C25 - Inclass Assignment

author & date

- author: Akshar Patel
- date: 4/13/2022

Q1. Update author name and date.

STRING

Q2. A line of code to return "price" strings from `inflation` variable

In [1]:

```
inflation = "increase in price"
```

In [8]:

```
print(inflation[12:18])
```

```
price
```

LIST

- A list of comma-separated items between square brackets [ ]
- Lists might contain items of different types
- Mutable: it is possible to change their content

In [9]:

```
# Run the following for Q3 and Q4
price = [100, 200, 300, "spike"]
```

Q3. A line of code to replace `"spike"` with `400`, and then print `price` variable to see the result

In [10]:

```
price[3] = 400
print(price)
```

```
[100, 200, 300, 400]
```

Q4. A line of code to add `"500"` to the last index, and then print

price

In [11]:

```
price.append(500)
print(price)
```

```
[100, 200, 300, 400, 500]
```

TUPLE

- A list of comma-separated items between parentheses ( )
- Immutable: it is not possible to change their content directly

Q5. A line of code to return the data type of `interest` variable

In [29]:

```
interest = ("SoFi", 2.5, "Chase", 4.2)
```

In [31]:

```
type(interest)
```

Out[31]:

```
tuple
```

RANGE

- Commonly used for looping a specific number of times in for loops
- Immutable sequence of numbers

Q6. Two lines of code 1) to create a variable `stock` from 0 to 100 with step of 15, and then 2) print it as a list

In [13]:

```
stock = range(0,100,15)
print(list(stock))
```

```
[0, 15, 30, 45, 60, 75, 90]
```

Q7. A line of code to find the position at the first occurence of `30` in `stock` variable

In [14]:

```
stock.index(30)
```

Out[14]:

```
2
```

DICT

- Dictionary
- Mapping object with braces like {'iphone': 100, 'ipad': 200, 'imac': 300}
- Structure: {key: value}
- Mutable

In [24]:

```
# Run this chunk to create `cap` dictionary variable for Q8, Q9, and Q10

cap = {'COST': 259, 'WMT': 424, 'TGT': 107}
```

Q8. A list of code to return the value of `COST` key

In [25]:

```
cap["COST"]
```

Out[25]:

```
259
```

Q9. A line of code to add `'HD': 317` to `cap` variable

In [33]:

```
cap["HD"] = 317
```

Q10. Create a for loop to return all the keys and values of `cap` variable

In [28]:

```
for i in cap.items():
  print(i)
```

```
('COST', 259)
('WMT', 424)
('TGT', 107)
('HD', 317)
```