Techtorial

Operator Precedence

- MULTIPLICATION, DIVISION AND REMAINDER WILL BE EXECUTED BEFORE THE ADDITION AND SUBSTRaCTION

- if the expression contains more than one operator with the same level of precedence they are evaluated from left to right

type() function

type() method returns class type of the argument(object) passed as parameter in Python.

Techtorial

Changing the Operator Precedence

- To change the order, we can use the parentheses
- We should always start from most inner parentheses

Techtorial

Assign Multiple Values In One Line

Pyhton allows us to assign multiple variables in a single line

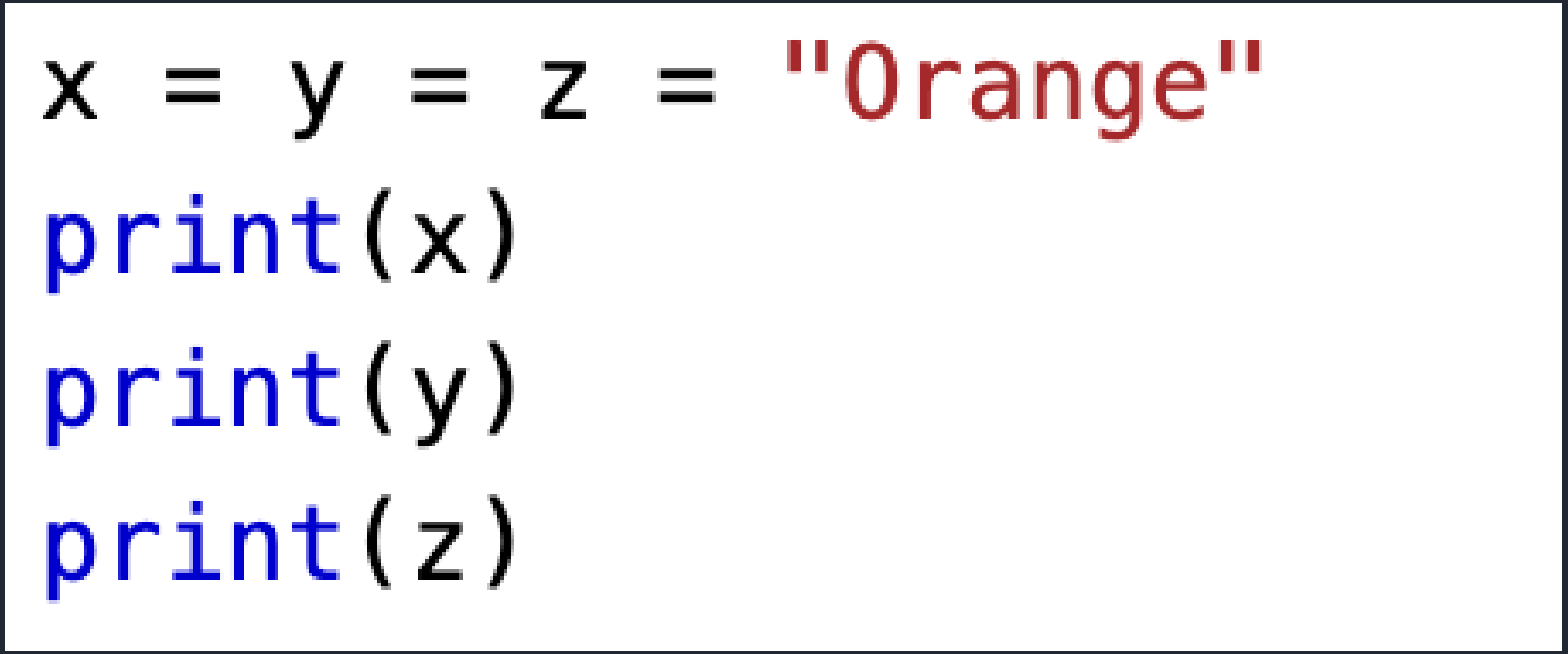
Assign Multiple Values In One Line

```python
x, y, z = "Orange", "Banana", "Cherry"
print(x)
print(y)
print(z)
```

Note: Make sure the number of variables matches the number of values, or else you will get an error.

Techtorial

Assign One Value to Multiple Variables



```
x = y = z = "Orange"
print(x)
print(y)
print(z)
```

Techtorial

Re-Assigning the values

After initializing the variables, we can reassign them again.

Techtorial

Assignment Operators In Python

=, +=, -=, *=, /=, %=, //=, **= , ...

Techtorial

Compound Assignment Operators

+=, -=, *=, /=, %=,//=,**=

Note: There are more compound assignment operators but these are the most common ones.

Techtorial

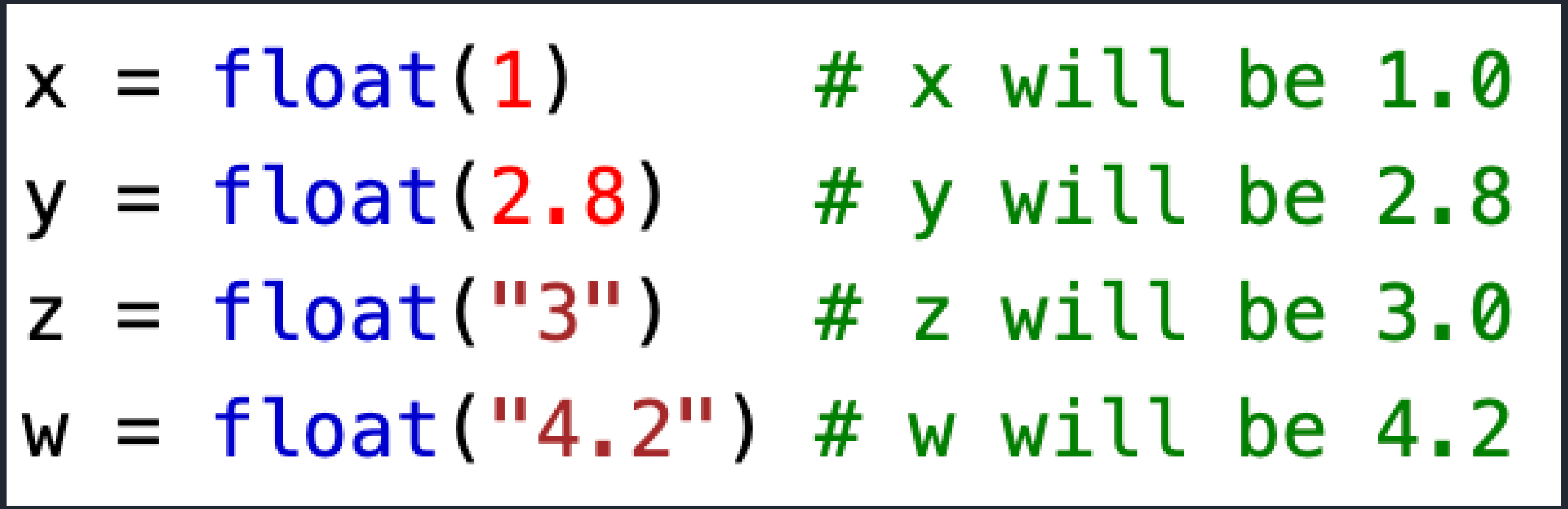
Python Casting

There may be times when you want to specify a type on to a variable. This can be done with casting. Python is an object-orientated language, and as such it uses classes to define data types, including its primitive types.
Casting in python is therefore done using constructor functions:

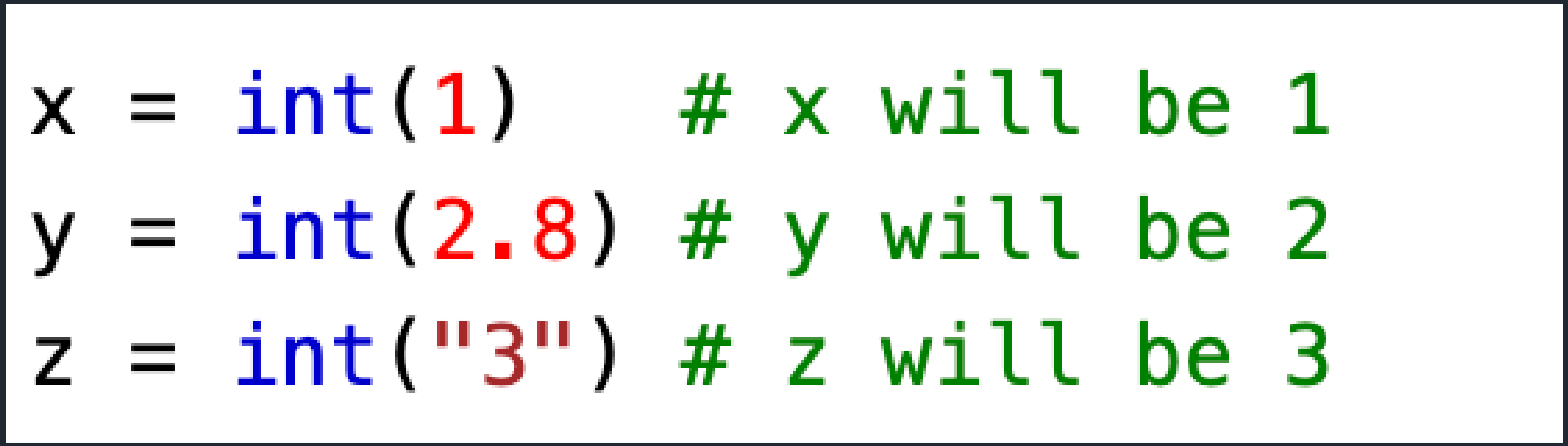
- int() - constructs an integer number from an integer literal, a float literal (by removing all decimals), or a string literal (providing the string represents a whole number)
- float() - constructs a float number from an integer literal, a float literal or a string literal (providing the string represents a float or an integer)

Techtorial

Python Casting



```python
x = float(1)     # x will be 1.0
y = float(2.8)   # y will be 2.8
z = float("3")   # z will be 3.0
w = float("4.2") # w will be 4.2
```



```python
x = int(1)   # x will be 1
y = int(2.8) # y will be 2
z = int("3") # z will be 3
```

Boolean

Boolean variables can only take 2 values
True or False.
It helps us to understand truth of the expressions

Numerically, True is equal to 1 and False is equal to0.

Boolean was invented to give some instructions for the computer.
For instance, if this condition is true we should continue, but in the opposite case, we should stop.

***

Every number except 0 means True. However,
it is better to use only 0 and 1, to avoid misunderstanding.***

It should be noted that if we write,
true instead of True
or
false instead of False,
it leads to an error.

Techtorial

Boolean Comparison Operators

>, <,!=, ==, <=, >=
are the key signs for boolean statements

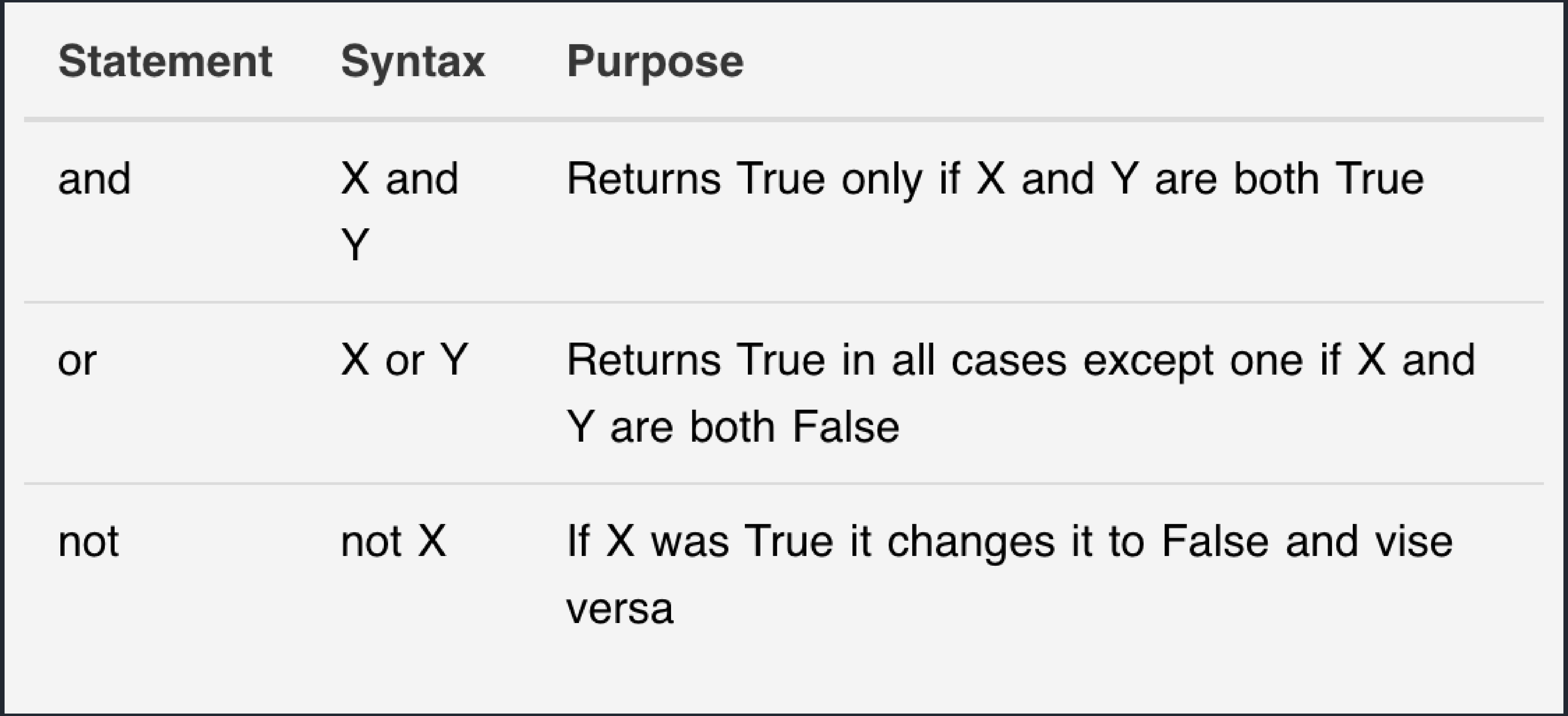
Sign	Syntax	Purpose
>	X > Y	Returns True if X greater than Y
<	X < Y	Returns True if X less than Y
!=	X != Y	Returns True if X not equal to Y
==	X == Y	Returns True if X is equal to Y
<=	X <= Y	Returns True if X is less or equal to Y
>=	X >= Y	Returns True if X is greater or equal to Y

Techtorial

Boolean Logical Operators

- And
- Or
- Not

Techtorial



Statement	Syntax	Purpose
and	X and Y	Returns True only if X and Y are both True
or	X or Y	Returns True in all cases except one if X and Y are both False
not	not X	If X was True it changes it to False and vise versa

Boolean Logical Operators

Note: and is multiplication for boolean statements and or is addition; hence, and has more prominence.

Techtorial

Boolean Logical Operators

Step by Step

1. ((True or True) and False) or True
2. (True and False) or True
3. False or True
4. True

Techtorial

Boolean Logical Operators

```
Input:
statement=((True or True) and False) or True
print(statement)

Output:
True
```

Techtorial

The bool() function

bool() function allows you to evaluate any value, and give you True or False in return

The bool() function

Most Values are True
Almost any value is evaluated to True if it has some sort of content.
Any string is True, except empty strings.
Any number is True, except 0.

Techtorial

The bool() function

Some Values are False

In fact, there are not many values that evaluate to False, except empty values, such as (), [], {}, "", the number 0, and the value None. And of course the value False evaluates to False

Techtorial