


# PYTHON FUNDAMENTALS

What if

# LEARNING OBJECTIVES

- To understand if/else syntax
  - To understand and use comparison operators
  - To write programs with single and multiple conditions
- 

# What if?





**Imagine there's some music on**

**How do you feel about the music?**



# Stupid question!



**Depends on what the music is!**

```
music = "classical"
```

```
if music == "classical":
```

```
    print("Oh no it's that classical again")
```

```
elif music == "no music":
```

```
    print("Arh, peace and quiet")
```

```
else:
```

```
    print("Nice and noisy")
```



Have you noticed that the  
code is formatted really  
nicely?







**It's not by accident.**

**Python is whitespace dependent.**



**That sounds fancy but it basically  
means it matters where there are  
indents and new lines**

```
if condition1:
```

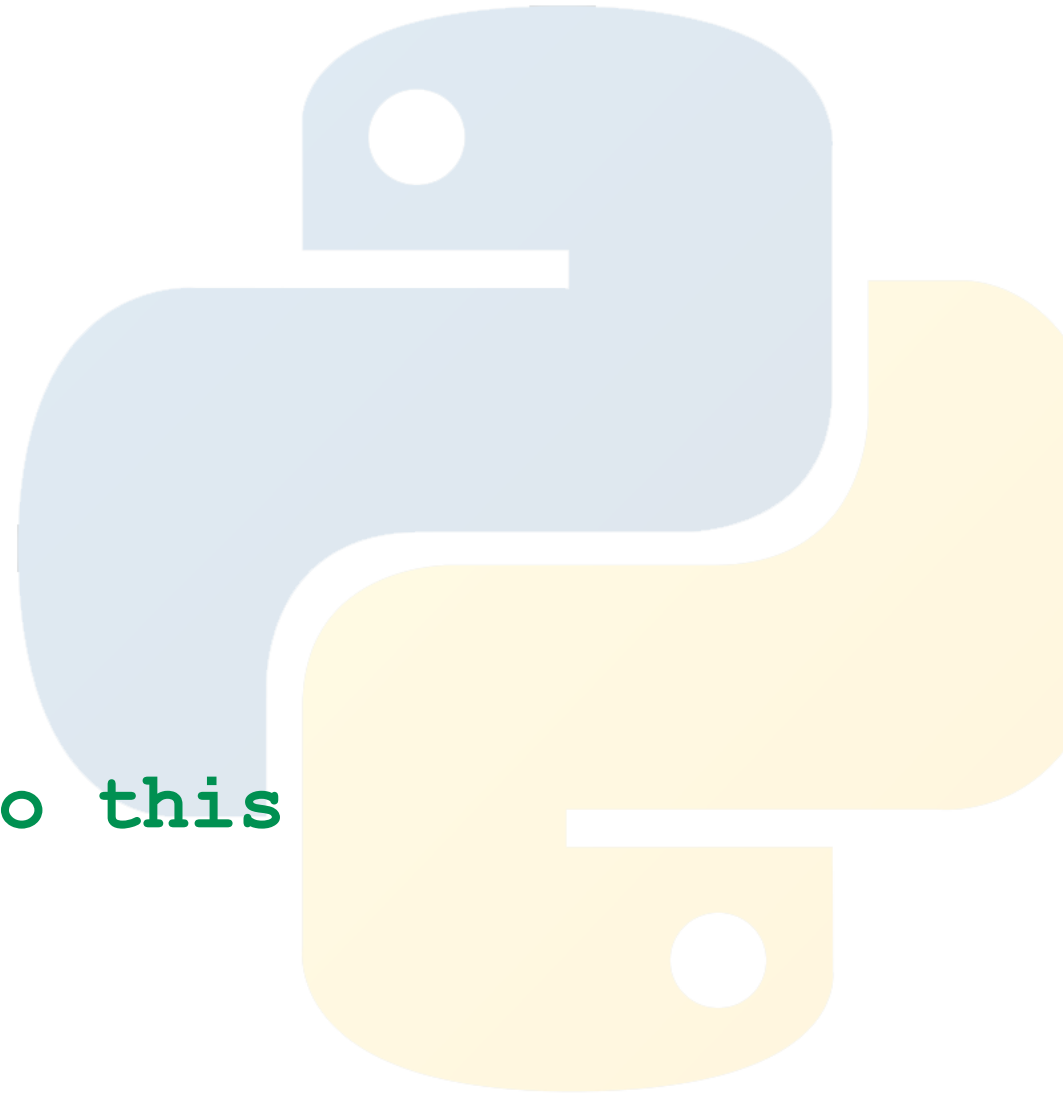
```
    #do this
```

```
elif condition2:
```

```
    #do this
```

```
else:
```

```
    #if nothing else matched do this
```



```
if music == "classical":  
    print("Oh no it's that classical again")
```

== ?



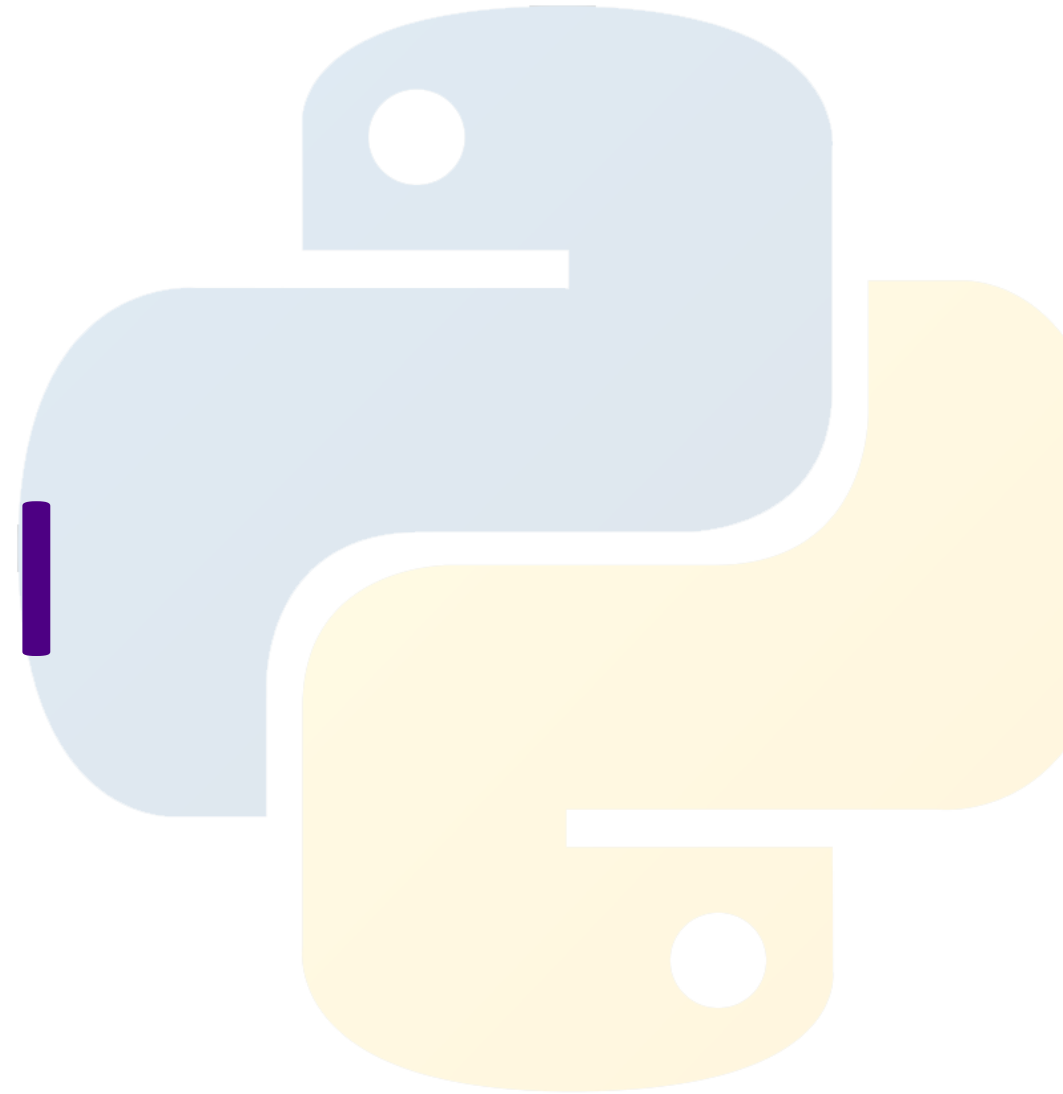
# Comparison Operators

**==**

**Equal**

**!=**

**Not equal**



$\geq$

$\leq$

$>$

$<$

```
music = "classical"
```

```
if music == "classical":
```

```
    print("Oh no it's that classical again")
```

```
elif music == "no music":
```

```
    print("Arh, peace and quiet")
```

```
else:
```

```
    print("Nice and noisy")
```





**To VSC**

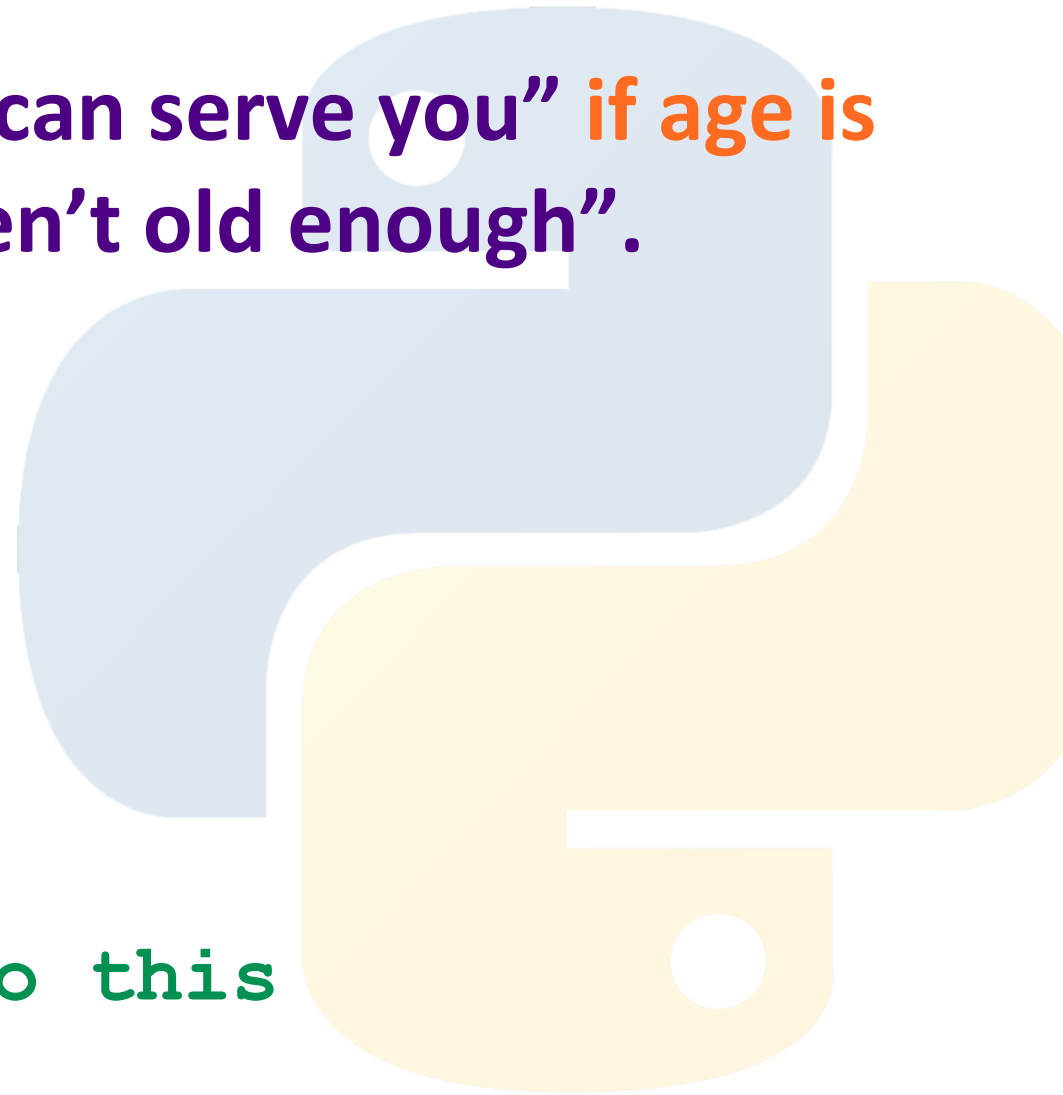


# Activity:

Create a variable called age.

Write an if statement that logs “Yes I can serve you” if age is greater than 17 and else logs “You aren’t old enough”.

```
if condition1:
    #do this
else:
    #if nothing else matched do this
```



# Possible solution

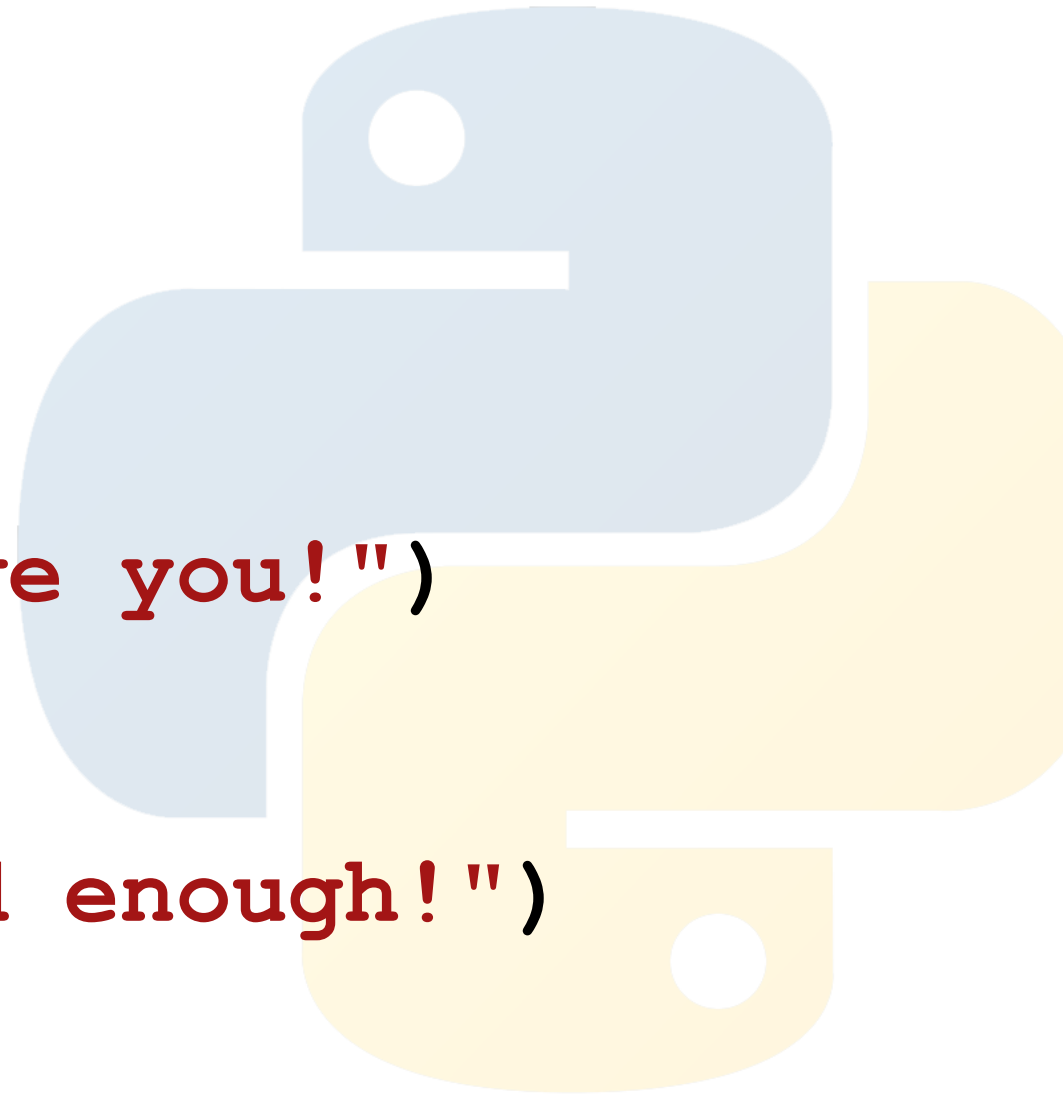
```
age = 20
```

```
if age > 17:
```

```
    print("Yes I can serve you!")
```

```
else:
```

```
    print("You aren't old enough!")
```



**And onto the next  
thing**



```
place = "MCR"
```

```
weather = "Cloudy"
```

```
if place == "MCR" and weather == "Sunny":
```

```
    print("Check again")
```

```
elif place == "MCR" and weather == "Rain":
```

```
    print("Obvs")
```

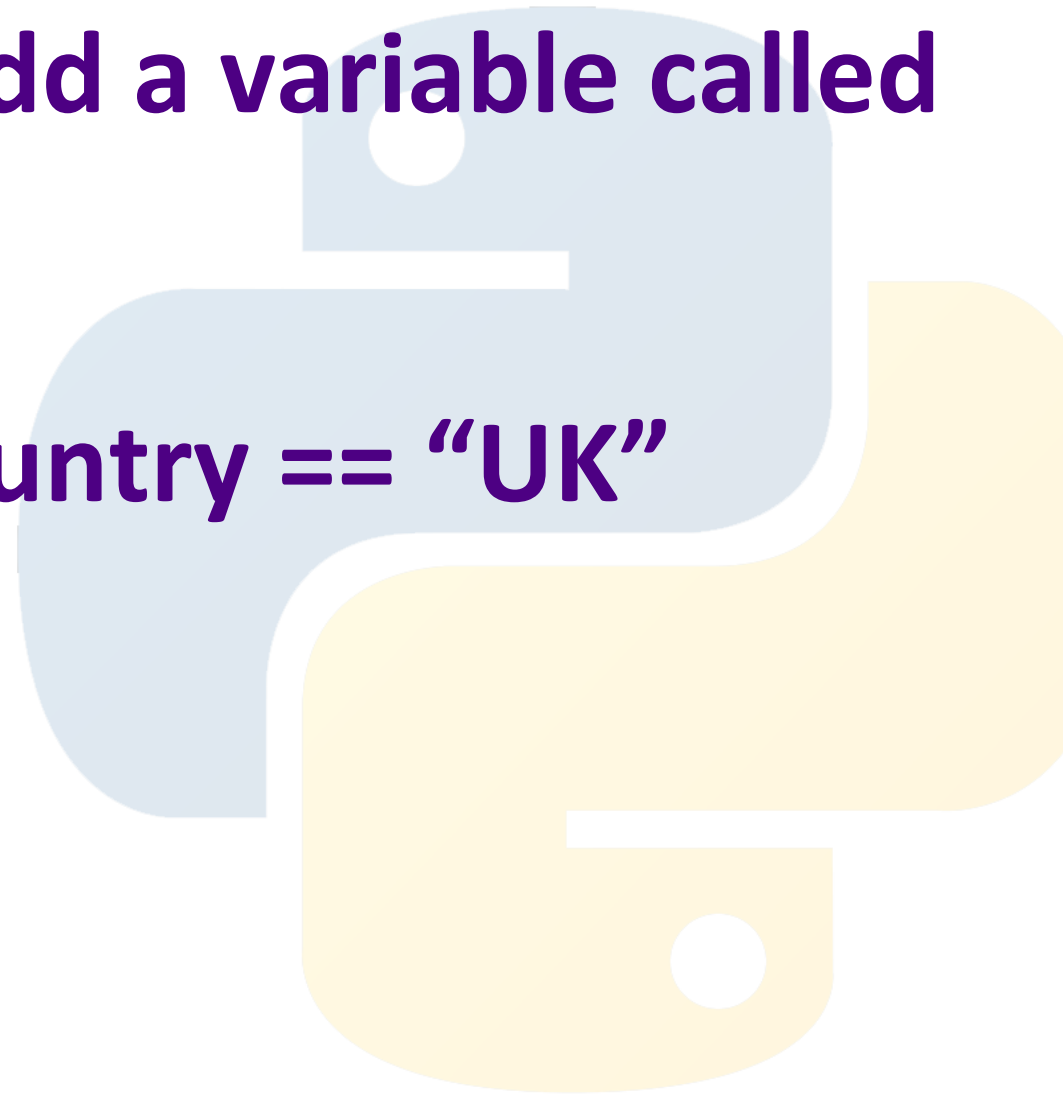
```
else:
```

```
    print("What it isn't raining?")
```

# Activity:

Take your if statement and add a variable called country.

Now check if `age > 17` and `country == "UK"`



# Possible solution (1)

```
age = 20
```

```
country = "UK"
```

```
if age > 17 and country == "UK":
```

```
    print("Yes I can serve you")
```

```
elif age > 17 and country != "UK":
```

```
    print("Where are you?")
```

```
else:
```

```
    print("You aren't old enough")
```



# Possible solution (2)

```
age = 20
```

```
country = "UK"
```

```
if age > 17 and country.lower() == "uk":
```

```
    print("Yes I can serve you")
```

```
elif age > 17 and country.lower() != "uk":
```

```
    print("Where are you?")
```

```
else:
```

```
    print("You aren't old enough")
```

Or not?





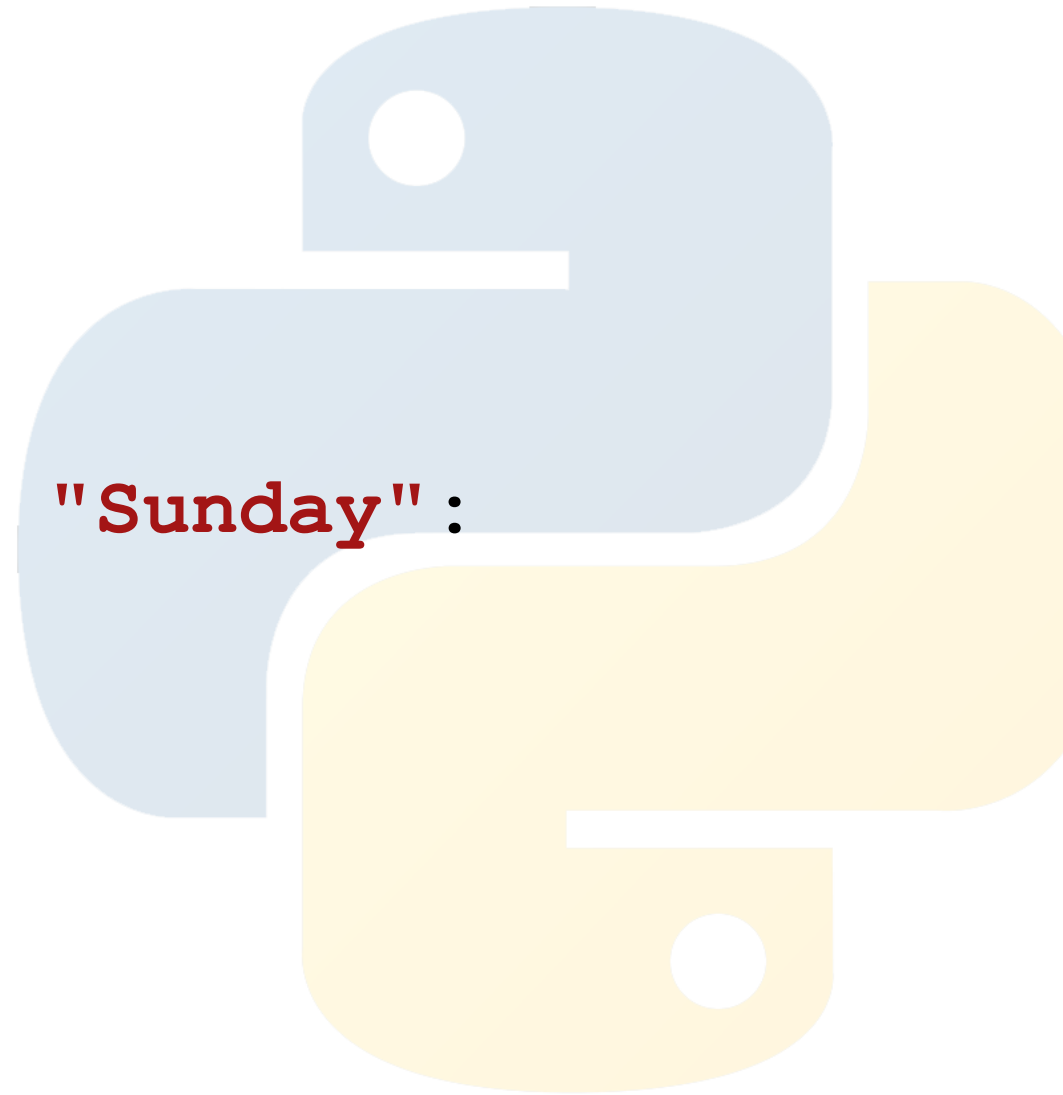
```
day = "Saturday"
```

```
if day == "Saturday" or day == "Sunday":
```

```
    print("It's weekend!")
```

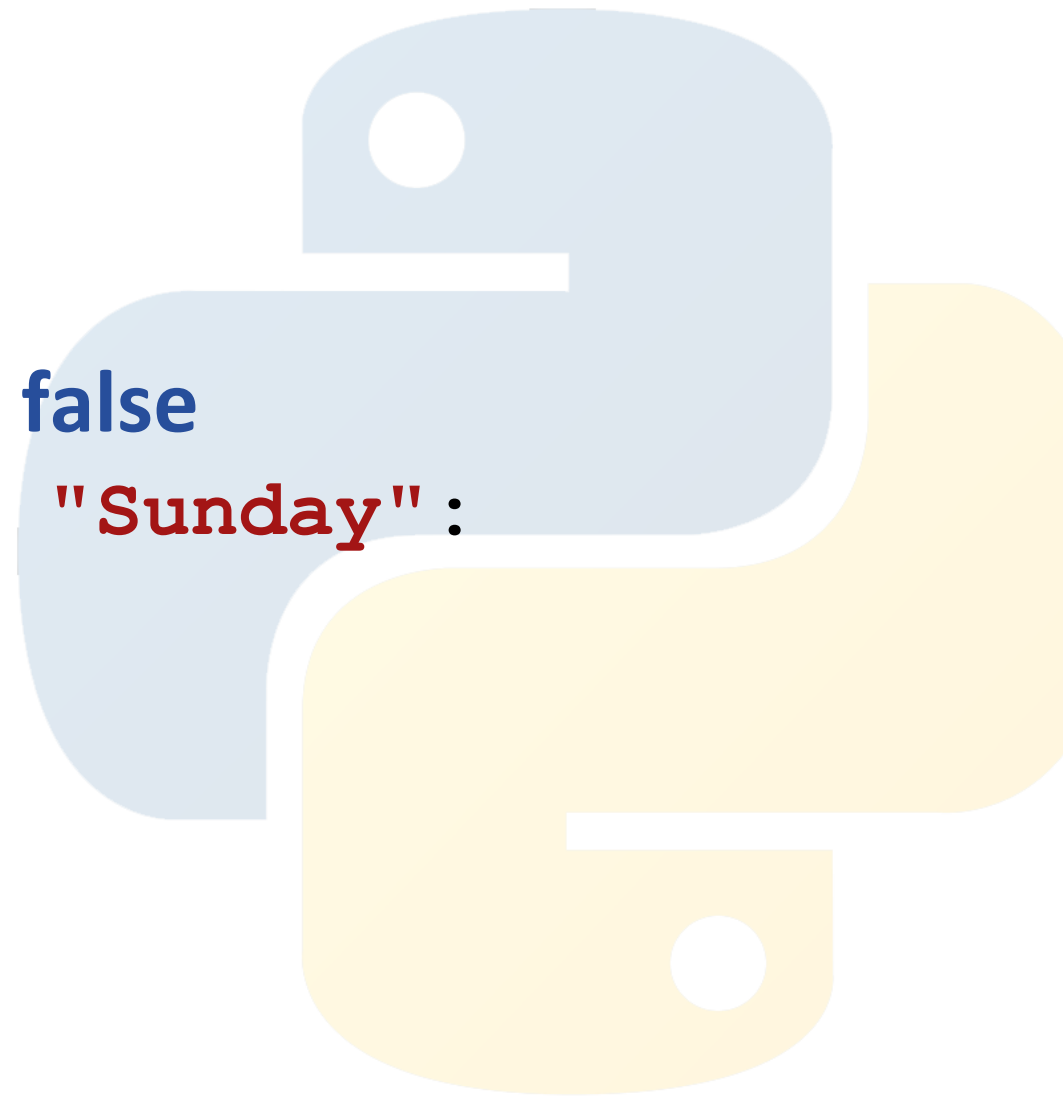
```
else:
```

```
    print("When's weekend?")
```



```
day = "Saturday"
```

```
                true    or    false
if day == "Saturday" or day == "Sunday":
    print("It's weekend!")
else:
    print("When's weekend?")
```



# In the condition we have

expression

To Be

Evaluated

logicalOperator

and/or

expression

To Be

Evaluated

**It's only logical**



# and

True and True —>  
True and False —>  
False and False —>

# and

True and True  $\rightarrow$  True  
True and False  $\rightarrow$  False  
False and False  $\rightarrow$  False

**or**


True and True —>  
True and False —>  
False and False —>

or

True and True  $\rightarrow$  True  
True and False  $\rightarrow$  True  
False and False  $\rightarrow$  False



# LEARNING OBJECTIVES

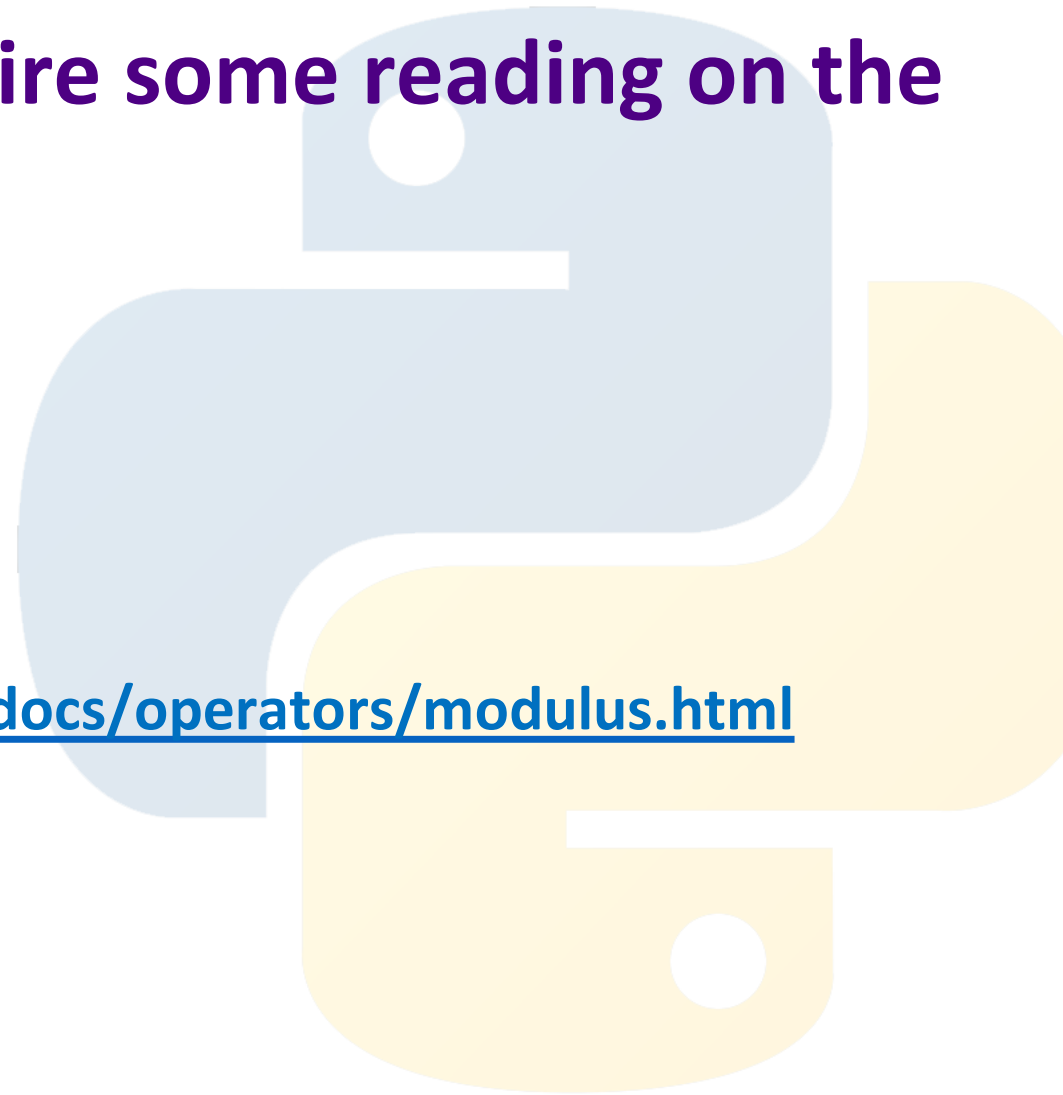
- To understand if/else syntax
  - To understand and use comparison operators
  - To write programs with single and multiple conditions
- 
- A large, solid orange shape occupies the bottom half of the slide. It has a jagged, mountain-like silhouette with several peaks and valleys, creating a modern, abstract background element.

# Pre-reading: modulus %

Some of the challenges below require some reading on the use of the modulus symbol %

This would be a starting point:

<https://python-reference.readthedocs.io/en/latest/docs/operators/modulus.html>



## Challenge 1:

Create a variable called password.

Check how many letters are in the password, if there are less than 8 print that the password is too short. Otherwise print the password.

## Challenge 2:

Create a variable called num.

Check if the variable is divisible by 3 or 5. If it is print “This number is divisible by 3 or 5” to the console. Otherwise log “This number is not divisible by 3 or 5”.

### Challenge 3:

Create a variable called num.

If num is divisible by 3 print “fizz”, if it’s divisible by 7 print “buzz”, if it’s divisible by both 3 and 7 print “fizz buzz”.

Otherwise print num.

### Challenge 4:

Create a variable called word that takes a string.

Create an if statement that checks if the last letter is the same as the first. If it is return true, otherwise return false.

## Challenge 5:

Create a variable called `time`, a variable called `place_of_work` and a variable called `town_of_home`. Create an if statement that prints where someone is at times of the day. E.g. if the time is 7 I'm at home, at 8 I'm commuting, at 9 I'm at work.

## Challenge 6:

Create two variables called `num1` and `num2`. Create an if statement that checks if the result of the sum is even. If it is, return a success message.

# Extra Challenges

## Challenge 7:

Create a variable called num.

Check if the number is a palindrome (looks the same forward as it does backwards e.g. 1001 or 20202).

# Extra

## Challenge 8:

Take the string

“jrfndklhgfndjkjlkgsperfijfhdknsadcvjhiihjfkledsopiuhgty  
ujwsdxcvhgfdjhiopiwquhejkdsoiufghedjwshi”.

Find the index of a last vowel in the string.

Extra reading

# Truthy and Falsey





Conditions can use Truthy and Falsey  
values for strings

# Let's take this in:

```
print("What is your name?")
```

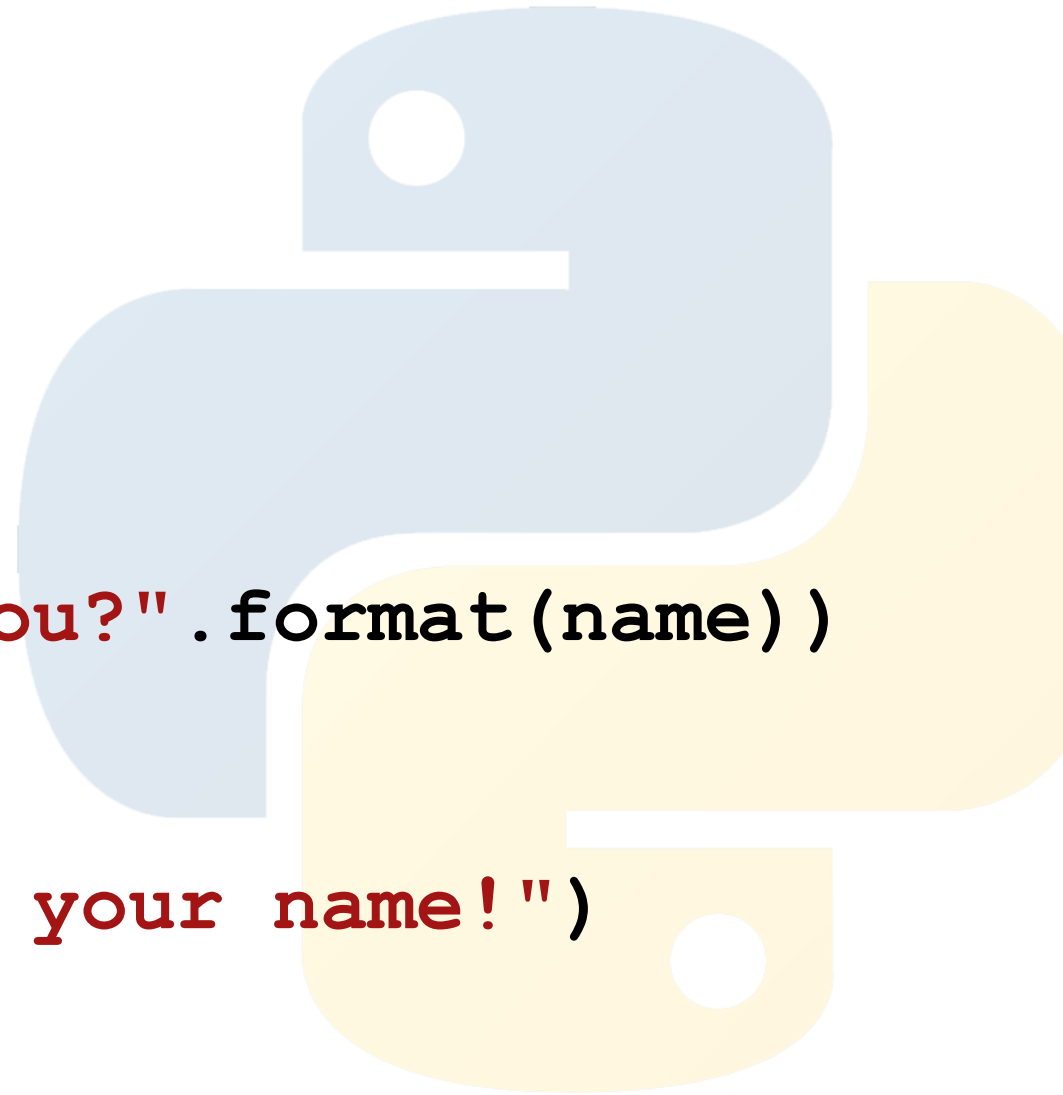
```
name = input()
```

```
if name:
```

```
    print("Hello {}, how are you?".format(name) )
```

```
else:
```

```
    print("You did not give me your name!")
```



# Let's take this in:

```
print("What is your name?")
```

```
name = input()
```

```
if name:
```

```
    print("You gave me your name!")
```

```
else:
```

```
    print("You did not give me your name!")
```

If this string is a truthy value, so  
this condition is met.

```
format(name) )
```

# Let's take this in:

```
print("What is your name?")
```

```
name = input()
```

```
if name:
```

```
    print
```

```
else:
```

```
    print("You did not give me your name!")
```

Blank string is falsey, and therefore  
else will be executed.

```
    print(name)
```

# Falsy values

- } Empty string
- } Value 0
- } Floating point value 0.0
- } Everything else is Truthy

