collections_Python

February 19, 2020

[1]:
```
from collections import namedtuple
from collections import  deque
from collections import ChainMap
from collections import Counter
from collections import OrderedDict
from collections import defaultdict
```

[17]:
```
## Nametuple
```

[2]:
```
a = namedtuple("corso", "nome, tecnologia")
```

[8]:
```
primo = a("andrea", "informatica")
secondo = a("paolo", "chimica")
terzo = a("francesca", "matematica")
```

[0]:
```
print(primo)
print()
print(secondo)
print()
print(terzo)
```

[13]:
```
print()
print(primo.nome)
print(terzo.tecnologia)
print(f"lo studente si chiama {primo.nome.upper()} e vuole studiare {primo.
 ↪tecnologia.upper()}")
print(f"lo studente si chiama {secondo.nome} e vuole studiare {secondo.
 ↪tecnologia.upper()}")
print(f"lo studente si chiama {terzo.nome} e vuole studiare {terzo.tecnologia}")
```

```
andrea
matematica
lo studente si chiama ANDREA e vuole studiare INFORMATICA
lo studente si chiama paolo e vuole studiare CHIMICA
lo studente si chiama francesca e vuole studiare matematica
```

0.1 Deque

[25]:
```
a = ["e", "d", "u", "r", "e", "k", "a"]
d = deque(a)
print(d)
```

```
deque(['e', 'd', 'u', 'r', 'e', 'k', 'a'])
```

[26]:
```
d.append("python")
print(d)
```

```
deque(['e', 'd', 'u', 'r', 'e', 'k', 'a', 'python'])
```

[27]:
```
d.appendleft("andrea")
print(d)
```

```
deque(['andrea', 'e', 'd', 'u', 'r', 'e', 'k', 'a', 'python'])
```

[28]:
```
d.popleft()
print(d)
```

```
deque(['e', 'd', 'u', 'r', 'e', 'k', 'a', 'python'])
```

0.2 chainmap

[32]:
```
a = {1: "eureka", 2: "python"}
b = {3: "ML", 4: "AI"}
a1 = ChainMap(a,b)
print(a1)
```

```
ChainMap({1: 'eureka', 2: 'python'}, {3: 'ML', 4: 'AI'})
```

0.3 counter

[43]:
```
a = [1,2,3,2,4,6,5,2,4,8,8,9,1,3,3,5,7, 12, 13, 12]
aC = Counter(a)
print(aC)
print(list(aC))
print(list(aC.elements()))
print(aC.most_common())
```

```
Counter({2: 3, 3: 3, 1: 2, 4: 2, 5: 2, 8: 2, 12: 2, 6: 1, 9: 1, 7: 1, 13: 1})
[1, 2, 3, 4, 6, 5, 8, 9, 7, 12, 13]
[1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 6, 5, 5, 8, 8, 9, 7, 12, 12, 13]
[(2, 3), (3, 3), (1, 2), (4, 2), (5, 2), (8, 2), (12, 2), (6, 1), (9, 1), (7,
1), (13, 1)]
```

[44]:
```
sub = {1:2 , 5:1}
print(aC.subtract(sub))
print(aC.most_common())
```

```
None
[(2, 3), (3, 3), (4, 2), (8, 2), (12, 2), (6, 1), (5, 1), (9, 1), (7, 1), (13,
1), (1, 0)]
```

0.4 orderedDict

[46]:
```
d = OrderedDict()
d[1] = "e"
d[2] = "d"
d[3] = "u"
d[4] = "r"
d[5] = "e"
d[6] = "k"
d[7] = "a"
print(d)
```

```
OrderedDict([(1, 'e'), (2, 'd'), (3, 'u'), (4, 'r'), (5, 'e'), (6, 'k'), (7,
'a')])
```

0.5 defaultdict

[48]:
```
d = defaultdict(int)
d[1] = "eureka"
d[2] = "virgo"
print(d)
print(d[2])
print(d[3]) # normal sorted dict return a key error! not default dict -->␣
 ↪return 0
```

```
defaultdict(<class 'int'>, {1: 'eureka', 2: 'virgo'})
virgo
0
```

[ ]:
```

```