Cheat Sheet

Data Structures

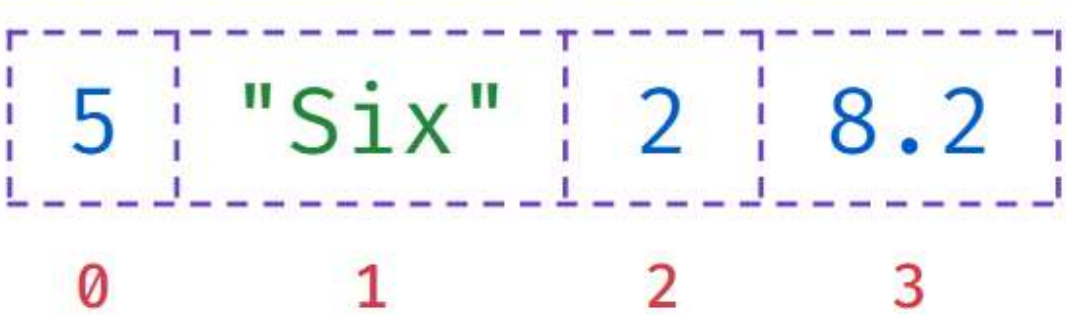
Data Structures allow us to store and organize data efficiently.
This will allow us to easily access and perform operations on the data.

In Python, there are four built-in data structures

- *List*
- *Tuple*
- *Set*
- *Dictionary*

List

List is the most versatile python data structure.
Holds an ordered sequence of items.



Creating a List

Created by enclosing elements within [square] brackets.
Each item is separated by a comma.

Code

PYTHON

```
a = 2
list_a = [5, "Six", a, 8.2]
print(type(list_a))
print(list_a)
```

Output

```
<class 'list'>
[5, 'Six', 2, 8.2]
```

Creating a List of Lists

Code

PYTHON

```python
a = 2
list_a = [5, "Six", a, 8.2]
list_b = [1, list_a]
print(list_b)
```

Output

```
[1, [5, 'Six', 2, 8.2]]
```

Length of a List

Code

PYTHON

```python
a = 2
list_a = [5, "Six", a, 8.2]
print(len(list_a))
```

Output

```
4
```

Accessing List Items

To access elements of a list, we use Indexing.

Code

PYTHON

```python
a = 2
list_a = [5, "Six", a, 8.2]
print(list_a[1])
```

Output

```
Six
```

Iterating Over a List

Code

PYTHON

```python
a = 2
list_a = [5, "Six", a, 8.2]
for item in list_a:
    print(item)
```

Output

```
5
Six
2
8.2
```

List Concatenation

Similar to strings,

+ operator concatenates lists.

Code

PYTHON

```python
list_a = [1, 2, 3]
list_b = ["a", "b", "c"]
list_c = list_a + list_b
print(list_c)
```

Output

```
[1, 2, 3, 'a', 'b', 'c']
```

Adding Items to List

Code

PYTHON

```python
list_a = []
print(list_a)
for i in range(1,4):
    list_a += [i]
print(list_a)
```

Output

```
[]
[1, 2, 3]
```

Repetition

* Operator repeats lists.

Code

PYTHON

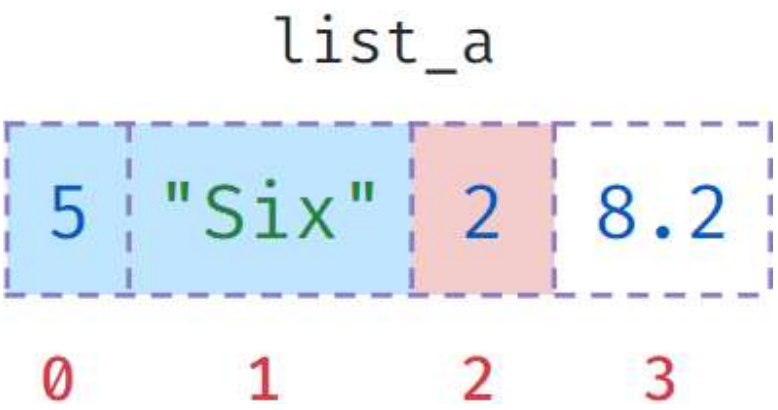
```
list_a = [1, 2]
list_b = list_a * 3
print(list_b)
```

Output

```
[1, 2, 1, 2, 1, 2]
```

List Slicing

Obtaining a part of a list is called List Slicing.



Code

PYTHON

```
list_a = [5, "Six", 2, 8.2]
list_b = list_a[:2]
print(list_b)
```

Output

```
[5, 'Six']
```

Extended Slicing

Similar to string extended slicing, we can extract alternate items using step.

Code

PYTHON

```python
list_a = ["R", "B", "G", "O", "W"]
list_b = list_a[0:5:3]
print(list_b)
```

Output

```
['R', 'O']
```

Converting to List

list(sequence) takes a sequence and converts it into list.

Code

PYTHON

```python
color = "Red"
list_a = list(color)
print(list_a)
```

Output

```
['R', 'e', 'd']
```

Code

PYTHON

```python
list_a = list(range(4))
print(list_a)
```

Output

```
[0, 1, 2, 3]
```

Lists are Mutable

- Lists can be modified.
- Items at any position can be updated.

Code

PYTHON

```python
list_a = [1, 2, 3, 5]
print(list_a)
list_a[3] = 4
print(list_a)
```

Output

```
[1, 2, 3, 5]
[1, 2, 3, 4]
```

Strings are Immutable

Strings are Immutable (Can't be modified).

Code

PYTHON

```python
message = "sea you soon"
message[2] = "e"
print(message)
```

Output

```
TypeError: 'str' object does not support item assignment
```

Submit Feedback