Welcome to python

Jamie Saxon

Introduction to Programming for Public Policy

October 2, 2017

What is Python?

Python is a popular, high-level interpreted programming language.

Interpreted: computer 'runs your instructions,' so you can:

- Run **interactively**: execute one line of code at a time.
- Or **script**: write down and save all of your commands.
- (It actually compiles itself, 'just-in-time.')

High level: python hides a lot of the complexity from you.

- You don't have to worry about moving bits (1s and 0s) around.
- You can accomplish a lot, with relatively little code.
- There are tons of libraries: it is simultaneously a basic programming language, an advanced computational engine, an excel buster, a web designer and scraper, a mapping platform, ...

Through the next two lecture, we'll discuss:

Types

Control

Assignment, Operators, and Methods

Some of them will be mixed together.

The Building Blocks

Notes: See Jupyter Notebook.

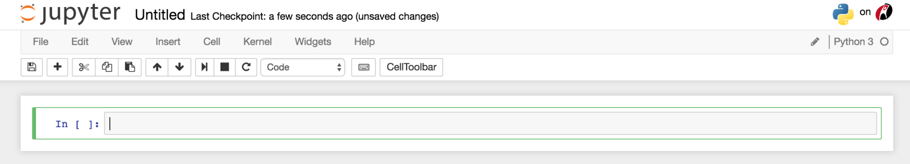
Launch python on your computer, or open a jupyter notebook:

Command line: ‘python’ (must be python 3).

Local Jupyter: Anaconda Navigator

Online Jupyter: tmpnb.org or try.jupter.org

Click ‘New’ in the upper right corner, then Python 3.



```
In [ ]:
```

An Aside – Boolean Arithmetic

- **True**/**False** values – booleans – are foundational to programming.
- We will often do something “if” a condition is satisfied (**True**), and not do it if the condition is **False**.
- It is **critical** to understand how they are combined (and how other variables may be converted into them).

True and True =	=
True and False =	=
False and True =	=
False and False =	=
True or True =	=
True or False =	=
False or True =	=
False or False =	=

An Aside – Boolean Arithmetic

- **True**/**False** values – booleans – are foundational to programming.
- We will often do something “if” a condition is satisfied (**True**), and not do it if the condition is **False**.
- It is **critical** to understand how they are combined (and how other variables may be converted into them).

True and True = True	=
True and False =	=
False and True =	=
False and False =	=
True or True =	=
True or False =	=
False or True =	=
False or False =	=

An Aside – Boolean Arithmetic

- **True**/**False** values – booleans – are foundational to programming.
- We will often do something "`if`" a condition is satisfied (**True**), and not do it if the condition is **False**.
- It is **critical** to understand how they are combined (and how other variables may be converted into them).

True and True = True	=
True and False = False	=
False and True =	=
False and False =	=
True or True =	=
True or False =	=
False or True =	=
False or False =	=

An Aside – Boolean Arithmetic

- True/False values – booleans – are foundational to programming.
- We will often do something "if" a condition is satisfied (True), and not do it if the condition is False.
- It is **critical** to understand how they are combined (and how other variables may be converted into them).

True	and	True	=	True		=	
True	and	False	=	False		=	
False	and	True	=	False		=	
False	and	False	=			=	
True	or	True	=			=	
True	or	False	=			=	
False	or	True	=			=	
False	or	False	=			=	

An Aside – Boolean Arithmetic

- **True**/**False** values – booleans – are foundational to programming.
- We will often do something "if" a condition is satisfied (**True**), and not do it if the condition is **False**.
- It is **critical** to understand how they are combined (and how other variables may be converted into them).

True and True = True	=
True and False = False	=
False and True = False	=
False and False = False	=
True or True =	=
True or False =	=
False or True =	=
False or False =	=

An Aside – Boolean Arithmetic

- **True**/**False** values – booleans – are foundational to programming.
- We will often do something "`if`" a condition is satisfied (**True**), and not do it if the condition is **False**.
- It is **critical** to understand how they are combined (and how other variables may be converted into them).

True	and	True	=	True		=	
True	and	False	=	False		=	
False	and	True	=	False		=	
False	and	False	=	False		=	
True	or	True	=	True		=	
True	or	False	=			=	
False	or	True	=			=	
False	or	False	=			=	

An Aside – Boolean Arithmetic

- **True**/**False** values – booleans – are foundational to programming.
- We will often do something "if" a condition is satisfied (**True**), and not do it if the condition is **False**.
- It is **critical** to understand how they are combined (and how other variables may be converted into them).

True	and	True	=	True		=
True	and	False	=	False		=
False	and	True	=	False		=
False	and	False	=	False		=
True	or	True	=	True		=
True	or	False	=	True		=
False	or	True	=			=
False	or	False	=			=

An Aside – Boolean Arithmetic

- **True**/**False** values – booleans – are foundational to programming.
- We will often do something "if" a condition is satisfied (**True**), and not do it if the condition is **False**.
- It is **critical** to understand how they are combined (and how other variables may be converted into them).

True and True = True	=
True and False = False	=
False and True = False	=
False and False = False	=
True or True = True	=
True or False = True	=
False or True = True	=
False or False =	=

An Aside – Boolean Arithmetic

- True/False values – booleans – are foundational to programming.
- We will often do something "if" a condition is satisfied (True), and not do it if the condition is False.
- It is **critical** to understand how they are combined (and how other variables may be converted into them).

True and True = True	=
True and False = False	=
False and True = False	=
False and False = False	=
True or True = True	=
True or False = True	=
False or True = True	=
False or False = False	=

An Aside – Boolean Arithmetic

- True/False values – booleans – are foundational to programming.
- We will often do something "if" a condition is satisfied (True), and not do it if the condition is False.
- It is **critical** to understand how they are combined (and how other variables may be converted into them).

True and True = True	not True =
True and False = False	not False =
False and True = False	True and not False =
False and False = False	False or (False or True) =
True or True = True	True and (False or True) =
True or False = True	False or False or False or True =
False or True = True	True and True and True and False =
False or False = False	not (False and not True) =

An Aside – Boolean Arithmetic

- True/False values – booleans – are foundational to programming.
- We will often do something "`if`" a condition is satisfied (True), and not do it if the condition is False.
- It is **critical** to understand how they are combined (and how other variables may be converted into them).

True and True	=	True	not True	=	False
True and False	=	False	not False	=	
False and True	=	False	True and not False	=	
False and False	=	False	False or (False or True)	=	
True or True	=	True	True and (False or True)	=	
True or False	=	True	False or False or False or True	=	
False or True	=	True	True and True and True and False	=	
False or False	=	False	not (False and not True)	=	

An Aside – Boolean Arithmetic

- `True`/`False` values – booleans – are foundational to programming.
- We will often do something "`if`" a condition is satisfied (`True`), and not do it if the condition is `False`.
- It is **critical** to understand how they are combined (and how other variables may be converted into them).

`True and True`	= `True`	`not True`	= `False`
`True and False`	= `False`	`not False`	= `True`
`False and True`	= `False`	`True and not False`	=
`False and False`	= `False`	`False or (False or True)`	=
`True or True`	= `True`	`True and (False or True)`	=
`True or False`	= `True`	`False or False or False or True`	=
`False or True`	= `True`	`True and True and True and False`	=
`False or False`	= `False`	`not (False and not True)`	=

An Aside – Boolean Arithmetic

- `True`/`False` values – booleans – are foundational to programming.
- We will often do something "`if`" a condition is satisfied (`True`), and not do it if the condition is `False`.
- It is **critical** to understand how they are combined (and how other variables may be converted into them).

`True and True`	=	`True`	`not True`	=	`False`
`True and False`	=	`False`	`not False`	=	`True`
`False and True`	=	`False`	`True and not False`	=	`True`
`False and False`	=	`False`	`False or (False or True)`	=	
`True or True`	=	`True`	`True and (False or True)`	=	
`True or False`	=	`True`	`False or False or False or True`	=	
`False or True`	=	`True`	`True and True and True and False`	=	
`False or False`	=	`False`	`not (False and not True)`	=	

An Aside – Boolean Arithmetic

- `True`/`False` values – booleans – are foundational to programming.
- We will often do something "`if`" a condition is satisfied (`True`), and not do it if the condition is `False`.
- It is **critical** to understand how they are combined (and how other variables may be converted into them).

True and True	= True	not True	= False
True and False	= False	not False	= True
False and True	= False	True and not False	= True
False and False	= False	False or (False or True)	= True
True or True	= True	True and (False or True)	=
True or False	= True	False or False or False or True	=
False or True	= True	True and True and True and False	=
False or False	= False	not (False and not True)	=

An Aside – Boolean Arithmetic

- True/False values – booleans – are foundational to programming.
- We will often do something “if” a condition is satisfied (True), and not do it if the condition is False.
- It is **critical** to understand how they are combined (and how other variables may be converted into them).

True and True	=	True	not True	=	False
True and False	=	False	not False	=	True
False and True	=	False	True and not False	=	True
False and False	=	False	False or (False or True)	=	True
True or True	=	True	True and (False or True)	=	True
True or False	=	True	False or False or False or True	=	
False or True	=	True	True and True and True and False	=	
False or False	=	False	not (False and not True)	=	

An Aside – Boolean Arithmetic

- **True**/**False** values – booleans – are foundational to programming.
- We will often do something "if" a condition is satisfied (**True**), and not do it if the condition is **False**.
- It is **critical** to understand how they are combined (and how other variables may be converted into them).

True and True	=	True	not True	=	False
True and False	=	False	not False	=	True
False and True	=	False	True and not False	=	True
False and False	=	False	False or (False or True)	=	True
True or True	=	True	True and (False or True)	=	True
True or False	=	True	False or False or False or True	=	True
False or True	=	True	True and True and True and False	=	
False or False	=	False	not (False and not True)	=	

An Aside – Boolean Arithmetic

- True/False values – booleans – are foundational to programming.
- We will often do something “if” a condition is satisfied (True), and not do it if the condition is False.
- It is **critical** to understand how they are combined (and how other variables may be converted into them).

True and True	=	True	not True	=	False
True and False	=	False	not False	=	True
False and True	=	False	True and not False	=	True
False and False	=	False	False or (False or True)	=	True
True or True	=	True	True and (False or True)	=	True
True or False	=	True	False or False or False or True	=	True
False or True	=	True	True and True and True and False	=	False
False or False	=	False	not (False and not True)	=	

An Aside – Boolean Arithmetic

- **`True`**/**`False`** values – booleans – are foundational to programming.
- We will often do something “`if`” a condition is satisfied (**`True`**), and not do it if the condition is **`False`**.
- It is **critical** to understand how they are combined (and how other variables may be converted into them).

`True and True`	=	`True`	`not True`	=	`False`
`True and False`	=	`False`	`not False`	=	`True`
`False and True`	=	`False`	`True and not False`	=	`True`
`False and False`	=	`False`	`False or (False or True)`	=	`True`
`True or True`	=	`True`	`True and (False or True)`	=	`True`
`True or False`	=	`True`	`False or False or False or True`	=	`True`
`False or True`	=	`True`	`True and True and True and False`	=	`False`
`False or False`	=	`False`	`not (False and not True)`	=	`True`