Bootcamp Session 2

July 4, 2020

0.0.1 Bootcamp Session 2 Notebook

By Subrat Kumar Swain

[ ]:

[2]:
```
l = [1, 2, 3, 4, 5]
k = l

print(id(l), id(k))

l[2] = 0
print(k)
```

```
97882720 97882720
[1, 2, 0, 4, 5]
```

[3]:
```
l = [1, 2, 3,4 ,5 ,6 , 7, 8, 9]
print(l[2 : 7])
```

```
[3, 4, 5, 6, 7]
```

[4]:
```
x = 10
y = 20
print(locals()['x'])
```

```
10
```

Alice and the boat trip: n = 5 <- No of trips

m = 2 <- No of boats

c = 2 <- Capacity of each boat

2 3 1 4 3 <- No of passengers in each trip

Output: You have to print 'yes' if all the trips are possible otherwise print 'no'

[8]:
```
print('Hello', end = "")

print('Hi')
```

```
HelloHi
```

[10]:
```
for i in range(1, 5):
    print('*' * i)

for i in range(4, 0, -1):
    print('*' * i)
```

```
*
**
***
****
****
***
**
*
```

[15]:
```
n = int(input())
for i in range(1, n // 2 + 2):
    for j in range(1, n + 1):
        if j == i or j == n - i + 1:
            print(j, end = "")
        else:
            print(" ", end = "")
    print()
```

```
5
1   5
 2 4
  3
```

[16]:
```
def fun(a):
    a += 10
    print(a)
    return

x = 10
fun(x)
print(x)
```

```
20
10
```

[17]:
```
def fun(k):
    k.append(4)
    return
l = [1, 2, 3]
fun(l)
```

```
print(l)
```

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

[19]:
```
def fun(a):
    print(id(a))
    a += 10
    print(id(a))
    return
x = 10
fun(x)
print(id(x))
```

```
1810457920
1810458080
1810457920
```

[25]:
```
def add(a, *b):
    sum = a
    for i in b:
        sum += i
    print(sum)
    return
add(10, 20, 30)
```

```
60
```

[29]:
```
def stud(name, **det):
    print(name, det)
    return

stud(name = 'harsh',roll = 55, mark = 87, city = 'Mumbai')
```

```
harsh {'roll': 55, 'mark': 87, 'city': 'Mumbai'}
```

[35]:
```
import math
n = int(input())
sum = 0.0
for i in range(1, n + 1):
    sum += math.log10(i)

print(math.floor(sum) + 1)
```

```
7
4
```

[38]:
```
print(math.floor(12.58))
```

12

[41]:
```
l = [9, 6, 3, 5, 3, 5, 9, 10, 20, 30, 20, 10, 40, 50, 60]
x = []
for i in l:
    if i not in x:
        x.append(i)
print(x)
```

[9, 6, 3, 5, 10, 20, 30, 40, 50, 60]

[ ]:
```
l = [1, 2, 3, 4]

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

[44]:
```
def outer(a):
    def inner(b):
        nonlocal a
        a += 1
        print(a + b)
    return inner

f = outer(5)
f(4)
```

10

[48]:
```
# def fun(a):
#     return a % 2 == 0

l = [1, 2,3, 4, 5, 6, 7, 8, 9]
print(list(map(lambda a: a * 2, l)))
```

[2, 4, 6, 8, 10, 12, 14, 16, 18]

[47]:
```
l = [1, 2,3, 4, 5, 6, 7, 8, 9]

print([i for i in l if i % 2 == 0])
```

[2, 4, 6, 8]

[49]:
```
l = [1, 2,3, 4, 5, 6, 7, 8, 9]

print([i * 2 for i in l])
```

[2, 4, 6, 8, 10, 12, 14, 16, 18]

[50]:
```
print([[0 for j in range(3)]for i in range(4)])
```

```
[[0, 0, 0], [0, 0, 0], [0, 0, 0], [0, 0, 0]]
```

[4]:
```
import functools
l = [1, 2, 3, 4, 5, 6]
print(functools.reduce(lambda a, b: a * b, l))
```

```
720
```

[5]:
```
l = [1, 2, 3, 4, 5, 6]
print(sum([i for i in l]))
```

```
21
```

[12]:
```
a = ['x', 'y', 'z']
b = 3

print(['{}{}'.format(j, i) for j in a for i in range(1, b + 1)])
```

```
['x1', 'x2', 'x3', 'y1', 'y2', 'y3', 'z1', 'z2', 'z3']
```

[13]:
```
l = [int(i) for i in input().split()]
print(l)
```

```
10 20 30 40 50
[10, 20, 30, 40, 50]
```

[ ]:
```

```