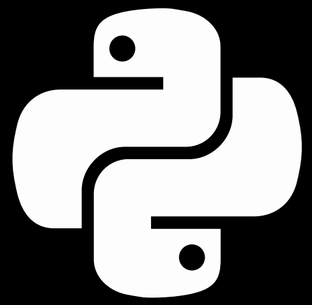


Python Basics

Python Basics

- Data types
- Variables
- Expression
- String operations

Data types

Data types

- Numbers
- Strings
- Boolean

Numbers

10

11.2

2 + 5j

```
10        --------> integer

11.2      --------> Float

2 + 5j    --------> Complex number
```

Number type conversion

int(5.32) → 5

float(11) → 11.0

complex(10) → 10 + 0j

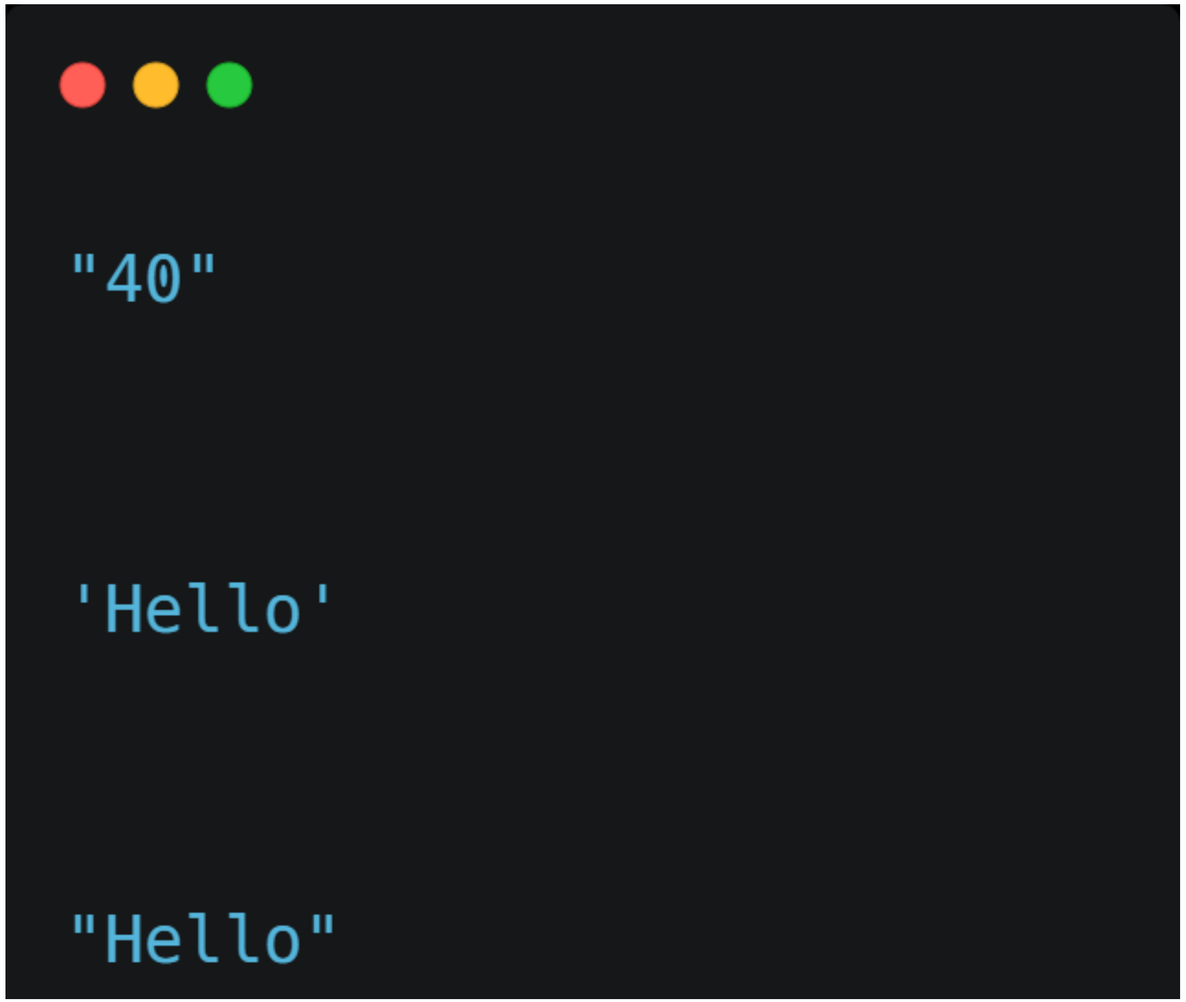
```
int(5.32)


float(11)


complex(10)
```

String

"40"

'Hello'

"Hello"

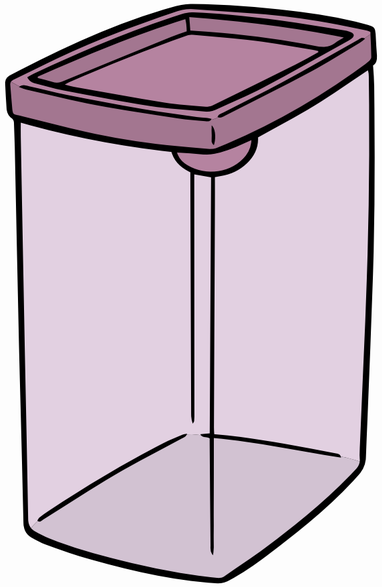


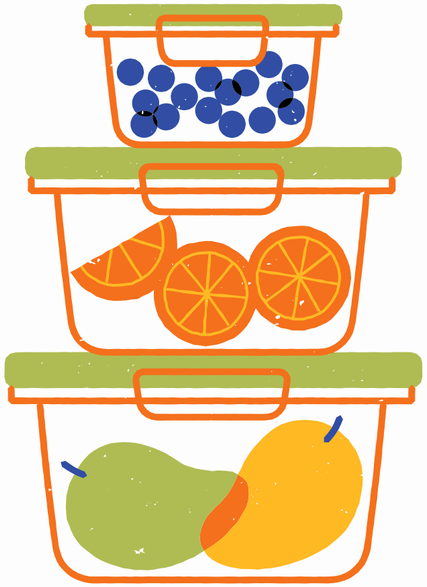
Boolean

```
True



False
```

Variables

Variables





Variables are containers for storing data values

variables

```
x = 5

y = "Hello"

z = True
```

- To create a variable, you simply need to assign a value to a name using the assignment operator (=)

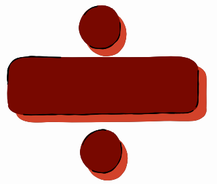
Rules for python variable

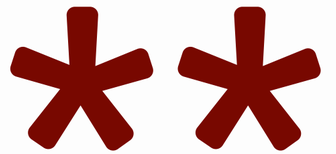
- variable names in Python have to start with a letter or an underscore
- They cannot start with a number
- They can't contain any special characters other than an underscore
- Variable names are case-sensitive (age, Age and AGE are three different variables)

Expressions

Arithmetic expression

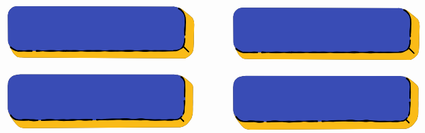
```
2 + 3      ----------> addition

43 - 25    ----------> subtraction

6 * 5      ----------> multiplication

120 / 3    ----------> division

10 % 3     ----------> modulus

2 ** 3     ----------> exponentiation

10 // 3    ----------> floor division
```

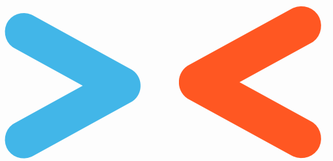




Relational expression

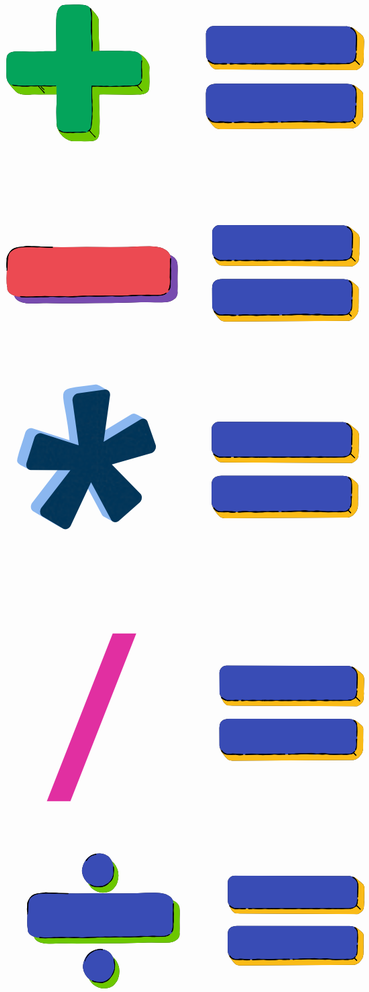
```
5 == 5     ------>  Equal to

5 != 5     ------>  Not equal to

5 > 3      ------>  Greater than

5 < 3      ------>  Less than

5 >= 5     ------>  Greater than or equal to

5 <= 3     ------>  Less than or equal to
```





Assignment expression

```
x = x + 1    ------>  x += 1

x = x - 1    ------>  x -= 1

x = x * 2    ------>  x *= 2

x = x / 3    ------>  x /= 3

x = x % 2    ------>  x %= 2
```



Logical expression

or

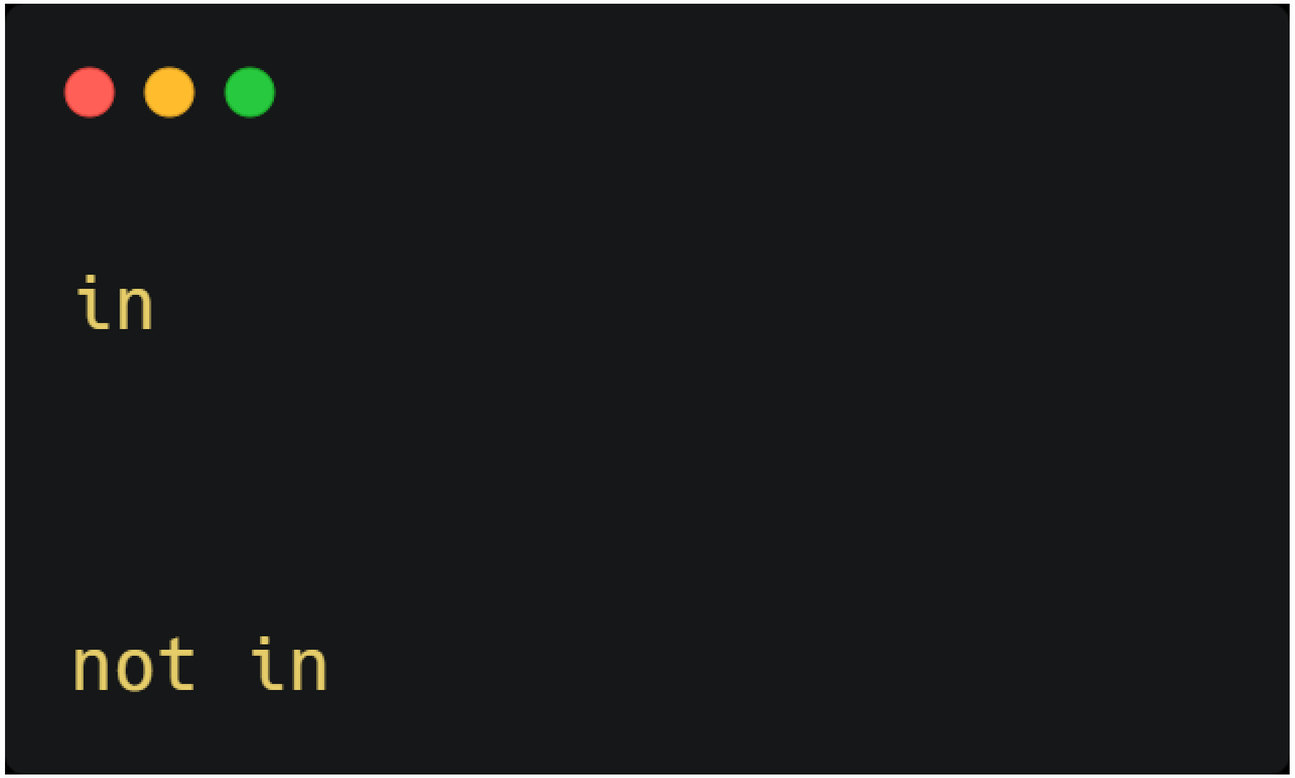
and

not

X	Y	X OR Y	X AND Y	NOT X
true	true	true	true	false
true	false	true	false	false
false	true	true	false	true
false	false	false	false	true

Identity expression

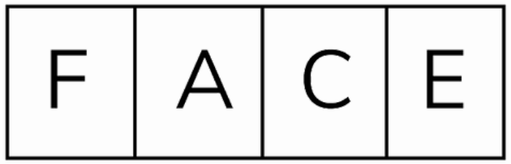
Membrership expression



String operations

Length of a string

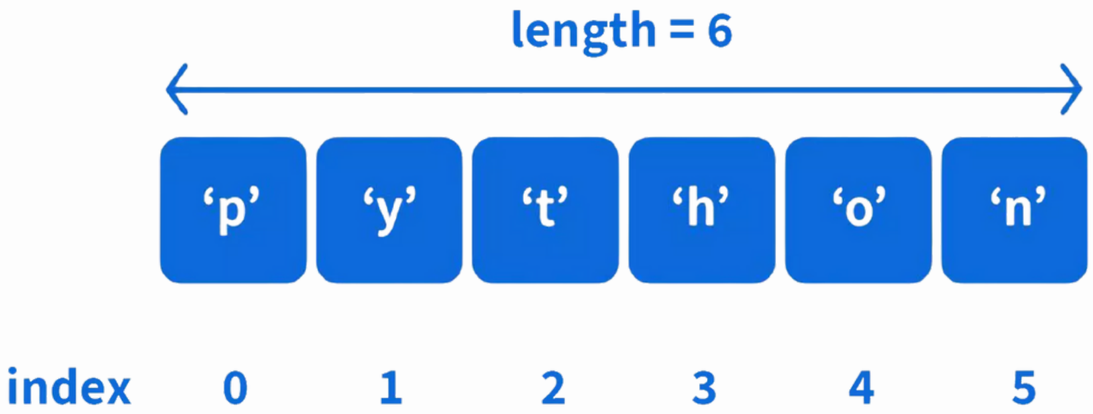
String ⇨



Length = 4

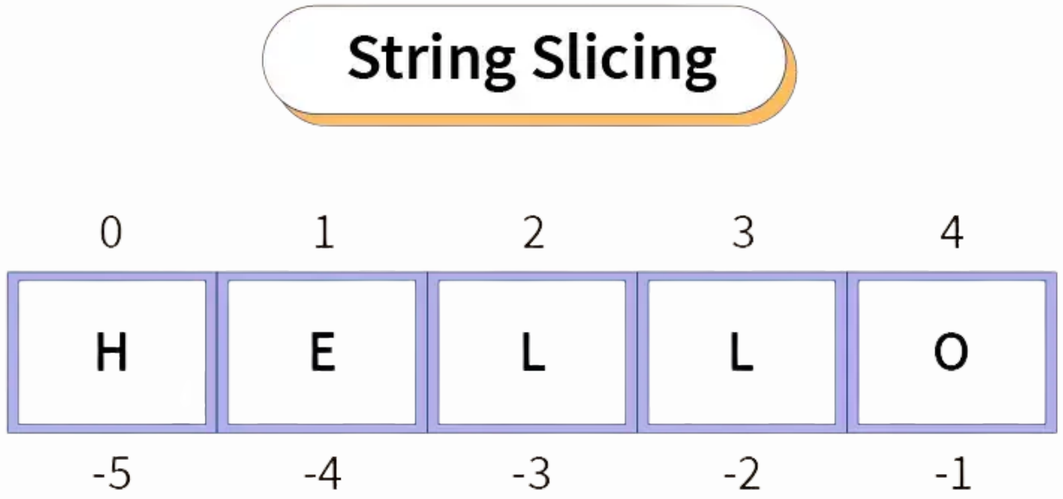
```
x = "Hello"


print(len(x))
```

Indexing



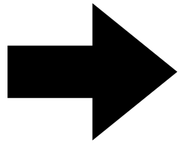

```
x = "Hello"


print(x[0])
```

string slicing




```
x = "Hello"

print(x[0:3])

print(x[:3])

print(x[2:])

print(x[:])
```

string concatenation

"Hello"  "world"  "Hello world"

```
x = "Hello"

y = "World"

print(x + y)
```

string format

"His age is" + 20

```
age = 20

print(f"His age is {age}")
```

- we cannot combine strings and numbers
- But we can combine strings and numbers by using the format method
- placeholders should be created with {}

Find the index

```
x = "Hello everyone, welcome to this class"


y = x.index("w")


print(y)
```

- This will return the index position of the specified string

Count

```
x = "Hello Hello Hello Hello"


y = x.count("Hello")


print(y)
```

- This will return the number of occurances of specified string

Capitalize

```
x = "hello world"


print(x.capitalize())
```

- The first character is converted to upper case, and the rest are converted to lower case

Upper

```
x = "hello world"


print(x.upper())
```

- It converts all characters present in the string into upper case

Lower

```
x = "Hello WORLD"


print(x.lower())
```

- It converts all characters present in the string into lower case

Replace

```
x = "Welcome to python course"


y = x.replace("python", "data science")


print(y)
```

- This method replaces a specified string with another specified string.

Replace

```
x = "Hello Hello Hello Hello"


y = x.replace("Hello", "Hi", 2)


print(y)
```

- The count can also be mentioned to specify how many occurrences of the old value you want to replace.

Split

```
x = "Welcome to this course"


y = x.split()


print(y)
```

- This will split each word into separate strings

Split

```
x = "Hi, Hello everyone, Welcome to this course"


y = x.split(", ")


print(y)
```

- You can also specify where to split

Center

```
x = "Hello world"


y = x.center(30)


print(y)
```

- This method will center allign the string based on the specified length

Find

```
x = "Hello everyone, welcome to this class"


y = x.find("e")


print(y)
```

- It finds the index of first occurance of specified string

Find

```
x = "Hello everyone, welcome to this class"


y = x.find("e", 10, 20)


print(y)
```

- You can also metion between the starting and ending index

rFind

```
x = "Hello everyone, welcome to this class"


y = x.rfind("e")


print(y)
```

- rfind is similar to find, but rfind searches from right side

Strip

```
x = ",,,,..    Hello  ...,.,. "


y = x.strip(".,  ")


print(y)
```

- This will remove the specified string in starting and ending position