NETWORKZ SYSTEMS
AN ISO 9001:2015 CERTIFIED COMPANY
CHINNAKADA , KOLLAM

INTRODUCTION TO DATATYPES

Python Numbers

Numbers in Python can be of 3 types: int , float and complex

```
age = 8
```

```
fraction = 0.1
```

```
complexNumber = 2+3j
```

Python Strings

String literals in python are surrounded by either single quotation marks, or double quotation marks

hello' is the same as "hello". You can display a string literal with the print() function:

```
name = "Roger"
```

Assign String to a Variable

Assigning a string to a variable is done with the variable name followed by an equal sign and the string

Example

```
a = "Hello"
print(a)
```

Multiline Strings

You can assign a multiline string to a variable by using three quotes

Example

You can use three double quotes:

```
a = """Lorem ipsum dolor sit amet,
        consectetur adipiscing elit, sed do
      eiusmod tempor incididunt ut labore et dolore magna aliqua."""
print(a)
```

Strings are Arrays

Get the character at position 1 (remember that the first character has the position 0):

```
message="hello world!"
print(message[0])
```

```
h
```

Example 2
Get the characters from position 2 to position 5 (not included):
b = "Hello, World!"
print(b[2:5])

Negative indexing of a string

```
message="hello world!"
print(message[-1])
```

```
!
```

Reversing a String

```
a=input("Enter a String: ")
print(a[::-1])
```

The len() method returns the length of a string:

```
a = "Hello, World!"
print(len(a))
```

The lower() method returns the string in lower case:

```
a = "Hello, World!"
print(a.lower())
```

The upper() method returns the string in upper case:

```
a = "Hello, World!"
print(a.upper())
```

To concatenate:

```
string1="Hello"
string2="World"
new_string = string1 +" "+ string2
print(new_string)
```

String Format

we cannot combine strings and numbers like this:

Example

```
age = 36
txt = "My name is John, I am " + age
print(txt)
```

But we can combine strings and numbers by using the format() method!

Example 1

```
age=36
txt="My name is John, and I am {}"
print(txt.format(age))
```

Example 2

quantity=7
itemno=567
price=49.95
myorder ="I want {} pieces of item {} for {} dollars."
print(myorder.format(quantity,itemno,price))

```
txt = "Hello {word}"
print(txt.format(word = 'World!'))

message1 = 'Hi, My name is {} and I am {} years old.'
print(message1.format('Bob', 36))

message2 = 'Hi, My name is {name} and I am {number} years old.'
print(message2.format(name ='Bob', number = 36))
```

Read value from a User

Example 1

```
a=input(“Enter your message”)
#Hello World
print(a)
```

Example 2

```
a=10
b=int(input(‘Enter any number’))
print(a+b)
```

Python Operators

Operators are used to perform operations on variables and values.
Python divides the operators in the following groups:

Arithmetic operators
Assignment operators
Comparison operators
Logical operators
Identity operators
Membership 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

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

Comparison operators

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)

`is` and `in`

`is` is called the **identity operator**. It is used to compare two objects and returns true if both are the same object. More on objects later.

`in` is called the **membership operator**. Is used to tell if a value is contained in a list, or another sequence. More on lists and other sequences later.

Python Collections (Arrays)

There are four collection data types in the Python programming language:

List is a collection which is ordered and changeable. Allows duplicate members.

Tuple is a collection which is ordered and unchangeable. Allows duplicate members.

Set is a collection which is unordered and unindexed. No duplicate members.

Dictionary is a collection which is unordered, changeable and indexed. No duplicate members

Lists

Create a List

```
thislist = ["apple", "banana", "cherry"]
print(thislist)
```

Access Items

```
thislist = ["apple", "banana", "cherry"]
print(thislist[1])
```

```
thislist = ["apple", "banana", "cherry"]
thislist.append("orange“)
print(thislist)
```

```
thislist = ["apple", "banana", "cherry"]
thislist.insert(1, "orange")
print(thislist)
```

```
thislist = ["apple", "banana", "cherry"]
thislist.remove("banana")
print(thislist)
```

```
thislist = ["apple", "banana", "cherry"]
thislist.pop()#index value optional
print(thislist)
```

```
thislist = ["apple", "banana", "cherry"]
del thislist[0]
print(thislist)
```

```
thislist = ["apple", "banana", "cherry"]
thislist.clear()
print(thislist)
```

```
thislist = ["apple", "banana", "cherry"]
mylist = thislist.copy()
print(mylist)
```

Change Item Value :

```
thislist = ["apple", "banana", "cherry"]
thislist[1] = "blackcurrant“ #replace
print(thislist)
```

Python List count()

The count() method returns the number of occurrences of an element in a list.

```
k = [‘L', ‘M', ‘N']
x = k.count(“M")
```

OR

```
points = [1, 4, 2, 9, 7, 8, 9, 3, 1]
x = points.count(9)
```

Python Tuples

A tuple is a collection which is ordered and unchangeable. In Python tuples are written with round brackets.

Example

```
thistuple = ("apple", "banana", "cherry")
print(thistuple)
print(thistuple[2])     o/p: cherry
print(thistuple[-1])    o/p: cherry
```

```
# Printing the each value in a tuple using both positive and negative indexing
tuple_1 = ('Hello', 'Python', 3.14, 1.618, True, False, 32, [1,2,3], {1,2,3}, {'A': 3, 'B': 8}, (0, 1))
print(tuple_1[0])
print(tuple_1[1])
print(tuple_1[2])
print(tuple_1[-1])
print(tuple_1[-2])
print(tuple_1[-3])
```

```
Hello
Python
3.14
(0, 1)
{'A': 3, 'B': 8}
{1, 2, 3}
```

Sets in Python

A set is a collection which is unordered, unindexed and unchangeable.
In Python sets are written with curly brackets.

Example Create a Set:

```
thisset = {"apple", "banana", "cherry"}
print(thisset)
```

Once a set is created, you cannot change its items, but you can add new items

To add one item to a set use the add() method.

```
thisset = {"apple", "banana", "cherry"}
thisset.add("orange")
print(thisset)
```

To add more than one item to a set use the update() method .

```
thisset = {"apple", "banana", "cherry"}
thisset.update(["orange", "mango", "grapes"])
print(thisset)
```

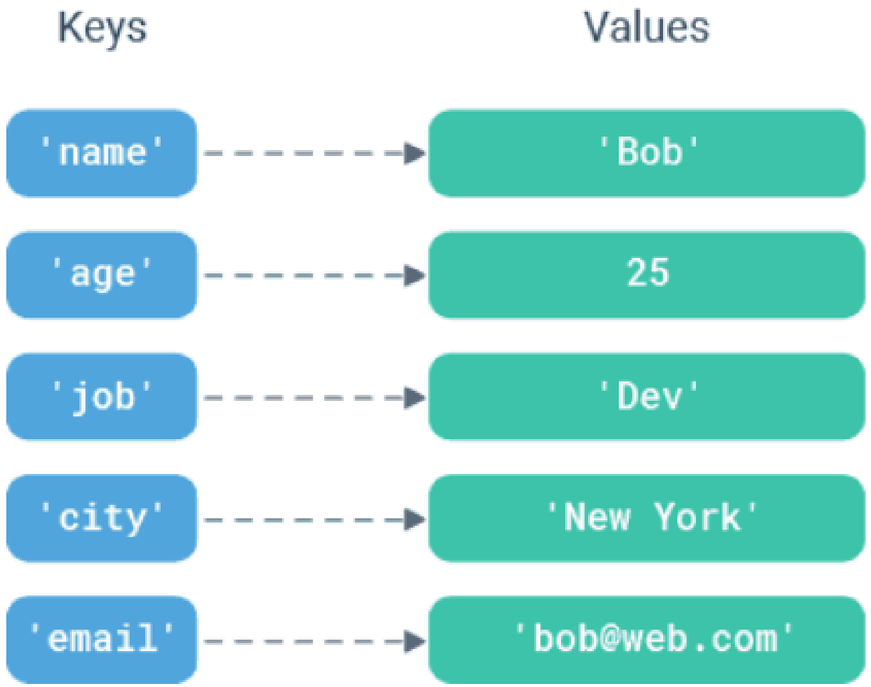
```
thisset = {"apple", "banana", "cherry"}
thisset.remove("banana")
print(thisset)
```

```
thisset = {"apple", "banana", "cherry"}
x = thisset.pop()
print(x)
print(thisset)
```

```
thisset = {"apple", "banana", "cherry"}
thisset.clear()
print(thisset)
```

Python Dictionaries

A dictionary is a collection which is unordered, changeable and indexed.
In Python dictionaries are written with curly brackets, and they have keys and values.



```
dog = { 'name': 'Roger', 'age': 8 }
```

thisdict = {
"brand": "Ford",
"model": "Mustang",
"year": 1964
}
print(thisdict)

You can access the items of a dictionary by referring to its key name, inside round brackets.
Get the value of the "model" key:

```
thisdict = {
"brand": "Ford",
"model": "Mustang",
"year": 1964
}
x = thisdict.get("model")
print(x)
```

You can change the value of a specific item by referring to its key name

```
thisdict = {
"brand": "Ford",
"model": "Mustang",
"year": 1964
}
thisdict["year"] = 2022
print(thisdict)
```

```
thisdict = {
"brand": "Ford",
"model": "Mustang",
"year": 1964
}
thisdict["color"] = "red"
print(thisdict)
```

```
thisdict = {
"brand": "Ford",
"model": "Mustang",
"year": 1964
}
thisdict.pop("model")
print(thisdict)
```

THANK YOU

NETWORKZ SYSTEMS

AN ISO 9001:2015 CERTIFIED COMPANY

CHINNAKADA , KOLLAM