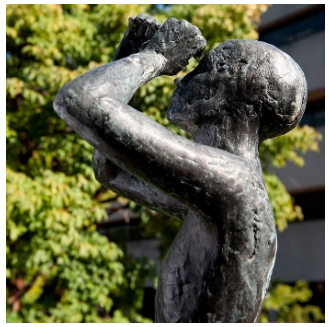


CS1117 – Introduction to Programming

Dr. Jason Quinlan,

School of Computer Science and Information Technology

A TRADITION OF INDEPENDENT THINKING

UCC

University College Cork, Ireland

Coláiste na hOllscoile Corcaigh

Announcements

Lab 1

How did you get on?

Too difficult, too easy?

Let’s find out…

LiveSlides web content

To view

Download the add-in.

liveslides.com/download

Start the presentation.

LiveSlides web content

To view

Download the add-in.

liveslides.com/download

Start the presentation.

Announcements

Lab 1

Okay, now we know how people found Lab 1

Let’s consider what the labs are for:

Announcements

Labs

The labs give you a chance to code what we cover in class

The labs get you thinking in CS logic

The labs get you used to deadlines

The labs are meant to be fun, well enjoyable at least…

Announcements

Labs – my advise

Read over all the exercise sheet before beginning

If there is something you are unsure of, put your hand up…

Ask lots and lots of questions…

After the lab, email me if you still have concerns…

Announcements

Lab 1

I'm going to re-open Lab 1

With a submission date of this coming Saturday @ 1am

If you haven't completed all three exercises, please try again

I have uploaded a video of me coding the answers to Lecture-2-Exercise-Sheet-submit.txt

So watch the video and you can see what is expected in each lab

Announcements

Lab 2

There will be a new lab this week (2 exercise sheets)

Make sure you can log into the G20 machines

Don’t leave your login details at home or in the lab…

Bring them and learn them…

You will need them for the next 4 years ☺

Lectures Recap – 2 weeks in

- We now know how to define variables
  - We looked at int, float and string
  - We looked at casting between types
- We can print(), type() and id() the variable values
- We looked at defining a function
  - With zero to multiple input parameters
  - And set default values to the parameters
  - One or more returned values
  - We defined docStrings
- We looked at Python operators and their mutability
- We looked at print() and its string manipulation
  - And also looked at special characters (\t and \n)
- We imported functions from Pythons library
  - as well as functions from files we created ourselves
    - Both from files in the same folder and from nested folders

Amazing…

- Coding has 2 main building blocks and you've now covered the majority of both of them.
- From today onwards, we will move beyond the structure of our code and into the logic
- So, let's return to "lecrture5_v2.py from last week"

if statement

Last week, we added two lines of code to the end of “lecture_5_v2.py”

lecture_5_v2.py | lecture_6.py

```
def average_of_two(number_1, number_2):
    """ this is my 'docstring'
    function to determine the average of two numbers
    number_1: first int to pass in
    number_2: second number to pass in
    """
    print("number_1 is ", number_1)
    print("number_2 is ", number_2)
    # determine the average of two numbers
    average = (number_1+number_2)/2
    # return the average number
    return average


def main():
    number_one = 2
    number_two = 4
    print(average_of_two(number_one, number_two))


if __name__ == "__main__":
    main()
```

if statement

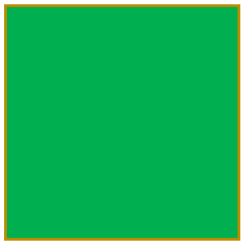
Last week, we added two lines of code to the end of “lecture_5_v2.py”

lecture_5_v2.py | lecture_6.py

```
def average_of_two(number_1, number_2):
    """ this is my 'docstring'
    function to determine the average of two numbers
    number_1: first int to pass in
    number_2: second number to pass in
    """
    print("number_1 is ", number_1)
    print("number_2 is ", number_2)
    # determine the average of two numbers
    average = (number_1+number_2)/2
    # return the average number
    return average


def main():
    number_one = 2
    number_two = 4
    print(average_of_two(number_one, number_two))


if __name__ == "__main__":
    main()
```

And I told you to ignore the line starting with if…
(we will come back to it, I said)

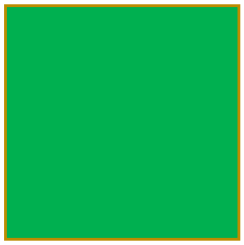
if statement



So let's get back to it. The lines of code were:

```
if __name__ == "__main__":
    main()
```

if statement



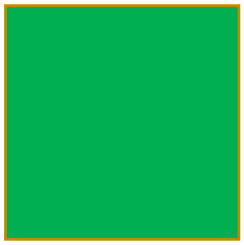
So let's get back to it. The lines of code were:

```
if __name__ == "__main__":
    main()
```

We understand the call to the function main()

Let's ignore that

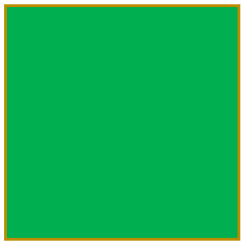
if statement



So let's get back to it. The lines of code were:

```
if __name__ == "__main__":
```

if statement



So let's get back to it. The lines of code were:

```
if __name__ == "__main__":
```

Now let's break this code down into its relevant parts

if

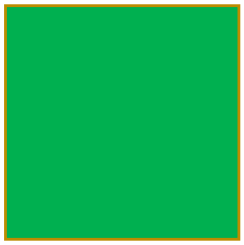
__name__

==

"__main__ ”

:

if statement



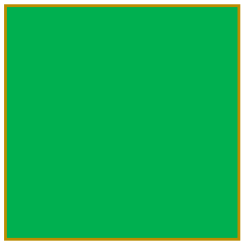
So let's get back to it. The lines of code were:

```
if __name__ == "__main__":
```

Now let's break this code down into its relevant parts

if	Keyword
__name__	
==	
"__main__"	
:	

Python reserved keywords

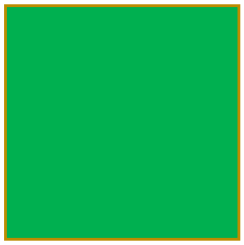


Just known that within Python

these 33 reserved keywords exist

False	def	if	raise
None	del	import	return
True	elif	in	try
and	else	is	while
as	except	lambda	with
assert	finally	nonlocal	yield
break	for	not	
class	from	or	
continue	global	pass	

if statement



So let's get back to it. The lines of code were:

```
if __name__ == "__main__":
```

Now let's break this code down into its relevant parts

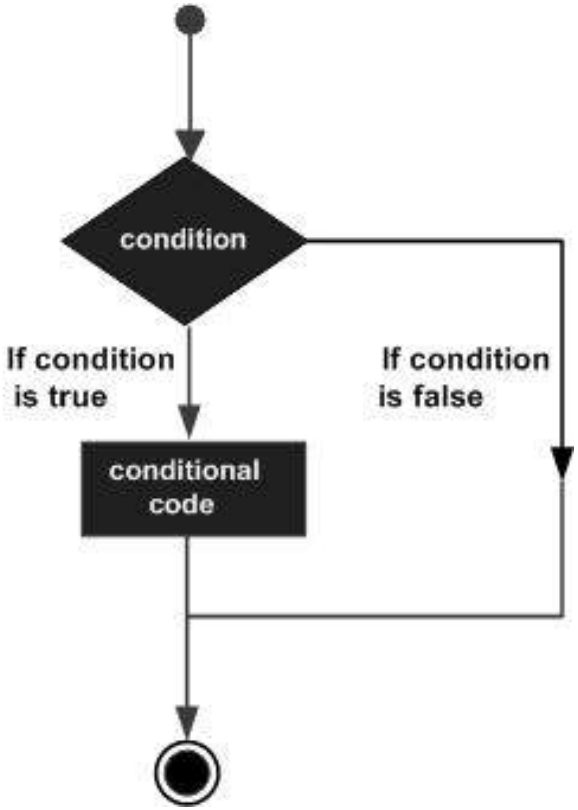
if	Keyword – start of a conditional statement
__name__	
==	
"__main__ "	
:	

if statement

if statements can be modeled as a flow chart.

If the condition is evaluated as true then execute the conditional code (statement block).

Otherwise skip that code.



if statement

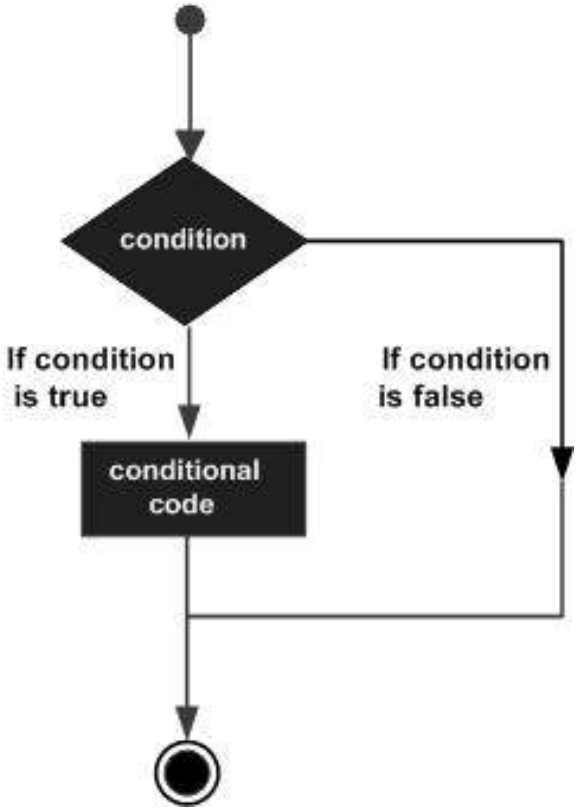
if statements can be modeled as a flow chart.

If the condition is evaluated as true then execute the conditional code (statement block).

Otherwise skip that code.

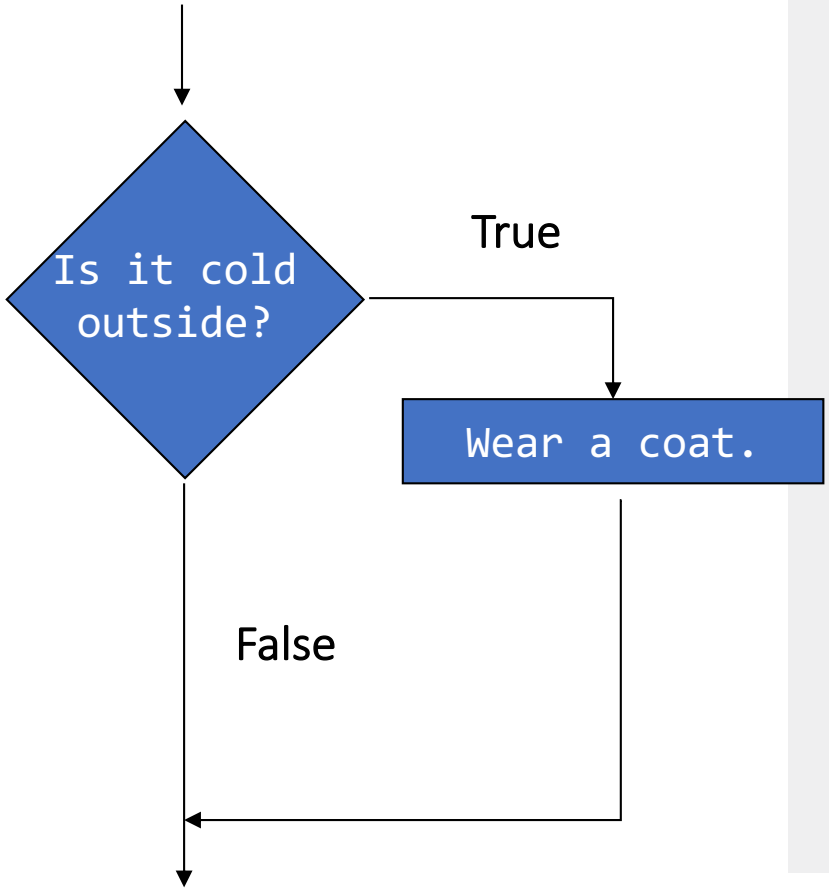
So, every if statement equates to either True or False

That's all…



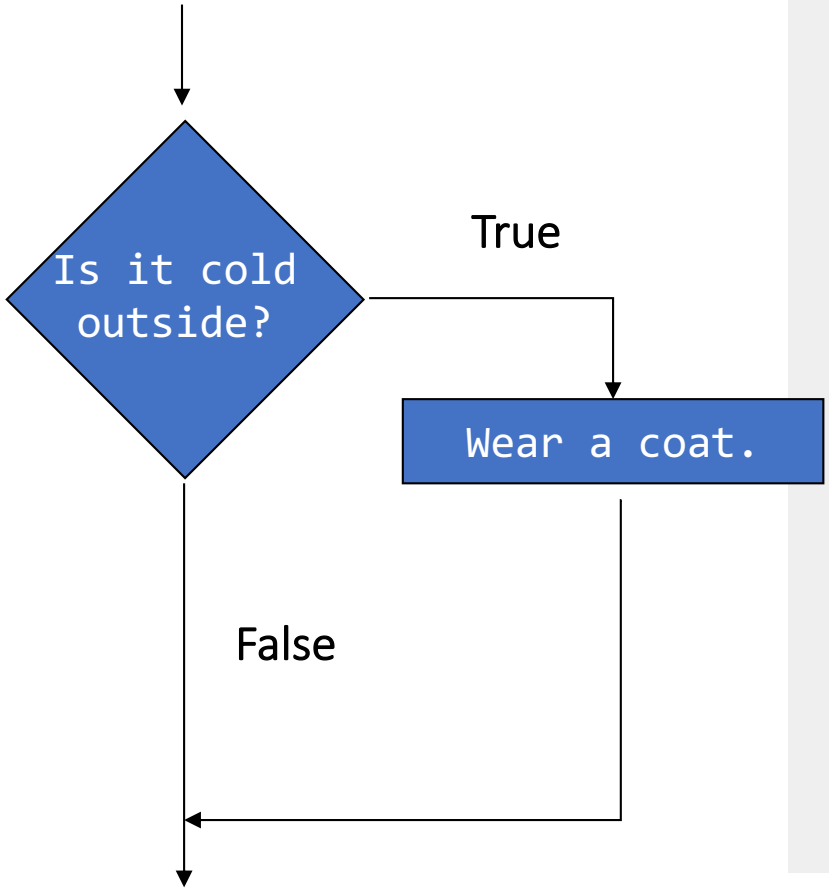
if statement

```
if cold_outside:
    print("wear coat")

# output
# wear coat
```



if statement

```
if cold_outside:
    print("wear coat")

# output
# wear coat
```

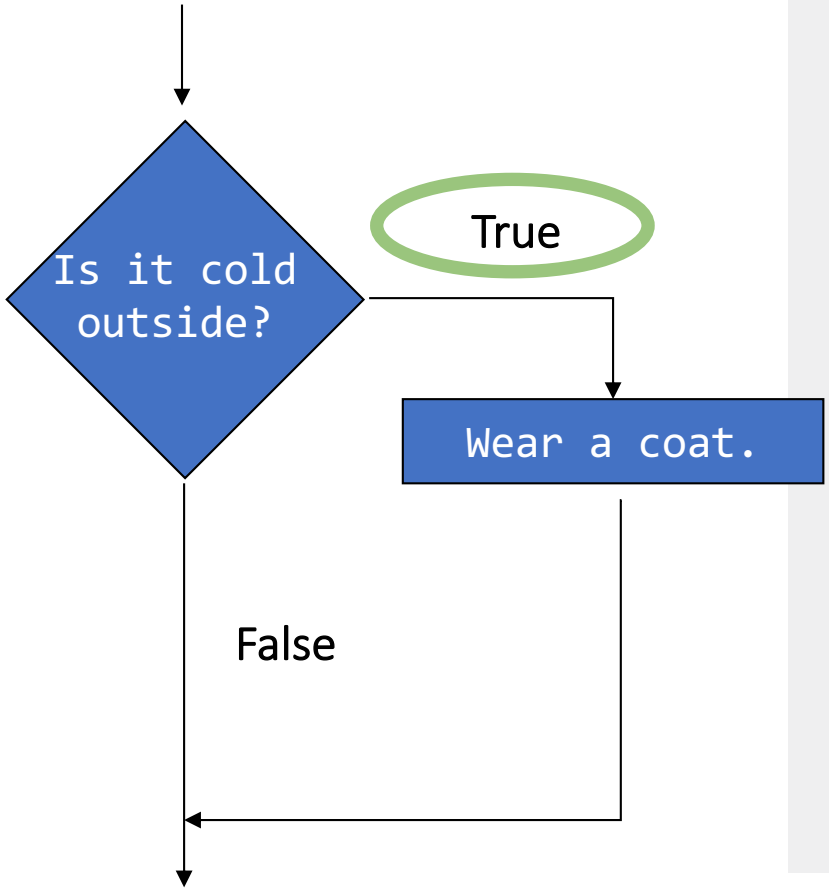


Whatever sits between the if and :

Must equate to a True or False

if statement

```
cold_outside = True

if cold_outside:
    print("wear coat")

# output
# wear coat
```

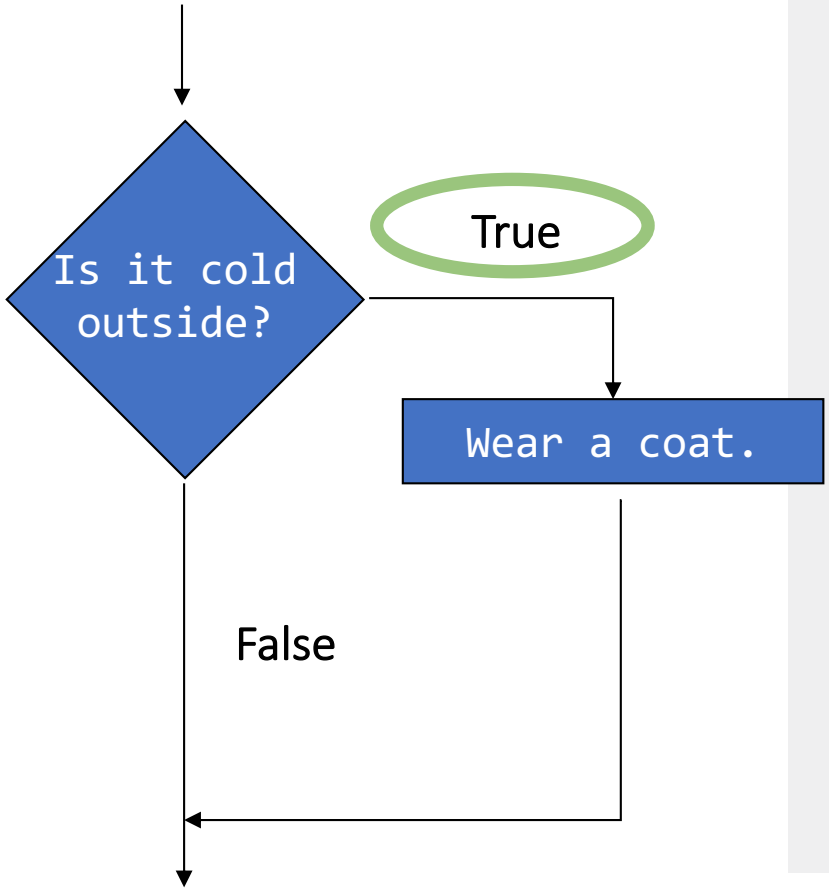


Whatever sits between the if and :

Must equate to a True or False

if statement

```
cold_outside = True

print(cold_outside)

if cold_outside:
    print("wear coat")

# output
# True
# wear coat
```

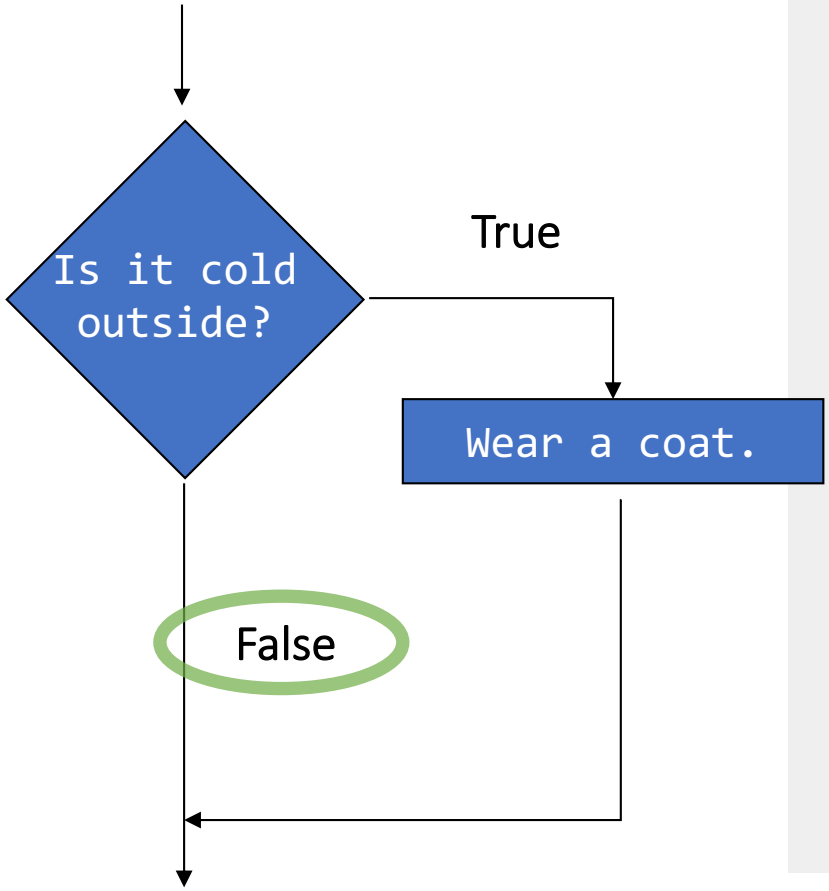


Whatever sits between the if and :

Must equate to a True or False

if statement

```
cold_outside = False

print(cold_outside)

if cold_outside:
    print("wear coat")

# output
# False
```



Whatever sits between the if and :

Must equate to a True or False

if statement

Just to clarify, if you forget to include the colon at the end of your if statement, you will see warnings and errors in PyCharm/Atom

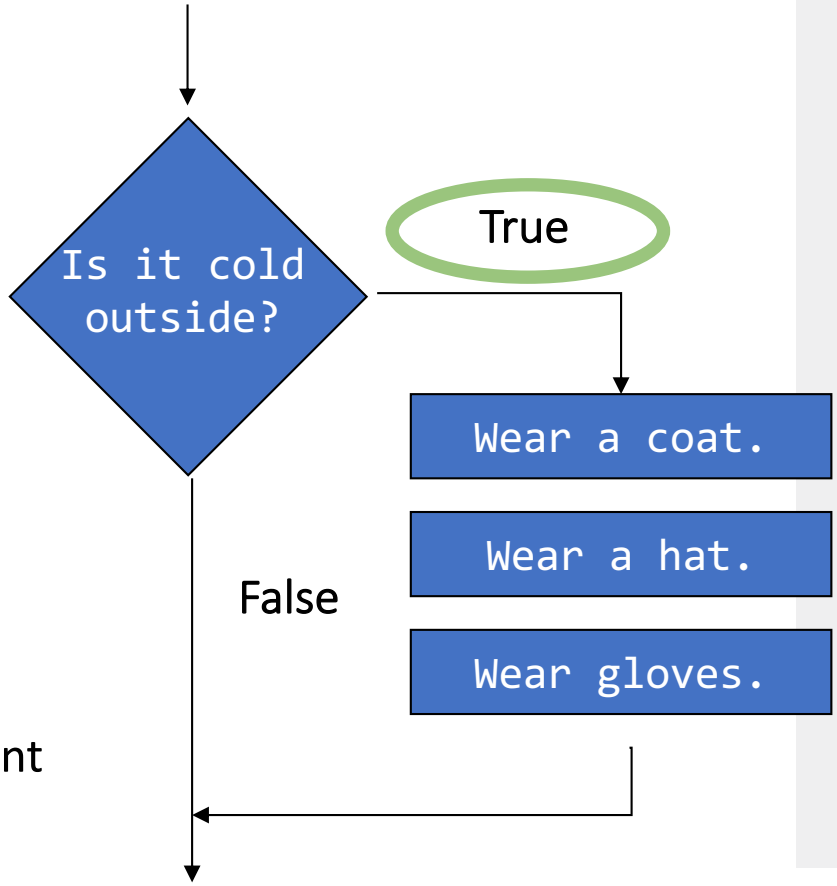
if statement

Just to clarify, if you forget to include the colon at the end of your if statement, you will see warnings and errors in PyCharm/Atom

Similar to Functions, the colon is there to tell Python that the next block of indented code belongs to this statement

if statement

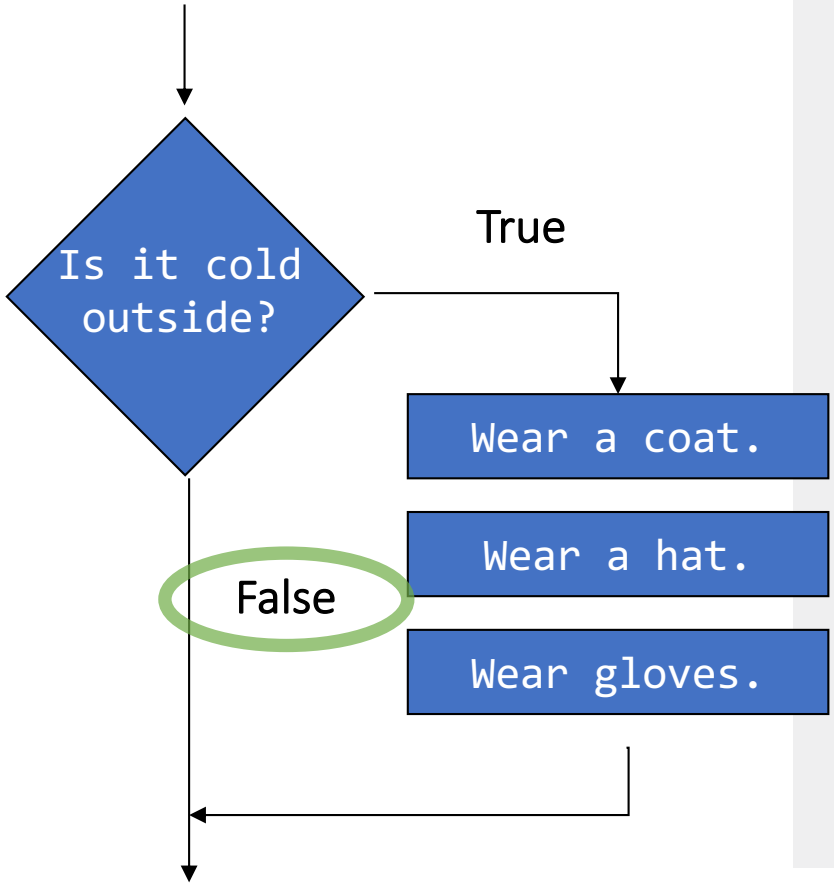
```
cold_outside = True

if cold_outside:
    print("wear coat")
    print("wear hat")
    print("wear gloves")

# output
# wear coat
# wear hat
# wear gloves
```



Like Functions, we can create statement blocks within the indented code…

if statement

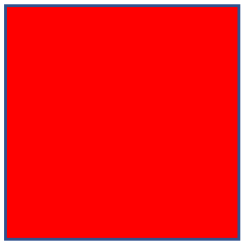
```
cold_outside = False

if cold_outside:
    print("wear coat")
    print("wear hat")
    print("wear gloves")

# output
```



In this example we set our conditional value to either True or False

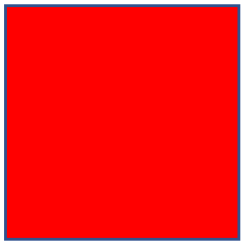
But typically these conditions using relational operators

if statement



Relational Operator	Meaning
>	is greater than
<	is less than
>=	is greater than or equal to
<=	is less than or equal to
==	is equal to
!=	is not equal to

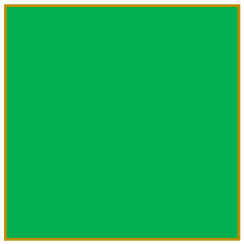
if statement



A condition is also called a *boolean expression* and is any variable or calculation that results in a `True` or `False` condition.

Expression	Meaning
`x > y`	Is x greater than y?
`x < y`	Is x less than y?
`x >= y`	Is x greater than or equal to y?
`x <= y`	Is x less than or equal to y.
`x == y`	Is x equal to y?
`x != y`	Is x not equal to y?

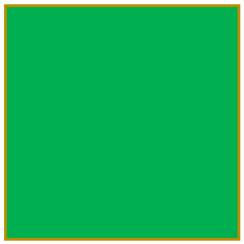
if statement



Let’s define some variables and some if statements

```
num_demogorgan = 0
num_demodog = 0
mind_flayer = 0

if num_demogorgan == 1:
    print("It's Stranger Things season 1, Eleven will save us")

if num_demodog == 10:
    print("It's Stranger Things season 2, Eleven will save us")

if mind_flayer == 1 and num_demogorgan == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")
```

if statement

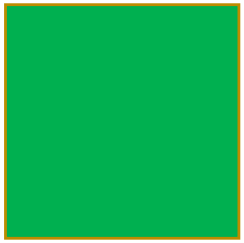


Let's define some variables and some if statements

```
num_demogorgan = 0
num_demodog = 0
mind_flayer = 0

if num_demogorgan == 1:
    print("It's Stranger Things season 1, Eleven will save us")

if num_demodog == 10:
    print("It's Stranger Things season 2, Eleven will save us")

if mind_flayer == 1 and num_demogorgan == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")
```

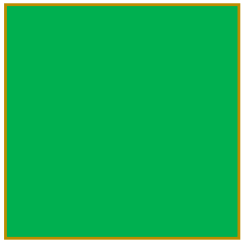
If you haven't watched
Stranger Things
Spoilers....

if statement



```
num_demogorgan = 1
num_demodog = 0
mind_flayer = 0

if num_demogorgan == 1:
    print("It's Stranger Things season 1, Eleven will save us")

if num_demodog == 10:
    print("It's Stranger Things season 2, Eleven will save us")

if mind_flayer == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")

# output
# It's Stranger Things season 1, Eleven will save us
```

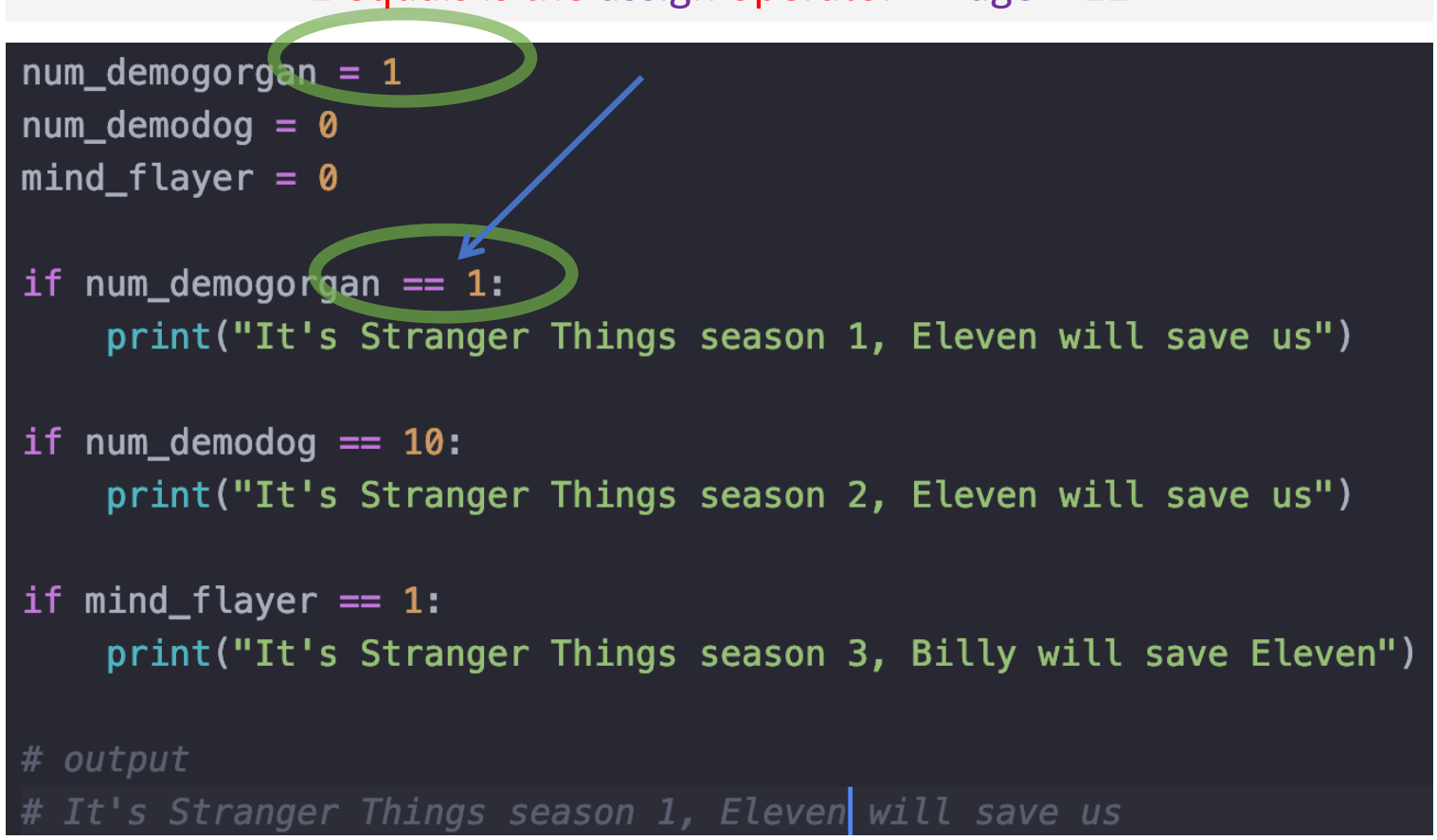
if statement



```
num_demogorgan = 1
num_demodog = 0
mind_flayer = 0

if num_demogorgan == 1:
    print("It's Stranger Things season 1, Eleven will save us")

if num_demodog == 10:
    print("It's Stranger Things season 2, Eleven will save us")

if mind_flayer == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")

# output
# It's Stranger Things season 1, Eleven will save us
```

if statement

Note: ‘is equal’ to is 2 equals signs ‘==’, not 1 equals ‘=‘
1 equals is the assign operator –> age = 12



```
num_demogorgan = 1
num_demodog = 0
mind_flayer = 0

if num_demogorgan == 1:
    print("It's Stranger Things season 1, Eleven will save us")

if num_demodog == 10:
    print("It's Stranger Things season 2, Eleven will save us")

if mind_flayer == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")

# output
# It's Stranger Things season 1, Eleven will save us
```

if statement

Note: 'is equal' to is 2 equals signs '==', not 1 equals '='
1 equals is the assign operator –> age = 12

```
num_demogorgan = 1
num_demodog = 0
mind_flayer = 0

if num_demogorgan == 1:
    print("It's Stranger Things season 1, Eleven will save us")

if num_demodog == 10:
    print("It's Stranger Things season 2, Eleven will save us")

if mind_flayer == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")

# output
# It's Stranger Things season 1, Eleven will save us
```

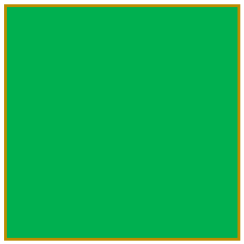
This is the biggest mistake people make when first coding if statements

if statement

Note: ‘is equal’ to is 2 equals signs ‘==’, not 1 equals ‘=‘
1 equals is the assign operator –> age = 12

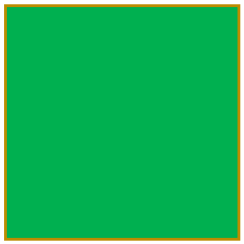
```
num_demogorgan = 1
num_demodog = 0
mind_flayer = 0

if num_demogorgan == 1:
    print("It's Stranger Things season 1, Eleven will save us")

if num_demodog == 10:
    print("It's Stranger Things season 2, Eleven will save us")

if mind_flayer == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")

# output
# It's Stranger Things season 1, Eleven will save us
```

if statement



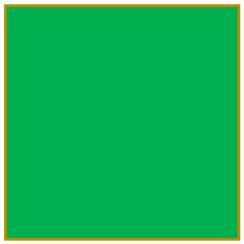
```
num_demogorgan = 0
num_demodog = 10
mind_flayer = 0

if num_demogorgan == 0:
    print("It's Stranger Things season 1, Eleven will save us")

if num_demodog == 10:
    print("It's Stranger Things season 2, Eleven will save us")

if mind_flayer == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")

# output
# It's Stranger Things season 2, Eleven will save us
```

if statement



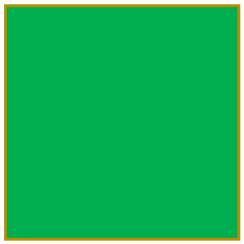
```
num_demogorgan = 0
num_demodog = 10
mind_flayer = 0

if num_demogorgan == 0:
    print("It's Stranger Things season 1, Eleven will save us")

if num_demodog == 10:
    print("It's Stranger Things season 2, Eleven will save us")

if mind_flayer == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")

# output
# It's Stranger Things season 2, Eleven will save us
```

if statement



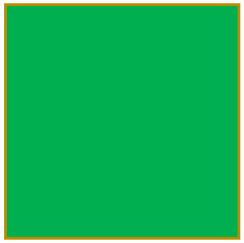
```
num_demogorgan = 0
num_demodog = 10
mind_flayer = 0

if num_demogorgan == 0:
    print("It's Stranger Things season 1, Eleven will save us")

if num_demodog == 10:
    print("It's Stranger Things season 2, Eleven will save us")

if mind_flayer == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")

# output
# It's Stranger Things season 2, Eleven will save us
```

if statement



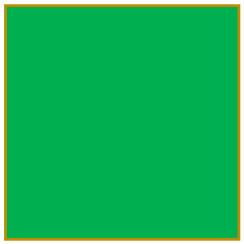
```
num_demogorgan = 0
num_demodog = 0
mind_flayer = 1

if num_demogorgan == 0:
    print("It's Stranger Things season 1, Eleven will save us")

if num_demodog == 10:
    print("It's Stranger Things season 2, Eleven will save us")

if mind_flayer == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")

# output
# It's Stranger Things season 3, Billy will save Eleven
```

if statement



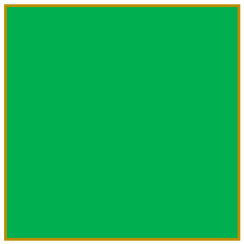
```
num_demogorgan = 0
num_demodog = 0
mind_flayer = 1

if num_demogorgan == 0:
    print("It's Stranger Things season 1, Eleven will save us")

if num_demodog == 10:
    print("It's Stranger Things season 2, Eleven will save us")

if mind_flayer == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")

# output
# It's Stranger Things season 3, Billy will save Eleven
```

if statement



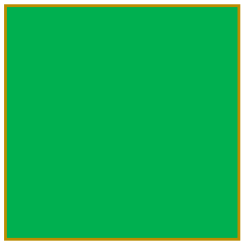
```
num_demogorgan = 0
num_demodog = 0
mind_flayer = 1

if num_demogorgan == 0:
    print("It's Stranger Things season 1, Eleven will save us")

if num_demodog == 10:
    print("It's Stranger Things season 2, Eleven will save us")

if mind_flayer == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")

# output
# It's Stranger Things season 3, Billy will save Eleven
```

if statement



```
num_demogorgan = 0
num_demodog = 10
mind_flayer = 0

if num_demogorgan == 1:
    print("It's Stranger Things season 1, Eleven will save us")

if num_demodog > 5:
    print("It's Stranger Things season 2, Eleven will save us")

if mind_flayer == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")

# output
# It's Stranger Things season 2, Eleven will save us
```

if statement

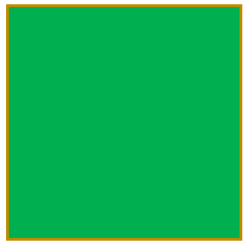


```python
num_demogorgan = 0
num_demodog = 10
mind_flayer = 0

if num_demogorgan == 1:
    print("It's Stranger Things season 1, Eleven will save us")

if num_demodog > 5:
    print("It's Stranger Things season 2, Eleven will save us")

if mind_flayer == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")

# output
# It's Stranger Things season 2, Eleven will save us
```

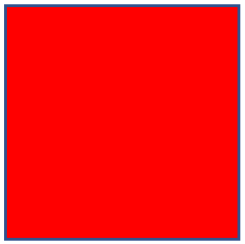
if statement

Note: ‘is greater than’ is ‘>’
So, if the value of num_demodog is greater than 5

```
num_demogorgan = 0
num_demodog = 10
mind_flayer = 0

if num_demogorgan == 1:
    print("It's Stranger Things season 1, Eleven will save us")

if num_demodog > 5:
    print("It's Stranger Things season 2, Eleven will save us")

if mind_flayer == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")

# output
# It's Stranger Things season 2, Eleven will save us
```

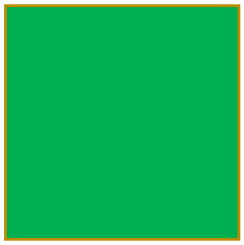


if statement



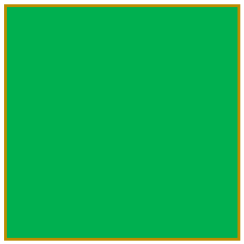
Relational Operator	Meaning
>	is greater than
<	is less than
>=	is greater than or equal to
<=	is less than or equal to
==	is equal to
!=	is not equal to

if statement



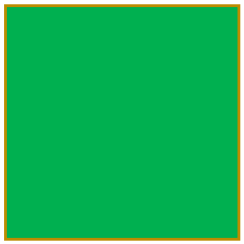
```
num_demogorgan = 0
num_demodog = 10
mind_flayer = 0

if num_demogorgan == 1:
    print("It's Stranger Things season 1, Eleven will save us")

if num_demodog > 5:
    print("It's Stranger Things season 2, Eleven will save us")

if mind_flayer == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")

# output
# It's Stranger Things season 2, Eleven will save us
```

if statement



```
num_demogorgan = 0
num_demodog = 5
mind_flayer = 0

if num_demogorgan == 0:
    print("It's Stranger Things season 1, Eleven will save us")

if num_demodog >= 5:
    print("It's Stranger Things season 2, Eleven will save us")

if mind_flayer == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")

# output
# It's Stranger Things season 2, Eleven will save us
```

if statement

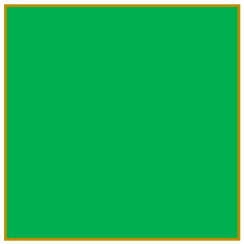


```
num_demogorgan = 0
num_demodog = 5
mind_flayer = 0

if num_demogorgan == 0:
    print("It's Stranger Things season 1, Eleven will save us")

if num_demodog >= 5:
    print("It's Stranger Things season 2, Eleven will save us")

if mind_flayer == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")

# output
# It's Stranger Things season 2, Eleven will save us
```

if statement

Note: ‘is greater than or equal to’ is ‘>=’
So, is the value of num_demodog greater than or equal to 5

```
num_demogorgan = 0
num_demodog = 5
mind_flayer = 0

if num_demogorgan == 0:
    print("It's Stranger Things season 1, Eleven will save us")

if num_demodog >= 5:
    print("It's Stranger Things season 2, Eleven will save us")

if mind_flayer == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")

# output
# It's Stranger Things season 2, Eleven will save us
```



if statement

Note: ‘is greater than or equal to’ is ‘>=’
So, is the value of the num_demodog greater than or equal to 5

```
num_demogorgan = 0
num_demodog = 5
mind_flayer = 0

if num_demogorgan == 0:
    print("It's Stranger Things season 1, Eleven will save us")

if num_demodog >= 5:
    print("It's Stranger Things season 2, Eleven will save us")

if mind_flayer == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")

# output
# It's Stranger Things season 2, Eleven will save us
```

Now we can have greater than (>) and less than (<) and also check for equal to (>=) (<=)

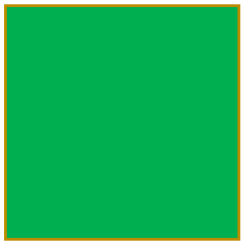
if statement

Note: 'is greater than or equal to' is '>='
So, is the value of the num_demodog greater than or equal to 5

```
num_demogorgan = 0
num_demodog = 5
mind_flayer = 0

if num_demogorgan == 0:
    print("It's Stranger Things season 1, Eleven will save us")

if num_demodog >= 5:
    print("It's Stranger Things season 2, Eleven will save us")

if mind_flayer == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")

# output
# It's Stranger Things season 2, Eleven will save us
```

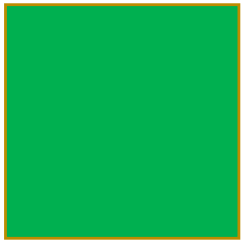
Now we can have greater than (>) and less than (<) and also check for equal to (>=) (<=)

if statement



```
num_demogorgan = 1
num_demodog = 0
mind_flayer = 0

if num_demogorgan != 0:
    print("It's Stranger Things season 1, Eleven will save us")

if num_demodog >= 5:
    print("It's Stranger Things season 2, Eleven will save us")

if mind_flayer == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")

# output
# It's Stranger Things season 1, Eleven will save us
```

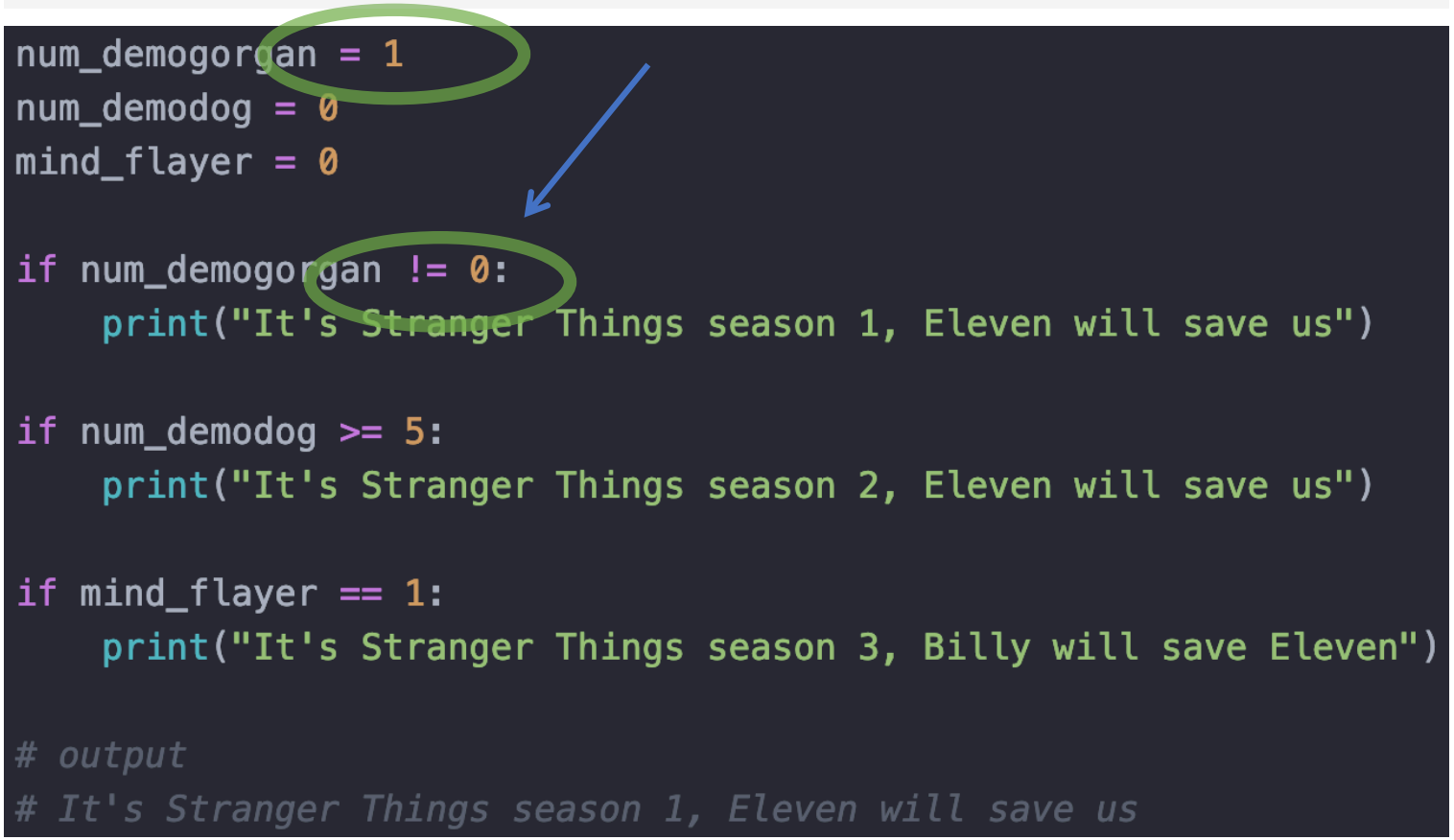
if statement



```
num_demogorgan = 1
num_demodog = 0
mind_flayer = 0

if num_demogorgan != 0:
    print("It's Stranger Things season 1, Eleven will save us")

if num_demodog >= 5:
    print("It's Stranger Things season 2, Eleven will save us")

if mind_flayer == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")

# output
# It's Stranger Things season 1, Eleven will save us
```

if statement

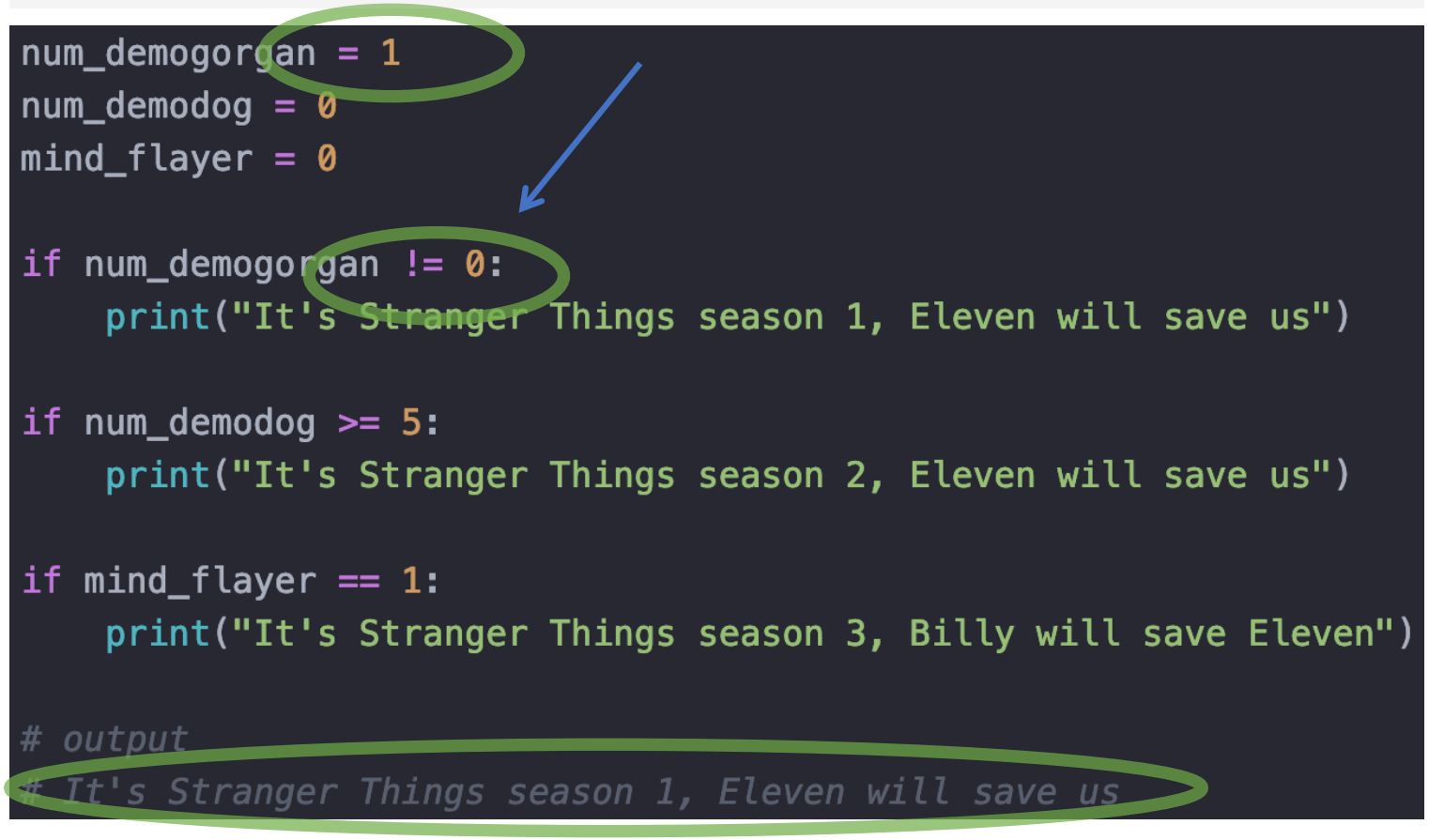
Note: we can also use '!=' to check if something is 'not equal to'
So, '!' is the same as 'not'



```
num_demogorgan = 1
num_demodog = 0
mind_flayer = 0

if num_demogorgan != 0:
    print("It's Stranger Things season 1, Eleven will save us")

if num_demodog >= 5:
    print("It's Stranger Things season 2, Eleven will save us")

if mind_flayer == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")

# output
# It's Stranger Things season 1, Eleven will save us
```

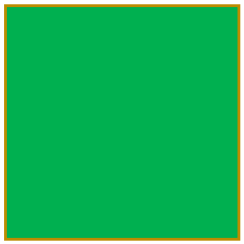
if statement

Note: we can also use '!=' to check if something is 'not equal to'
So, '!' is the same as 'not'



```
num_demogorgan = 1
num_demodog = 0
mind_flayer = 0

if num_demogorgan != 0:
    print("It's Stranger Things season 1, Eleven will save us")

if num_demodog >= 5:
    print("It's Stranger Things season 2, Eleven will save us")

if mind_flayer == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")

# output
# It's Stranger Things season 1, Eleven will save us
```

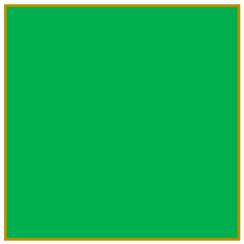
if statement



Let’s look at ‘!=‘ a little bit more

```
if num_demogorgan != 1:
    print("It's Stranger Things season 1, Eleven will save us")
```

if statement



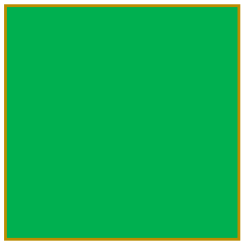
Let's look at '!=' a little bit more

```
if num_demogorgan != 1:
    print("It's Stranger Things season 1, Eleven will save us")
```

We can group this conditional statement in brackets

```
if (num_demogorgan != 1):
    print("It's Stranger Things season 1, Eleven will save us")
```

if statement



Let's look at '!=' a little bit more

```
if num_demogorgan != 1:
    print("It's Stranger Things season 1, Eleven will save us")
```

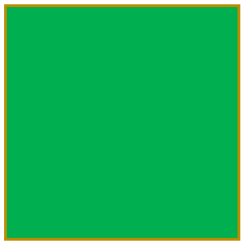
We can group this conditional statement in brackets

```
if (num_demogorgan != 1):
    print("It's Stranger Things season 1, Eleven will save us")
```

Without any change to the output

```
# output
# It's Stranger Things season 1, Eleven will save us
```

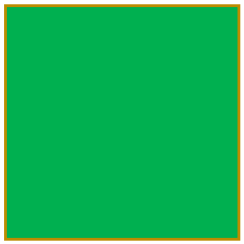
if statement



Python likes to use readable code where possible, so we can

```
if (num_demogorgan != 1):
    print("It's Stranger Things season 1, Eleven will save us")
```

if statement



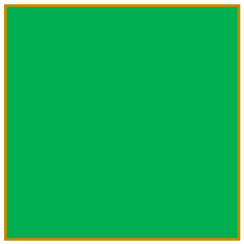
Python likes to use readable code where possible, so we can

```
if (num_demogorgan != 1):
    print("It's Stranger Things season 1, Eleven will save us")
```

Change the ‘not equal to (!=)’ to ‘equal to (==)’

```
if (num_demogorgan == 1):
    print("It's Stranger Things season 1, Eleven will save us")
```

if statement



Python likes to use readable code where possible, so we can

```
if (num_demogorgan != 1):
    print("It's Stranger Things season 1, Eleven will save us")
```

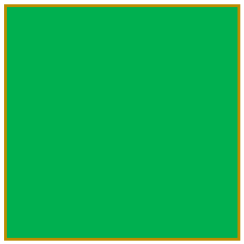
Change the ‘not equal to (!=)’ to ‘equal to (==)’

```
if (num_demogorgan == 1):
    print("It's Stranger Things season 1, Eleven will save us")
```

And place not outside the brackets

```
if not (num_demogorgan == 1):
    print("It's Stranger Things season 1, Eleven will save us")
```

if statement



Python likes to use readable code where possible, so we can

```
if (num_demogorgan != 1):
    print("It's Stranger Things season 1, Eleven will save us")
```

Change the ‘not equal to (!=)’ to ‘equal to (==)’

```
if (num_demogorgan == 1):
    print("It's Stranger Things season 1, Eleven will save us")
```

And place not outside the brackets

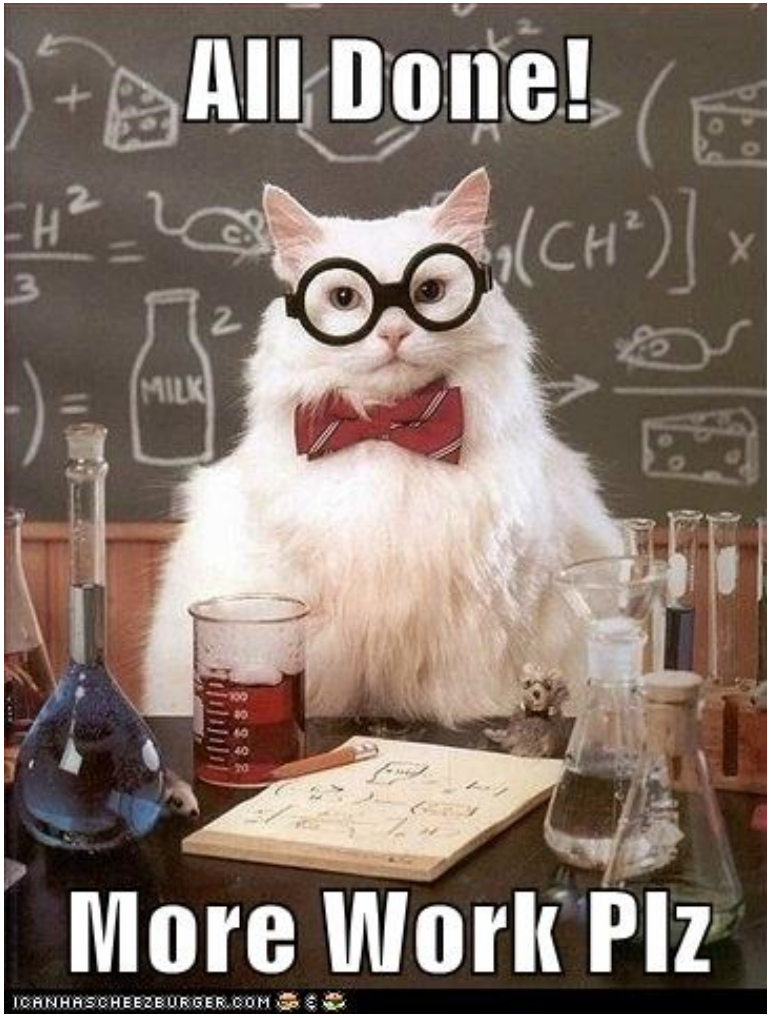
```
if not (num_demogorgan == 1):
    print("It's Stranger Things season 1, Eleven will save us")
```

And if you even remove the brackets, the output stays the same

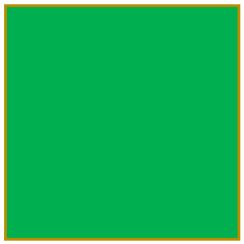
```
if not num_demogorgan == 0:
    print("It's Stranger Things season 1, Eleven will save us")
```

Canvas Student App

Let’s Sign into this lecture now



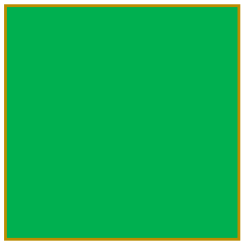
if statement



If you remember Season 3, one demogorgan existed

```
num_demogorgan = 1
num_demodog = 0
mind_flayer = 1

if num_demogorgan == 1:
    print("It's Stranger Things season 1, Eleven will save us")

if num_demodog == 10:
    print("It's Stranger Things season 2, Eleven will save us")

if mind_flayer == 1 and num_demogorgan == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")
```

if statement

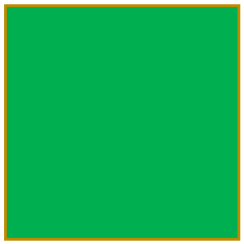


If you remember Season 3, one demogorgan existed

So our code should check for this

```
num_demogorgan = 1
num_demodog = 0
mind_flayer = 1

if num_demogorgan == 1:
    print("It's Stranger Things season 1, Eleven will save us")

if num_demodog == 10:
    print("It's Stranger Things season 2, Eleven will save us")

if mind_flayer == 1 and num_demogorgan == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")
```

if statement

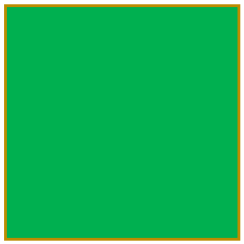


If you remember Season 3, one demogorgan existed

So our code should check for this

```
num_demogorgan = 1
num_demodog = 0
mind_flayer = 1

if num_demogorgan == 1:
    print("It's Stranger Things season 1, Eleven will save us")

if num_demodog == 10:
    print("It's Stranger Things season 2, Eleven will save us")

if mind_flayer == 1 and num_demogorgan == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")
```

if statement



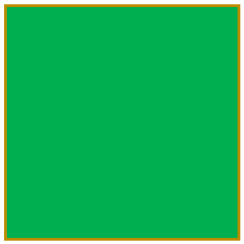
If you remember Season 3, one demogorgan existed

So our code should check for this

```
num_demogorgan = 1
num_demodog = 0
mind_flayer = 1

if num_demogorgan == 1:
    print("It's Stranger Things season 1, Eleven will save us")

if num_demodog == 10:
    print("It's Stranger Things season 2, Eleven will save us")

if mind_flayer == 1 and num_demogorgan == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")
```

For our code to check for more than one condition, we can use and

if statement

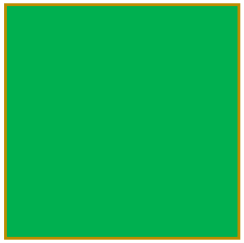


If you remember Season 3, one demogorgan existed

So let's check our output

```
num_demogorgan = 1
num_demodog = 0
mind_flayer = 1

if num_demogorgan == 1:
    print("It's Stranger Things season 1, Eleven will save us")

if num_demodog == 10:
    print("It's Stranger Things season 2, Eleven will save us")

if mind_flayer == 1 and num_demogorgan == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")

# output
# It's Stranger Things season 1, Eleven will save us
# It's Stranger Things season 3, Billy will save Eleven
```

if statement



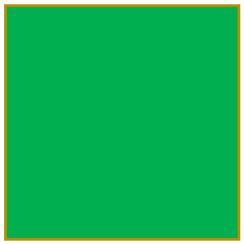
If you remember Season 3, one demogorgan existed

So let's check our output

```
num_demogorgan = 1
num_demodog = 0
mind_flayer = 1

if num_demogorgan == 1:
    print("It's Stranger Things season 1, Eleven will save us")

if num_demodog == 10:
    print("It's Stranger Things season 2, Eleven will save us")

if mind_flayer == 1 and num_demogorgan == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")

# output
# It's Stranger Things season 1, Eleven will save us
# It's Stranger Things season 3, Billy will save Eleven
```

Ooops…

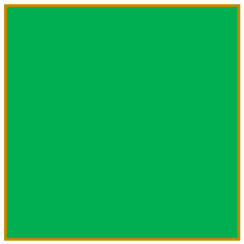
if statement



To fix this we need to add a second condition check on Season 1

```
num_demogorgan = 1
num_demodog = 0
mind_flayer = 1

if num_demogorgan == 1 and mind_flayer != 1:
    print("It's Stranger Things season 1, Eleven will save us")

if num_demodog == 10:
    print("It's Stranger Things season 2, Eleven will save us")

if mind_flayer == 1 and num_demogorgan == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")

# output
# It's Stranger Things season 3, Billy will save Eleven
```

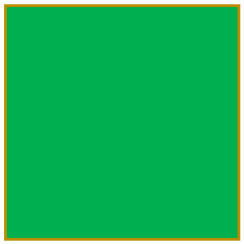
if statement



To fix this we need to add a second condition check on Season 1

```
num_demogorgan = 1
num_demodog = 0
mind_flayer = 1

if num_demogorgan == 1 and mind_flayer != 1:
    print("It's Stranger Things season 1, Eleven will save us")

if num_demodog == 10:
    print("It's Stranger Things season 2, Eleven will save us")

if mind_flayer == 1 and num_demogorgan == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")

# output
# It's Stranger Things season 3, Billy will save Eleven
```

if statement

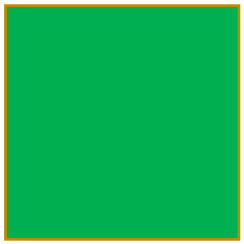


To fix this we need to add a second condition check on Season 1

```
num_demogorgan = 1
num_demodog = 0
mind_flayer = 1

if num_demogorgan == 1 and mind_flayer != 1:
    print("It's Stranger Things season 1, Eleven will save us")

if num_demodog == 10:
    print("It's Stranger Things season 2, Eleven will save us")

if mind_flayer == 1 and num_demogorgan == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")

# output
# It's Stranger Things season 3, Billy will save Eleven
```

Print ‘Season 1’ if there is one demogorgan
and there is not one mind flayer

if statement



- The addition of and and not make the code much easier to read

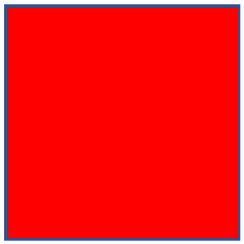
if statement

- The addition of and and not make the code much easier to read
- not will negate the output of the condition
  - So if the condition is True
  - cold_outside is True
  - not cold_outside is equal to False

if statement

- The addition of and and not make the code much easier to read
- not will negate the output of the condition
  - So if the condition is True
  - cold_outside is True
  - not cold_outside is equal to False
- and mandates that all the conditions must be True
  - cold_outside is True
  - raining_outside is True
  - if cold_outside and raining_outside are True, the condition is True

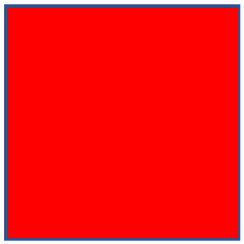
and



For and all expressions (conditions) must be True

Expression 1	Expression 2	Expression1 || Expression2
true	true	true
true	false	false
false	true	false
false	false	false

and



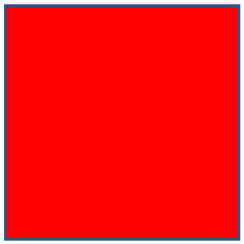
For and all expressions (conditions) must be True

If one or both condition(s) are False

The result is False

Expression 1	Expression 2	Expression1 || Expression2
true	true	true
true	false	false
false	true	false
false	false	false

and



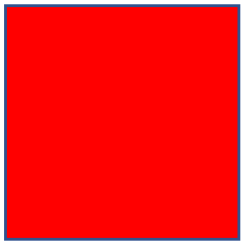
For and all expressions (conditions) must be True

If one or both condition(s) are False

The result is False

Expression 1	Expression 2	Expression1 || Expression2
true	true	true
true	false	false
false	true	false
false	false	false

and



For and all expressions (conditions) must be True

If one or both condition(s) are False

The result is False

Expression 1	Expression 2	Expression1 || Expression2
true	true	true
true	false	false
false	true	false
false	false	false

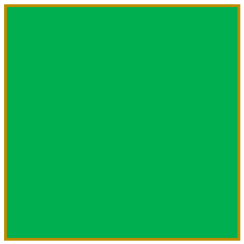
if statement

- The addition of and and not make the code much easier to read
- not will negate the output of the condition
  - So if the condition is True
  - cold_outside is True
  - not cold_outside is equal to False
- and mandates that all the conditions must be True
  - cold_outside is True
  - raining_outside is True
  - if cold_outside and raining_outside are True, the condition is True
- and and not are known as Boolean operators
  - They produce a value that can have at most 2 values

if statement

- The addition of and and not make the code much easier to read
- not will negate the output of the condition
  - So if the condition is True
  - cold_outside is True
  - not cold_outside is equal to False
- and mandates that all the conditions must be True
  - cold_outside is True
  - raining_outside is True
  - if cold_outside and raining_outside are True, the condition is True
- and and not are known as Boolean operators
  - They produce a value that can have at most 2 values
- We have one more Boolean operator
  - or mandates if one condition or the other condition is True
  - The entire condition is True

if statement



Let's look at or

I want my season checker to now print out
if a demogorgan appeared in the season

```
num_demogorgan = 1
num_demodog = 0
mind_flayer = 0

if num_demogorgan == 1:
    print("It's Stranger Things season 1, Eleven will save us")
    print("This season had a Demogorgan")
```

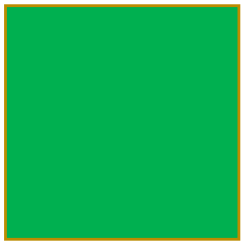
if statement

Let's look at or

I want my season checker to now print out
if a demogorgan appeared in the season

```
num_demogorgan = 1
num_demodog = 0
mind_flayer = 0

if num_demogorgan == 1:
    print("It's Stranger Things season 1, Eleven will save us")
    print("This season had a Demogorgan")
```

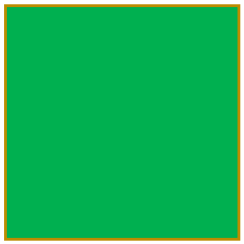
if statement



And now I want to add a check for this in season 3

```
num_demogorgan = 1
num_demodog = 0
mind_flayer = 0

if num_demogorgan == 1:
    print("It's Stranger Things season 1, Eleven will save us")
    print("This season had a Demogorgan")

if num_demodog == 10:
    print("It's Stranger Things season 2, Eleven will save us")
    print("This season had no Demogorgan")

if mind_flayer == 1 or num_demogorgan == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")
    print("This season had a Demogorgan")
```

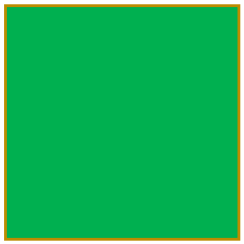
if statement



I want to print Season 3, if I have 1 demogorgan or 1 mind flayer

```
num_demogorgan = 1
num_demodog = 0
mind_flayer = 0

if num_demogorgan == 1:
    print("It's Stranger Things season 1, Eleven will save us")
    print("This season had a Demogorgan")

if num_demodog == 10:
    print("It's Stranger Things season 2, Eleven will save us")
    print("This season had no Demogorgan")

if mind_flayer == 1 or num_demogorgan == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")
    print("This season had a Demogorgan")
```

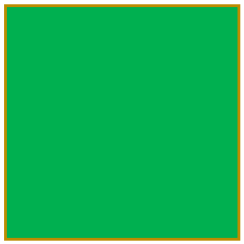
if statement



I want to print Season 3, if I have 1 demogorgan or 1 mind flayer
So, I can use or

```
num_demogorgan = 1
num_demodog = 0
mind_flayer = 0

if num_demogorgan == 1:
    print("It's Stranger Things season 1, Eleven will save us")
    print("This season had a Demogorgan")

if num_demodog == 10:
    print("It's Stranger Things season 2, Eleven will save us")
    print("This season had no Demogorgan")

if mind_flayer == 1 or num_demogorgan == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")
    print("This season had a Demogorgan")
```

if statement

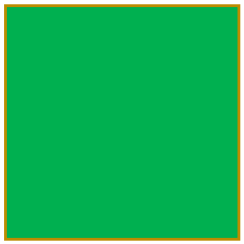


But I want to print Season 3, if I have 1 demogorgan or 1 mind flayer
So, I can use or

```
num_demogorgan = 1
num_demodog = 0
mind_flayer = 0

if num_demogorgan == 1:
    print("It's Stranger Things season 1, Eleven will save us")
    print("This season had a Demogorgan")

if num_demodog == 10:
    print("It's Stranger Things season 2, Eleven will save us")
    print("This season had no Demogorgan")

if mind_flayer == 1 or num_demogorgan == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")
    print("This season had a Demogorgan")
```

So, if mind_flyer is equal to 1 or num_demogorgan is equal to 1
I will print season 3

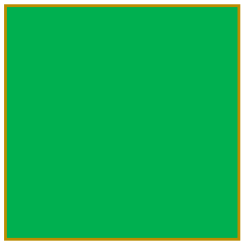
if statement



And season 1 and season 3 print out…

```
num_demogorgan = 1
num_demodog = 0
mind_flayer = 0

if num_demogorgan == 1:
    print("It's Stranger Things season 1, Eleven will save us")
    print("This season had a Demogorgan")

if num_demodog == 10:
    print("It's Stranger Things season 2, Eleven will save us")
    print("This season had no Demogorgan")

if mind_flayer == 1 or num_demogorgan == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")
    print("This season had a Demogorgan")

# output
# It's Stranger Things season 1, Eleven will save us
# This season had a Demogorgan
# It's Stranger Things season 3, Billy will save Eleven
# This season had a Demogorgan
```

if statement

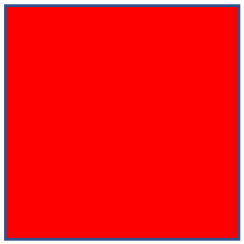


What happens when there is one demogorgan and one mind flayer

```
num_demogorgan = 1
num_demodog = 0
mind_flayer = 1

if num_demogorgan == 1:
    print("It's Stranger Things season 1, Eleven will save us")
    print("This season had a Demogorgan")

if num_demodog == 10:
    print("It's Stranger Things season 2, Eleven will save us")
    print("This season had no Demogorgan")

if mind_flayer == 1 or num_demogorgan == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")
    print("This season had a Demogorgan")

# output
# It's Stranger Things season 1, Eleven will save us
# This season had a Demogorgan
# It's Stranger Things season 3, Billy will save Eleven
# This season had a Demogorgan
```

We get the exact same result…. Why?

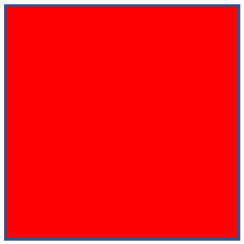
or



For or only one expressions (conditions) needs to be True

Expression 1	Expression 2	Expression1 || Expression2
true	true	true
true	false	true
false	true	true
false	false	false

or

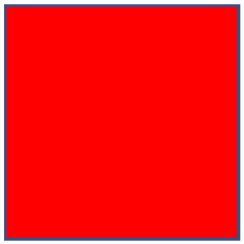


For or only one expressions (conditions) needs to be True

If one or both condition(s) are True -> the result is True

Expression 1	Expression 2	Expression1 || Expression2
true	true	true
true	false	true
false	true	true
false	false	false

or

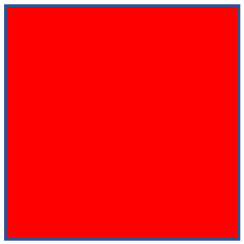


For or only one expressions (conditions) needs to be True

If one or both condition(s) are True -> the result is True

Expression 1	Expression 2	Expression1 || Expression2
true	true	true
true	false	true
false	true	true
false	false	false

or

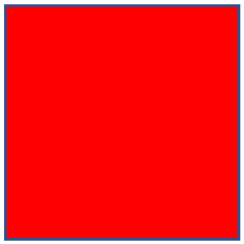


For or only one expressions (conditions) needs to be True

If one or both condition(s) are True -> the result is True

Expression 1	Expression 2	Expression1 || Expression2
true	true	true
true	false	true
false	true	true
false	false	false

or

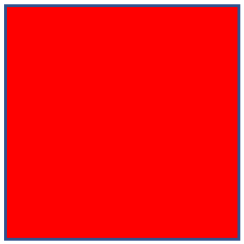


For or only one expressions (conditions) needs to be True

If one or both condition(s) are True -> the result is True

Expression 1	Expression 2	Expression1 || Expression2
true	true	true
true	false	true
false	true	true
false	false	false

or



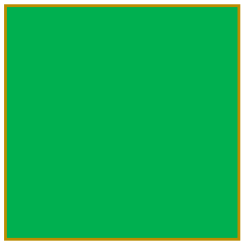
For or only one expressions (conditions) needs to be True

If one or both condition(s) are True -> the result is True

If both condition(s) are False -> the result is False

Expression 1	Expression 2	Expression1 || Expression2
true	true	true
true	false	true
false	true	true
false	false	false

if statement

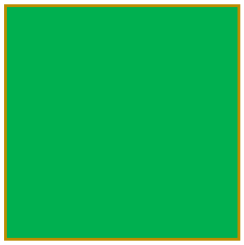


This happens when there is one demogorgan and one mind flayer

```
num_demogorgan = 1
num_demodog = 0
mind_flayer = 1

if num_demogorgan == 1:
    print("It's Stranger Things season 1, Eleven will save us")
    print("This season had a Demogorgan")

if num_demodog == 10:
    print("It's Stranger Things season 2, Eleven will save us")
    print("This season had no Demogorgan")

if mind_flayer == 1 or num_demogorgan == 1:
    print("It's Stranger Things season 3, Billy will save Eleven")
    print("This season had a Demogorgan")

# output
# It's Stranger Things season 1, Eleven will save us
# This season had a Demogorgan
# It's Stranger Things season 3, Billy will save Eleven
# This season had a Demogorgan
```

Both conditions are True, so the if statement is True

if statement



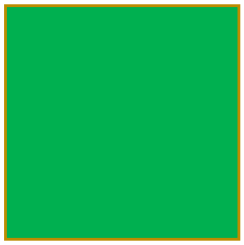
Now let's go back and look at our new line of code

```
if __name__ == "__main__":
```

We can see that this is a conditional check to see

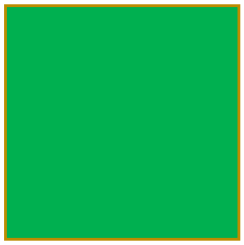
if the variable __name__ has the same value
(is equal to) the string "__main__"

if statement



- One final comment on conditional checks

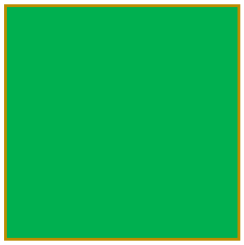
if statement



- One final comment on conditional checks
- We just saw a conditional check on strings

```
if __name__ == "__main__":
```

if statement

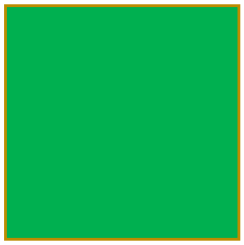


- One final comment on conditional checks
- We just saw a conditional check on strings

```
if __name__ == "__main__":
```

- So, how is this achieved?

if statement

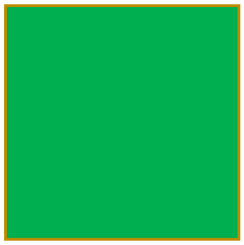


- One final comment on conditional checks
- We just saw a conditional check on strings

```
if __name__ == "__main__":
```

- So, how is this achieved?
- Are we checking the object id() for each ?

if statement



- One final comment on conditional checks
- We just saw a conditional check on strings

```
if __name__ == "__main__":
```

- So, how is this achieved?
- Are we checking the object id() for each ?
- Lets see:

if statement

- One final comment on conditional checks
- We just saw a conditional check on strings

if __name__ == "__main__":

- So, how is this achieved
- Are we checking the object id() for each?
- Lets see:

```
if __name__ == "__main__":
    print("I'm main")
    print(id(__name__))
    print(id("__main__"))
    print(__name__)
    print("__main__")

# output
# I'm main
# 4318795056
# 4318090352
# __main__
# __main__
```

if statement

- One final comment on conditional checks
- We just saw a conditional check on strings

if __name__ == "__main__":

- So, how is this achieved
- Are we checking the object id() for each?
- Lets see:

```
if __name__ == "__main__":
    print("I'm main")
    print(id(__name__))
    print(id("__main__"))
    print(__name__)
    print("__main__")

# output
# I'm main
# 4318795056
# 4318090352
# __main__
# __main__
```

if statement

- One final comment on conditional checks
- We just saw a conditional check on strings

```
if __name__ == "__main__":
```

- So, how is this achieved
- Are we checking the object id() for each?
- Lets see:
- No we don't, as they are not the same…

```
if __name__ == "__main__":
    print("I'm main")
    print(id(__name__))
    print(id("__main__"))
    print(__name__)
    print("__main__")

# output
# I'm main
# 4318795056
# 4318090352
# __main__
# __main__
```

if statement

- One final comment on conditional checks
- We just saw a conditional check on strings

if __name__ == "__main__":

- So, how is this achieved
- Are we checking the object id() for each?
- Lets see:
- No we don't, as they are not the same…

```
if __name__ == "__main__":
    print("I'm main")
    print(id(__name__))
    print(id("__main__"))
    print(__name__)
    print("__main__")

# output
# I'm main
# 4318795056
# 4318090352
# __main__
# __main__
```

if statement

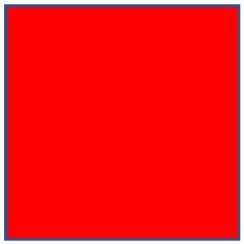
- One final comment on conditional checks
- We just saw a conditional check on strings

```
if __name__ == "__main__":
```

- So, how is this achieved
- Are we checking the object id() for each?
- Lets see:
- No we don't, as they are not the same…
- But the string values are the same…

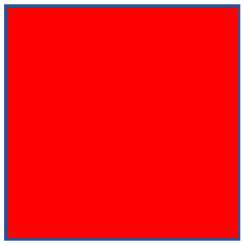
```
if __name__ == "__main__":
    print("I'm main")
    print(id(__name__))
    print(id("__main__"))
    print(__name__)
    print("__main__")

# output
# I'm main
# 4318795056
# 4318090352
# __main__
# __main__
```

String comparison



- When we compare strings using == and !=
- We are actually comparing the string ASCII values

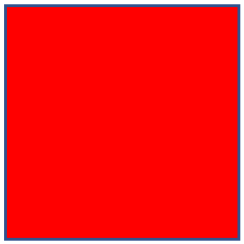
String comparison



- When we compare strings using == and !=
- We are actually comparing the string ASCII values
- **ASCII** stands for American Standard Code for Information Interchange.

- Each character is assigned a unique ASCII value
- ‘A’ (65) has a lower value than ‘Z’ (90)
- ‘A’ (65) and ‘a’ (97) are not the same

String comparison



- When we compare strings using == and !=
- We are actually comparing the string ASCII values
- **ASCII** stands for American Standard Code for Information Interchange.

- Each character is assigned a unique ASCII value
- ‘A’ (65) has a lower value than ‘Z’ (90)
- ‘A’ (65) and ‘a’ (97) are not the same

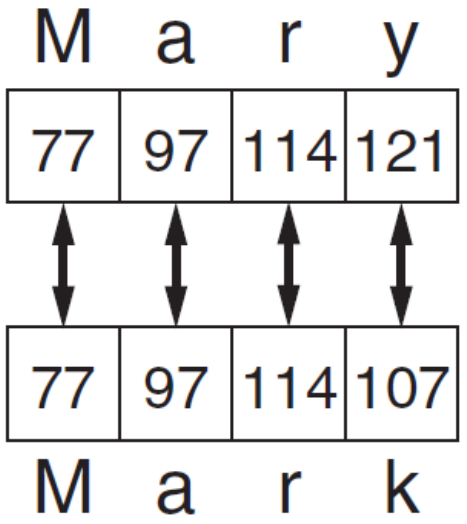
- Let’s look at an example:

String comparison

- When we compare strings using == and !=
- We are actually comparing the string ASCII values
- **ASCII** stands for American Standard Code for Information Interchange.

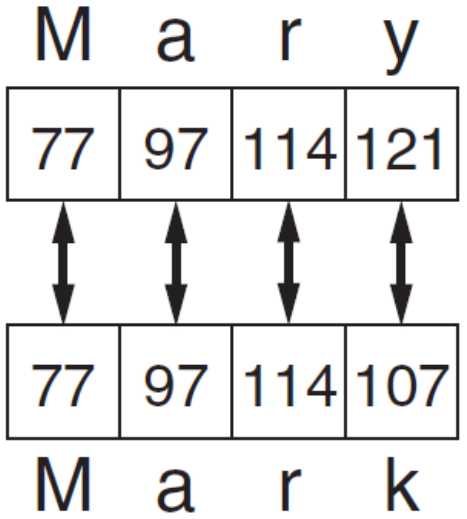
- Each character is assigned a unique ASCII value
- ‘A’ (65) has a lower value than ‘Z’ (90)
- ‘A’ (65) and ‘a’ (97) are not the same

- Let’s look at an example:



String comparison

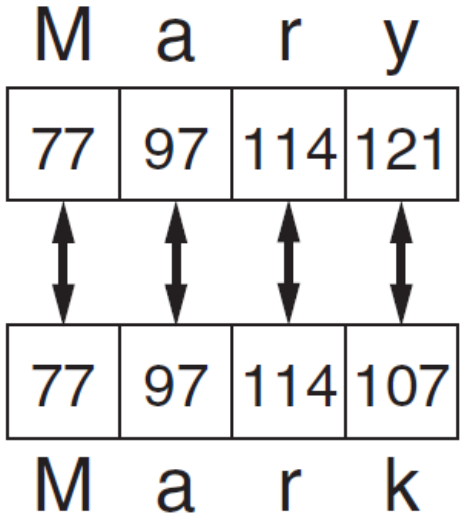
- Mary = 77, 97, 114, and 121
- Mark = 77, 97, 114, and 107



String comparison

- Mary = 77, 97, 114, and 121
- Mark = 77, 97, 114, and 107

```
print("M has the ascii value:", ord('M'))
print("a has the ascii value:", ord('a'))
print("r has the ascii value:", ord('r'))
print("y has the ascii value:", ord('y'))

# output
# M has the ascii value: 77
# a has the ascii value: 97
# r has the ascii value: 114
# y has the ascii value: 121
```

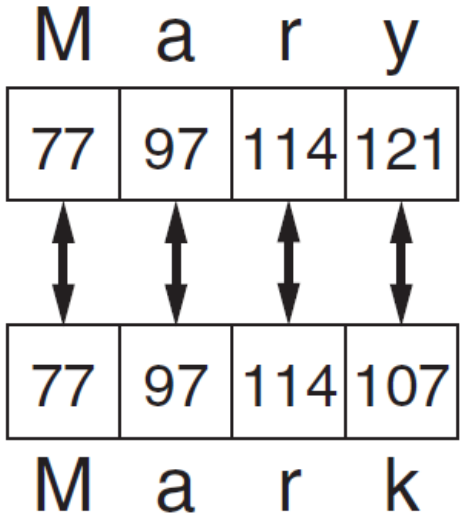


String comparison

- Mary = 77, 97, 114, and 121
- Mark = 77, 97, 114, and 107

```
print("M has the ascii value:", ord('M'))
print("a has the ascii value:", ord('a'))
print("r has the ascii value:", ord('r'))
print("y has the ascii value:", ord('y'))

# output
# M has the ascii value: 77
# a has the ascii value: 97
# r has the ascii value: 114
# y has the ascii value: 121
```

We can use ord() to view the ASCII Unicode value for a single character

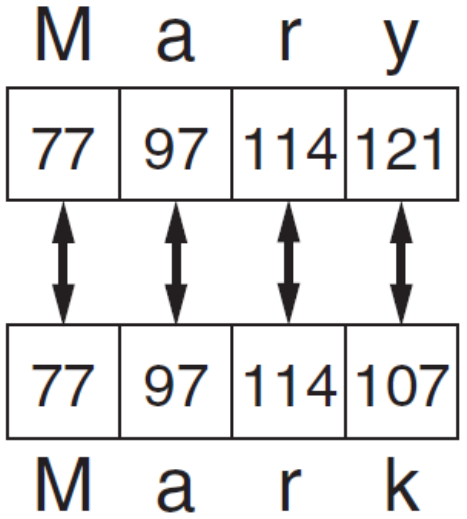


String comparison

- Mary = 77, 97, 114, and 121
- Mark = 77, 97, 114, and 107

- Mark has a lower value than Mary so they are not the same

- As Mark is lower than Mary we can now also use



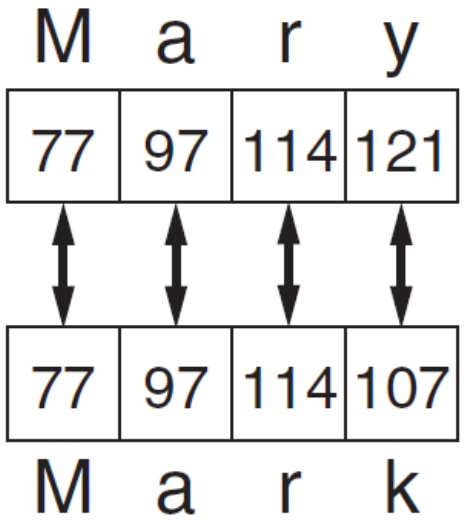
String comparison

- Mary = 77, 97, 114, and 121
- Mark = 77, 97, 114, and 107

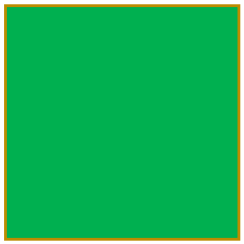
- Mark has a lower value than Mary so they are not the same

- As Mark is lower than Mary we can now also use

- <
- >
- <=
- >=

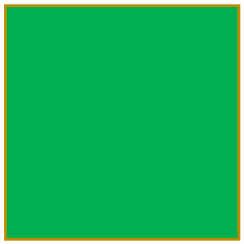


String comparison



- One final comment on string comparison
- If you do want to make sure two values are the same object, i.e., same id()

String comparison



- One final comment on string comparison
- If you do want to make sure two values are the same object, i.e., same id()
- You can use is

- This checks to make sure the underlying objects are the same

String comparison

- One final comment on string comparison
- If you do want to make sure two values are the same object, i.e., same id()
- You can use is

- This checks to make sure the underlying objects are the same

```
name = "__main__"

if name is "__main__":
    print("I'm main")
    print(id(name))
    print(id("__main__"))
    print(name)
    print("__main__")

# output
# I'm main
# 4449166448
# 4449166448
# __main__
# __main__
```

String comparison

- One final comment on string comparison
- If you do want to make sure two values are the same object, i.e., same id()
- You can use is
- This checks to make sure the underlying objects are the same
- Python will give you a warning when you use this ☺

```
name = "__main__"

if name is "__main__":
    print("I'm main")
    print(id(name))
    print(id("__main__"))
    print(name)
    print("__main__")

# output
# I'm main
# 4449166448
# 4449166448
# __main__
# __main__
```

Recap

- We introduced comparison statements
  - if – allows us to check if a condition is True or False

Recap

- We introduced comparison statements
  - if – allows us to check if a condition is True or False
- if is constructed similar to functions
  - if condition:

    indent – statement block of code

Recap

- We introduced comparison statements
  - if – allows us to check if a condition is True or False
- if is constructed similar to functions
  - if condition:

    indent – statement block of code
- We introduced relational operators
  - < <= == != > >=
  - Permits comparison of different values (variables)

Recap

- We introduced comparison statements
  - if – allows us to check if a condition is True or False
- if is constructed similar to functions
  - if condition:

    indent – statement block of code
- We introduced relational operators
  - < <= == != > >=
  - Permits comparison of different values (variables)
- We introduced Boolean operators
  - and – both expressions must be True for condition to be True
  - not – only one expression must be True for condition to be True
  - or – negates the expression/condition
    - From True to False, or False to True

Recap

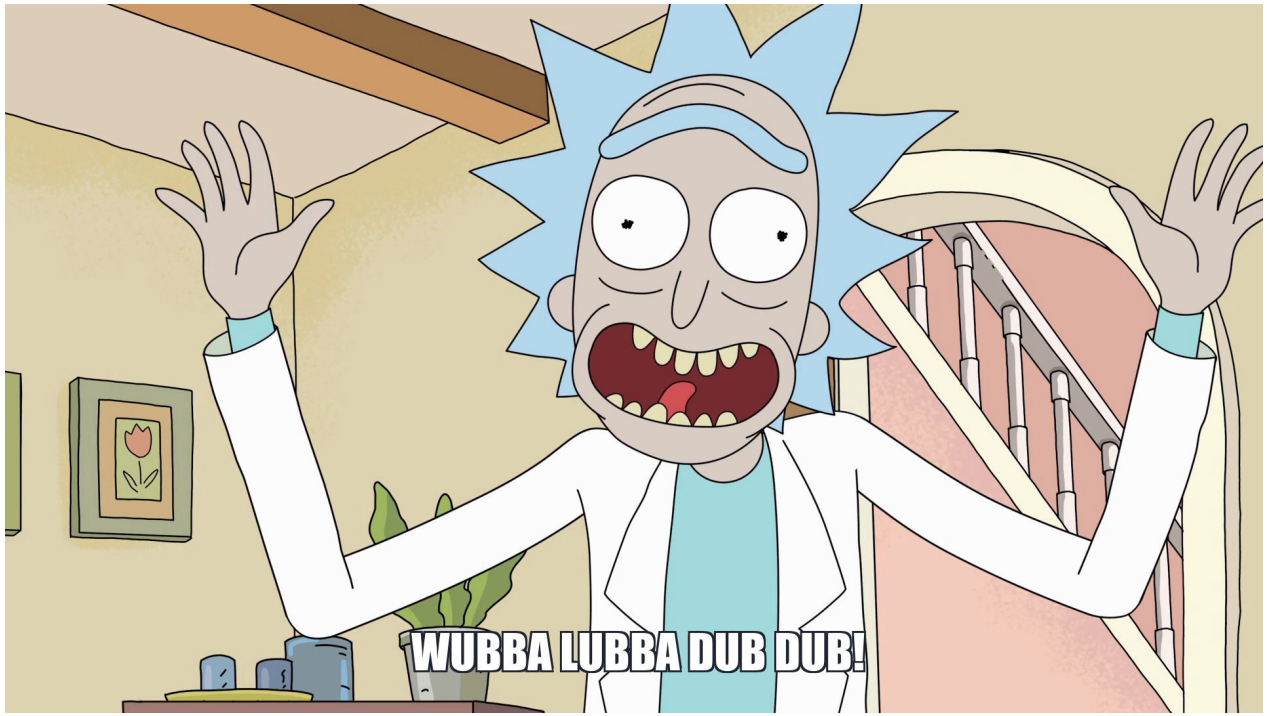
- We introduced comparison statements
  - if – allows us to check if a condition is True or False
- if is constructed similar to functions
  - if condition:

    indent – statement block of code
- We introduced relational operators
  - < <= == != > >=
  - Permits comparison of different values (variables)
- We introduced Boolean operators
  - and – both expressions must be True for condition to be True
  - not – only one expression must be True for condition to be True
  - or – negates the expression/condition
    - From True to False, or False to True
- We saw how if compares Strings
  - Using ASCII characters (not via object values)

Recap

- We introduced comparison statements
  - if – allows us to check if a condition is True or False
- if is constructed similar to functions
  - if condition:

    indent – statement block of code
- We introduced relational operators
  - < <= == != > >=
  - Permits comparison of different values (variables)
- We introduced Boolean operators
  - and – both expressions must be True for condition to be True
  - not – only one expression must be True for condition to be True
  - or – negates the expression/condition
    - From True to False, or False to True
- We saw how if compares Strings
  - Using ASCII characters (not via object values)
- We can use is to compare Strings using object values



UCC

University College Cork, Ireland
Coláiste na hOllscoile Corcaigh