

Python Lab Batch S13 Date 14/02/2022, Batch Prof. Dr. Shachi Natu

In [1]:

```
1 def add_num():
2     x=10
3     y=20
4     c=x+y
5     print(c)
```

In [2]:

```
1 add_num()
```

30

In [13]:

```
1 def add_num(x):
2     y=20
3     c=x+y
4     print(c)
```

In [15]:

```
1 c = add_num(20)
2 c
```

40

In [6]:

```
1 def welcome(x):
2     print("Welcome ",x)
```

In [7]:

```
1 welcome("Shachi")
```

Welcome Shachi

In [8]:

```
1 def add_num(x):
2     y=20
3     c=x+y
4     return(c)
```

In [10]:

```
1 addition = add_num(30)
2 print(addition)
```

50

In [16]:

```
1 def add_sub(x,y):  
2     return(x+y, x-y)  
3
```

In [17]:

```
1 a,s = add_sub(50,60)
```

In [18]:

```
1 print(a,s)
```

110 -10

Positional and Keyword arguments in Python

In [22]:

```
1 def average(a,b,c):  
2     print(a,b,c)  
3     return((a+b+c)/3)
```

In [23]:

```
1 avg = average(10,20,30)  
2 avg
```

10 20 30

Out[23]:

20.0

In [24]:

```
1 avg = average(30,10,20)
```

30 10 20

In [25]:

```
1 avg
```

Out[25]:

20.0

In [26]:

```
1 def show(greeting, name):  
2     print(greeting, name, "!!")
```

In [27]:

```
1 show('hello', 'Shachi')# positional argument
```

hello Shachi !!

In [28]:

```
1 show(name = 'Shachi', greeting= 'Hi') # keyword argument
```

Hi Shachi !!

Variable length arguments in Python

In [29]:

```
1 def add_integers(a,b,c):
2     return(a+b+c)
```

In [30]:

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

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-30-f55570d38072> in <module>()
----> 1 add_integers(2,3)
```

TypeError: add_integers() missing 1 required positional argument: 'c'

In [31]:

```
1 add_integers(2,3,4)
```

Out[31]:

9

In [32]:

```
1 add_integers(2,3,4,5,6)
```

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-32-98f8eb70342f> in <module>()
----> 1 add_integers(2,3,4,5,6)
```

TypeError: add_integers() takes 3 positional arguments but 5 were given

In [33]:

```
1 def add_integers(integer):
2     sum=0
3     for element in integer:
4         sum+=element
5     return(sum)
```

In [34]:

```
1 add_integers([1,2,3,4,5,6])
```

Out[34]:

21

In [35]:

```
1 def add_integers(*args):
2     sum=0
3     for arg in args:
4         sum+=arg
5     return(sum)
```

In [37]:

```
1 add_integers(1,2,3,4,5)
```

Out[37]:

15

In [38]:

```
1 def concat_strings(*args):
2     sum=""
3     for arg in args:
4         sum=sum+arg+" "
5     return(sum)
```

In [40]:

```
1 concat_strings("Hello", "World", "r", "u", "there")
```

Out[40]:

'Hello World r u there '

In [41]:

```
1 def concat_strings(**kwargs):
2     sum=""
3     for arg in kwargs.values():
4         sum=sum+arg+" "
5     return(sum)
```

In [43]:

```
1 concat_strings(a="Hello", b="World", c="r", d="u", e="there")
```

Out[43]:

```
'Hello World r u there '
```

In [49]:

```
1 def add_num(*args):  
2     #sum=x+y  
3     sum=0  
4     for arg in args:  
5         sum+= arg  
6     return(sum)
```

In [50]:

```
1 add_num()
```

Out[50]:

```
0
```

In [51]:

```
1 def concat_strings(x,y,**kwargs):  
2     sum=x+" "+y  
3     for arg in kwargs.values():  
4         sum=sum+" "+arg  
5     return(sum)
```

In [55]:

```
1 concat_strings('hello','world',z='welcome')
```

Out[55]:

```
'hello world welcome'
```

In []:

```
1
```