Module 2: Numbers, Strings, and Lists

1

Knowledge Points

- Creating variables (Assigning values to them)
- Numbers and Arithmetic Operations
- Boolean and testing conditions
- Stings and Polymorphism:
  - Length
  - Slicing and Indexing
  - Concatenation
- Lists:
  - Indexing
  - Slicing
  - Nested
  - Sorting
- Types and Mutability

2

Using Variables

```
num_quarters = 7
num_nickels = 10
num_dimes = 5

total_change =  num_quarters*.25 +\
                num_nickels*0.05+\
                num_dimes*0.1

total_change
```

```
2.75
```

3

Conditional Tests

- Sets the variables x equal to 5.

```
x = 5
```

- Asks if x is equal to 5. Returns boolean.

```
x == 5
```

```
True
```

- Asks if x is less than or equal to 4. Returns boolean.

```
x <= 4
```

```
False
```

4

Slicing Strings

We can access the characters of the string through their **index**

```
sentence = 'Charlie likes walks.'
```

```
sentence[7]
```

```
' '
```

```
len(sentence)
```

```
20
```

Returns the number of characters in the string

5

String Concatenation

- I can combine strings using the + operator.
- So the + operator between two numbers add them and the + operator between two strings concatenates them! This is called **polymorphism.**

```
first = "Jake"
middle = "Belinkoff"
last = "Feldman"

full_name = first + middle + last
full_name
```

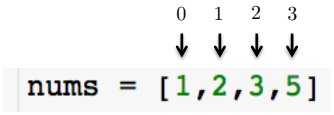
```
'JakeBelinkoffFeldman'
```

- If we want a space, we have to say so.

6

Slicing Lists

- Slicing for lists I also very similar to strings



Returns lists

```
#Get elements at index 1,2
nums[1:3]
```

```
[2, 3]
```

```
#Get element at index 0,1
nums[:2]
```

```
[1, 2]
```

```
len(nums)
```

```
4
```

7

Sorting Lists

- We can sort lists with the built-in sorted() function.

```
#Build list
L = [3,4,2,1,5]

#keyword reverse
sorted(L , reverse = True)
```

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

- Sort list descending
- Default is reverse = False

Next session we will see how to sort L “inplace”.

8

Converting Types

```
y = 5.5
type(y)
```

```
float
```

```
#Convert float to integer
int_y  = int(y)
int_y
```

```
5
```

Built in int() function

```
#Check type
type(int_y)
```

```
int
```

- int() is one way to perform a floor operation

9

Example of Immutability

```
#Create a string
name = "jake"
name
```

```
'jake'
```

Let's say I want to change the first letter of name to a "J"

```
#How I access the first letter
name[0]
```

```
'j'
```

```
#Ituitively...
name[0] = "J"
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-28-35bdf32ef360> in <module>()
      1 #Ituitively...
----> 2 name[0] = "J"

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

Can't change name once it is created!

10