User Input, Numbers, and Operators

User input and output

When you are about to open or execute a program, file or software, you either double-click, click and press enter, look it up at the Start menu and open it from their, etc. Pointing with the mouse and doing the actions that are mentioned is the first thing you are taught on how to use a computer, but we never pondered why does that happen? Well what you did there is just you, the user doing a specific input on the computer.

An **input** is any data or information that is sent to the computer to be processed, to send data or information to the computer what you need is an input device. **Input devices are** hardware[1] devices allowing you to interact with the computer, most common and used input devices are the mouse and keyboard.

The mouse and keyboard, one of the most common input devices

After you've put an input in the computer, that data or is gonna be processed, any data that is sent back to the user is called an **output.** But in order to see the output you need an output device. **Output devices** are hardware that sends data to other devices or users. What you are seeing right now on the screen is the output and

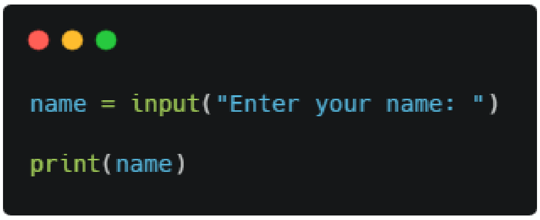
[1] Hardware is any physical component of a computer

the screen you are using right now called the monitor is the output device.

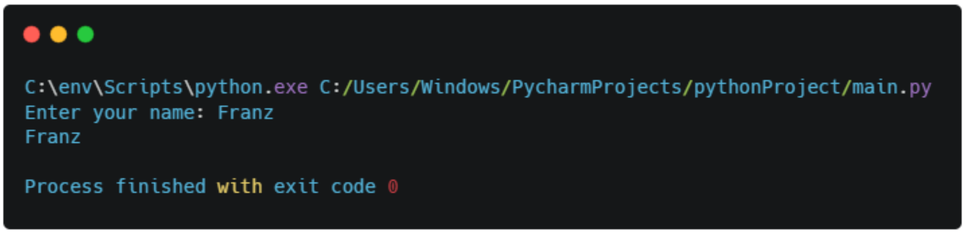
When creating a program or software, there are times that a program needs the user to get an input, like entering the activation key to activate Windows, a video game tutorial prompting you to press a button to do a certain action in order to continue etc. But in this lesson we will be only focusing on command in the Python console.

input()

input() is the command for Python to get the users input, Python stops executing until the user has entered in an input; Inside the parentheses known as parameter holders is where you can put a variable or string. When you use an input, it does not automatically appear on the screen, for it to show up you must use the command **print()**.



```
name = input("Enter your name: ")

print(name)
```

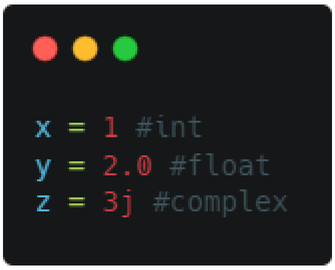


```
C:\env\Scripts\python.exe C:/Users/Windows/PycharmProjects/pythonProject/main.py
Enter your name: Franz
Franz

Process finished with exit code 0
```

We can use input in different types of scenarios, with using numbers and operators which I will be discussing next.

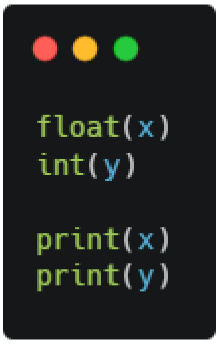
Numbers

Other than integer there are other data types that represent a number, **float** is a numeric data type either positive or negative that contains one or more decimal values, **complex** is a numeric data type to represent an imaginary value.

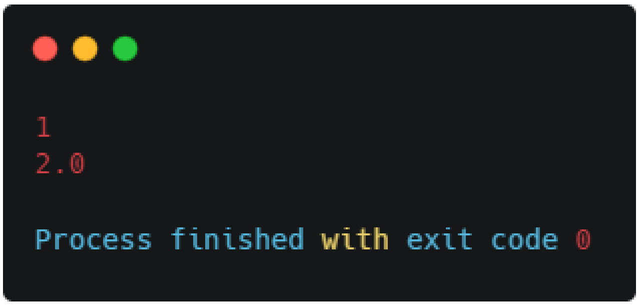


```
x = 1 #int
y = 2.0 #float
z = 3j #complex
```

We can convert integer to a float and complex, we can convert a float to an integer and complex but we cannot convert a complex to either data types.

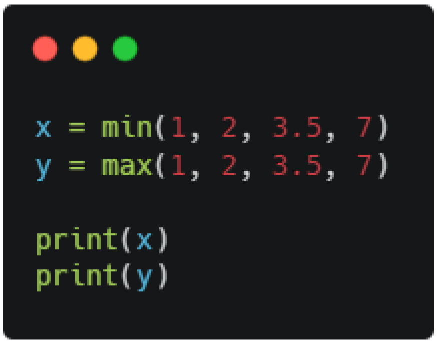


```
float(x)
int(y)

print(x)
print(y)
```

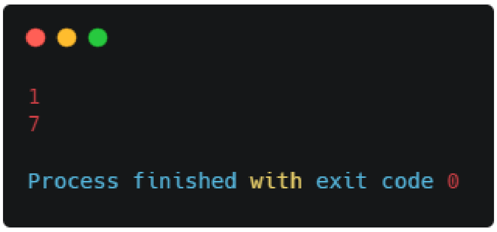


```
1
2.0

Process finished with exit code 0
```

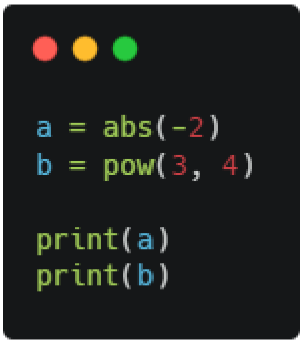
Python has a built-in math functions and a math module that the user can use to do math. **min()** gets the lowest value while **max()** is the opposite, it gets the highest value, **abs()** gives us the absolute value of a specified number and **pow()** where it gives us the value of number and raised to the power of a number, where the first argument is the number and the second argument is the exponent.



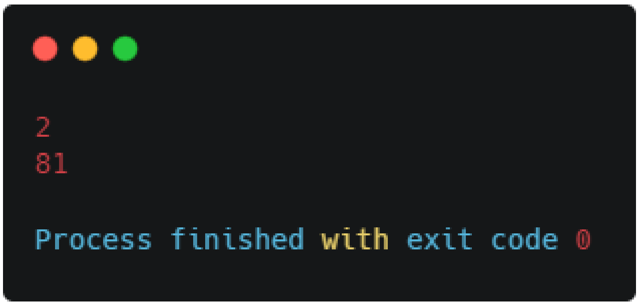
```python
x = min(1, 2, 3.5, 7)
y = max(1, 2, 3.5, 7)

print(x)
print(y)
```



```
1
7

Process finished with exit code 0
```



```python
a = abs(-2)
b = pow(3, 4)

print(a)
print(b)
```

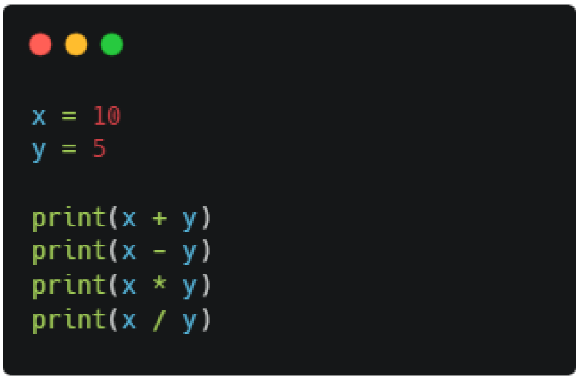


```
2
81

Process finished with exit code 0
```

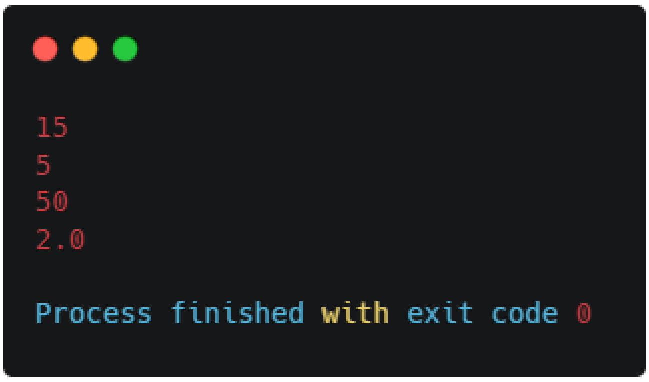
Operators

Operators are used when we need to perform a specific action to variables and values, There are many types of operators but we are only focusing on arithmetic, assignment, comparison and logical operators. **Arithmetic operators** are operators that you use to do basic math.

Operator	Operation
+	Addition
-	Subtraction
*	Multiplication
**	Exponentiation
/	Division
//	Floor Division
%	Modulus



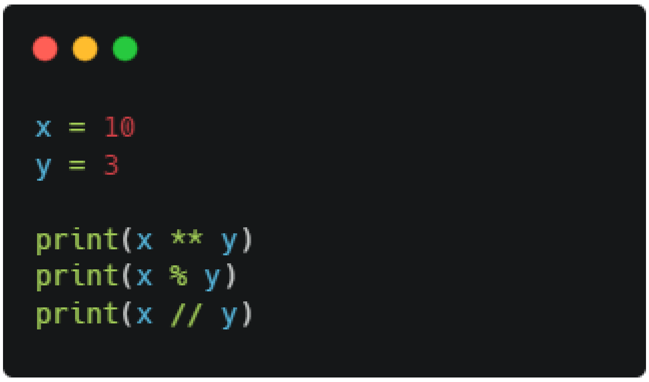
```
x = 10
y = 5

print(x + y)
print(x - y)
print(x * y)
print(x / y)
```



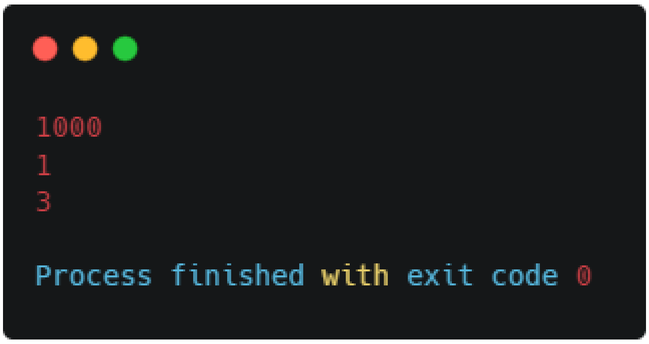
```
15
5
50
2.0

Process finished with exit code 0
```

Note: When dividing values it will always return a float.

Modulus divides both values but gets the remainder, **floor division** divides both values rounds it down to the nearest whole number and **exponentiation** is just a short way of writing how many times is a number gonna be multiplied by itself.



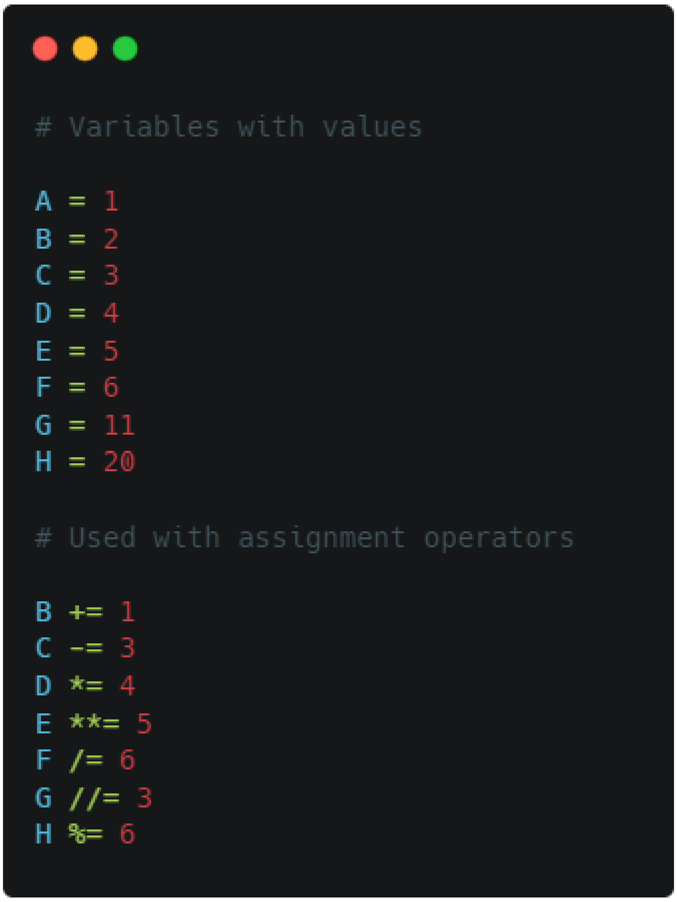
```
x = 10
y = 3

print(x ** y)
print(x % y)
print(x // y)
```



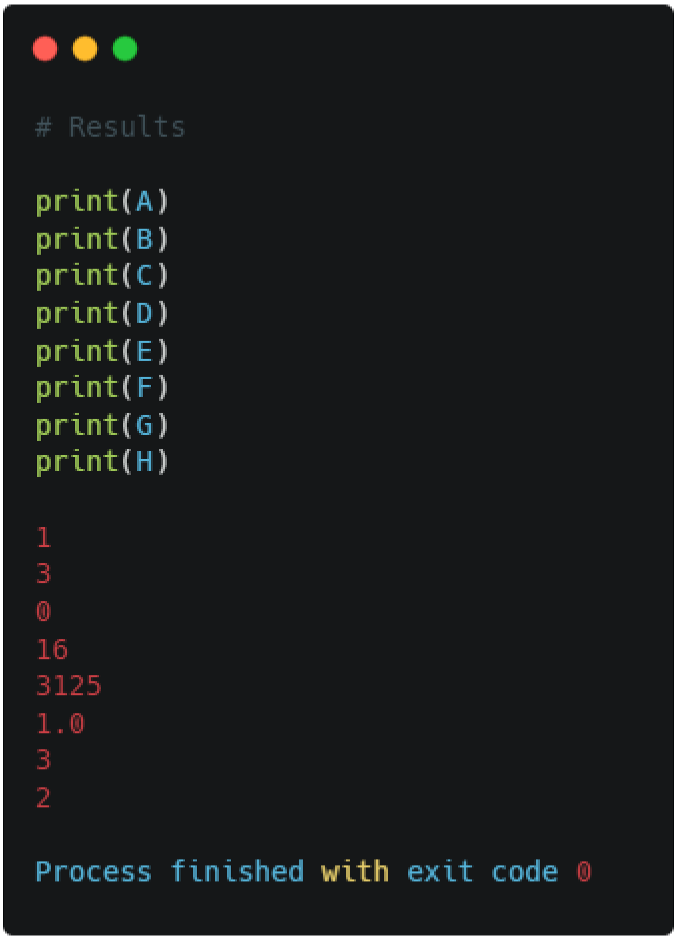
```
1000
1
3

Process finished with exit code 0
```

Assignment operators are operators that assign any value to a variable. If a variable already contains a numeric data (i.e. an **int**, **float** or **complex**) it will do the following:

Operator	Example	Similar Syntax
=	C = 8	None
+=	A += 3	A = A + 3
-=	D -= 4	D = D - 4
*=	B *= 2	B = B * 2
**=	E **= 0	E = E**0
/=	F /= 2	F = F / 2
//=	H //= 3	H = H // 2
%=	G %= 3	G = G % 5



```
# Variables with values

A = 1
B = 2
C = 3
D = 4
E = 5
F = 6
G = 11
H = 20

# Used with assignment operators

B += 1
C -= 3
D *= 4
E **= 5
F /= 6
G //= 3
H %= 6
```



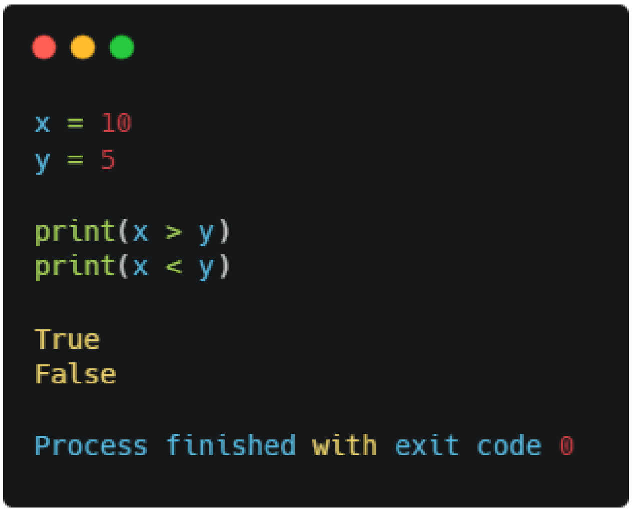
```
# Results

print(A)
print(B)
print(C)
print(D)
print(E)
print(F)
print(G)
print(H)

1
3
0
16
3125
1.0
3
2

Process finished with exit code 0
```

Comparison operators from the word itself “comparison” are used when we want to compare two values. It can only compare numeric data and variables that contain them.

Operator	Definition	Example
==	Known as “is equal” it compares if both values/variables are the same	X == Y
!=	Known as “not equal” it compares if one of the value/variable is not the same	B != F
>	Known as “greater than” it compares	2 > 1

	if the first value/variable has a larger value than the second.	
<	Known as “less than” it compares if the second value/variable is larger than the first value/variable.	4 < 5
>=	Known as “greater than or equal to” it compares if the first value/variable is greater than or equal to the second value/variable.	C >= 12
<=	Known as “less than or equal to” it compares if the first value/variable is less than or equal to the second value/variable.	K <= 32

After we compare the values based on what operator we used if both values satisfy the condition it will return **True**, if not it will return **False**.

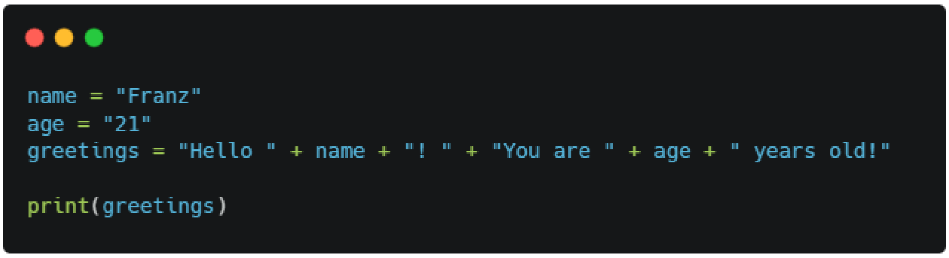


```
x = 10
y = 5

print(x > y)
print(x < y)

True
False

Process finished with exit code 0
```

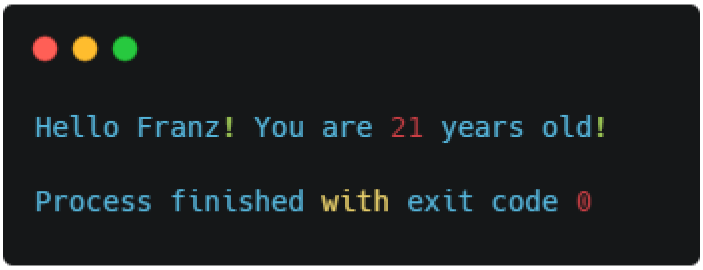
Lastly, **Logical operators** are operators used when we need to combine conditions. Conditional statements will be further discussed on the next lesson.

Operator	Definition
and	If both conditions are **True**, return **True**. Otherwise return **False**.
or	If one condition is **True**, return **True**. Otherwise return **False**.
not	Reverses the result; returns **False** if the condition is **True**.

Other than the operations mentioned earlier, we can use the "+" operator to concatenate. **Concatenation**, in terms of programming is combining multiple strings.

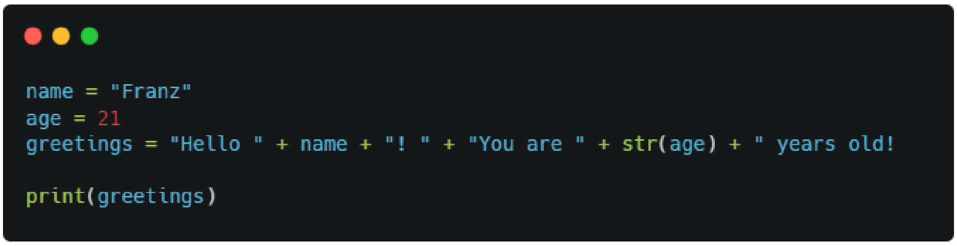


```
name = "Franz"
age = "21"
greetings = "Hello " + name + "! " + "You are " + age + " years old!"

print(greetings)
```



```
Hello Franz! You are 21 years old!

Process finished with exit code 0
```

Notice that the variable age is a **String** and not any numeric data type? It's because you cannot concatenate a **String** to an **Integer**, it's because it is trying to do an arithmetic operator to it. To concatenate an **Integer** or any numeric data, you need to convert it to a **String**; to convert it to a String you need to use the **string()** command.



```
name = "Franz"
age = 21
greetings = "Hello " + name + "! " + "You are " + str(age) + " years old!

print(greetings)
```

Sources

- BBC. (2021, October 28). *What are input and output devices?* BBC Bitesize. Retrieved December 17, 2021, from https://www.bbc.co.uk/bitesize/topics/zf2f9j6/articles/zx8hpv4
- Computer Hope. (2021, July 6). *What is Input?* Retrieved December 17, 2021, from https://www.computerhope.com/jargon/i/input.htm
- Computer Hope. (2021b, August 16). *What is an Input Device?* Retrieved December 17, 2021, from https://www.computerhope.com/jargon/i/inputdev.htm
- Techopedia. (2020, April 23). *Output Device*. Techopedia.Com. Retrieved December 17, 2021, from https://www.techopedia.com/definition/3538/output-device
- *Python Numbers*. (n.d.). W3Schools Online Web Tutorial. Retrieved December 17, 2021, from https://www.w3schools.com/python/python_numbers.asp
- *Python Operators*. (n.d.). W3Schools Online Web Tutorial. Retrieved December 17, 2021, from https://www.w3schools.com/python/python_operators.asp