Pi515

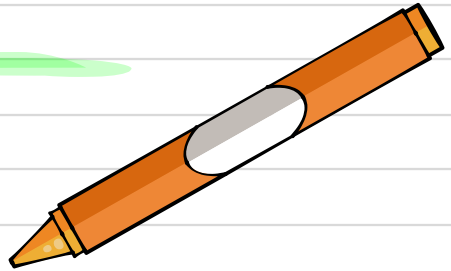
Tech Mentorship

Unit 6

Booleans, Comparison Operators, and Logical Operators

01

Booleans



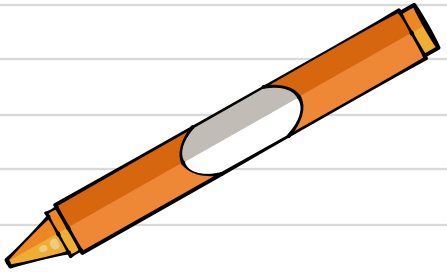
What is a Boolean?

A Boolean is a data type with two possible values: True or False

True	False
1	0

02

Comparison Operators



What are Comparison Operators?

Comparison operators are special characters that allow us to compare values and check the relationship between them.

`>`	`<`	`==`
Greater Than	Less Than	Equal to
`>=`	`<=`	`!=`
Greater Than or Equal to	Less than or Equal to	Not Equal to

Comparison Operators in Action Pt.1

01 `10 > 4`
Answer:

02 `11 == 4`
Answer:

03 `3 < 3`
Answer:

04 `4 == 4`
Answer:

05 `4 < 12`
Answer:

06 `6 > 6`
Answer:

Comparison Operators in Action Pt.1

01 `10 > 4`
Answer: True

02 `11 == 4`
Answer: False

03 `3 < 3`
Answer: False

04 `4 == 4`
Answer: True

05 `4 < 12`
Answer: True

06 `6 > 6`
Answer: False

Comparison Operators in Action Pt.2

01 `10 >= 10`
Answer:

02 `2 != 3`
Answer:

03 `10 <= 3`
Answer:

04 `5 >= 10`
Answer:

05 `5 != 5`
Answer:

06 `9 <= 9`
Answer:

Comparison Operators in Action Pt.2

01 `10 >= 10`
Answer: True

02 `2 != 3`
Answer: True

03 `10 <= 3`
Answer: False

04 `5 >= 10`
Answer: False

05 `5 != 5`
Answer: False

06 `9 <= 9`
Answer: True

Comparison Operators in Python

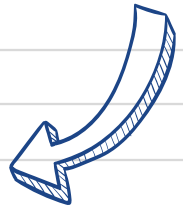
main.py

```
#====================
# Test out the ">" & ">=" Logical Operator in Python

print("5 is greater than 2:", 5 > 5)
print("5 is greater than or equal to 5:", 5 >= 5)
print()

#====================
# Test out the "<" & "<=" Logical Operator in Python

print("5 is less than 10:", 5 < 10)
print("3 is less than or equal to 3", 3 <= 3)
print()

#====================
# Test out the "==" & "!=" Logical Operator in
Python

print("5 equals 6:", 5 == 6)
print("10 does not equal 7", 10 != 7)
```

Shell | Console

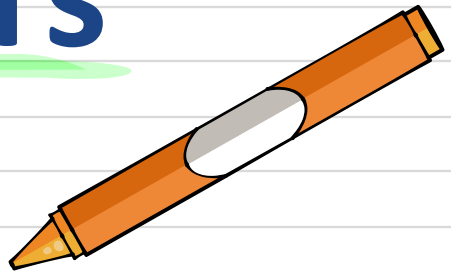
```
5 is greater than 2: False
5 is greater than or equal to 5: True

5 is less than 10: True
3 is less than or equal to 3 True

5 equals 6: False
10 does not equal 7 True
>
```

Notice how we can use **Booleans** and **Comparison Operators** in Python



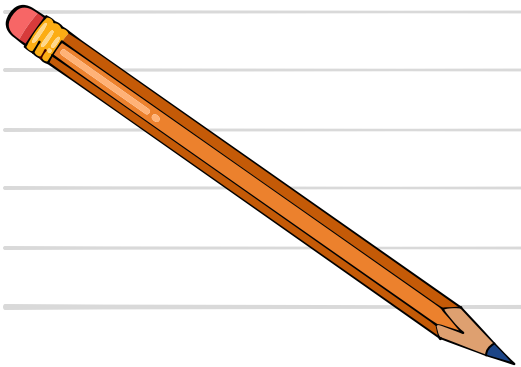
03

Logical Operators



What are Logical Operators

Logical operators are special keywords used to combine multiple conditions and determine the overall truth value of the combined conditions.



And

The and operator in Python evaluates to True only if both conditions on its left and right are True; otherwise, it evaluates to False.

Or

It evaluates to True if at least one of the conditions is True. If both conditions are False, it evaluates to False.

Not

The not operator is used to negate or reverse the value of a condition.

“and” Operator in Action

True and True

Answer: True

True and False

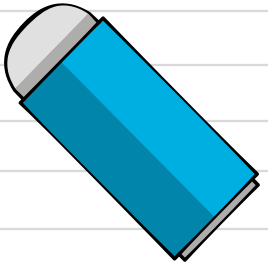
Answer: False

False and True

Answer: False

False and False

Answer: False



“or” Operator in Action

True or True

Answer: True

True or False

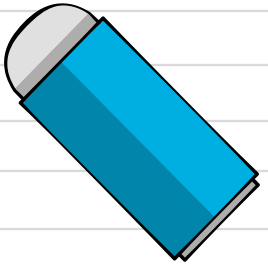
Answer: True

False or True

Answer: True

False or False

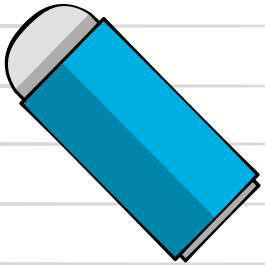
Answer: False



“Not” Operator in Action

Not True

Answer: False

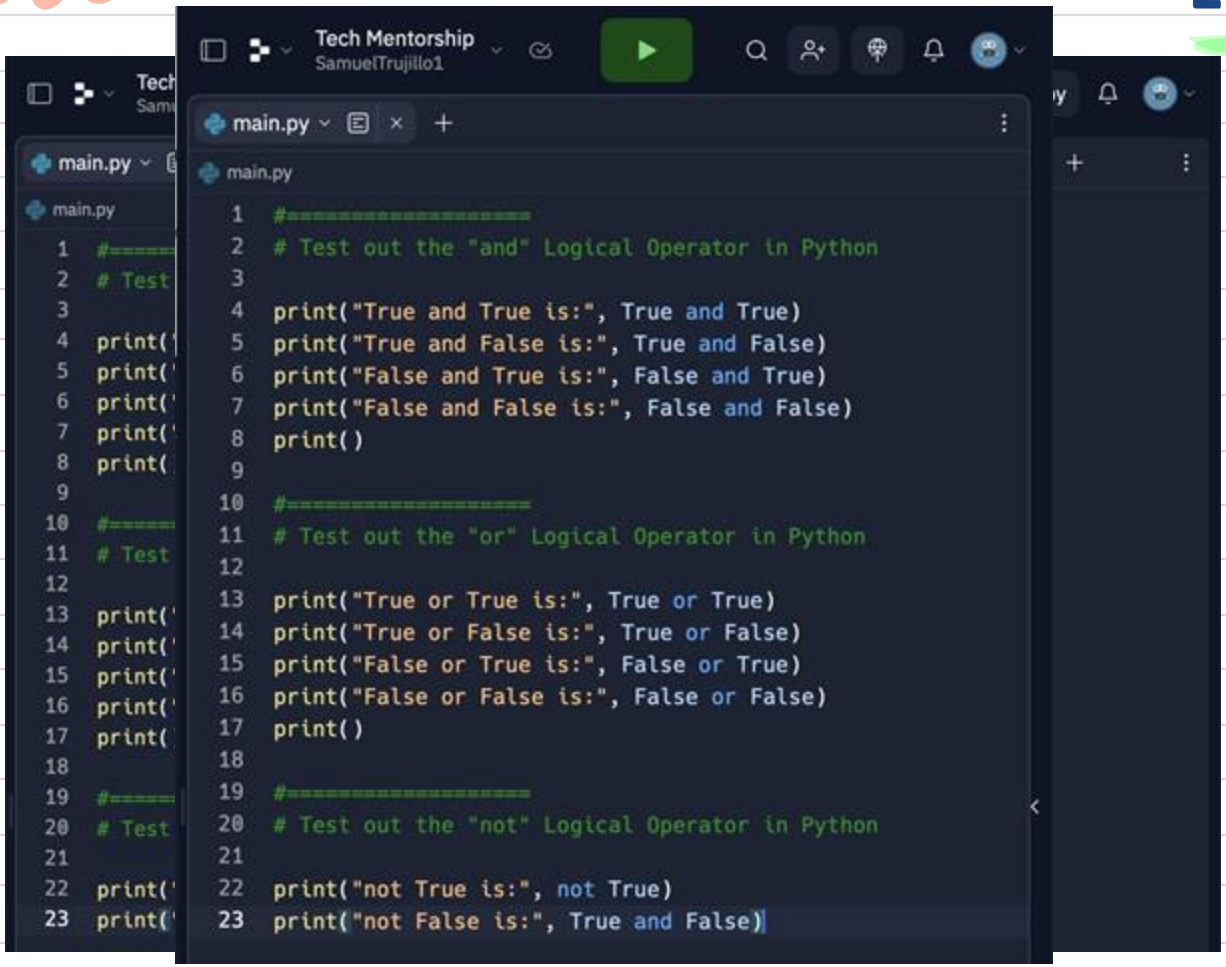


$$x = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

Not False

Answer: True

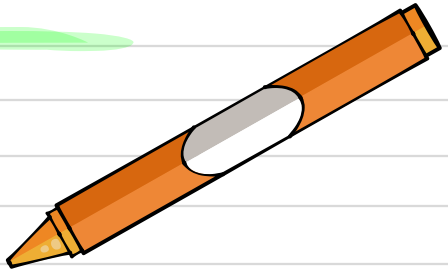
Logical Operators in Python



Take a Look at how we can use Booleans and Comparison Operators in Python

04

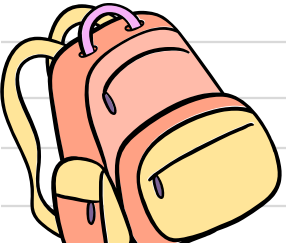
Comparison & Logical Operators



Both Operators Together Pt.1

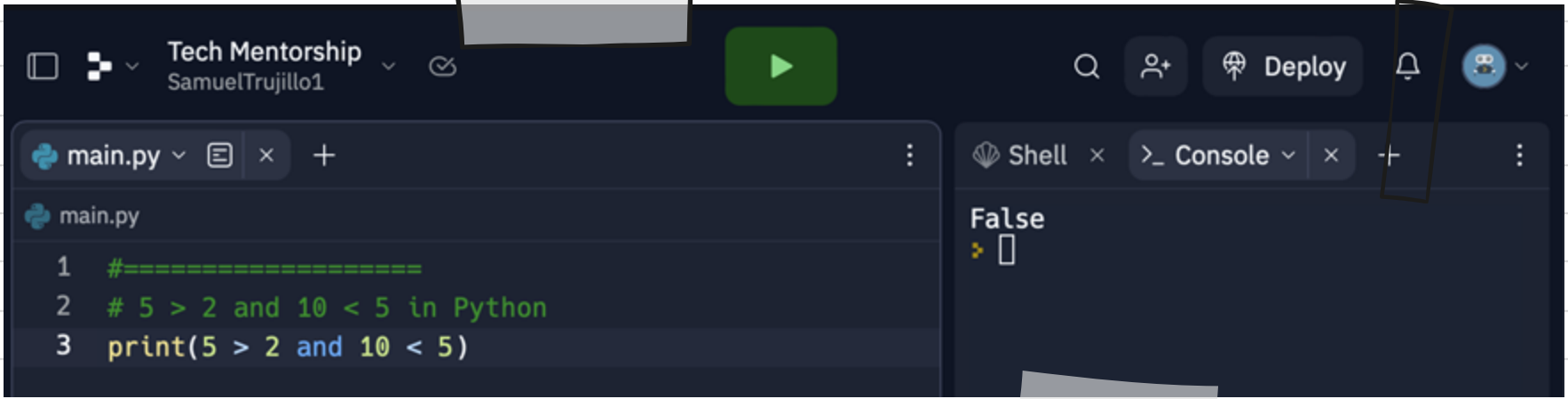
5 > 2 and 10 < 5

5 > 2	10 < 5	T and F	False
Solve the Left Side First	Then Solve the right Side	Apply to the booleans with the "and"	The Overall Statement is False
5 > 2: True	**5 > 2: False**	**T and F: F**	**5 > 2 and 10 < 5: F**



Both Operators in Python Pt.1

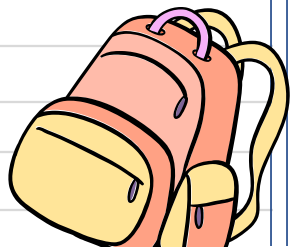
Here is the previous slide in Python



Both Operators Together Pt.2

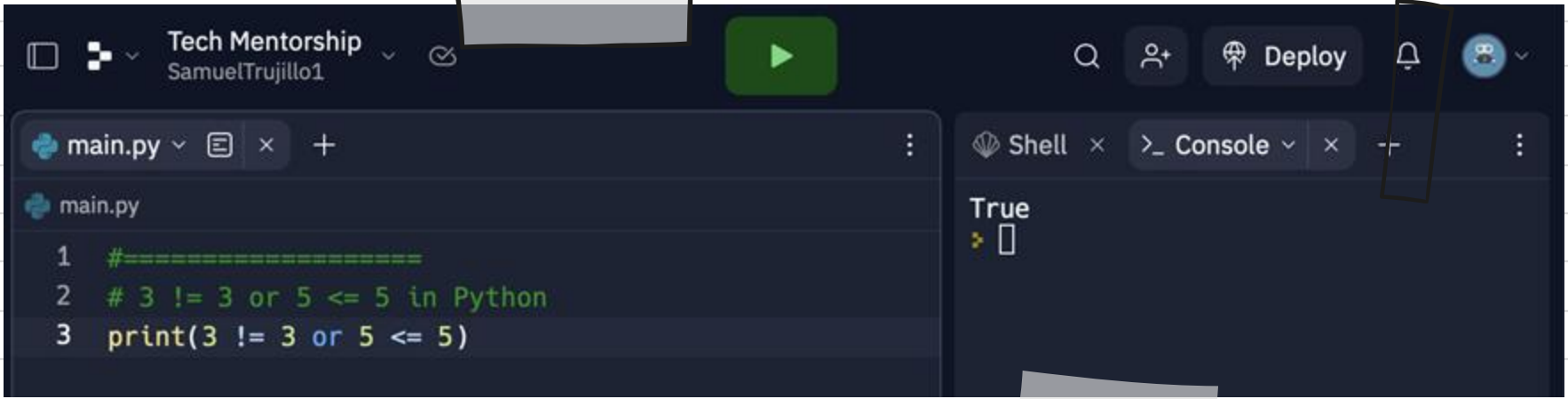
3 != 3 or 5 <= 5

3 != 3	5 <= 5	F or T	True
Solve the Left Side First	Then Solve the right Side	Apply to the booleans with the "and"	The Overall Statement is True
3 != 3: False	**5 <= 5: True**	**F or T: True**	**3 != 3 or 5 <=5: True**



Both Operators in Python Pt.2

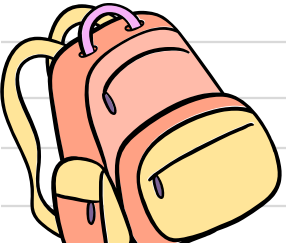
Here is the previous slide in Python



Both Operators Together Pt.3

not(5 <= 2 and 4 == 4)

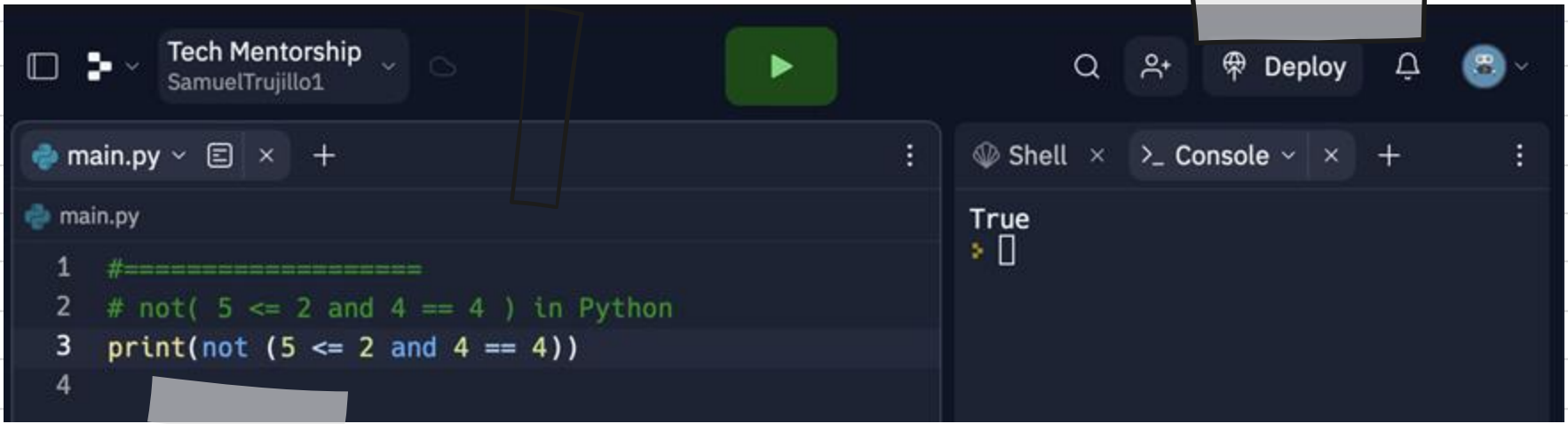
5 <= 2	4 == 4	F and T	not(F)
Solve the Left Side First	Then Solve the right Side	Apply to the booleans with the "and"	Apply the "not"
5 <= 2: False	4 == 4: True	F and T: F	not(F): T



not(5 <= 2 and 4 == 4): True

Both Operators in Python Pt.3

Here is the previous slide in Python



Practice & Review

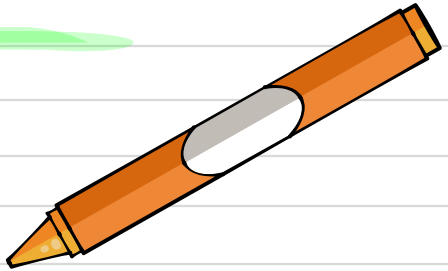
01 **not(7 < 2 or 5 != 5)**
Answer: True

02 **Not (True) and True**
Answer: False

03 **4 > 3 and 2 != 1**
Answer: True

05

Boolean Algebra Challenges



Boolean Algebra Formulas

Drawing on similarities with traditional math allows us to do more complex Boolean Algebra

Treat variables as either 0 or 1 (AKA F/T)

Treat ‘or’ like addition

Treat ‘and’ like multiplication

Name	AND form	OR form
Identity law	$1A = A$	$0 + A = A$
Null law	$0A = 0$	$1 + A = 1$
Idempotent law	$AA = A$	$A + A = A$
Inverse law	$A\overline{A} = 0$	$A + \overline{A} = 1$
Commutative law	$AB = BA$	$A + B = B + A$
Associative law	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Distributive law	$A + BC = (A + B)(A + C)$	$A(B + C) = AB + AC$
Absorption law	$A(A + B) = A$	$A + AB = A$
De Morgan's law	$\overline{AB} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A}\,\overline{B}$