Structure and Interpretation of Computer Programs with Python

Lesson 6

-Presented By: Sean Li

Project Day!

Create a Rock, Paper, Scissors Game

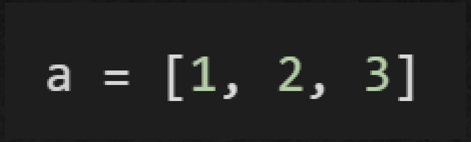
- You vs computer
- Computer generates its play using a random function
- Best 3 out of 5
- Print the message stating whether you win or lose
- Print the score for both players after each round
- After one player winning 3 times, stop the game, and display the result

Hints

1. from random import choice	1. Import the library
2. plays = ["Rock", "Paper", "Scissors"]	2. Create a list of plays
3. computer = choice(plays)	3. Choose one play at random using the “choice” library
4. player = input()	4. Wait and read user input. Will not move forward until it gets an input string
5. str(5)	5. Convert integer 5 to string “5”
6. = vs ==	6. = assigns the value on the right to the variable on the left == compares the values and evaluates to True or False

Consider using a while loop, many if-elif-else statements. Don’t forget about indentation and :

Lists



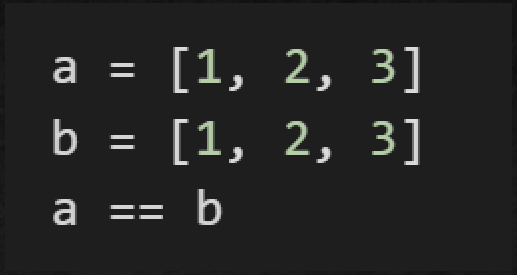
```
a = [1, 2, 3]
```

Definition

A list is a collection which is ordered and changeable. In Python, lists are written with square brackets. It's a built-in data structure.

Properties

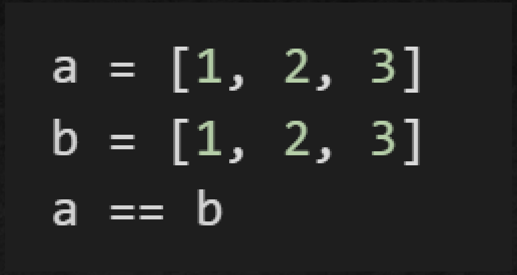
- Ordered using index



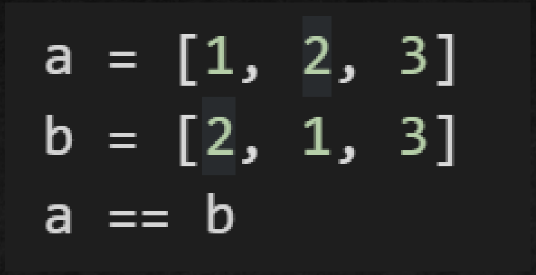
```
a = [1, 2, 3]
b = [1, 2, 3]
a == b
```

Properties

- Ordered using index



```
a = [1, 2, 3]
b = [1, 2, 3]
a == b
```



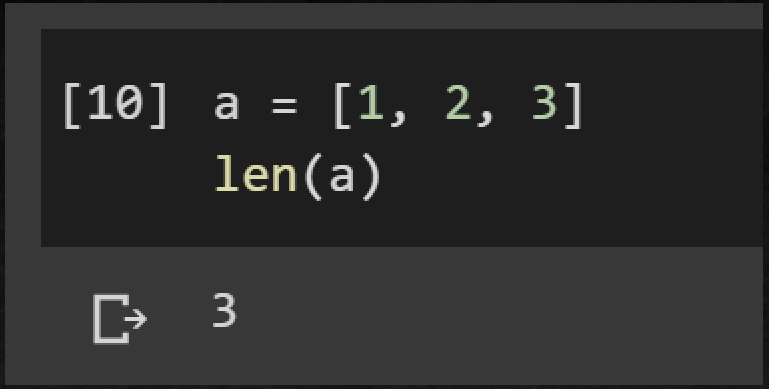
```
a = [1, 2, 3]
b = [2, 1, 3]
a == b
```

Properties

- Changeable/Mutable

Work with Lists

- Length:

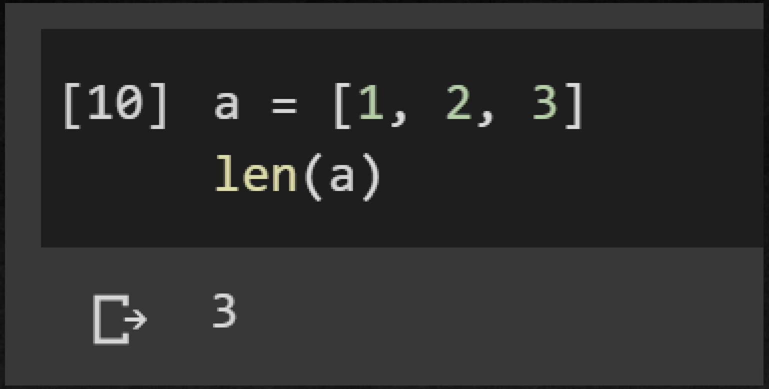


```
[10] a = [1, 2, 3]
     len(a)
```

```
3
```

Work with Lists

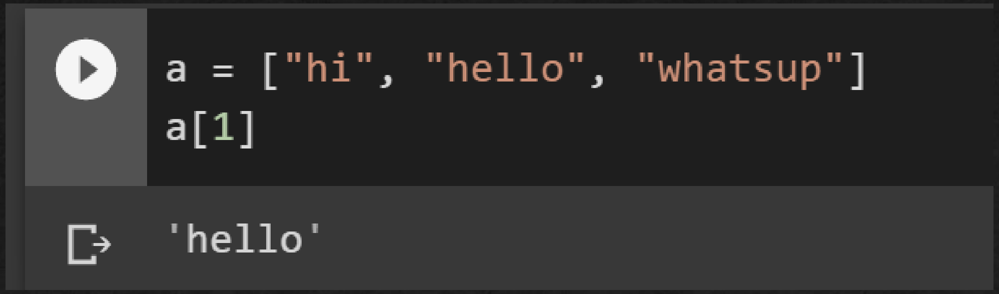
- Length:



```
[10] a = [1, 2, 3]
     len(a)
```

```
3
```

- Access an element using its index

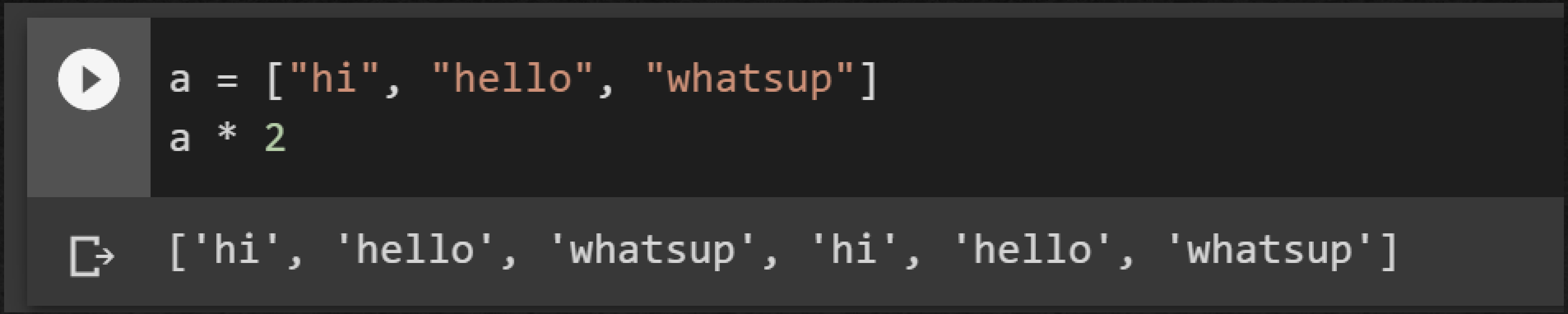


```
a = ["hi", "hello", "whatsup"]
a[1]
```

```
'hello'
```

Work with Lists

- Concatenation & Repetition

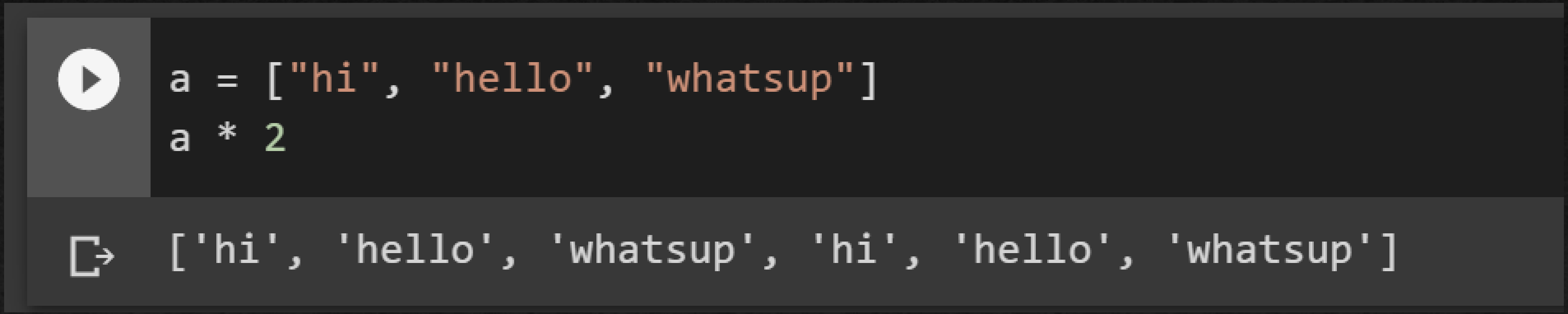


```
a = ["hi", "hello", "whatsup"]
a * 2
```

```
['hi', 'hello', 'whatsup', 'hi', 'hello', 'whatsup']
```

Work with Lists

- Concatenation & Repetition

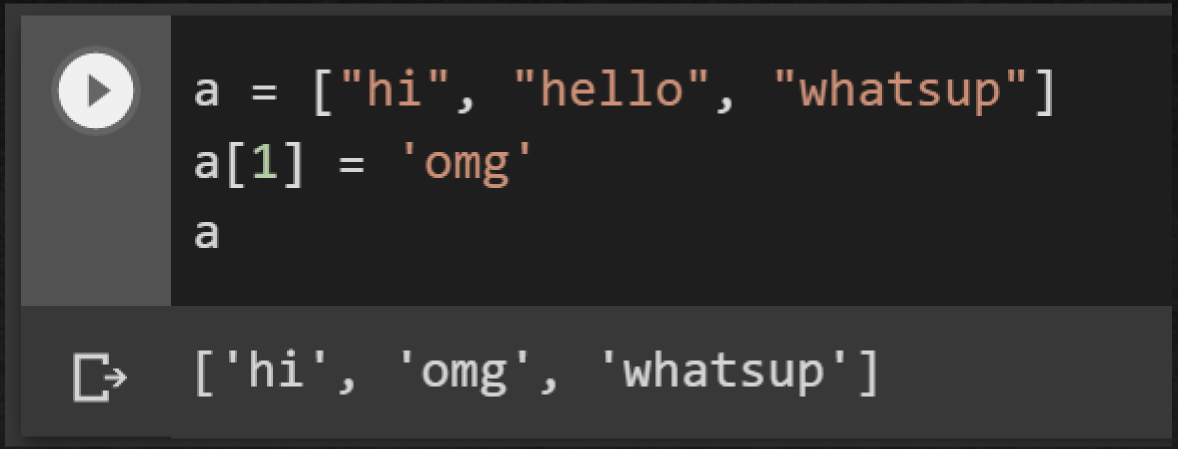


```
a = ["hi", "hello", "whatsup"]
a * 2
```

```
['hi', 'hello', 'whatsup', 'hi', 'hello', 'whatsup']
```

Work with Lists

- Mutation

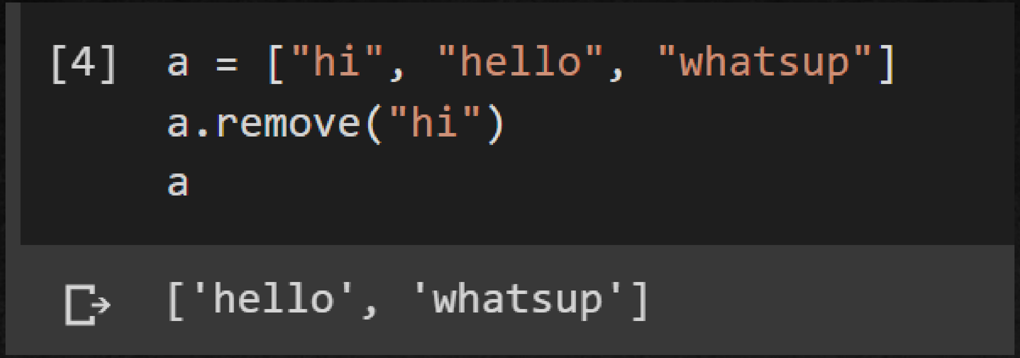


```
a = ["hi", "hello", "whatsup"]
a[1] = 'omg'
a
```

```
['hi', 'omg', 'whatsup']
```

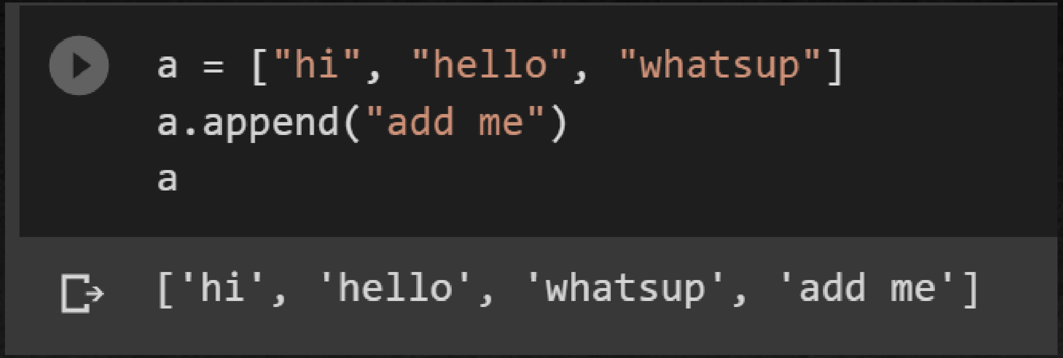
Other built-in Methods

https://www.w3schools.com/python/python_lists.asp



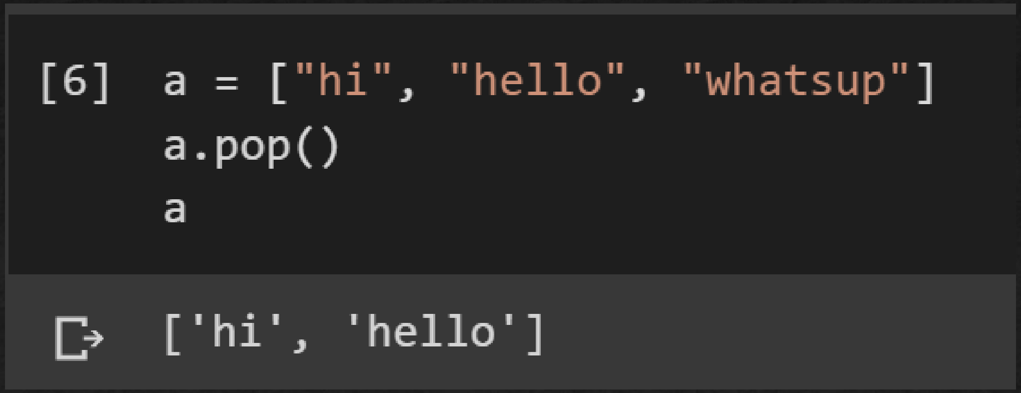
```
[4]  a = ["hi", "hello", "whatsup"]
     a.remove("hi")
     a
```

```
['hello', 'whatsup']
```



```
a = ["hi", "hello", "whatsup"]
a.append("add me")
a
```

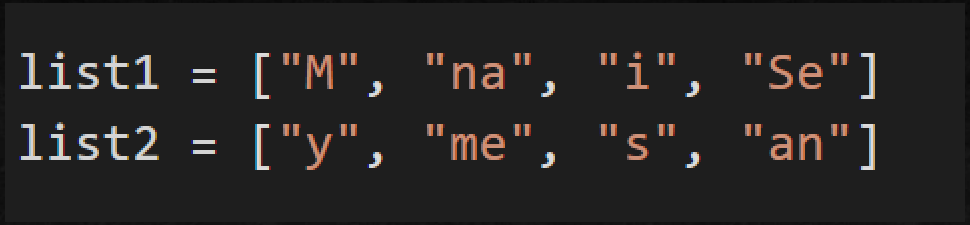
```
['hi', 'hello', 'whatsup', 'add me']
```



```
[6]  a = ["hi", "hello", "whatsup"]
     a.pop()
     a
```

```
['hi', 'hello']
```

Exercise:



```
list1 = ["M", "na", "i", "Se"]
list2 = ["y", "me", "s", "an"]
```

Create a new list that combines list1 and list2 together.

The result should have a human understandable meaning

Exercise:

You have a given list with unknown integer elements (like [1, 2, 3, 4, 5]). Iterate through the list using a for loop and find the sum of all elements

Exercise:

Create a program that creates an empty list by default. In each round, it takes in user input "add" or "remove", then it asks for the value user wants to add or remove, then execute the user's command. After each round, display the current list.