CoGrammar

Control Structures & Operators

SKILLS FOR LIFE

SKILLS BOOTCAMPS

Department for Education

Data Science Lecture Housekeeping

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly. **(FBV: Mutual Respect.)**
- No question is daft or silly - **ask them!**
- There are **Q&A sessions** midway and at the end of the session, should you wish to ask any follow-up questions. Moderators are going to be answering questions as the session progresses as well.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Open Classes. You can submit these questions here: [Open Class Questions]()

Data Science Lecture Housekeeping cont.

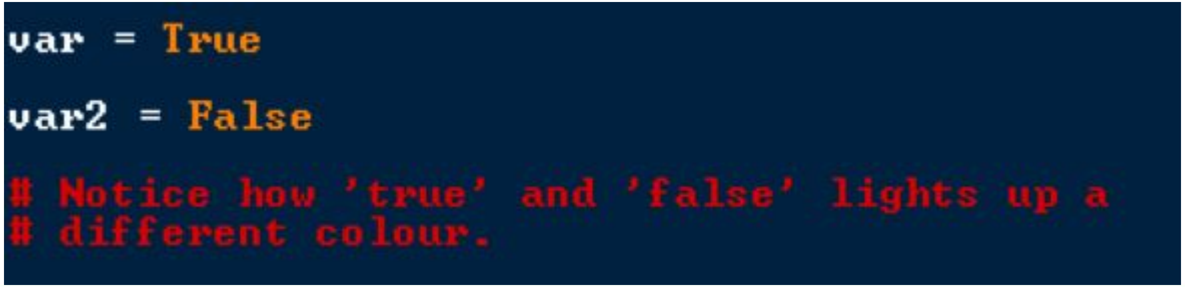
- For all **non-academic questions**, please submit a query: www.hyperiondev.com/support
- Report a **safeguarding** incident: www.hyperiondev.com/safeguardreporting
- We would love your **feedback** on lectures: Feedback on Lectures

Lecture Objectives

- Learning how booleans operate.
- Learn how to use if, elif and else statements to make decisions in your programs

Booleans

- Booleans can only store one of two values : True or False.
- These are mainly used for conditional checks.
- Booleans should be declared with capitals. Using lowercase for booleans will return an error in Python.
- Example :



```
var = True

var2 = False

# Notice how 'true' and 'false' lights up a
# different colour.
```

Truthiness

In Python, all conditional checks will resolve to either True or False.

For Example :

X = 1

X is 1 >> True

X is 0 >> False

Truthiness

We can call values that result in True as “truthy”, or values that result in False as “falsy”

Besides False conditional checks, there are other things that are naturally falsy. These include : empty strings, None and Zero.

Integers & Floats as Booleans

★ Both integers and floating point numbers can be converted to boolean using the bool() function.

★ An int, float, or complex number set to zero will return as False.

★ An int, float, or complex number set to any other value that is not zero, returns True.

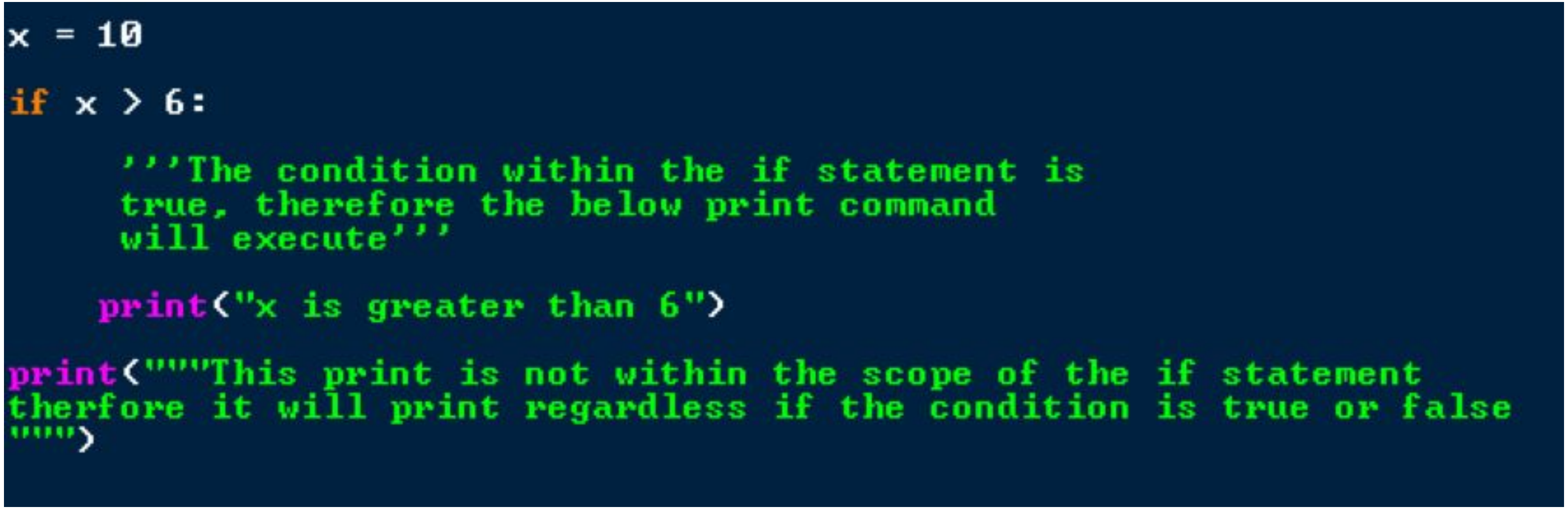
Control Structures

- ★ **Control structures are code that will analyse variables and then choose a direction to follow based on the provided input.**

- ★ **Think of it as a form of branching : depending on the provided input, your program will have one of *x* branches to follow.**
    - ○ **E.g. “If I finish my work early, I will go to bed. Else, I will have to work through the night.”**

If Statement & Syntax

```
if <condition>:
      <statement>
```

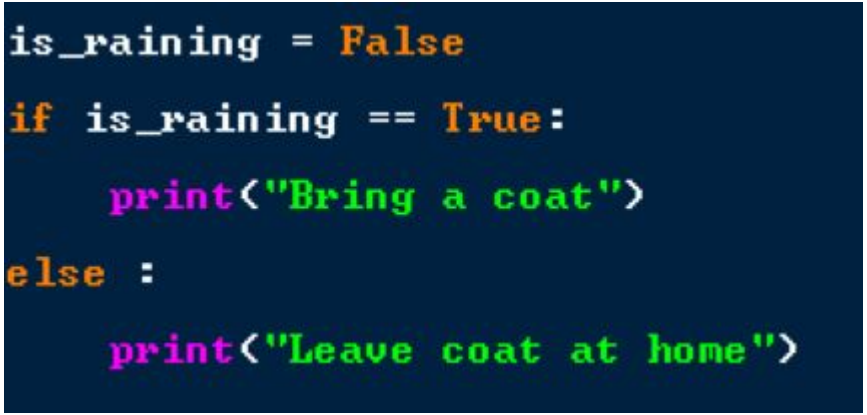


```
x = 10

if x > 6:

    '''The condition within the if statement is
    true, therefore the below print command
    will execute'''

   print("x is greater than 6")

print("""This print is not within the scope of the if statement
therfore it will print regardless if the condition is true or false
""")
```

Else Statements

- We now know that we can use if statements to control the flow of our programs.
- What if we wanted and alternative outcome?
- This is where the else statement comes in.
  - E.g. if it is raining, I shall bring my coat, else I shall leave my coat at home.

Else Example



```
is_raining = False
if is_raining == True:
    print("Bring a coat")
else :
    print("Leave coat at home")
```

Elif Statements

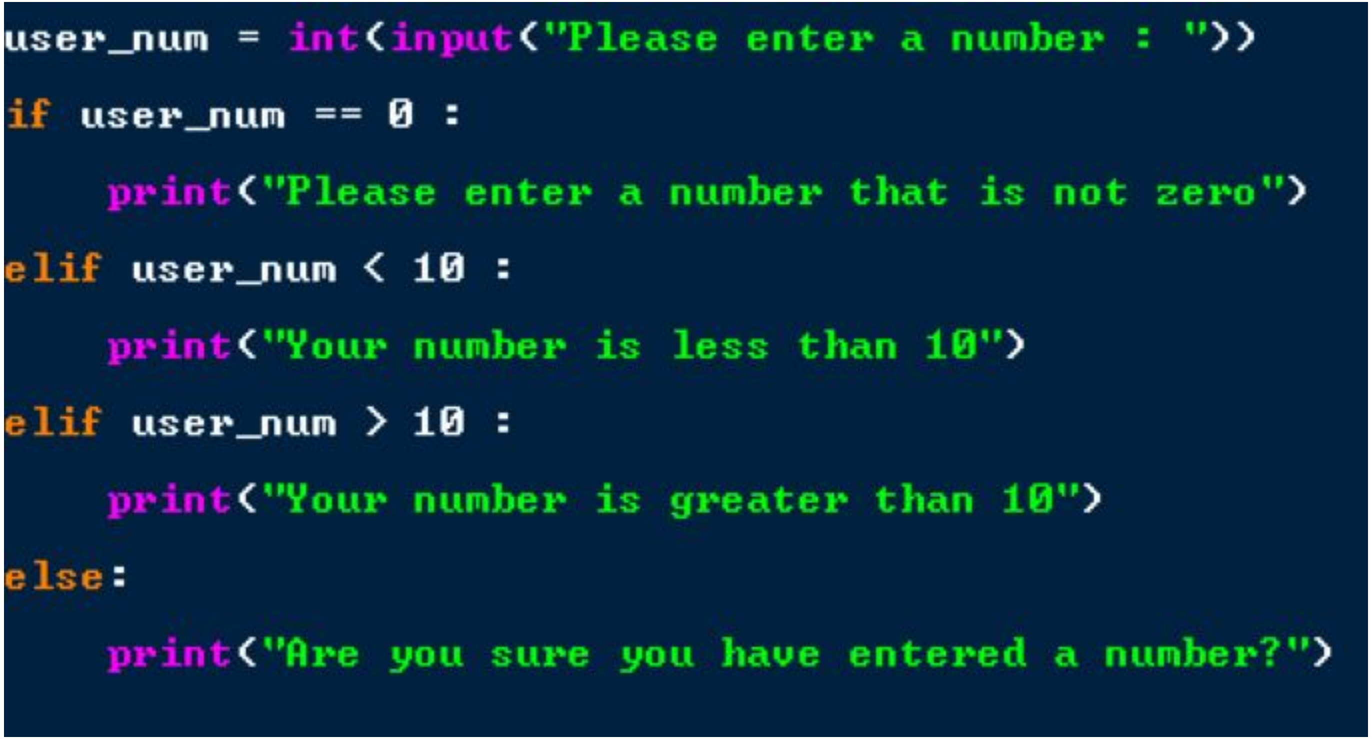
★ What if there was a situation where we could have multiple statements that are True?

★ This is where elif comes into play : Else if → elif

★ Elif statements are mainly used to handle the case when multiple True statements are present.

★ Note that you can have multiple elif statements in an if-else block.

Elif Statement Example



```
user_num = int(input("Please enter a number : "))
if user_num == 0 :
    print("Please enter a number that is not zero")
elif user_num < 10 :
    print("Your number is less than 10")
elif user_num > 10 :
    print("Your number is greater than 10")
else:
    print("Are you sure you have entered a number?")
```

Things to Note

★ There is no limit to the number of elif statements one can have in an if-else block.

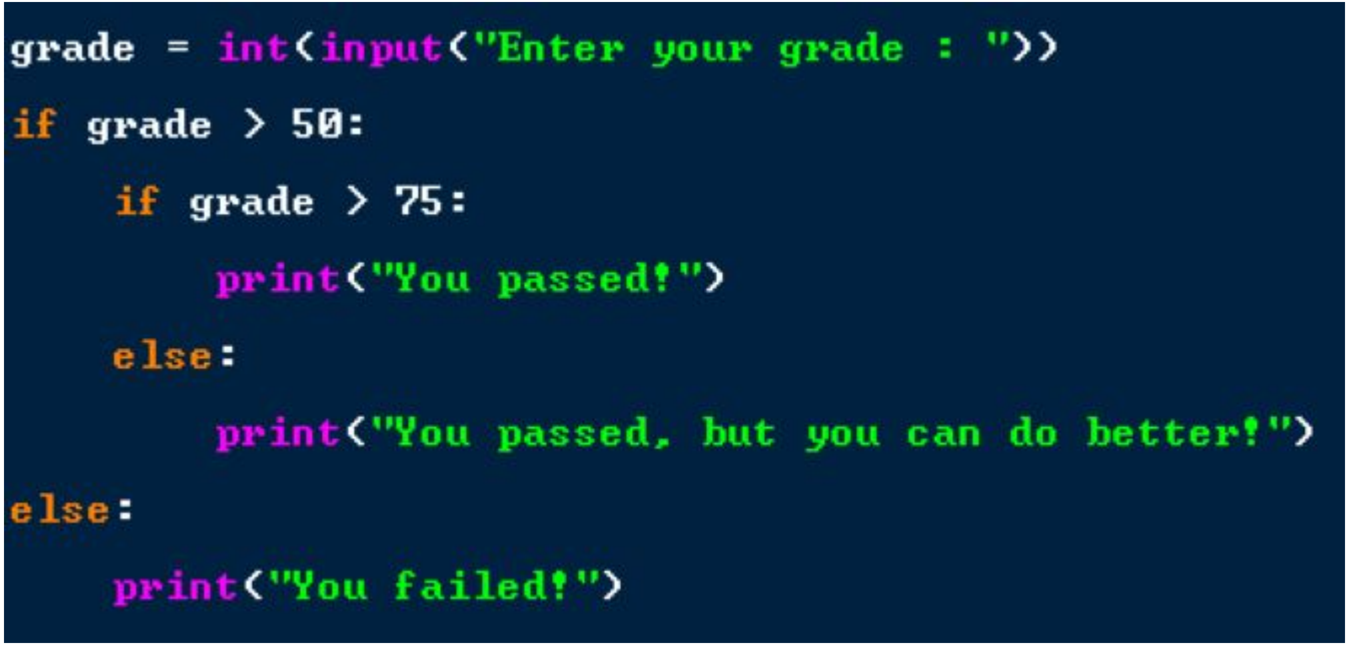
★ Only one final else statement is allowed, per block.

★ Each condition is checked in order.

★ If one condition is true, that branch executes, and the statement ends.

★ Even if there are multiple conditions that are True, only the first True branch will execute.

Nested if Statements



```
grade = int(input("Enter your grade : "))
if grade > 50:
    if grade > 75:
        print("You passed!")
    else:
        print("You passed, but you can do better!")
else:
    print("You failed!")
```

Let’s Breathe

Let’s take a small break before moving on to the next topic.

- ★ So far, we have used a few operators, namely:
  - ○ Assignment (=)
  - ○ Equal to (==)
  - ○ Greater than (>)
  - ○ Less than (<)
- ★ Here we will cover more operators available to us and how to utilise them.

Comparison Operators

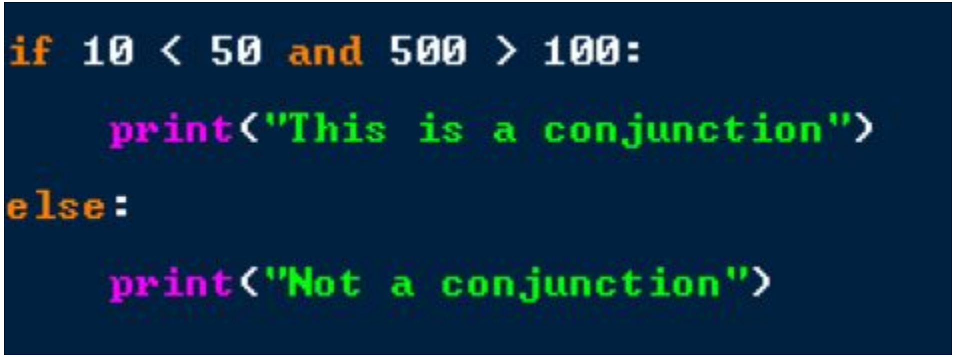
OPERATOR	OPERATION	EXAMPLE
== Equal to	True if **x** has the same value as **y**	x == y # True
!= Not equal to	True if **x** does **NOT** have the same value as **y**	x != y # False
>= greater than or equal to	True if **x** is greater than or equal to **y**	x >= y # True
<= Less than or equal to	True if **x** is less than or equal to **y**	x <= y # True

Logical Operators

OPERATOR	OPERATION	EXAMPLE
and	True if both **x AND y** are true (logical conjunction)	**If x and y:** **print(z)**
or	True if either **x OR y** are true (**logical disjunction**)	**If x or y** **print(z)**
not	True if the **opposite** of **x** is true (**logical negation**)	**If not x** **print(y)**

and Operator

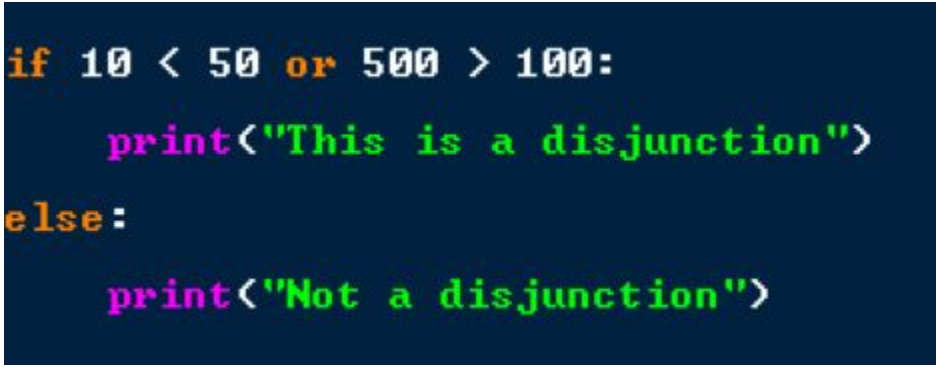
- ★ Returns as True when both conditions specified are met.
- ★ Example:



```
if 10 < 50 and 500 > 100:
    print("This is a conjunction")
else:
    print("Not a conjunction")
```

or Operator

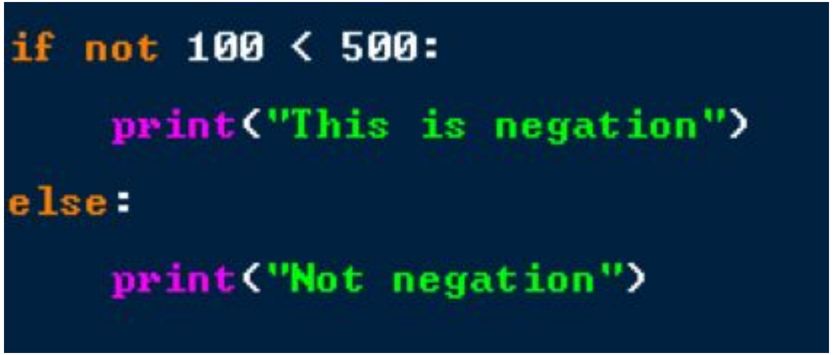
- ★ Returns True if either of the specified conditions are met.
- ★ Example:



```
if 10 < 50 or 500 > 100:
    print("This is a disjunction")
else:
    print("Not a disjunction")
```

not Operator

- ★ Changes the condition from True to False and vice versa.
- ★ Example:



```
if not 100 < 500:
    print("This is negation")
else:
    print("Not negation")
```

CoGrammar

Q & A SECTION

Please use this time to ask any questions relating to the topic, should you have any.

CoGrammar

Thank you for joining!