



Programming for DA

Joseba Carricas
jcarricasga@external.unav.es

- ~~11/1 - Introduction~~
- ~~18/1 - Working with data files~~
- ~~25/1 - Working with data files~~
- 1/2 - Environment setup
- 8/2 - Basics: conditions
- 15/2 - First data analysis & presentation
- 22/2 - First data analysis & presentation

- 1/3 - Partial exam

- 8/3 - Basics: loops and functions
- 15/3 - Manipulating strings
- 22/3 - Manipulating strings
- 29/3 - Data structures
- 5/4 - Data structures
- 26/4 - Data structures

Syntax & Main concepts

Variables	Containers for storing data values
Quotation	Text delimiter: “ or ‘
Text vs Numbers	“10” is a text and 10 is a number
Comments	#This is a comment.
Reserved words	You cannot use them as variable name
Indentation	How to tell Python what’s inside a loop or condition

A good reference:

<https://www.w3schools.com/python/default.asp>

Variables

A variable can have a short name (like x and y) or a more descriptive name (age, carname, total_volume).

Rules for Python variables:

- A variable name must start with a letter or the underscore character.
- A variable name cannot start with a number.
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _).
- Variable names are case-sensitive (age, Age and AGE are three different variables).

If you want to specify the data type of a variable, this can be done with casting:

```
x = str(3)      # x will be '3'
y = int(3)      # y will be 3
z = float(3)    # z will be 3.0
```

Reserved words

and	exec	not
assert	finally	or
break	for	pass
class	from	print
continue	global	raise
def	if	return
del	import	try
elif	in	while
else	is	with
except	lambda	yield

Indentation

Syntax error:

```
if 5 > 2:  
print("Five is greater than two!")
```

Correct syntax:

```
if 5 > 2:  
    print("Five is greater than two!")  
if 5 > 2:  
    print("Five is greater than two!")
```

Conditions

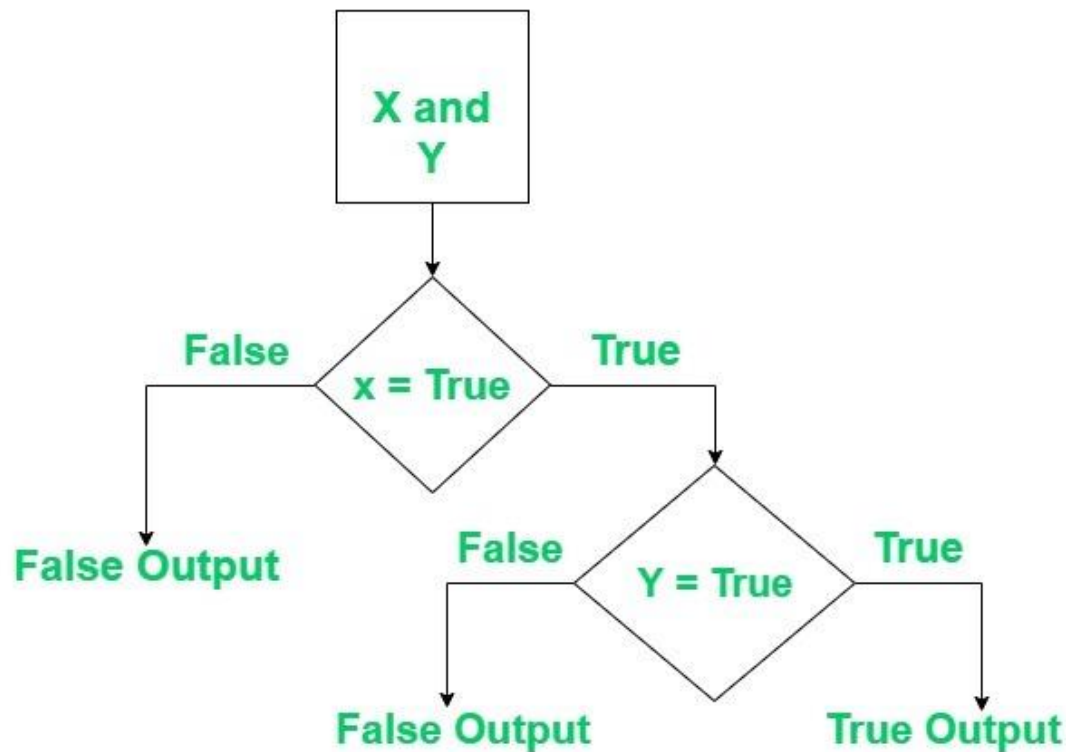
DESCRIPTION	
<	Less than
>	Greater than
!=	Not equals
<=	Less than or equal to
>=	Greater than or equal to
==	Equal to

Logical operators

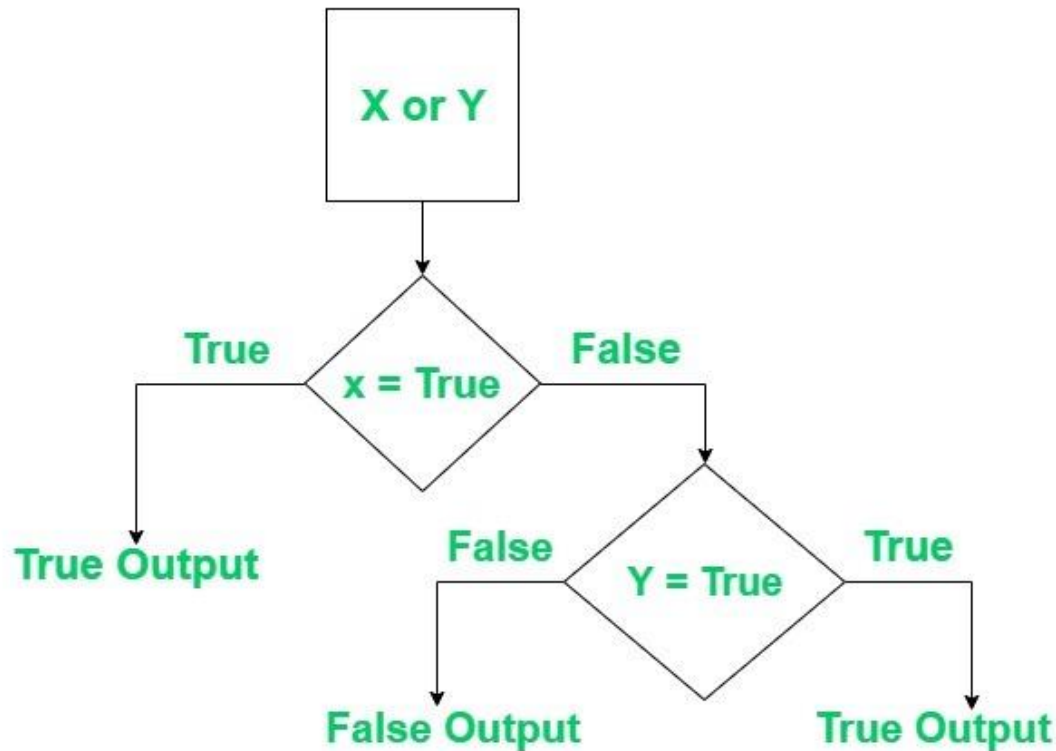
DESCRIPTION	
and	Every condition must be true
or	Any condition can be true
not	Logical NOT

<https://www.geeksforgeeks.org/python-logical-operators-with-examples-improvement-needed/>

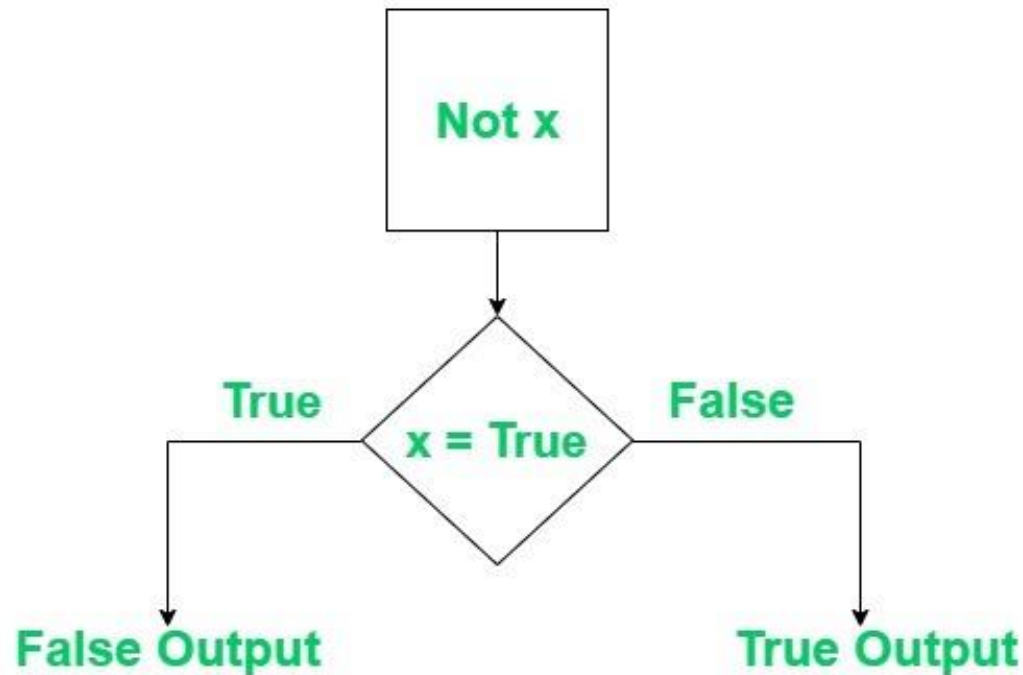
Logical operators: and



Logical operators: or



Logical operators: not



Goal: use data files.

1. Open the “base_code.py” file and run it to get a working base code. Pay attention to:
 - How to open a file.
 - How to go through each line of the file.
 - How to convert a string into a number.
2. Extend the base code to get the total sum of revenue.
3. Extend your code to get the percentage of revenue vs income.

Exercise 1

Goal: start using flow control.

1. Using the dataset “tripadvisor_museum_world.csv”, get the number of museums with a rating above 4.
 1. “Plan” your code/algorithm.
 2. Implement.
 3. Validate the result: 836

* Tip: go through the data file to understand the structure

Exercise 2

Goal: start using flow control.

1. Using the dataset “tripadvisor_museum_world.csv”, get the number of museums with a rating above 4 and more than 1000 reviews.
 1. “Plan” your code/algorithm.
 2. Implement.
 3. Validate the result: 485

Exercise 3

Goal: mix of flow control and variables.

1. Get the number of museums grouped by number of reviews (<1000, 1000 to 10000, >10000).

1. “Plan” your code/algorithm.

2. Implement.

3. Validate the result:

```
Museums by number of reviews:  
>10000          110  
1000-10000      432  
<1000           470
```

Go through some videos

Automate the Boring Stuff with Python:

https://www.youtube.com/watch?v=&list=PL0-84-yl1fUnRuXGFe_F7qSH1LEnn9LkW

Lesson 6: while loops.

Lesson 7: for loops.



Programming for DA

Joseba Carricas
jcarricasga@external.unav.es