

Python 3 Cheatsheet: Data Structures

Lists

```

test = [0] * 100      # Initialize with 100 0's
rtn.append([])        # 2D Array setup
rtn[0].append(1)      # [[1]]
ss.reverse()          # Reverse in-place
list1 + list2         # Join lists

```

Tuples

Ordered and immutable.

```

t = ("apple", "banana")
t[1]                      # "banana"
# Useful as Dict keys
d = {}
key = tuple(sorted("cba")) # ('a', 'b', 'c')
d[key] = "anagram"

```

Dictionaries

Hashtables basics.

```

d = {'key': 'value'}
d['k'] = 'v'
d.get('k', default)    # Safe access
d.items()              # [('key', 'value'), ...]
d.pop('k')              # Remove and return
del d['k']                # Delete

# Dict of Lists (Vote counting)
votes = ["ABC", "CBD", "BCA"]
rnk = {v:[0] * len(votes[0]) for v in votes[0]}

```

Sets

Unique elements.

```

s = set()
s.add(3)
s.remove(3)            # Error if missing

```

```
s.discard(3)          # No error if missing
s.pop()                # Remove and return random element
s1 | s2                # Union
s1 & s2                # Intersection
s1 - s2                # Difference
s1 ^ s2                # Symmetric Difference
```

Collections Module

Deque (Double-ended Queue)

```
from collections import deque
q = deque([1, 2, 3])
q.appendleft(0)
q.popleft()           # O(1)
```

Counter

```
from collections import Counter
c = Counter("hello")    # {'l': 2, 'h': 1, 'e': 1, 'o': 1}
c.most_common(2)         # [('l', 2), ('h', 1)]
c.elements()             # Iterator over elements
# Math on Counters
c1 + c2; c1 - c2; c1 & c2 (min); c1 | c2 (max)
```

DefaultDict

```
from collections import defaultdict
d = defaultdict(int)     # Default value 0
d['new'] += 1            # No Key Error
d = defaultdict(list)
d['k'].append(1)
```