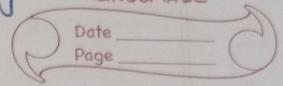


PYTHON

Kaggle \rightarrow Python CLASSMAKER \Rightarrow Tutorial



- Dynamically Typed: Type of variable is determined at runtime.
- Garbage collector: Automatically memory management.
- $5 // 2 \Rightarrow 2$
 \hookrightarrow always give integer value
- Swaping Technique Unique

$$a = 10$$

$$b = 20$$

$$a, b = b, a$$

$$a \Rightarrow 20$$

$$b \Rightarrow 10$$

- Default Argument

```
def ( total_candies , number = 3 ) :
```

```
    return total_candies / number
```

--> fxn(100, 4) => 0
--> fxn(100) => 1

LIST []

List fxn

- ① len(L)
- ② sorted(fruits)
↳ if string sort it in alphabetic order
- ③ sum(L)
- ④ max(primes)
- ⑤ min(primes)

LIST

i) Append

ii) Pop

iii) Index
↳ g

iv) fn =
L
D
- arr

TUPLE

D
Diff

LIST OPERATIONS

- i) Append \Rightarrow planets.append('Pluto')
- ii) Pop \Rightarrow planets.pop()
- iii) Index \Rightarrow planets.index('Earth')
↳ gives index
- iv) In \Rightarrow "Earth" in planets \rightarrow True / False
↳ Determine list contain that particular value

TUPLE () \rightarrow

↳ $t = (1, 2, 3)$ } same
↳ $t = 1, 2, 3$

Q Difference in List & Tuple?

↳ can't be modified
↳ so they are immutable

STRING OPERATION

i) `isupper()` \Rightarrow check if character is all uppercase.

ii) `islower()`

iii) `upper()` (iv) `lower()`

LIST UPDATION

• `square = [n**2 for n in range(10)]`

• `p = [planet for planet in planets
if len(planet) <= 6]`

* Python is Best in String

Manipulation .

• " " "

" " "

→ let us include newlines literally just by hitting enter

triplequote = """Hello
World""""

--> Print (triplequote) => Hello
World.

- print ("pluto", end='')
- L → instead of writing in new
line it will write in same line

pluto print ("plutos", end='')

O/P

→ plutoplutos

STRING OPERATION

1) planet = 'Pluto'

planet[0] → 'P'

planet[-3:] => uto

len(planet) => 5.

★ Difference in List & string is
string is immutable.

- a.lower()
 - a.upper()
 - a.split()
 - ' '.join([])
 - +
 - a.isdigit()
 - rstrip(' ') → strip from back
 - count(' ')
 - count()
- like Map in C++.

DICTIONARY ⇒ Map keys to value

2 number = { 'one': 1, 'two': 2,
--- }

→ Mutable.

planets = ['Mercury', 'Venus', 'Earth', 'Mars',
'Jupiter', 'Saturn', 'Uranus',
'Neptune']

String

L>

L>

L>

W

C-

le

P is a dictionary

P = { planet : planet[0] for planet in planets }

P = { 'Mercury' : 'M',
 'Venus' : 'V',

 --- y }

trip from
Back

String

L> \' \equiv '

L> \n \equiv \nline

L> \" \equiv "

• what is the length

c = 'it\'s ok' \xrightarrow{len} it's ok.

len = 7

Enumerate Concept IN FOR Loop

↳ Syntax

↳ `enumerate (iterable, start=0)`

↳ It basically add a counter with while looping (iterating) in python

for index, mark in enumerate(marks):

```
    print(mark)
    if (index == 3):
        print("HARRY, awesome")
```

→ if you want index to start from 1

↳ `enumerate(marks, start=1)`

PRINT

① `print ("The result is : {} ".format(42))`

② `print (".... {} ".format("example"))`

index = 0

for mark in marks:

 print(mark)

 if (index == 3):

 print("Harry, Awesome")

 index = index + 1

works):

1

)

form st(42)) => the result is : 42

....example

⇒ Python
List

LIST v/s NUMPY ARRAY

classmate

Date _____

Page _____

① $[3, 4, 1, 2, 2, 1] + 10 \rightarrow X$
(Not allowed)

② rolls = ~~random.~~ np.random.randint(1, 6, 10)
 \hookrightarrow rolls = array([3, 4, 3, 4, 5, ..., 10])

rolls + 10 \Rightarrow array([13, 14, 13, 14, 15, ...])

List v/s Numpy Array

① Adding a number to the list is not allowed

① Adding number to each element of array.

③ print(rolls <= 3) \hookrightarrow array

o/p \hookrightarrow array([True, False, True, ...])

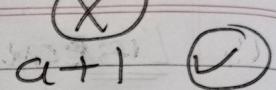
List $L[1, -1] \Rightarrow$ 2nd List last element.

NIGHT
python.
Ran

PYTHON IN 2 HR

classmate

Date _____
Page _____



Q

Che

S

s

if

cle

re

RANDO

L

Numpy

MATH FXN

$a++$

$a = a + 1$

X

✓

① $m.\exp(-200) \Rightarrow e^{-200}$

② $m.\log(100, 2) \Rightarrow \log_2(100)$

③ $m.\log(100) \Rightarrow \log_{10}(100)$

④ $m.\cos(30)$

⑤ $m.\sin(30)$

⑥ $m.\tan(30)$

⑦ $m.\sqrt{324} \Rightarrow \sqrt{324}$

⑧ $m.\text{ceil}(89.9) \Rightarrow 90$
 $(89.4) \Rightarrow 90$

⑨ $m.\text{floor}(89.9) \Rightarrow 89$

Q

Check Palindrome in Python

s = 'Babbar'

s = 'Madam'

if (s == s[::-1]):

return true

else:

return false

RANDOM

↳ import random as r.

↳ Take two arguments only

r.randint(1, 100)

Numpy => random

↳ np.random.randint(1, 100, 10)

↳ [5, 9, 55, 77, ...]

→ Generate real number using random

(i) ↳ `r.uniform(1,100)` ⇒ 77.19233

(ii) ↳ `x = uniform
round(r.uniform(1,100), 2)` ⇒ 77.19

→ SAMPLE

(i) `r.sample(A, 2)`
↳ Take two sample

(ii) `r.sample(range(-100,100), 2)`

EXCEPTION HANDLING

↳ use concept of

↳ Try

try :

except :

list

* For iter printing

→ `del(L)`
→ ~~del(L,L)~~
→ ~~del(L,L)~~

→ ~~L + L~~

~~L + L~~

~~2 * L~~

→ ~~L.sort~~

~~min(L)~~

random

1233:

> 77.19

for i in range (-5, 6):

try:

print (100/i)

except:

print ('error')

list

★ For iteration in List we do for printing all elements of list

→ del (L) ⇒ delete the list

→ del (L[1])

→ del (L[1]) ⇒ Delete the first 2nd element from list

→ L + L ⇒ [1 2 3] + [1 2 3]

L + L ⇒ [1 2 3 1 2 3]

→ 2 * L ⇒ [2 4 6]

→ L.sort()

→ L.sort (reverse = True)

→ min (max (L))

LIST

★ ↳ We can do direct modification in Numpy array but not in List
To do modification in List we need to iterate it

$$L = [1 \ 2 \ 3]$$

① $L + L = [1 \ 2 \ 3 \ 1 \ 2 \ 3]$

② $2L \Rightarrow [1 \ 2 \ 3 \ 1 \ 2 \ 3]$

For modification in elements of List

↳ $L = [i * 5 \text{ for } i \text{ in } L]$

↳ $L = [5 \ 10 \ 15]$

DICTIONARY TRAVERSAL Skippy

for k in CGPA:

print(CGPA[k])

classmate

Date _____

Page _____

★ CGPA.
★ CGPA.
★ del
★ CGPA.
★ del C

Tuple

↳ To A
et e
again
Delete
/-->
Ag

SET

↳ or
↳ syn

- * CGPA.keys()
- * CGPA.values()
- * del CGPA[1]
- * CGPA.clear()
- * del CGPA

Tuple

- ↳ To Add element in Tuple convert it into List and then convert it again to Tuple
- ↳ Deleting is also not possible in Tuple
 - ! --> del(T[1]) X
 - ↳ Again for that convert it into its List.

SET

- ↳ only stores unique element even syntax if you store duplicate
- ↳ s = { 'A', 'B', 'A', 'C', 'E' }

print(s) $\Rightarrow \{ 'A', 'B', 'C', 'E' \}$

SET Operation

① $A \cup B \Rightarrow A.union(B)$

set set

② $A \cap B \Rightarrow A.intersection(B)$

③ $A - B \Rightarrow A - B$

④ set $\Rightarrow a = \{ \}$

- i) a.add()
- ii) a.remove()
- iii) a.pop()
- iv)