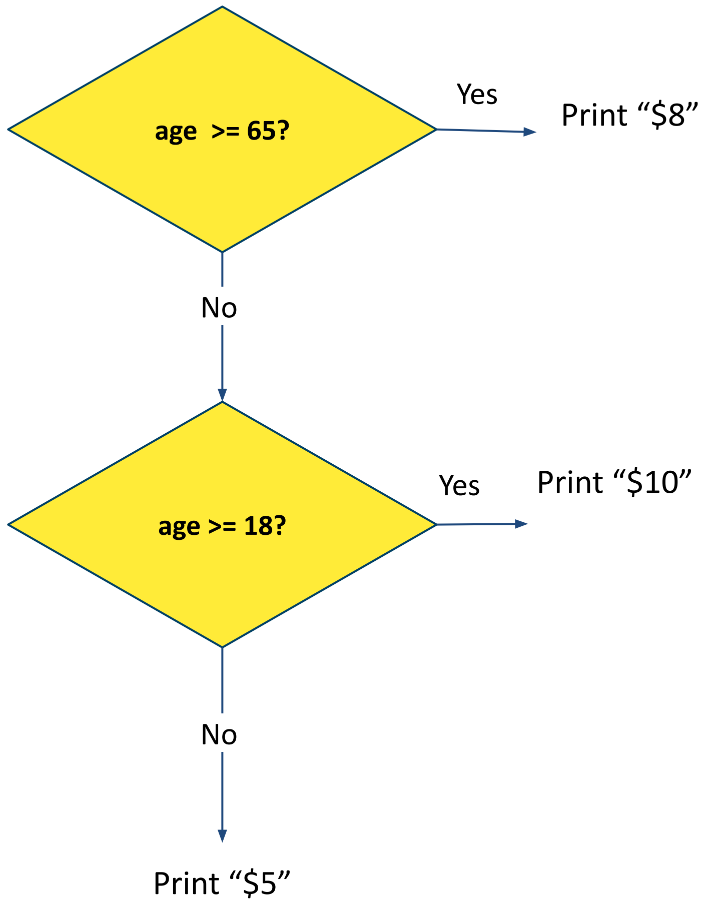
This block only gets executed if condition3 is True AND all previous conditions (condition1 & condition2) are False

```
if condition1:
    action1()
    action2()
    ...
elif condition2:
    alt_action1()
    ...
elif condition3:
    alt_action2()
else:
    alt_action3()
```

Elif Statement

Let’s see it in action:

```
age = int(input("How old are you?"))
if age >= 65:
    print("The movie ticket will be $8")
elif age >= 18:
    print("The movie ticket will be $10")
else:
    print("The movie ticket will be $5")
print("Enjoy the movie")
```



Coding Example

example_3.py

Breakout Rooms

1. Begin working on Problem Set 2