



Jamie Saxon

Introduction to Programming for Public Policy

October 2, 2017

## 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).

<b>True</b> and <b>True</b> =	=
<b>True</b> and <b>False</b> =	=
<b>False</b> and <b>True</b> =	=
<b>False</b> and <b>False</b> =	=
<b>True</b> or <b>True</b> =	=
<b>True</b> or <b>False</b> =	=
<b>False</b> or <b>True</b> =	=
<b>False</b> or <b>False</b> =	=

# 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).

<b>True</b> and <b>True</b> = <b>True</b>	=
<b>True</b> and <b>False</b> =	=
<b>False</b> and <b>True</b> =	=
<b>False</b> and <b>False</b> =	=
<b>True</b> or <b>True</b> =	=
<b>True</b> or <b>False</b> =	=
<b>False</b> or <b>True</b> =	=
<b>False</b> or <b>False</b> =	=

# 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).

<b>True</b> and <b>True</b> = <b>True</b>	=
<b>True</b> and <b>False</b> = <b>False</b>	=
<b>False</b> and <b>True</b> =	=
<b>False</b> and <b>False</b> =	=
<b>True</b> or <b>True</b> =	=
<b>True</b> or <b>False</b> =	=
<b>False</b> or <b>True</b> =	=
<b>False</b> or <b>False</b> =	=

# 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).

<b>True</b> and <b>True</b> = <b>True</b>	=
<b>True</b> and <b>False</b> = <b>False</b>	=
<b>False</b> and <b>True</b> = <b>False</b>	=
<b>False</b> and <b>False</b> =	=
<b>True</b> or <b>True</b> =	=
<b>True</b> or <b>False</b> =	=
<b>False</b> or <b>True</b> =	=
<b>False</b> or <b>False</b> =	=

# 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).

<b>True</b> and <b>True</b> = <b>True</b>	=
<b>True</b> and <b>False</b> = <b>False</b>	=
<b>False</b> and <b>True</b> = <b>False</b>	=
<b>False</b> and <b>False</b> = <b>False</b>	=
<b>True</b> or <b>True</b> =	=
<b>True</b> or <b>False</b> =	=
<b>False</b> or <b>True</b> =	=
<b>False</b> or <b>False</b> =	=

# 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).

<b>True</b> and <b>True</b> = <b>True</b>	=
<b>True</b> and <b>False</b> = <b>False</b>	=
<b>False</b> and <b>True</b> = <b>False</b>	=
<b>False</b> and <b>False</b> = <b>False</b>	=
<b>True</b> or <b>True</b> = <b>True</b>	=
<b>True</b> or <b>False</b> =	=
<b>False</b> or <b>True</b> =	=
<b>False</b> or <b>False</b> =	=

# 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).

<b>True</b> and <b>True</b> = <b>True</b>	=
<b>True</b> and <b>False</b> = <b>False</b>	=
<b>False</b> and <b>True</b> = <b>False</b>	=
<b>False</b> and <b>False</b> = <b>False</b>	=
<b>True</b> or <b>True</b> = <b>True</b>	=
<b>True</b> or <b>False</b> = <b>True</b>	=
<b>False</b> or <b>True</b> =	=
<b>False</b> or <b>False</b> =	=



# 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).

<b>True</b> and <b>True</b> = <b>True</b>	=
<b>True</b> and <b>False</b> = <b>False</b>	=
<b>False</b> and <b>True</b> = <b>False</b>	=
<b>False</b> and <b>False</b> = <b>False</b>	=
<b>True</b> or <b>True</b> = <b>True</b>	=
<b>True</b> or <b>False</b> = <b>True</b>	=
<b>False</b> or <b>True</b> = <b>True</b>	=
<b>False</b> or <b>False</b> =	=

# 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).

<b>True</b> and <b>True</b> = <b>True</b>	=
<b>True</b> and <b>False</b> = <b>False</b>	=
<b>False</b> and <b>True</b> = <b>False</b>	=
<b>False</b> and <b>False</b> = <b>False</b>	=
<b>True</b> or <b>True</b> = <b>True</b>	=
<b>True</b> or <b>False</b> = <b>True</b>	=
<b>False</b> or <b>True</b> = <b>True</b>	=
<b>False</b> or <b>False</b> = <b>False</b>	=

# 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).

<b>True</b> and <b>True</b> = <b>True</b>	not <b>True</b> =
<b>True</b> and <b>False</b> = <b>False</b>	not <b>False</b> =
<b>False</b> and <b>True</b> = <b>False</b>	<b>True</b> and not <b>False</b> =
<b>False</b> and <b>False</b> = <b>False</b>	<b>False</b> or ( <b>False</b> or <b>True</b> ) =
<b>True</b> or <b>True</b> = <b>True</b>	<b>True</b> and ( <b>False</b> or <b>True</b> ) =
<b>True</b> or <b>False</b> = <b>True</b>	<b>False</b> or <b>False</b> or <b>False</b> or <b>True</b> =
<b>False</b> or <b>True</b> = <b>True</b>	<b>True</b> and <b>True</b> and <b>True</b> and <b>False</b> =
<b>False</b> or <b>False</b> = <b>False</b>	not ( <b>False</b> and not <b>True</b> ) =

# 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).

<b>True</b> and <b>True</b> = <b>True</b>	not <b>True</b> = <b>False</b>
<b>True</b> and <b>False</b> = <b>False</b>	not <b>False</b> =
<b>False</b> and <b>True</b> = <b>False</b>	<b>True</b> and not <b>False</b> =
<b>False</b> and <b>False</b> = <b>False</b>	<b>False</b> or ( <b>False</b> or <b>True</b> ) =
<b>True</b> or <b>True</b> = <b>True</b>	<b>True</b> and ( <b>False</b> or <b>True</b> ) =
<b>True</b> or <b>False</b> = <b>True</b>	<b>False</b> or <b>False</b> or <b>False</b> or <b>True</b> =
<b>False</b> or <b>True</b> = <b>True</b>	<b>True</b> and <b>True</b> and <b>True</b> and <b>False</b> =
<b>False</b> or <b>False</b> = <b>False</b>	not ( <b>False</b> and not <b>True</b> ) =

# 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).

<b>True</b> and <b>True</b> = <b>True</b>	not <b>True</b> = <b>False</b>
<b>True</b> and <b>False</b> = <b>False</b>	not <b>False</b> = <b>True</b>
<b>False</b> and <b>True</b> = <b>False</b>	<b>True</b> and not <b>False</b> =
<b>False</b> and <b>False</b> = <b>False</b>	<b>False</b> or ( <b>False</b> or <b>True</b> ) =
<b>True</b> or <b>True</b> = <b>True</b>	<b>True</b> and ( <b>False</b> or <b>True</b> ) =
<b>True</b> or <b>False</b> = <b>True</b>	<b>False</b> or <b>False</b> or <b>False</b> or <b>True</b> =
<b>False</b> or <b>True</b> = <b>True</b>	<b>True</b> and <b>True</b> and <b>True</b> and <b>False</b> =
<b>False</b> or <b>False</b> = <b>False</b>	not ( <b>False</b> and not <b>True</b> ) =

# 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).

<b>True</b> and <b>True</b> = <b>True</b>	not <b>True</b> = <b>False</b>
<b>True</b> and <b>False</b> = <b>False</b>	not <b>False</b> = <b>True</b>
<b>False</b> and <b>True</b> = <b>False</b>	<b>True</b> and not <b>False</b> = <b>True</b>
<b>False</b> and <b>False</b> = <b>False</b>	<b>False</b> or ( <b>False</b> or <b>True</b> ) =
<b>True</b> or <b>True</b> = <b>True</b>	<b>True</b> and ( <b>False</b> or <b>True</b> ) =
<b>True</b> or <b>False</b> = <b>True</b>	<b>False</b> or <b>False</b> or <b>False</b> or <b>True</b> =
<b>False</b> or <b>True</b> = <b>True</b>	<b>True</b> and <b>True</b> and <b>True</b> and <b>False</b> =
<b>False</b> or <b>False</b> = <b>False</b>	not ( <b>False</b> and not <b>True</b> ) =

# 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).

<b>True</b> and <b>True</b> = <b>True</b>	not <b>True</b> = <b>False</b>
<b>True</b> and <b>False</b> = <b>False</b>	not <b>False</b> = <b>True</b>
<b>False</b> and <b>True</b> = <b>False</b>	<b>True</b> and not <b>False</b> = <b>True</b>
<b>False</b> and <b>False</b> = <b>False</b>	<b>False</b> or ( <b>False</b> or <b>True</b> ) = <b>True</b>
<b>True</b> or <b>True</b> = <b>True</b>	<b>True</b> and ( <b>False</b> or <b>True</b> ) =
<b>True</b> or <b>False</b> = <b>True</b>	<b>False</b> or <b>False</b> or <b>False</b> or <b>True</b> =
<b>False</b> or <b>True</b> = <b>True</b>	<b>True</b> and <b>True</b> and <b>True</b> and <b>False</b> =
<b>False</b> or <b>False</b> = <b>False</b>	not ( <b>False</b> and not <b>True</b> ) =

# 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).

<b>True</b> and <b>True</b> = <b>True</b>	not <b>True</b> = <b>False</b>
<b>True</b> and <b>False</b> = <b>False</b>	not <b>False</b> = <b>True</b>
<b>False</b> and <b>True</b> = <b>False</b>	<b>True</b> and not <b>False</b> = <b>True</b>
<b>False</b> and <b>False</b> = <b>False</b>	<b>False</b> or ( <b>False</b> or <b>True</b> ) = <b>True</b>
<b>True</b> or <b>True</b> = <b>True</b>	<b>True</b> and ( <b>False</b> or <b>True</b> ) = <b>True</b>
<b>True</b> or <b>False</b> = <b>True</b>	<b>False</b> or <b>False</b> or <b>False</b> or <b>True</b> =
<b>False</b> or <b>True</b> = <b>True</b>	<b>True</b> and <b>True</b> and <b>True</b> and <b>False</b> =
<b>False</b> or <b>False</b> = <b>False</b>	not ( <b>False</b> and not <b>True</b> ) =



# 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).

<b>True</b> and <b>True</b> = <b>True</b>	not <b>True</b> = <b>False</b>
<b>True</b> and <b>False</b> = <b>False</b>	not <b>False</b> = <b>True</b>
<b>False</b> and <b>True</b> = <b>False</b>	<b>True</b> and not <b>False</b> = <b>True</b>
<b>False</b> and <b>False</b> = <b>False</b>	<b>False</b> or ( <b>False</b> or <b>True</b> ) = <b>True</b>
<b>True</b> or <b>True</b> = <b>True</b>	<b>True</b> and ( <b>False</b> or <b>True</b> ) = <b>True</b>
<b>True</b> or <b>False</b> = <b>True</b>	<b>False</b> or <b>False</b> or <b>False</b> or <b>True</b> = <b>True</b>
<b>False</b> or <b>True</b> = <b>True</b>	<b>True</b> and <b>True</b> and <b>True</b> and <b>False</b> =
<b>False</b> or <b>False</b> = <b>False</b>	not ( <b>False</b> and not <b>True</b> ) =

# 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).

<b>True</b> and <b>True</b> = <b>True</b>	not <b>True</b> = <b>False</b>
<b>True</b> and <b>False</b> = <b>False</b>	not <b>False</b> = <b>True</b>
<b>False</b> and <b>True</b> = <b>False</b>	<b>True</b> and not <b>False</b> = <b>True</b>
<b>False</b> and <b>False</b> = <b>False</b>	<b>False</b> or ( <b>False</b> or <b>True</b> ) = <b>True</b>
<b>True</b> or <b>True</b> = <b>True</b>	<b>True</b> and ( <b>False</b> or <b>True</b> ) = <b>True</b>
<b>True</b> or <b>False</b> = <b>True</b>	<b>False</b> or <b>False</b> or <b>False</b> or <b>True</b> = <b>True</b>
<b>False</b> or <b>True</b> = <b>True</b>	<b>True</b> and <b>True</b> and <b>True</b> and <b>False</b> = <b>False</b>
<b>False</b> or <b>False</b> = <b>False</b>	not ( <b>False</b> and not <b>True</b> ) =

# 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).

<b>True</b> and <b>True</b> = <b>True</b>	not <b>True</b> = <b>False</b>
<b>True</b> and <b>False</b> = <b>False</b>	not <b>False</b> = <b>True</b>
<b>False</b> and <b>True</b> = <b>False</b>	<b>True</b> and not <b>False</b> = <b>True</b>
<b>False</b> and <b>False</b> = <b>False</b>	<b>False</b> or ( <b>False</b> or <b>True</b> ) = <b>True</b>
<b>True</b> or <b>True</b> = <b>True</b>	<b>True</b> and ( <b>False</b> or <b>True</b> ) = <b>True</b>
<b>True</b> or <b>False</b> = <b>True</b>	<b>False</b> or <b>False</b> or <b>False</b> or <b>True</b> = <b>True</b>
<b>False</b> or <b>True</b> = <b>True</b>	<b>True</b> and <b>True</b> and <b>True</b> and <b>False</b> = <b>False</b>
<b>False</b> or <b>False</b> = <b>False</b>	not ( <b>False</b> and not <b>True</b> ) = <b>True</b>

# 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.

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](http://tmpnb.org) or [try.jupyter.org](http://try.jupyter.org)**

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

