If / else statements

SHARP SIGHT

What you'll learn

- Basic conditional statements
  - `if`
  - `if` / `else`
  - `if` / `elif` / `else`

- How code blocks work in Python
  - white space is syntactically meaningful in Python

What are conditional statements?

Conditional statements

- 3 foundational types:
  - `if`
  - `if` / `else`
  - `if` / `elif` / `else`

- Conditional statements have the same basic structure:
  - Check condition
  - If the condition is true, then execute code

- To make these work, you need to know how to work with boolean logic

Why we use conditional execution

- Conditional statements allow us to change the operation of the program on the basis of different conditions

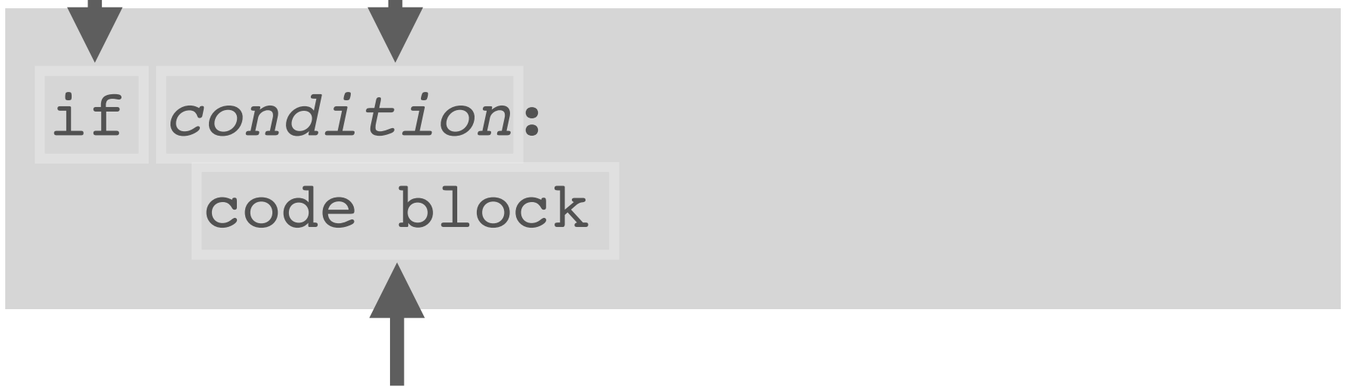
- Change the operation of code *if* a condition is true
  - If *this* is true, then do *something*
  - If *that* is true, then do *something else*
  - etc ...

The basic "if" statement

`if` STATEMENT SYNTAX

Use the `if` keyword to start an if statement

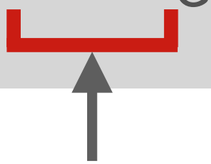
The *condition* is some boolean expression that evaluates to **`True`** or **`False`**



This code block will only be executed if the condition evaluates to `True`

The code blocks of an `if` statement must be indented

```
if condition:
    code block
```



This indentation must be present ....

The white space (i.e., indentation) is syntactically meaningful in Python

The best practice is to use 4 spaces to indent code blocks

Example: simple `if` statement

- Example: if `volume` equals 11, then print '`This one goes to eleven.`'

```
volume = 11
if volume == 11:
    print('This one goes to eleven.')
This one goes to eleven.
```

Example: simple `if` statement

This step checks if the statement is `True`

```
if volume == 11:
    print('This one goes to eleven.')
```

This code will only execute *if* the condition is `True`

... it will print out the message `'This one goes to eleven.'`

Example: simple `if` statement

This "condition" is a comparison statement

```
if volume == 11:
    print('This one goes to eleven.')
```

Remember: you learned about comparison operators and logic in a separate presentation

Example: simple `if` statement

This "condition" is `True`

```
volume = 11
if volume == 11:
    print('This one goes to eleven.')
This one goes to eleven.
```

... therefore, the `print()` statement will be executed.

If/else statements

`if/else` Statements

- **`if`/`else`** is similar to an **`if`** statement
  - We still check some condition

- However, **`if`** / **`else`** as *two* alternatives
  - code that's executed if the condition is **`True`**
  - also an alternative piece of code, executed if the condition is **`False`**

- There are two possible "branches" that the code can take

Example: `if/else`

- There are two fundamental alternatives
  - if the condition is `True`, then execute first code block
  - if the condition is `False`, (i.e., `else`) then execute the second code block

```
if x > 0:
    print('greater than 0')
else:
    print('less than or equal to 0')
```

Example: `if/else`

The `if` statement will check if this condition is `True`

If the condition is `True`, this indented block of code will execute

```
if x > 0:
    print('greater than 0')
else:
    print('less than or equal to 0')
```

If the condition is `False`, the indented block of code will execute

The condition of an if statement can be very complex

Note that this logical statement is very simple...

... but they can be much more complex!

```
if x > 0:
    print('greater than 0')
else:
    print('less than or equal to 0')
```

If/elif/else statements

`if`/`elif` Statements

- Same basic structure as **`if`** and **`if`** / **`else`**
  - conditions and code blocks

- However, **`if`**/**`elif`** has several conditions
  - each condition has separate code block
  - code block is only executed if the related condition is true

- Note: there's no limit on the number of **`elif`** conditions

Example: `if/elif`

- This example is a variation on “Hello World”
  - it prints different statements depending on conditions

```
if city == 'Cleveland':
    print('Hello Cleveland!')
elif city == 'New York':
    print('Hello New York!’)
elif city == 'Chicago':
    print('Hello Chicago!’)
```

first condition

second condition

third condition

if/elif/else

- You can put an **else** statement after **if** / **elif**
  - **else** is *not required*

```
if city == 'Cleveland':
    print('Hello Cleveland!')
elif city == 'New York':
    print('Hello New York!’)
elif city == 'Chicago':
    print('Hello Chicago!’)
else:
    print('Where are we?')
```

Which branch will the code take?

- The program will evaluate each condition in order
  - if the first condition is false, it will move on to the next one

- The program will execute the code block for the first true condition

- The program will *exit* the **`if`** / **`elif`** statement after finding the first true condition
  - i.e., for the first true condition, it will execute the code, and then exit

- Even if multiple conditions are true, *only the code for the first true condition gets executed*

Recap

Recap of what we learned

- Basic conditional statements
  - `if`
  - `if` / `else`
  - `if` / `elif` / `else`

- White space is syntactically meaningful in Python