String Immutability

Grow Data Skills

String data types are immutable. Which means a string value cannot be updated.

```
# Can not reassign
t= "GrowDataSkills"
type(t)
```

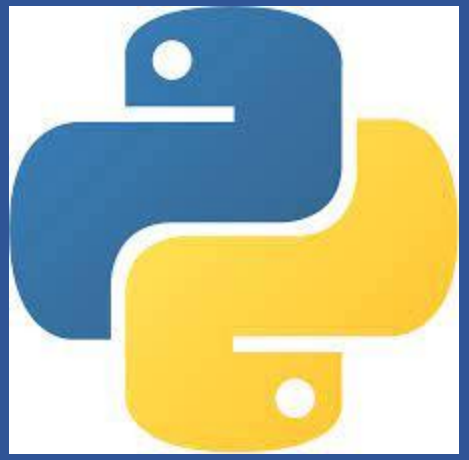
```
str
```

```
t[0] = "M"
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In [2], line 1
----> 1 t[0] = "M"

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

Data Structures in Python



- Tuple
- List
- Set
- Dictionary

Tuples in Python

Grow Data Skills

Tuples is an ordered collection of elements enclosed in ()

tup1=('John',45,5.9)

Tuples are immutable

Getting Individual elements


```
tup1 = ('a', 'b', 'c', 'd', 'e')
tup1[0]

# Output: 'a'
```

```
tup1 = ('a', 'b', 'c', 'd', 'e')
tup1[-1]

# Output: ‘e'
```

```
tup1 = ('a', 'b', 'c', 'd', 'e')
tup1[1:4]

# Output: (‘b’, ’c’, ’d’)
```

Modifying a Tuple

You cannot modify a tuple since it’s immutable

```
tup1= ('a', 'b', 'c', 'd', 'e')
```

```
tup1[2]='f'
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[2], line 1
----> 1 tup1[2]='f'

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

Tuple Basic Operations

Grow Data Skills

Finding length of a Tuple

```
tup1 = ('a', 'b', 'c', 'd', 'e')
len(tup1)
```

```
5
```

Concatenating Tuples

```
tup1 = ('a', 'b', 'c', 'd', 'e')
tup2 = ('f', 'g')
tup1+tup2
```

```
('a', 'b', 'c', 'd', 'e', 'f', 'g')
```

Minimum value

```
tup1 = ('a', 'b', 'c', 'd', 'e')
min(tup1)
```

```
'a'
```

Maximum value

```
tup1 = ('a', 'b', 'c', 'd', 'e')
max(tup1)
```

```
'e'
```

List in Python

Grow Data Skills

List is an ordered collection of elements enclosed in []

lis1=[Apple’,10,True, 3.14]

Lists are mutable

Getting Individual elements


```
l1 = ['a', '1', 'b', '2', 'c']
l1[0]

# Output: 'a'
```

```
l1 = ['a', '1', 'b', '2', 'c']
l1[-1]

# Output: ‘c'
```

```
l1 = ['a', '1', 'b', '2', 'c']
l1[2:4]

# Output: ['b', '2’, ]
```

List Basic Operations

Grow Data Skills

Changing the element at 0th index

```
l1 = ['a', '1', 'b', '2', 'c']
l1[0]=100
l1
```

```
[100, '1', 'b', '2', 'c']
```

Popping the last element

```
l1 = ['a', '1', 'b', '2', 'c']
l1.pop()
l1
```

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

Appending a new element

```
l1 = ['a', '1', 'b', '2', 'c']
l1.append("hello")
l1
```

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

Concatenating lists

```
l1 = ['a', '1', 'b', '2', 'c']
l2=[True, False,'3.14']
l1+l2
```

```
['a', '1', 'b', '2', 'c', True, False, '3.14']
```

List Basic Operations

Grow Data Skills

Reverse elements of a list

```
l1 = ['a', '1', 'b', '2', 'c']
l1.reverse()
l1
```

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

Sorting a list

```
l1= ["banana", "cherry", "apple", "cherry"]
l1.sort()
l1
```

```
['apple', 'banana', 'cherry', 'cherry']
```

Inserting element at specified index

```
l1 = ['a', '1', 'b', '2', 'c']
l1.insert(1,"hello")
l1
```

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

Dictionary in Python

Dictionary is an unordered collection of key-value pairs enclosed in {}

Dictionary is mutable

```
dict1 = {'name': 'John Doe', 'age': 20,'grade': 'A’}
```

Keys and values in Dictionary

Extracting keys from a Dictionary

```
my_dict = {
    'name': 'John',
    'age': 30,
    'city': 'New York'
}
my_dict.keys()
```

```
dict_keys(['name', 'age', 'city'])
```

Extracting values from a Dictionary

```
my_dict = {
    'name': 'John',
    'age': 30,
    'city': 'New York'
}
my_dict.values()
```

```
dict_values(['John', 30, 'New York'])
```

Modifying a Dictionary

Adding a new element

```
my_dict = {
    'name': 'John',
    'age': 30,
    'city': 'New York'
}

my_dict['occupation'] = 'Engineer'
my_dict
```

```
{'name': 'John', 'age': 30, 'city': 'New York', 'occupation': 'Engineer'}
```

Changing an existing element

```
my_dict = {
    'name': 'John',
    'age': 30,
    'city': 'New York'
}

my_dict['age'] = 31
my_dict
```

```
{'name': 'John', 'age': 31, 'city': 'New York'}
```

Set in Python

Grow Data Skills

Set is an unordered and unindexed collection of elements enclosed in {}

s1={10,”abcd”, True}

Duplicates are not allowed in Set

Set Basic Operations

Add new elements to a set

```
my_set = {1, "Monday", 3.14, 5}
my_set.add("hello")
my_set
```

```
{1, 3.14, 5, 'Monday', 'hello'}
```

Updating multiple elements

```
my_set = {1, "Monday", 3.14, 5}
my_set.update([10,False])
my_set
```

```
{1, 10, 3.14, 5, False, 'Monday'}
```

Removing an element

```
my_set = {1, "Monday", 3.14, 5}
my_set.remove(3.14)
my_set
```

```
{1, 5, 'Monday'}
```

Set Functions

Union of 2 Sets

```
s1={1,2,3}
s2={"a", "b", "c"}

s1.union(s2)
```

```
{1, 2, 3, 'a', 'b', 'c'}
```

Intersections of 2 Sets

```
s1={1,2,3,4,5,6,7}
s2={5,6,7,8,9,10}

s1.intersection(s2)
```

```
{5, 6, 7}
```