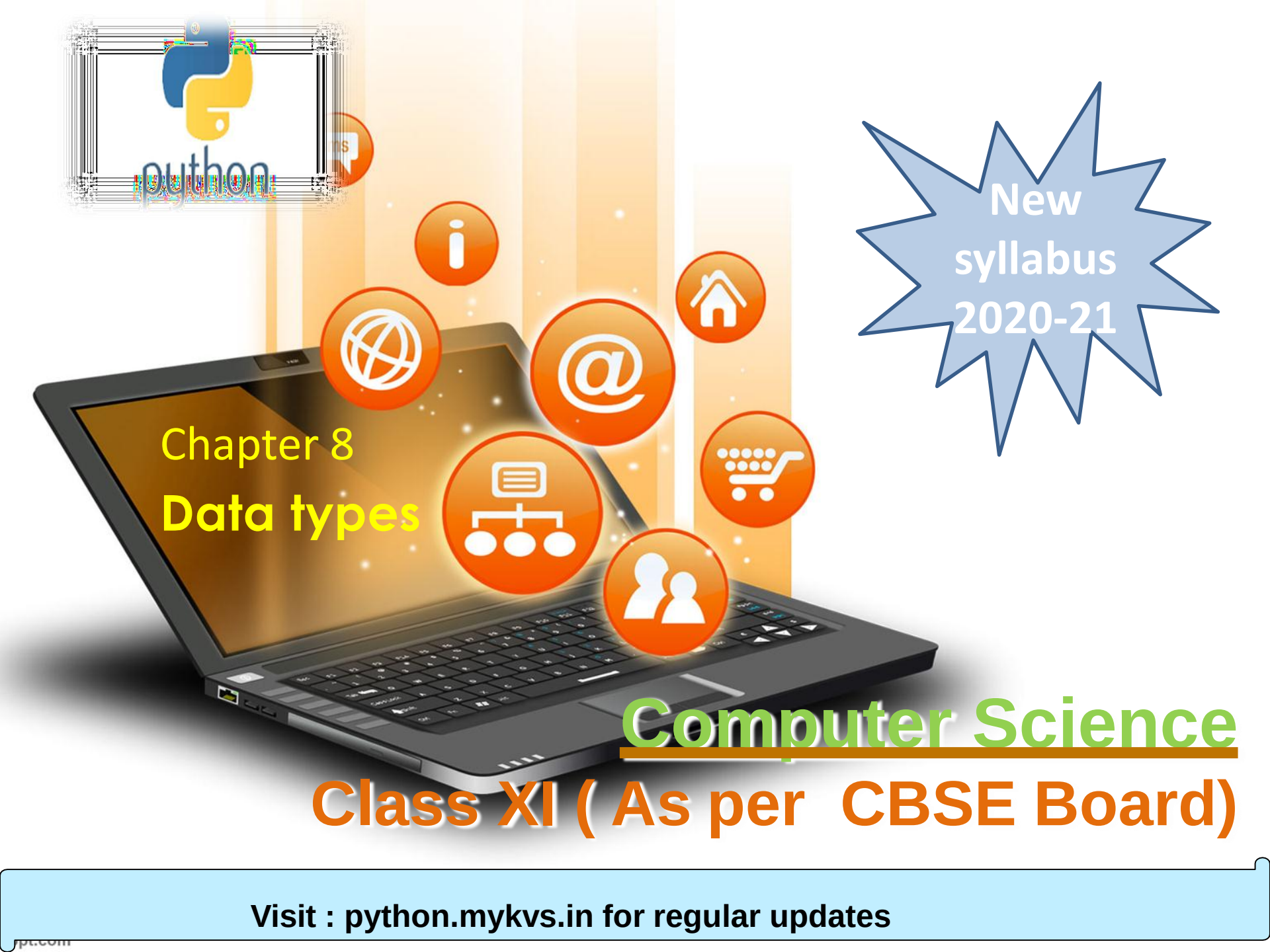

python
New syllabus 2020-21
Chapter 8
Data types
Computer Science
Class XI (As per CBSE Board)
Visit : python.mykvs.in for regular updates

Data handling

Most of the computer programming language support data type, variables,operator and expression like fundamentals.Python also support these.

Data Types

Data Type specifies which type of value a variable can store. type() function is used to determine a variable's type in Python.

Data type continue

Data Types In Python

1. Number
2. String
3. Boolean
4. List
5. Tuple
6. Set
7. Dictionary

Data type continue

Mutable and Immutable Data type

A mutable data type can change its state or contents and immutable data type cannot.

Mutable data type:

list, dict, set, byte array

Immutable data type:

int, float, complex, string, tuple, frozen set [note: immutable version of set], bytes

Mutability can be checked with id() method.

```
x=10
print(id(x))
x=20
print(id(x))
#id of both print statement is different as integer is immutable
```

Data type continue

1. Number In Python

It is used to store numeric values

Python has three numeric types:

1. Integers
2. Floating point numbers
3. Complex numbers.

Data type continue

1. Integers

Integers or int are positive or negative numbers with no decimal point. Integers in Python 3 are of unlimited size.

e.g.

```
a= 100
b= -100
c= 1*20
print(a)
print(b)
print(c)
```

Output :-

```
100
-100
200
```

Data type continue

Type Conversion of Integer

int() function converts any data type to integer.

e.g.

```
a = "101" # string
b=int(a) # converts string data type to integer.
c=int(122.4) # converts float data type to integer.
print(b)
print(c)Run Code
```

Output :-

101

122

Data type continue

2. Floating point numbers

It is a positive or negative real numbers with a decimal point.

e.g.

```
a = 101.2
b = -101.4
c = 111.23
d = 2.3*3
print(a)
print(b)
print(c)
print(d)Run Code
```

Output :-

```
101.2
-101.4
111.23
6.8999999999999995
```

Data type continue

Type Conversion of Floating point numbers

float() function converts any data type to floating point number.

e.g.

```
a='301.4' #string
b=float(a) #converts string data type to floating point number.
c=float(121) #converts integer data type to floating point number.
print(b)
print(c)Run Code
```

Output :-

```
301.4
121.0
```

Data type continue

3. Complex numbers

Complex numbers are combination of a real and imaginary part.Complex numbers are in the form of X+Yj, where X is a real part and Y is imaginary part.

e.g.

```
a = complex(5) # convert 5 to a real part val and zero imaginary part
print(a)
b=complex(101,23) #convert 101 with real part and 23 as imaginary part
print(b)Run Code
```

Output :-

```
(5+0j)
(101+23j)
```

Data type continue

2. String In Python

A string is a sequence of characters. In python we can create string using single (' ') or double quotes (" ").Both are same in python.

e.g.

```
str='computer science'
print('str-',   str) # print string
print('str[0]-',   str[0]) # print first char 'h'
print('str[1:3]-',   str[1:3]) # print string from postion 1 to 3 'ell'
print('str[3:]-',   str[3:]) # print string staring from 3rd char 'llo world'
print('str *2-',   str *2 ) # print string two times
print("str +'yes'-",   str +'yes') # concatenated string
```

Output

```
str- computer science
str[0]- c
str[1:3]- om
str[3:]- puter science
str *2- computer sciencecomputer science
str +'yes'- computer scienceyes
```

Data type continue

Iterating through string

e.g.

```
str='comp sc'
for i in str:
    print(i)
```

Output

```
c
o
m
p

s
c
```

Data type continue

3. Boolean In Python

It is used to store two possible values either true or false

e.g.

```
str="comp sc"
boo=str.isupper() # test if string contains upper case
print(boo)
```

Output

False

Data type continue

4.List In Python

List are collections of items and each item has its own index value.

5. Tuple In Python

List and tuple, objects mean you cannot modify the contents of a tuple once it is assigneboth are same except ,a list is mutable python objects and tuple is immutable Python objects. Immutable Python d.

e.g. of list

```
list =[6,9]
list[0]=55
print(list[0])
print(list[1])
```

OUTPUT

```
55
9
```

e.g. of tuple

```
tup=(66,99)
Tup[0]=3 # error message will be displayed
print(tup[0])
print(tup[1])
```

Data type continue

6. Set In Python

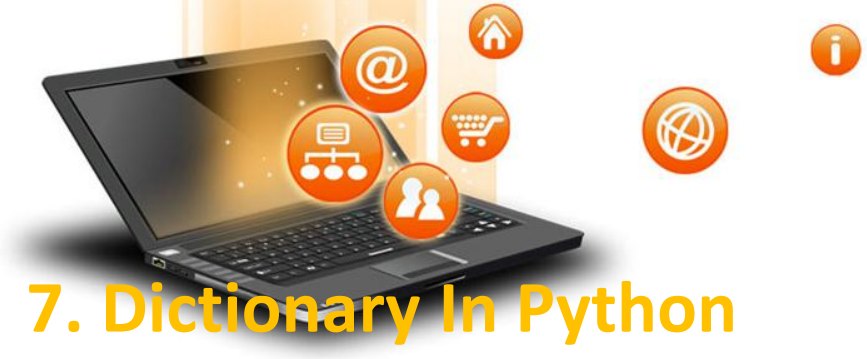
It is an unordered collection of unique and immutable (which cannot be modified)items.

e.g.

```
set1={11,22,33,22}
print(set1)
```

Output

{33, 11, 22}



Data type continue

7. Dictionary In Python

It is an unordered collection of items and each item consist of a key and a value.

e.g.

```
dict = {'Subject': 'comp sc', 'class': '11'}
print(dict)
print ("Subject : ", dict['Subject'])
print ("class : ", dict.get('class'))
```

Output

```
{'Subject': 'comp sc', 'class': '11'}
Subject :  comp sc
class :  11
```

Type conversion

The process of converting the value of one data type (integer, string, float, etc.) to another data type is called type conversion.

Python has two types of type conversion.

- Implicit Type Conversion
- Explicit Type Conversion

Implicit Type Conversion:

In Implicit type conversion, Python automatically converts one data type to another data type. This process doesn't need any user involvement.

e.g.

```
num_int = 12
num_flo = 10.23
num_new = num_int + num_flo
print("datatype of num_int:",type(num_int))
print("datatype of num_flo:",type(num_flo))
print("Value of num_new:",num_new)
print("datatype of num_new:",type(num_new))
```

OUTPUT

```
('datatype of num_int:', <type 'int'>)
('datatype of num_flo:', <type 'float'>)
('Value of num_new:', 22.23)
('datatype of num_new:', <type 'float'>)
```

Type conversion

Explicit Type Conversion:

In Explicit Type Conversion, users convert the data type of an object to required data type. We use the predefined functions like int(),float(),str() etc.

e.g.

```
num_int = 12
num_str = "45"
print("Data type of num_int:",type(num_int))
print("Data type of num_str before Type Casting:",type(num_str))
num_str = int(num_str)
print("Data type of num_str after Type Casting:",type(num_str))
num_sum = num_int + num_str
print("Sum of num_int and num_str:",num_sum)
print("Data type of the sum:",type(num_sum))
```

OUTPUT

```
('Data type of num_int:', <type 'int'>)
('Data type of num_str before Type Casting:', <type 'str'>)
('Data type of num_str after Type Casting:', <type 'int'>)
('Sum of num_int and num_str:', 57)
('Data type of the sum:', <type 'int'>)
```