HOW TO DOWNLOAD PYTHON-

Go to this link- https://www.python.org/downloads/

Download latest version of python and install it

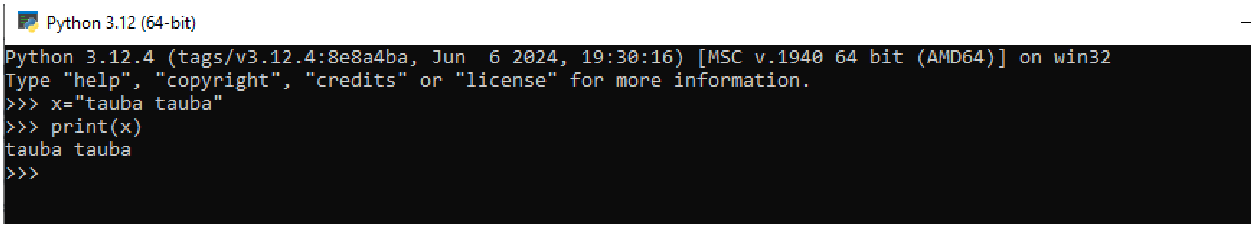
Day 1-

Constants- Fixed values such as Numbers, letters and strings

You can use single quotes(‘) or double quotes (“) for string constants

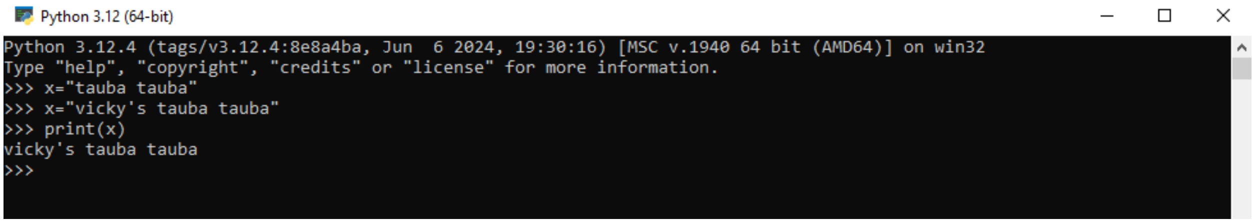
Variable- variable basically oka space where you can put your own value and call that whenever you want. It can be changed whenever you want

Assigning value to a variable-



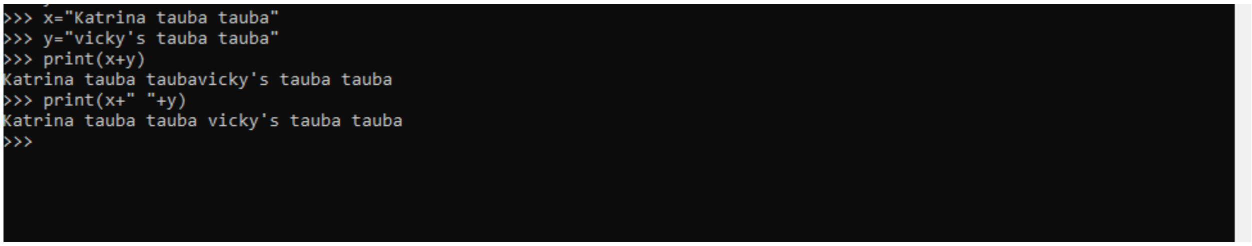
```
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun  6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> x="tauba tauba"
>>> print(x)
tauba tauba
>>>
```

We assigned tauba tauba to x



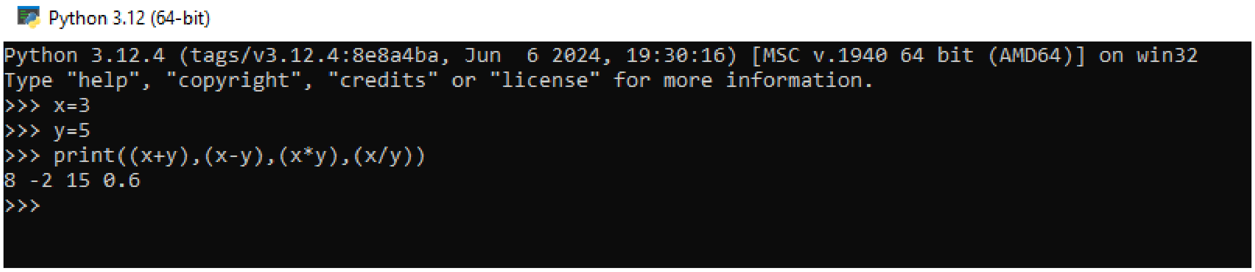
```
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun  6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> x="tauba tauba"
>>> x="vicky's tauba tauba"
>>> print(x)
vicky's tauba tauba
>>>
```

You can also add two strings



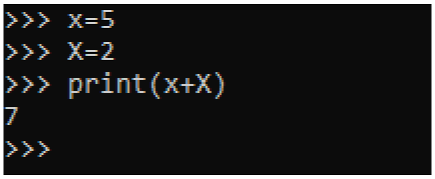
```
>>> x="Katrina tauba tauba"
>>> y="vicky's tauba tauba"
>>> print(x+y)
Katrina tauba taubavicky's tauba tauba
>>> print(x+" "+y)
Katrina tauba tauba vicky's tauba tauba
>>>
```

You can also do addition, subtraction, multiplication and division of numbers-



```
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun  6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> x=3
>>> y=5
>>> print((x+y),(x-y),(x*y),(x/y))
8 -2 15 0.6
>>>
```

Python is Case sensitive-



```
>>> x=5
>>> X=2
>>> print(x+X)
7
>>>
```

Successfully completed day 1! GOOD JOB

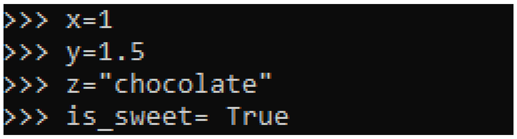
DAY2 PYTHON

**Integers**: Counting apples in a basket.

**Floats**: Measuring ingredients in a recipe (e.g., 1.5 cups of flour).

**Strings**: Writing a name or a sentence.

**Booleans**: Answering yes/no questions (e.g., Is the light on? True/False).



```
>>> x=1
>>> y=1.5
>>> z="chocolate"
>>> is_sweet= True
```

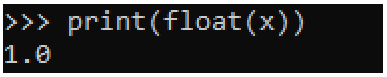
X is int

Y is float

Z is string

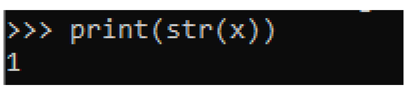
Is_sweet is Boolean

Conversion- INT to FLOAT-



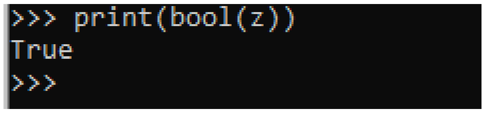
```
>>> print(float(x))
1.0
```

Conversion- INT to STRING-



```
>>> print(str(x))
1
```

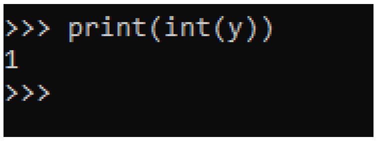
Conversion- STRING to Boolean



```
>>> print(bool(z))
True
>>>
```

True because z has “chocolate” saved if z was empty (z=””) then the value would return false

Conversion- Float to INT



```
>>> print(int(y))
1
>>>
```

Int datatype considers the value 1.5 as whole number 1

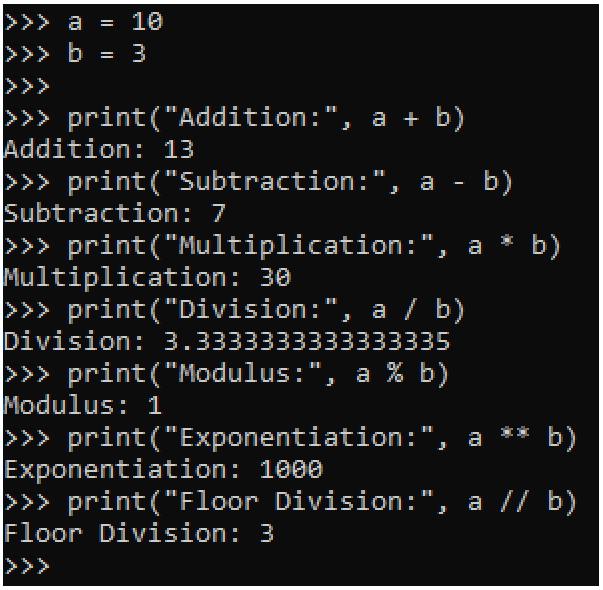
Congratulations GUYS DAY 2 completed.

DAY3 PYTHON

- Arithmetic operators
- Assignment operators
- Comparison operators
- Logical operators

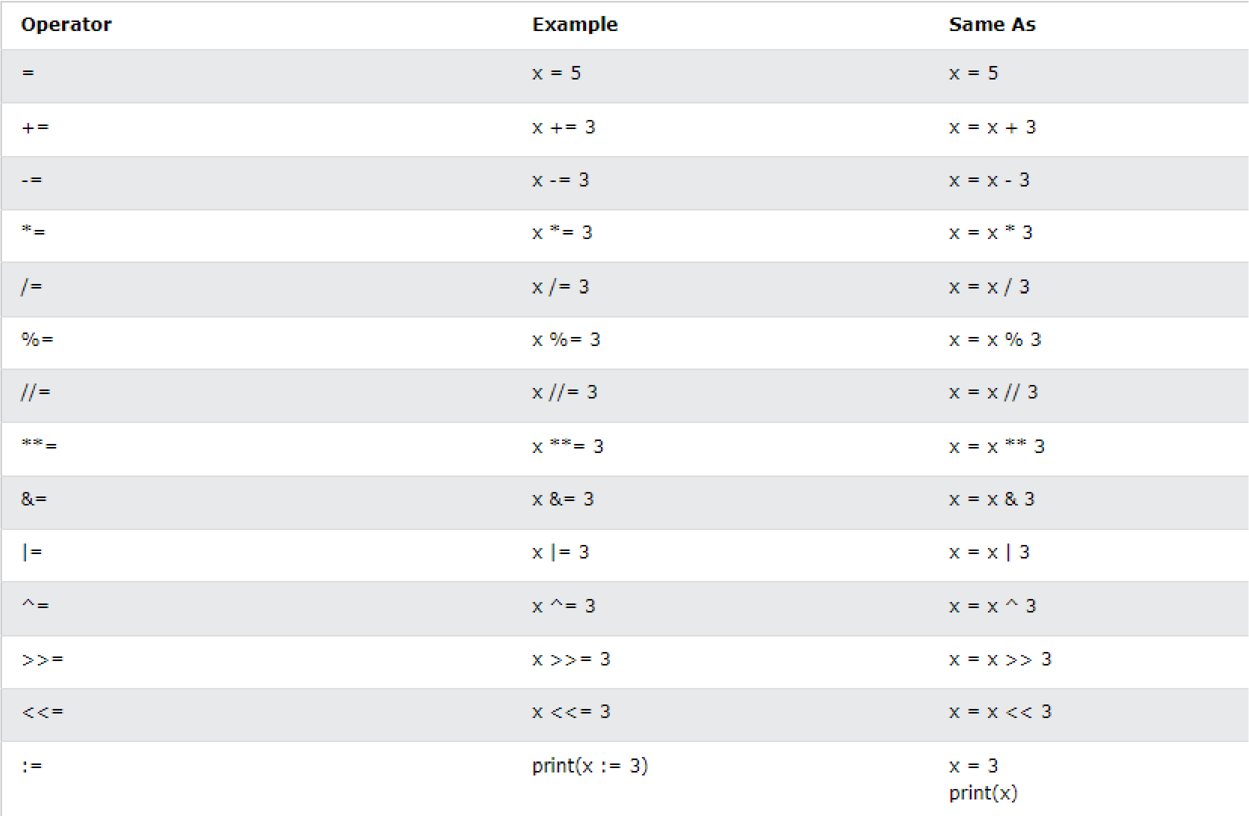
Arithmetic operators

| **Operator** | **Name** | **Example** |
|---|---|---|
| + | Addition | x + y |
| - | Subtraction | x - y |
| * | Multiplication | x * y |
| / | Division | x / y |
| % | Modulus | x % y |
| ** | Exponentiation | x ** y |
| // | Floor division | x // y |



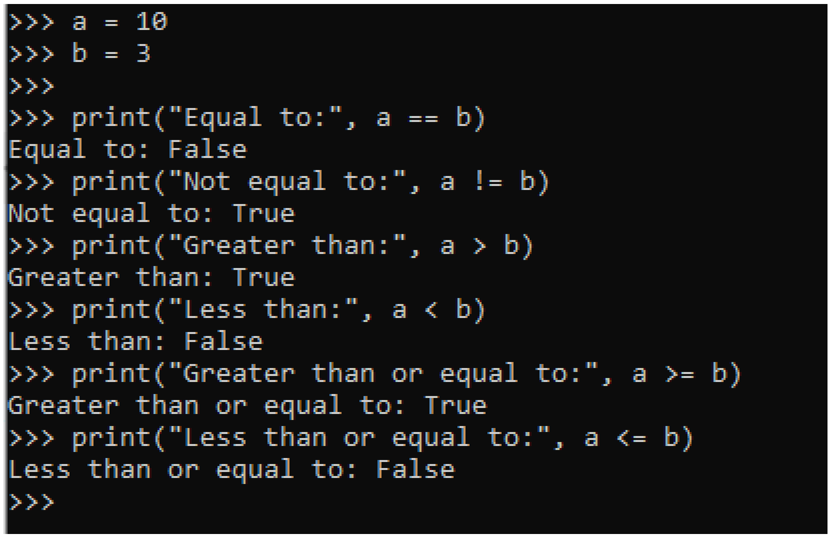
```
>>> a = 10
>>> b = 3
>>>
>>> print("Addition:", a + b)
Addition: 13
>>> print("Subtraction:", a - b)
Subtraction: 7
>>> print("Multiplication:", a * b)
Multiplication: 30
>>> print("Division:", a / b)
Division: 3.3333333333333335
>>> print("Modulus:", a % b)
Modulus: 1
>>> print("Exponentiation:", a ** b)
Exponentiation: 1000
>>> print("Floor Division:", a // b)
Floor Division: 3
>>>
```

Assignment operators

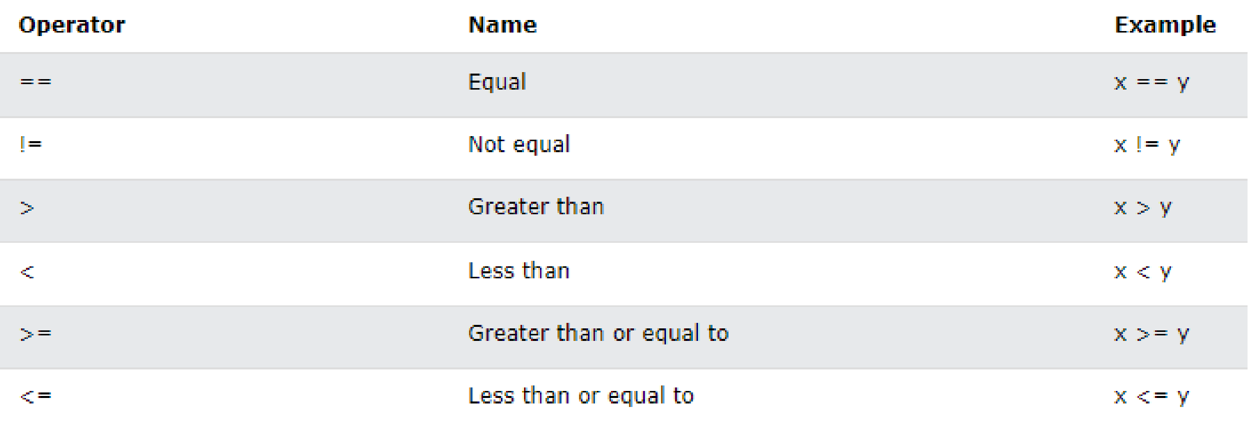


| Operator | Example | Same As |
|---|---|---|
| = | x = 5 | x = 5 |
| += | x += 3 | x = x + 3 |
| -= | x -= 3 | x = x - 3 |
| *= | x *= 3 | x = x * 3 |
| /= | x /= 3 | x = x / 3 |
| %= | x %= 3 | x = x % 3 |
| //= | x //= 3 | x = x // 3 |
| **= | x **= 3 | x = x ** 3 |
| &= | x &= 3 | x = x & 3 |
| \|= | x \|= 3 | x = x \| 3 |
| ^= | x ^= 3 | x = x ^ 3 |
| >>= | x >>= 3 | x = x >> 3 |
| <<= | x <<= 3 | x = x << 3 |
| := | print(x := 3) | x = 3<br>print(x) |

Comparison operators-

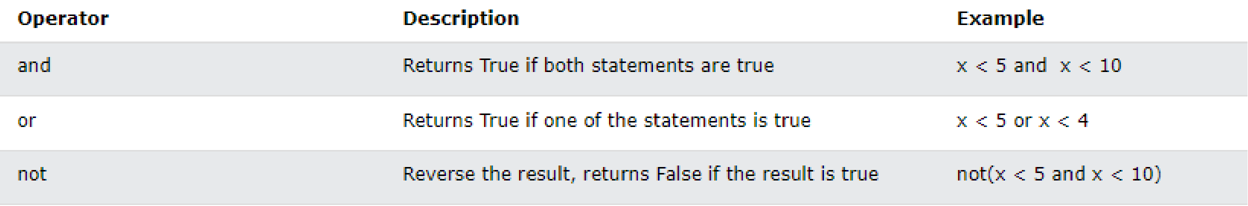


```
>>> a = 10
>>> b = 3
>>>
>>> print("Equal to:", a == b)
Equal to: False
>>> print("Not equal to:", a != b)
Not equal to: True
>>> print("Greater than:", a > b)
Greater than: True
>>> print("Less than:", a < b)
Less than: False
>>> print("Greater than or equal to:", a >= b)
Greater than or equal to: True
>>> print("Less than or equal to:", a <= b)
Less than or equal to: False
>>>
```

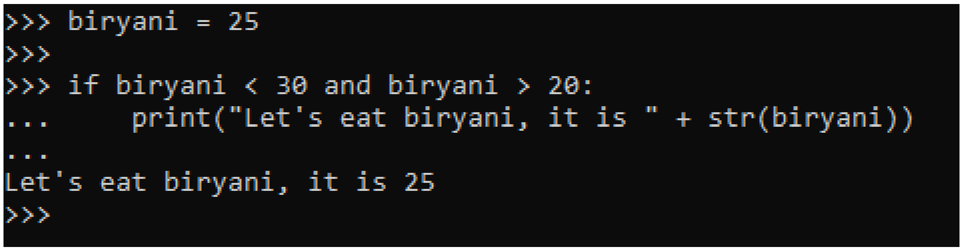


| Operator | Name | Example |
|---|---|---|
| == | Equal | x == y |
| != | Not equal | x != y |
| > | Greater than | x > y |
| < | Less than | x < y |
| >= | Greater than or equal to | x >= y |
| <= | Less than or equal to | x <= y |

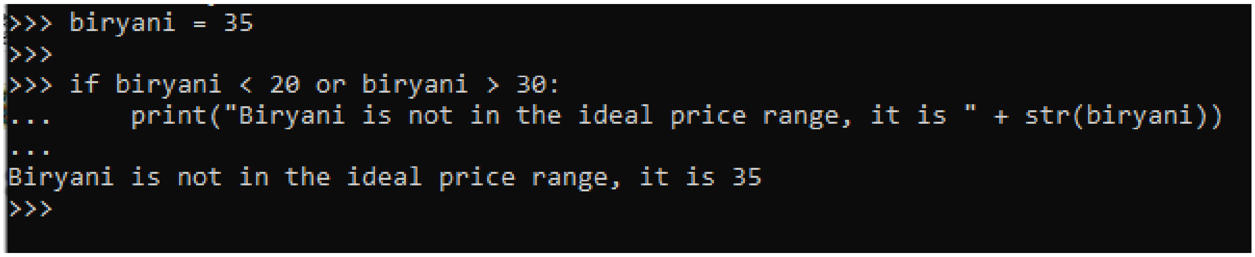
## Logical operators



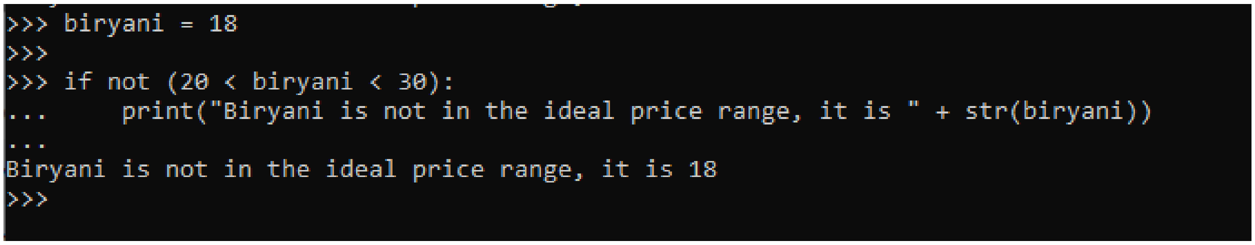
| Operator | Description | Example |
|---|---|---|
| and | Returns True if both statements are true | x < 5 and  x < 10 |
| or | Returns True if one of the statements is true | x < 5 or x < 4 |
| not | Reverse the result, returns False if the result is true | not(x < 5 and x < 10) |



```
>>> biryani = 25
>>>
>>> if biryani < 30 and biryani > 20:
...     print("Let's eat biryani, it is " + str(biryani))
...
Let's eat biryani, it is 25
>>>
```



```
>>> biryani = 35
>>>
>>> if biryani < 20 or biryani > 30:
...     print("Biryani is not in the ideal price range, it is " + str(biryani))
...
Biryani is not in the ideal price range, it is 35
>>>
```



```
>>> biryani = 18
>>>
>>> if not (20 < biryani < 30):
...     print("Biryani is not in the ideal price range, it is " + str(biryani))
...
Biryani is not in the ideal price range, it is 18
>>>
```