Double-click (or enter) to edit

Tuples in Python

1. tuples are immutable
2. tuple is a ordered collecttion of elements

```
my_tuple = (1, "a", 3)
# my_tuple[1] = "b"; #tuples are immutable, giving error
print(my_tuple[0:3])
```

```
(1, 'a', 3)
```

replace() method

```
my_tuple = (1, "a", 3)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
/tmp/ipython-input-7-2284161407.py in <cell line: 0>()
      1 my_tuple = (1, "a", 3)
----> 2 my_tuple.replace("a", "b")

AttributeError: 'tuple' object has no attribute 'replace'
```

SETS

1. sets are unordered collection of elements
2. sets conatin only immutable elements
3. indexing and slicing dont work in sets
4. sets contain only unique elements

```
my_set = {1,2,(3,4)} # can define tuple inside a set
print(my_set)
# my_set1 = {1,2,[3,4]} #cannot define a list inside a set because they are mutable
# print(my_set1)
my_set2 = {1,2,{3,4}}
print(my_set2)
```

```
{1, 2, (3, 4)}
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
/tmp/ipython-input-11-1354637214.py in <cell line: 0>()
      3 # my_set1 = {1,2,[3,4]} #cannot define a list inside a set because they are mutable
      4 # print(my_set1)
----> 5 my_set2 = {1,2,{3,4}}
      6 print(my_set2)

TypeError: unhashable type: 'set'
```

add() method

```
my_set = {1,"a", 3}
my_set.add(1) #add only one element in set
print(my_set)
```

```
{'a', 1, 3}
```

update() method

```
my_set = {1, "a", 2, 3}
my_set.update({2,4,5}) #add multiple elements into the set
print(my_set)
```

```
{1, 2, 3, 4, 5, 'a'}
```

remove() method

```
my_set = {1, "a", 2, "b"}
my_set.remove("a")
print(my_set)
```

```
{1, 2, 'b'}
```

difference() method

```
my_set = {1, "a", 2, "b"}
my_set2 = {3,4,1,"a"}
my_set2.difference(my_set)
```

```
{3, 4}
```

DICTIONARY IN PYTHON

```
my_dictionary = {"a": "apple", "b": "mango", "c": "banana"}
print(my_dictionary)
my_dictionary["a"] = "papaya";
print(my_dictionary)
```

```
{'a': 'apple', 'b': 'mango', 'c': 'banana'}
{'a': 'papaya', 'b': 'mango', 'c': 'banana'}
```

update() method

bold text

```
my_dictionary = {"a": "apple", "b": "mango", "c": "banana"}
my_dictionary.update({"city": "New York", "occupation": "Engineer"})
my_dictionary.update({"a" : "papaya"}) #update a key:value pair or add multiple key:value pairs
print(my_dictionary)
```

```
{'a': 'papaya', 'b': 'mango', 'c': 'banana', 'city': 'New York', 'occupation': 'Engineer'}
```

list & tuples inside a dictionary

```
my_dict = {"a": "apple", "b": "mango", "c": "banana", "d": ["hello", "world"]}
my_dict["d"][1]
my_dict1 = {"a": "apple", "b": "mango", "c": ("my", "world2", "yes"), "d": ["hello", "world"]}
print(my_dict1["c"][1])
```

```
world2
```

keys(), values() and items() method

```
my_dict = {"a": "apple", "b": "mango", "c": "banana", "d": ["hello", "world"]}
my_dict.keys()
my_dict.values()
my_dict.items()
```

```
dict_items
```

Typecasting

```
my_dict = {"a": "apple", "b": "mango", "c": "banana", "d": ["hello", "world"]}
print(list(my_dict.values())) #typecasted to list
```

```
['apple', 'mango', 'banana', ['hello', 'world']]
```

get() method

```
my_dict = {"a": "apple", "b": "mango", "c": "banana", "d": ["hello", "world"]}
my_dict.get("e", "Not found") #returns a value which passed as a second argument in get() if not found and return
```

```
'Not found'
```

Typecasting to list and dictionary by using zip() method

```
country_list = ["india", "USA", "brazil", "russia"]
city_list = ["delhi", "washington DC", "ABC", "XYZ"]
new_dict = dict(zip(country_list, city_list))
print(new_dict)
new_list = list(zip(country_list, city_list))
print(new_list)
number_list = [1,2,3,4]
```

```
number_list2 = [5,6,7,8]
number_list_dict = dict(zip(number_list, number_list2))
print(number_list_dict)
# number_dict = {1: "apple"}
# print(number_dict)
```

```
{'india': 'delhi', 'USA': 'washington DC', 'brazil': 'ABC', 'russia': 'XYZ'}
[('india', 'delhi'), ('USA', 'washington DC'), ('brazil', 'ABC'), ('russia', 'XYZ')]
{1: 5, 2: 6, 3: 7, 4: 8}
```

pop() method in dictionary

```
my_dict = {"a": "apple", "b": "mango", "c": "banana", "d": ["hello", "world"]}
my_dict.pop("a")
print(my_dict.pop("e", "not found"))
print(my_dict)
```

```
not found
{'b': 'mango', 'c': 'banana', 'd': ['hello', 'world']}
```

CONDITIONALS STATEMENTS

```
# age = int(input())
# print(age)
m,n = map(int, input().split())
print(m)
print(n)
```

```
1 2
1
2
```

```
m,n = map(str, input().split())
print(m+" "+n)
```

```
tanishak singhal
tanishak singhal
```

```
names = eval(input())
print(names)
```

```
["tanishak", "rahul", "sachine"]
['tanishak', 'rahul', 'sachine']
```

```
names = input().strip()
print(names)
```

```
"tanishak" "singhal"
"tanishak" "singhal"
```

```
age = int(input("Enter your age: "))
if(age >= 18) :
    print("eligible for vote")
    print("Can have driving license")
else :
```

```
    print("Not eligible for vote and driving license")
```

```
Enter your age: 12
Not eligible for vote and driving license
```

```
number = int(input("Enter a number: "))
if(number % 2 == 0) :
  print("Even")
else :
  print("Odd")
```

```
Enter a number: 23
Odd
```

```
# number = "121"
# start=0
# end=len(number)-1
# while(start<=end) :
#   if(number[start] != number[end]) :
#     print("not palindrome")
#     break;
#   start++
#   end--

# print("palindrome")
```

```
  File "/tmp/ipython-input-18-4148372947.py", line 8
    start++
          ^
SyntaxError: invalid syntax
```

```
#elif
age=int(input())
if(age>=18):
    print("adult")
elif(age<=18 & age>=16):
    print("teenager")
else:
  print("child")
```

```
18
adult
```

```
marks = int(input("Enter marks between 0 to 100: "))
if(marks>100) :
  print("couldn't you understand! between 0 and 100")
elif(marks>=90) :
  print("A")
elif(marks >= 70 and marks < 90) :
  print("B")
elif(marks>=40 and marks < 70) :
  print("C")
elif(marks<40):
  print("F")
```

```
Enter marks between 0 to 100: 101
couldn't you understand! between 0 and 100
```

```
number1 = int(input("Enter first number: "))
number2 = int(input("Enter second number: "))
operator = input("Enter a operator: ")
if(operator == "+") :
  print(number1+number2)
elif(operator == "-"):
  print(number1-number2)
elif(operator == "*"):
  print(number1*number2)
elif(operator == "/"):
  print(number1/number2)
elif(operator == "%"):
  percentage = ((number1+number2)/100)
  print(percentage)
```

```
Enter first number: 23
Enter second number: 34
Enter a operator: %
0.57
```

Double-click (or enter) to edit