

WEEK 4

Lists

Python List

- Python list is an ordered container.
- A list is created by using square brackets ([]).
- The objects are placed inside those brackets and are separated by commas (,).

Example, `l = []
l = [1, 2, 3]
print(l)`

Output
[1, 2, 3]

- A list can contain mixed data types.

Example, `x = ['dog', 21, True]
print(x)`

Output: ['dog', 21, True]

INDEXING

- Indexing is used to access the items of a list.
- Indexing uses square brackets and numbers to access individual items of a list.

- Where 0 refers to the first item, 1 refers to the second item, and so on.

CODE: $l = [1, 2, 3]$
 $\text{print}(l[0], l[1], l[2])$

OUTPUT : 1 2 3

* Negative Indexing

- Negative indexing is used to access the items of a list using negative numbers.
- Where -1 refers to the last item, -2 refers to the second to the last item, and so on.

CODE: $l = [2, 4, 6, 8]$ OUTPUT : 8
 $\text{print}(l[-1])$
 $\text{print}(l[-2])$ 6

* Range of Indexes

- By using a colon (:), we can access a range of items at least once.
- Simply separate two indexes using the colon .
- First Index : start of the range
Second Index : end of the range (not included)
- If you do not specify the first index, the range starts from index 0.

- If you don't specify the last index, the range ends with the last item of the list.

CODE 1: $l = [1, 2, 3, 4, 5]$

OUTPUT

$x = l[2:5]$

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

$y = l[:4]$

$z = l[2:]$

`print(x, y, z)`

ADDING ITEMS TO LISTS

- The append() method adds an item to the end of the list.

Example: $l = [2, 4, 6]$
 $l.append(8)$
`print(l)`

Output: [2,4,6,8]

- The insert() method inserts an item at a specific index.

Example: $l = [2, 4, 6]$
 $l.insert(0, 0)$
 $l.insert(4, 8)$
`print(l)`

Output: [0,2,4,6,8]

DELETING ITEMS FROM A LIST

- The pop() method removes the last item from a list.

Example: $l = [2, 3, 4, 5]$ Output : [2, 3, 4]
 $l.pop()$
print(l)

- The remove() method removes the specified item value.

Example: $l = [2, 3, 4, 5]$ Output : [2, 4, 5]
 $l.remove(3)$
print(l)

- To delete specified index , use the del keyword.

Example: $l = [2, 3, 4, 5]$ Output : [2, 3, 5]
 $del l[2]$
print(l)

GETTING THE LENGTH OF THE LIST

To get the length or the no. of items in a list , use the `len()` method .

Example: $l = [4, 8, 2, 1]$ Output : 4
print(len(l))

CHANGING AN ITEM'S VALUE

To change an item's value, access the index first and use the assignment operator.

CODE : `l = [1, 2, 8, 6, 12, 14]` OUTPUT : `[1, 2, 8, 9, 12, 14]`

`l[3] = 9`

`print(l)`

CHECKING IF AN ITEM EXISTS

To check if an item exists in a list, use the in operator.

It returns True or False, if the item is found or not found, resp.

CODE : `pets = ['dog', 'cat', 'rabbit']` OUTPUT : `True`

`print('dog' in pets)`

`print('python' in pets)` OUTPUT : `False`

EXTENDING A LIST

The `extend()` method adds all items from a list to another list.

CODE : `l1 = [1, 2, 3]`
`l2 = [4, 5, 6]`

~~pass them~~

```
l1 = extend(l2)
```

```
print(l1)
```

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

LOOPING THROUGH A LIST

Looping through a list basically means accessing all ~~the~~ its items one-by-one.

The for loop is used to loop through a list.

CODE:

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

```
for i in l:  
    print(i)
```

OUTPUT :

```
1  
2  
3
```

Date
June 1, 2021

classmate

Date _____
Page _____

Birthday Paradox

CODE :

```
import random
```

```
l = []
```

```
# create an empty list
```

```
for i in range (30) :
```

```
    l.append (random.randint (1,365))
```

```
# append random no.s between 1 to 365
```

```
# append 30 of them
```

```
l.sort()
```

```
print (l)
```

```
i = 0
```

```
flag = 0
```

```
while ( i < len (l) - 1 ) :
```

```
    if ( l[i] == l[i+1] ) :
```

```
        print ("Repeating", l[i])
```

```
        flag = 1
```

```
        break
```

```
i += 1
```

```
if flag == 0 :
```

```
    print ("There is no repetition")
```

Naive Search in a List

CODE :

```
import random as r
```

```
l = []
```

```
for i in range (100000) :  
    l.append (r.randint (1,1000000))
```

```
n = 0
```

```
while (n > -1) :
```

```
    print ("Enter a no. , type -1 to exit")
```

```
n = int (input ())
```

```
flag = 0
```

```
for i in range (len(l)) :
```

```
    if (n == l[i]) :
```

```
        print ("Found")
```

```
        flag = 1
```

```
        break
```

```
if (flag == 0) :
```

```
    print ("Not found")
```

Sorting

CODE :

```
l = [12, 10, 7, 18, 6, 42, 8, 5, 35, 7]  
x = []
```

```
while (len(l) > 0):  
    min = l[0]  
    for i in range(len(l)):  
        if l[i] < min:  
            min = l[i]  
    x.append(min)  
    l.remove(min)
```

```
print(x)
```

OUTPUT : [5, 6, 7, 7, 8, 10, 12, 18, 35, 42]

Dot-Product

CODE : $x = [1, 7, 3, 4]$
 $y = [8, 6, 3, 2]$

if $\text{len}(x) == \text{len}(y)$
 dot_product = 0

for i in range ($\text{len}(x)$):

 dot_product += $x[i] * y[i]$

print (dot_product)

else :

 print ('Dot product can't be calculated')

Matrix - Addition

CODE :

Matrix Addition by first principle

dim = 3

$m_1 = [1, 2, 3]$

$m_2 = [4, 5, 6]$

$m_3 = [7, 8, 9]$

$s_1 = [1, 2, 1]$

$s_2 = [6, 2, 3]$

$s_3 = [4, 2, 1]$

A = []

A.append (e1)

A.append (e2)

A.append (e3)

B = []

B.append (s1)

B.append (s2)

B.append (s3)

print (A)

print (B)

We need to add A and B

C = [[0, 0, 0], [0, 0, 0], [0, 0, 0]]

for i in range (dim) :

 for j in range (dim) :

 C[i][j] = A[i][j] + B[i][j]

print (C).

Output : [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
[[1, 2, 1], [6, 2, 3], [4, 2, 1]]
[[2, 4, 4], [10, 7, 9], [11, 10, 10]]

Matrix Multiplication

CODE:

```

r1 = [1, 2, 3]
r2 = [4, 5, 6]
r3 = [7, 8, 9]
    
```

```

s1 = [1, 2, 1]
s2 = [6, 2, 3]
s3 = [4, 2, 1]
    
```

```

A = []
B = []
A.append(r1)
A.append(r2)
A.append(r3)

B.append(s1)
B.append(s2)
B.append(s3)
    
```

$C = [[0, 0, 0], [0, 0, 0], [0, 0, 0]]$

$\text{dim} = 3$

$C[2][1]$ is the dot product of the 2nd row of A and the
first column of B

for i in range(dim) :

 for j in range(dim) :

 for k in range(dim) :

$c[i][j] += A[i][k] * B[k][j]$

print (c)

Output : $\begin{bmatrix} 25, 12, 10 \\ 58, 30, 25 \\ 91, 48, 40 \end{bmatrix}$

Shortcut for Calculating Matrix Multiplication

import numpy

Output $\begin{bmatrix} 25, 12, 10 \\ 58, 30, 25 \\ 91, 48, 40 \end{bmatrix}$

X = numpy.mat(A)
Y = numpy.mat(B)

print (X * Y)