Feb-09

```
>>> dict
<class 'dict'>
>>> capitals = (("India", "New Delhi"),
         ("England", "London"),
         ("Sri Lanka", "Colombo"),
         ("Japan", "Tokyo"))
>>> capitals
(('India', 'New Delhi'), ('England', 'London'), ('Sri Lanka',
'Colombo'), ('Japan', 'Tokyo'))
>>> capitals_dict = dict(capitals)
>>> capitals_dict
{'India': 'New Delhi', 'England': 'London', 'Sri Lanka': 'Colombo',
'Japan': 'Tokyo'}
>>> capitals = ((["India"], "New Delhi"),
         ("England", "London"),
         ("Sri Lanka", "Colombo"),
         ("Japan", "Tokyo"))
>>> capitals_dict = dict(capitals)
Traceback (most recent call last):
  File "<pyshell#9>", line 1, in <module>
    capitals_dict = dict(capitals)
TypeError: unhashable type: 'list'
>>> capitals_dict
{'India': 'New Delhi', 'England': 'London', 'Sri Lanka': 'Colombo',
'Japan': 'Tokyo'}
>>> capitals_dict["India"] = "Delhi"
>>> capitals_dict
{'India': 'Delhi', 'England': 'London', 'Sri Lanka': 'Colombo',
'Japan': 'Tokyo'}
>>> capitals_dict["USA"] = "Washington DC"
>>> capitals_dict
{'India': 'Delhi', 'England': 'London', 'Sri Lanka': 'Colombo',
'Japan': 'Tokyo', 'USA': 'Washington DC'}
>>> capitals_dict.keys()
dict_keys(['India', 'England', 'Sri Lanka', 'Japan', 'USA'])
>>> dir({})
['__class__', '__class_getitem__', '__contains__', '__delattr__',
'__delitem__', '__dir__', '__doc__', '__eq__', '__format__',
'__ge__', '__getattribute__', '__getitem__', '__gt__', '__hash__',
'__init__', '__init_subclass__', '__ior__', '__iter__', '__le__',
'__len__', '__lt__', '__ne__', '__new__', '__or__', '__reduce__',
'__reduce_ex__', '__repr__', '__reversed__', '__ror__',
'__setattr__', '__setitem__', '__sizeof__', '__str__',
'__subclasshook__', 'clear', 'copy', 'fromkeys', 'get', 'items',
'keys', 'pop', 'popitem', 'setdefault', 'update', 'values']
>>> capitals_dict.items()
dict_items([('India', 'Delhi'), ('England', 'London'), ('Sri Lanka',
'Colombo'), ('Japan', 'Tokyo'), ('USA', 'Washington DC')])
>>> x = capitals_dict.items()
>>> x
dict_items([('India', 'Delhi'), ('England', 'London'), ('Sri Lanka',
'Colombo'), ('Japan', 'Tokyo'), ('USA', 'Washington DC')])
\
>>> dir(x)
['__and__', '__class__', '__contains__', '__delattr__', '__dir__',
'__doc__', '__eq__', '__format__', '__ge__', '__getattribute__',
'__gt__', '__hash__', '__init__', '__init_subclass__', '__iter__',
'__le__', '__len__', '__lt__', '__ne__', '__new__', '__or__',
'__rand__', '__reduce__', '__reduce_ex__', '__repr__',
'__reversed__', '__ror__', '__rsub__', '__rxor__', '__setattr__',
'__sizeof__', '__str__', '__sub__', '__subclasshook__', '__xor__',
'isdisjoint']
>>> capitals_dict
{'India': 'Delhi', 'England': 'London', 'Sri Lanka': 'Colombo',
'Japan': 'Tokyo', 'USA': 'Washington DC'}
>>> for i in capitals_dict:
     print(i)


India
England
Sri Lanka
Japan
USA
>>> for key,value in capitals_dict.items():
     print(value)


Delhi
London
Colombo
Tokyo
Washington DC
>>> for key,value in capitals_dict.items():
     print(key, value)


India Delhi
```

```
England London
Sri Lanka Colombo
Japan Tokyo
USA Washington DC
>>> capitals_dict["Nepal"]
Traceback (most recent call last):
  File "<pyshell#29>", line 1, in <module>
    capitals_dict["Nepal"]
KeyError: 'Nepal'
>>> capitals_dict["Nepal"] = "Kathmandu"
>>> capitals_dict["Nepal"]
'Kathmandu'
>>> bool
<class 'bool'>
>>>  # Nested Dictionary

>>> states = {{'a'}: 'b'}
Traceback (most recent call last):
  File "<pyshell#34>", line 1, in <module>
    states = {{'a'}: 'b'}
TypeError: unhashable type: 'set'
>>> states = {{'a':'a'}: 'b'}
Traceback (most recent call last):
  File "<pyshell#35>", line 1, in <module>
    states = {{'a':'a'}: 'b'}
TypeError: unhashable type: 'dict'
>>> states = {"Karnataka": {'capital': 'Bengaluru',}}
>>> states = {"Karnataka": {'capital': 'Bengaluru', 'sweet':'Mysore
Pak'}, "Maharashtra": {"capital": "Mumbai", "sweet": ("Jalebi",
"Gulab Jamun")}, 'Telanagana': {'capital': "Hyderabad",
'sweet':'kurbanika meta'}}
>>> states
{'Karnataka': {'capital': 'Bengaluru', 'sweet': 'Mysore Pak'},
'Maharashtra': {'capital': 'Mumbai', 'sweet': ('Jalebi', 'Gulab
Jamun')}, 'Telanagana': {'capital': 'Hyderabad', 'sweet': 'kurbanika
meta'}}
>>> states['Karnataka']
{'capital': 'Bengaluru', 'sweet': 'Mysore Pak'}
>>> states['Karnataka']['sweet']
'Mysore Pak'
>>> states['Maharashtra']['sweet']
('Jalebi', 'Gulab Jamun')
>>> states['Telangana']['sweet']
Traceback (most recent call last):
  File "<pyshell#42>", line 1, in <module>
```

```
    states['Telangana']['sweet']
KeyError: 'Telangana'
>>> states['Telanagana']['sweet']
'kurbanika meta'
>>> del states['Telanagana']
>>> states
{'Karnataka': {'capital': 'Bengaluru', 'sweet': 'Mysore Pak'},
'Maharashtra': {'capital': 'Mumbai', 'sweet': ('Jalebi', 'Gulab
Jamun')}}
>>> states.keys()
dict_keys(['Karnataka', 'Maharashtra'])
>>> states['Telangana']= {'capital': 'Hyderabad', 'sweet': 'kurbanika
meta'}
>>> states.keys()
dict_keys(['Karnataka', 'Maharashtra', 'Telangana'])
>>> states['Telangana']
{'capital': 'Hyderabad', 'sweet': 'kurbanika meta'}

>>> ## Find sum of digits
>>> 12233
12233
>>> 1+2+2+3+3
11
>>> x = 21321321
>>> ## Convert integer to a list(or any sequence)
>>> x_l = [x]  # convert into list

>>> x_l
[21321321]
>>> l = 12233
>>> s = 0
>>> for i in str(l):
     s = s+ int(i)


>>> s
11
>>> for i in list(str(l)):
     i


'1'
'2'
'2'
'3'
```

```
'3'
>>> s = 0
>>> for i in str(l):
     s += int(i)  # s = s + i


>>> s
11
>>> r = "abc"
>>> r * 2
'abcabc'
>>> r
'abc'
>>> #r = r * 2
>>> r *= 2
>>> r
'abcabc'
>>> l1 = [1,2,3,4]
>>> l2 = [1, 2, 9, 6]
>>> for i in l1:
     if i in l2:
          l1.remove(i)


>>> l2
[1, 2, 9, 6]
>>> l1
[2, 3, 4]
>>> # l1[0] -> 1
>>> # l1 = [1,2,3,4]
>>> l1 = [1,2,3,4]
>>> for i in l1:
     print(f'outside if i = {i}, l1 = {l1}')
     if i in l2:
          l1.remove(i)
          print(f'inside if i = {i}, l1 = {l1}')
     print(f'after if i = {i}, l1 = {l1}')


outside if i = 1, l1 = [1, 2, 3, 4]
inside if i = 1, l1 = [2, 3, 4]
after if i = 1, l1 = [2, 3, 4]
outside if i = 3, l1 = [2, 3, 4]
after if i = 3, l1 = [2, 3, 4]
outside if i = 4, l1 = [2, 3, 4]
```

```
after if i = 4, l1 = [2, 3, 4]
>>> # ASCII values
>>> ord('a')
97
>>> ord('z')
122
>>> ord('A')
65
>>> ord('Z')
90
>>> s = "safsafdsafdsfdsf4dsfewrfrw"
>>> s = "safsafdsafdsfdsfAdsfewrfrw"
>>> # break the loop if a character is not lowercase  using ASCII
>>> # value using break keyword
>>> for i in s:
     if ord(i) < 97 or ord(i) > 122:
           break
     print(i)


s
a
f
s
a
f
d
s
a
f
d
s
f
d
s
f
>>> # break the loop if a character is not lowercase  using
.islower() method value using break keyword
>>> for i in s:
     if not i.islower():
           break
     print(i)


s
```

a
f
s
a
f
d
s
a
f
d
s
f
d
s
f