Python

Python is a programming language

```
print("Hello World")
```

```
Hello World
```

Addition of two numbers

You have to enter two numbes to get a sum

```
a = int(input("Enter a number "))
b = int(input("Enter another number "))
print(f"Sum={a+b}")
```

```
Enter a number 3
Enter another number 5
Sum=8
```

Sequesnce Data Types

- list
- String
- tuple
- set
- dictionary

List

```
l = [1,·2.3,·"Hello",·True,·[1,2,3]] # collection of heterogeniuos data

l[0]
```

```
1
```

```
l[3]
```

```
True
```

```
l1 = []
l1[0]·=·10
```

```
---------------------------------------------------------------------------
--
IndexError                                Traceback (most recent call
last)
<ipython-input-7-3ad20ed43d36> in <module>
      1 l1 = []
----> 2 l1[0] = 10

IndexError: list assignment index out of range
```

```
dir(l)
```

```
['__add__',
 '__class__',
 '__contains__',
 '__delattr__',
 '__delitem__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__getitem__',
 '__gt__',
 '__hash__',
 '__iadd__',
 '__imul__',
 '__init__',
 '__init_subclass__',
 '__iter__',
 '__le__',
 '__len__',
 '__lt__',
 '__mul__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__reversed__',
 '__rmul__',
 '__setattr__',
 '__setitem__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
 'append',
 'clear',
 'copy',
 'count',
 'extend',
 'index',
 'insert',
 'pop',
 'remove',
 'reverse',
 'sort']
```

```
l.append(101)
```

```
l
```

```
[1, 2.3, 'Hello', True, [1, 2, 3], 101]
```

```
l[2]="Python"
```

```
l
```

```
[1, 2.3, 'Python', True, [1, 2, 3], 101]
```

```
# creating list dynamically
n = int(input("Number of elements "))
l = []
#·l·=·list()
for i in range(n):
  x = int(input("Enter a number "))
  l.append(x)

print(l)
```

```
Number of elements 5
Enter a number 1
Enter a number 2
Enter a number 3
Enter a number 4
Enter a number 5
[1, 2, 3, 4, 5]
```

```
for i in range(n):
  print(l[i], end=" ")
```

```
1 2 3 4 5
```

```
s=0
for i in range(n):
  s += l[i]
print("Sum=",s)
```

```
Sum= 15
```

```
print("Sum = ",sum(l))
```

```
Sum =  15
```

```
max(l)
```

```
5
```

```
min(l)
```

```
1
```

```
l = [1,4,2,7,8,3,4,2,5,8,9,10,1,3]
len(l)
```

```
14
```

```
for i in range(len(l)):
  print(l[i], end=" ")
```

```
1 4 2 7 8 3 4 2 5 8 9 10 1 3
```

```
for e in l:
  print(e, end=" ")
```

```
1 4 2 7 8 3 4 2 5 8 9 10 1 3
```

Membership operator

```
l
```

```
[1, 4, 2, 7, 8, 3, 4, 2, 5, 8, 9, 10, 1, 3]
```

```
100 in l
```

```
False
```

```
3 in l
```

```
True
```

```
1 not in l
```

```
False
```

```
123 not in l
```

```
True
```

```
l.index(8)
```

```
4
```

```
l.index(102)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-30-084ca63cdf52> in <module>
----> 1 l.index(102)

ValueError: 102 is not in list

SEARCH STACK OVERFLOW
```

```
l
```

```
[1, 4, 2, 7, 8, 3, 4, 2, 5, 8, 9, 10, 1, 3]
```

```
l.count(8)
```

```
2
```

```
l.count(103)
```

```
0
```

```
l1 = [1,2,3]
l2 = [4,5,6]
#l1.append(l2) # not working
# for e in l2:
#   l1.append(e)
l1.extend(l2)
l1
```

```
[1, 2, 3, 4, 5, 6]
```

```
l1
```

```
[1, 2, 3, 4, 5, 6]
```

```
del l1[3]
```

```
l1
```

```
[1, 2, 3, 5, 6]
```

```
l
```

```
[1, 4, 2, 7, 8, 3, 4, 2, 5, 8, 9, 10, 1, 3]
```

```
l.sort()
```

```
l
```

```
[1, 1, 2, 2, 3, 3, 4, 4, 5, 7, 8, 8, 9, 10]
```

```
l = [1, 4, 2, 7, 8, 3, 4, 2, 5, 8, 9, 10, 1, 3]
```

```
sorted(l)
```

```
[1, 1, 2, 2, 3, 3, 4, 4, 5, 7, 8, 8, 9, 10]
```

```
l
```

```
[1, 4, 2, 7, 8, 3, 4, 2, 5, 8, 9, 10, 1, 3]
```

Slicing

```
# list[start : stop : step]
# list[0 : len : 1]
```

```
l = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
l[2:7]
```

```
[2, 3, 4, 5, 6]
```

```
l[2:8:2]
```

```
[2, 4, 6]
```

```
l[:6]
```

```
[0, 1, 2, 3, 4, 5]
```

```
l[2:]
```

```
[2, 3, 4, 5, 6, 7, 8, 9]
```

```
l[:]
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
l[::3]
```

```
[0, 3, 6, 9]
```

```
l[8:3:-1]
```

```
[8, 7, 6, 5, 4]
```

```
l[5::-1]
```

```
[5, 4, 3, 2, 1, 0]
```

```
l[::-1]
```

```
[9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
```

String

```
s = "Hello Python"
```

```
s[3]
```

```
'l'
```

```
s[3]='x' # Immutable Object
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-61-512e1e3ef74d> in <module>
----> 1 s[3]='x'

TypeError: 'str' object does not support item assignment

SEARCH STACK OVERFLOW
```

```
s
```

```
'Hello Python'
```

```
s[2:6]
```

```
'llo '
```

```
s[::-1]
```

```
'nohtyP olleH'
```

```
s
```

```
'Hello Python'
```

```
len(s)
```

```
12
```

```
dir(s)
```

```
'__reduce__',
'__reduce_ex__',
'__repr__',
'__rmod__',
'__rmul__',
'__setattr__',
'__sizeof__',
'__str__',
'__subclasshook__',
'capitalize',
'casefold',
'center',
'count',
'encode',
'endswith',
'expandtabs',
'find',
'format',
'format_map',
'index',
'isalnum',
'isalpha',
'isascii',
'isdecimal',
'isdigit',
'isidentifier',
'islower',
'isnumeric',
'isprintable',
'isspace',
'istitle',
'isupper',
'join',
'ljust',
'lower',
'lstrip',
'maketrans',
'partition',
'replace',
'rfind',
'rindex',
'rjust',
'rpartition',
'rsplit',
'rstrip',
'split',
'splitlines',
'startswith',
'strip',
'swapcase',
'title',
'translate'
```

s.lower()

```
'hello python'
```

s.upper()

```
'HELLO PYTHON'
```

s

```
'Hello Python'
```

s.startswith('H')

```
True
```

s.startswith('x')

```
False
```

s

```
'Hello Python'
```

```
v = "aeiouAEIOU"
for c in s:
  if c in v:
    print(c)
```

```
e
o
o
```

```
s1 = "Apple"
s1[0] in v
```

```
True
```

```
s = "we are learing python"
words = s.split(" ")

words
```

```
['we', 'are', 'learing', 'python']
```

```
for word in words:
  if word[0] not in v:
    print(word)
```

```
we
learing
python
```

```
words
```

```
['we', 'are', 'learing', 'python']
```

```
s2 = "=".join(words)

s2
```

```
'we=are=learing=python'
```

Tuple

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

t
```

```
(1, 2, 3)
```

```
t[1] = 100
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-88-95a1c443bf88> in <module>
----> 1 t[1] = 100
```

```
a=1
b=2
c=3

# a,b,c = (1,2,3) #tuple unpacking
a,b,c = 1,2,3 #tuple unpacking
```

```
a
```

```
1
```

```
b
```

```
2
```

```
c
```

```
3
```

```
t = (1,2,3,4,5,6,7,8,9)
```

```
l = list(t)
```

```
l
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
l[3] =40
```

```
l
```

```
[1, 2, 3, 40, 5, 6, 7, 8, 9]
```

```
t = tuple(l)
```

```
t
```

```
(1, 2, 3, 40, 5, 6, 7, 8, 9)
```

Set

```
s = {1,2,3,4,1,5,3,2,5,6}
```

```
s
```

```
{1, 2, 3, 4, 5, 6}
```

```
s.add(10)

s
```

```
{1, 2, 3, 4, 5, 6, 10}
```

```
s1 = {1,2,3,4}
s2 = {3,4,5,6}

s1.union(s2)
```

```
{1, 2, 3, 4, 5, 6}
```

```
s1
```

```
{1, 2, 3, 4}
```

```
s1.intersection(s2)
```

```
{3, 4}
```

```
s1.difference(s2)
```

```
{1, 2}
```

```
s2.difference(s1)
```

```
{5, 6}
```

```
s3 =  {1,2}
s4 = {1,2,3,4,5}
s3.issubset(s4)
```

```
True
```

```
s4.issubset(s3)
```

```
False
```

```
s4.issuperset(s3)
```

```
True
```

Dictionary

```
# dict = {key: value}
d = {1: "Sam", 2: "Jhon", 3: "Sara"}

x = {}
type(x)
```

```
dict
```

```
x = {1}
type(x)
```

```
set
```

```
d[2]
```

```
'Jhon'
```

```
d.get(2)
```

```
'Jhon'
```

```
d[10]
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-123-76a0c2738598> in <module>
----> 1 d[10]

KeyError: 10
```

SEARCH STACK OVERFLOW

```
d.get(10)
```

```
dir(d)
```

```
['__class__',
 '__contains__',
 '__delattr__',
 '__delitem__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__getitem__',
 '__gt__',
 '__hash__',
 '__init__',
 '__init_subclass__',
 '__iter__',
 '__le__',
 '__len__',
 '__lt__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__reversed__',
 '__setattr__',
 '__setitem__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
 'clear',
 'copy',
 'fromkeys',
 'get',
 'items',
 'keys',
 'pop',
 'popitem',
 'setdefault',
 'update',
 'values']
```

```
d[5] = 'Riya'
```

```
d
```

```
{1: 'Sam', 2: 'Jhon', 3: 'Sara', 5: 'Riya'}
```

```
d.keys()
```

```
dict_keys([1, 2, 3, 5])
```

```
for i in d.keys():
  print(d.get(i))
```

```
Sam
Jhon
Sara
Riya
```

```
d.values()
```

```
dict_values(['Sam', 'Jhon', 'Sara', 'Riya'])
```

```
for n in d.values():
  print(n)
```

```
Sam
Jhon
Sara
Riya
```

```
d.items()
```

```
dict_items([(1, 'Sam'), (2, 'Jhon'), (3, 'Sara'), (5, 'Riya')])
```

```
for item in d.items():
  print(item)
```

```
(1, 'Sam')
(2, 'Jhon')
(3, 'Sara')
(5, 'Riya')
```

```
for item in d.items():
  print(item[0], item[1])
```

```
1 Sam
2 Jhon
3 Sara
5 Riya
```

```
for item in d.items():
  r,n = item
  print(f"Roll: {r} Name:{n}")
```

```
Roll: 1 Name:Sam
Roll: 2 Name:Jhon
Roll: 3 Name:Sara
Roll: 5 Name:Riya
```

```
for r,n in d.items():
  print(f"Roll: {r} Name:{n}")
```

```
Roll: 1 Name:Sam
Roll: 2 Name:Jhon
Roll: 3 Name:Sara
Roll: 5 Name:Riya
```

```
std = {1: ['c', 'c++'], 2: ['Python', 'Django', 'java'], 3: ['html', 'c
```

```
for r, sub in std.items():
  print(f"Roll {r} knows ", end=" ")
  for s in sub:
    print(s, end=" ")
  print()
```

```
Roll 1 knows  c c++
Roll 2 knows  Python Django java
Roll 3 knows  html css
```

Functions

```
def hello():
  print("Hello World!")

hello()
```

```
Hello World!
```

```
def greet(name):
  print("Hello, ", name)

greet("Tom")
```

```
Hello,  Tom
```

```
def add(a,b):
  print("Sum = ",(a+b))

add(2,3)
```

```
Sum =  5
```

```
def add(a,b):
  return a+b


c = add(10,20)
print(f"Sum= {c}")
```

```
Sum= 30
```

```
def calc(a,b):
  return a+b, a-b, a*b

s,d,m = calc(3,2)
print(s,d,m)
```

```
5 1 6
```

✓ 0s completed at 6:17 PM