C8 Structured Data

Lists

1. created using a pair of square brackets around a list of values separated by commas
2. mutable (can add, delete and change values)

```
def main():
    game = [ 'Rock', 'Paper', 'Scissors', 'Lizard', 'Spock' ]
    print(game[1:5:2])
    #search index
    i = game.index('Paper')
    print(game[i])
    #add element in list
    game.append("Computer")
    #add element by defining the place in the list
    game.insert(0, "Water")
    #remove element from the list
    game.remove("Spock")
    #remove element from the end of the list and print out what has been removed
    x = game.pop()
    print(x)
    #remove element by using index
    game.pop(3)
    #or..
    del game[2]
    #remove element by slice
    del game[1:5]
    #join the list
    print(",".join(game))
    #count number of element
    print(len(game))
    print_list(game)

def print_list(o):
    for i in o: print(i, end=' ', flush=True)
    print()

if __name__ == '__main__': main()
```

Tuple

1. like a list but it is immutable
2. created by using parentheses

```
def main():
    game = game = ( 'Rock', 'Paper', 'Scissors', 'Lizard', 'Spock' )
    print_list(game)

def print_list(o):
    for i in o: print(i, end=' ', flush=True)
    print()

if __name__ == '__main__': main()
```

Dictionary

1. created using sequance of keywords
2. also known as associative array or hashed array
3. created using curly brackets
4. format → key : value

```
def main():
    #using dictionary constructor (more convenient)
    animals = dict(kitten= 'meow', puppy= 'ruff!', lion= 'grrr',
        giraffe= 'I am a giraffe!', dragon= 'rawr')
    #print keys
    for k in animals.keys():
        print(k)
    #search items by using key instaed of index
    print(animals["lion"])
    #change values
    animals["lion"] = "I'm a lion"
    #add new element
    animals["monkey"] = "haha"
    #search key by uisng the in operator
    print("lion" in animals)
    #search key by using conditional expression
    print("found!" if "lion" in animals else "nope!")
    #search for key that is not exist without ant exception
    print(animals.get("godzilla"))
    #normal way to create dictionary
    animals = { 'kitten': 'meow', 'puppy': 'ruff!', 'lion': 'grrr',
        'giraffe': 'I am a giraffe!', 'dragon': 'rawr' }
    print_dict(animals)

def print_dict(o):
    #for x in o: print(f'{x}: {o[x]}')
    #readable way to print keys and values
    for k, v in o.items():
        print(f"{k}: {v}")

if __name__ == '__main__': main()
```

Sets

1. unordered list of unique values
2. indicate with curly brackets
3. does not allow duplicate elements

```python
def main():
    a = set("We're gonna need a bigger boat.")
    b = set("I'm sorry, Dave. I'm afraid I can't do that.")
    print_set(a)
    print_set(b)
    #sort the results
    print_set(sorted(a))
    print_set(sorted(b))
    #check elements in set a but not in set b by using "-"
    print_set(a - b)
    #check elements in set a or in set b or both by using "|"
    print_set(a | b)
    #check elements not in a and b
    print_set(a ^ b)
    #check elements in a and b
    print_set(a & b)


def print_set(o):
    print('{', end = '')
    for x in o: print(x, end = '')
    print('}')

if __name__ == '__main__': main()
```