

CHP 02 "Python data Types"

1- Python Numbers :- The number data type are used to store the numeric value. Python supports integers, float, complex numbers.

1.1-Type conversion :- The process of converting one data type into another is called type conversion. There are two types of type conversion.

i)- Implicit type conversion

ii)- Explicit type conversion

2- String :- A string is a sequence of characters. For example "Ayz" is a string containing a sequence of characters "A", "y", "A", "z". We use single or double quotes to create a string.

2.1. Access string characters

i)- indexing :- We can access Python

String through indexing. For ex:

a = 'Ayaž'

Print(a[1]) # "y"

ii)-Negative indexing: Python also allow negative indexing. For ex:

Print(a[-4]) # "A"

iii)-Slicing: Access a range of characters in a string by the slicing operator colon : . For ex:

a = "Ayaž"

Print(a[1:4]) # "yaž"

Python String are immutable

For ex:

gxcet = "Hello"

gxcet[0] = "A"

Print(gxcet)

This will show an error

2.2 - Python multiline string

We use triple double quotes
''' or single triple quotes.

For ex:-

message = "

Hi my name is

Ajaz Khan"

Print(message)

2.3 - Python string methods

There are so many methods of string. Some important are:

i) - Capitalize():- Convert the first character of the string to uppercase.

str1 = "hi"

Print(str1.capitalize())

ii) - Count():- Return a no of time specified value occurs. For ex:-

str1 = "hello"

Print(str1.count("l"))

iii) format() :- Returns the formatted version of the string. For ex:-

name = "Aje²z"

age = 21

Print ("my name is {} and
my age is {}".format(name, age))

iv) len() :- Returns the length of the string. For ex:-

greet = "Hello"

Print (len(greet))

v) upper() :- Convert the string to upper

greet = "Aje²z"

Print (greet.upper())

vi) lower() :- Convert the string to lower

greet = "Aje²z"

Print (greet.lower())

vii) split() :-

text = my name is Aje²z

Print (text.split())

(viii) startswith() :- Returns True if a string with a specified prefix (String)

message = 'Hi my name is Aje²z'

Print (message.startswith('Aje²z'))

3- List :- Lists are used to store multiple data at once.
With list we can do the same things like we did with string.
For ex:- Slicing, indexing and etc.

Python list are not immutable

```
list1 = [1, 2, 3]
```

```
list1[0] = 30
```

```
Print(list1)
```

```
# 30, 2, 3
```

3.1 - List methods :- These are so many list method. Some important are as following.

i) - append() : Append an element at the end of the list.

```
list = [1, 2, 3, 4]
```

```
list.append(5)
```

```
Print(list)
```

ii) - extend() : add two lists

```
list1 = [1, 2, 3]
```

```
list2 = [4, 5, 6]
```

list1.extend([list2])

Print(list1)

iii) - insert(): Insert an element into the list at a specific place

list = [1, 2, 3]

list.insert(1, 4)

index element

Print(list)

iv) - pop(): Remove and return the element at specified index.

list = [1, 2, 3]

element = list.pop()

Print(list) # [1, 3]

Print(element) # 2

v) - remove(): Remove the first occurrence of the specified element.

list = [1, 2, 2, 3]

list.remove(2)

Print(list) # [1, 2, 3]

vi) - sort(): Sort list in ascending order

list = [1, 3, 2, 5, 4]

list.sort()

Print(list) # [1, 2, 3, 4, 5]

4. Tuple : A tuple in Python

is a similar to list. The difference between in that we cannot change the element of a tuple once it is assigned.

Creating a Tuple

```
tup = (1, 2, 3)
```

```
print(tup)
```

Important Thing

```
var1 = ("Hello") # String
```

```
var2 = ("Hello",) # Tuple
```

We can do the same thing with tuple also like we did with string

4.1 - Tuple methods

i) - index () :

```
tup = (1, 2, 3)
```

```
ind = tup.index(2)
```

```
Print(ind)
```

Output
1

ii)- Count :-

tub = (1, 2, 2, 3)

Print(tub.count(2)) #

5- Sets

i)- Sets are unordered collections of objects.

ii)- Sets are unchangeable.

iii)- Sets contain different type of sets + types.

Empty Set

a = Set()

Print(a)

Creating a Set

Student_marks = [50, 90, 70, 40]

Print(Student_marks)

Sets does not contain any of the duplicate items.

#01 union

Set1 = {1, 2, 4}

Set2 = {3, 5, 6}

Print(Set1.union(Set2))

Output: 1, 2, 3, 4, 5, 6

#02 update

a = S1.update(S2)

Print(a)

The value of S2 will come into Set1 or S1 so the update S1 will be {1, 2, 3, 4, 5, 6}

#03 intersection

Cities = {"Pak", "Aus", "Eng"}

Scities = {"Pak", "india", "Aus"}

Print(Cities.intersection(scities))

Output: Pak, Aus

#04 update

a = cities.intersection.update(scities)

Print(a)

Output: "Pak", "Aus"

#05 Symmetric

In symmetric method those sets will be print who are not common.

$$\text{Group A} = \{\text{"Park"}, \text{"Ban"}, \text{"ind"}\}$$

$$\text{Group B} = \{\text{"Ban"}, \text{"Aus"}, \text{"Eng"}\}$$

Output:- Park, ind, Aus, Eng

#06 Difference

The elements will print who are just in Set A but also element must be not common.

$$\text{Set 1} = \{1, 2, 3\}$$

$$\text{Set 2} = \{2, 3, 4\}$$

Output :- 1

#07 isdisjoint

$$x\$ = \text{Set 1} \text{.isdisjoint(Set 2)}$$

Print (x\$)

Output :- False

Bcz 2 and 3 is common in disjoint set there is no relationship

#08 issubset

$$\text{Set 1} = \{1, 2, 3\}$$

$$\text{Set 2} = \{2, 3\}$$

`Print (Set1 . issubset (Set2))`

Output :- True

A SubSet is a set that contains all elements of set b plus some additional elements too.

09. isSubSet

`SetA = {1, 2, 3}`

`SetB = {1, 2, 3, 4, 5}`

`Print (SetA . isSubSet (SetB))`

Output :- True

10. add

`SetA.add (32)`

`Print (SetA)`

Output :- 1, 2, 3, 32

11. Remove / discard

The main difference b/w is remove will through error but discard will not.

`SetA.remove (1)`

`SetA.remove (23)` # This will through error

`SetA.discard (23)` # This will not

#12. Pop

```
city = {"Pak", "ind", "Tokyo")
```

```
item = city.pop()
```

```
print(city)
```

```
print(item)
```

Random element will remove.

13. del

```
del city #del whole set
```

#14. clear

```
set1 = {1, 2, 3}
```

```
xcl = set1.clear()
```

```
print(xcl)
```

Output: set()

The whole set will be clear

Creating Dict

```
items = { '1': "AYAZ", '2': "Ali", '3':  
"Ahmed", '4': "Hassan"}
```

1, 2, 3 and 4 are the keys
and "AYAZ" ... "Hassan" are the values.

Printing keys

```
Print(items.keys())
```

This will Print the keys

Printing values

```
Print(items.values())
```

This will Print the values

Change a value

```
items['1'] = "KHAN"
```

This will change the value of
first key.

Removing a value

```
del items['Pakistan']  
Ayaz
```

```
Print(items)
```

This will remove Pakistan, Ayaz

Output Ali, Ahmed, Hassan

item

```
it = { 'Fruits': "Apple", 'Food':  
"Rice", 'Milkshakes': "BananaShake"}
```

`Print (it.items())`

Output :- (Fruit, Apple), (Milkshake,
BananaShake), (Food, Kachhi)

Printing values and keys

for key,value in it.items():

`Print (f"The value corresponding
to {key} is {value}")`

Output:- The value corresponding
to Fruit is Apples

The value corresponding to
Milkshake is Banana shake

The value corresponding to
Food is Kachhi

"Python dictionary is ordered
and mutable"

1. update()

eP1 = {1: 69, 2: 72, 3: 84}

eP2 = {4: 89, 5: 72}

eP1.update(eP2)

Print("Updated", eP1)

2. clear()

numbers = {1: 69, 2: 72}

a = numbers.clear()

Print(a)

Output: - {}

3. copy()

This will create shallow copy

marks = {'Physics': 67, 'Computer': 80, 'Urdu': 79}

copy_marks = marks.copy()

4. Pop()

a = eP1.pop(3)

Print(eP1)

Print(a)

This will Pop key value

5. Popitem()

```
eP1 = { "1": "Id", "Salary": 35000,  
"name": "AVAI2" }
```

```
eP1.popitem()
```

```
Print(eP1)
```

The last item will automatically
removed.

#6. del

```
del eP1[0] del eP1[1]
```

Del whole dict also del item

7. get

```
marks = { 'Physics': 69, 'Urdu': 92 }
```

```
Print(marks.get('Physics'))
```

8. keys

```
m_keys = marks.keys()
```

This will get keys

9. values

```
m_values = marks.values()
```

```
Print(m_values.)
```