

Week 4

CS106R

Sabri **Eyuboglu** & Geoffrey **Angus**

Writing Functions with Objects

Objects and Functions

Functions **receive** objects



```
age = input_int("How old are you?")
```



Functions **return** objects

Objects and Functions

Functions **receive** objects

`cap` = `capitalize_string`(`"hello"`)

Functions **return** objects

Objects and Functions

How do we write functions that:
receive objects
and
return objects?

Objects and Functions

Example: Pythagoras 2.0

IMPORTANT IDEA

Every **function** has its own variables

Variables cannot exist across
functions

Scope

Code

```
def compute_pythag(a, b):  
    c_squared = a*a + b*b  
    c = square_root(c_squared)  
    return c  
  
def main():  
    side_1 = input_float("Enter side one:")  
    side_2 = input_float("Enter side two:")  
    hypotenuse = compute_pythag(side_1, side_2)  
    print(hypotenuse)
```

Output



Memory

Variables

Objects

Operators

Code

```
def compute_pythag(a, b):  
    c_squared = a*a + b*b  
    c = square_root(c_squared)  
    return c  
  
def main():  
    side_1 = input_float("Enter side one:")  
    side_2 = input_float("Enter side two:")  
    hypotenuse = compute_pythag(side_1, side_2)  
    print(hypotenuse)
```

Output

```
❏  
Enter side one:
```

Memory

Variables

Objects

Operators

Code

```
def compute_pythag(a, b):  
    c_squared = a*a + b*b  
    c = square_root(c_squared)  
    return c  
  
def main():  
    side_1 = input_float("Enter side one:")  
    side_2 = input_float("Enter side two:")  
    hypotenuse = compute_pythag(side_1, side_2)  
    print(hypotenuse)
```

Output

```
Enter side one: 3
```

Memory

Variables

Objects

3.0

float

Operators

Code

```
def compute_pythag(a, b):  
    c_squared = a*a + b*b  
    c = square_root(c_squared)  
    return c  
  
def main():  
    side_1 = input_float("Enter side one:")  
    side_2 = input_float("Enter side two:")  
    hypotenuse = compute_pythag(side_1, side_2)  
    print(hypotenuse)
```

Output

```
❏  
Enter side one: 3
```

Memory

Variables

side_1

Objects

3.0
float

Operators

Code

```
def compute_pythag(a, b):  
    c_squared = a*a + b*b  
    c = square_root(c_squared)  
    return c  
  
def main():  
    side_1 = input_float("Enter side one:")  
    side_2 = input_float("Enter side two:")  
    hypotenuse = compute_pythag(side_1, side_2)  
    print(hypotenuse)
```

Output

```
❏  
Enter side one: 3  
Enter side two:
```

Memory

Variables

side_1

Objects

3.0
float

Operators

Code

```
def compute_pythag(a, b):  
    c_squared = a*a + b*b  
    c = square_root(c_squared)  
    return c  
  
def main():  
    side_1 = input_float("Enter side one:")  
    side_2 = input_float("Enter side two:")  
    hypotenuse = compute_pythag(side_1, side_2)  
    print(hypotenuse)
```

Output

```
❏  
Enter side one: 3  
Enter side two: 4
```

Memory

Variables

side_1

Objects

3.0
float

4.0
float

Operators

Code

```
def compute_pythag(a, b):  
    c_squared = a*a + b*b  
    c = square_root(c_squared)  
    return c  
  
def main():  
    side_1 = input_float("Enter side one:")  
    side_2 = input_float("Enter side two:")  
    hypotenuse = compute_pythag(side_1, side_2)  
    print(hypotenuse)
```

Output

```
❏  
Enter side one: 3  
Enter side two: 4
```

Memory

Variables

side_1

side_2

Objects

3.0
float

4.0
float

Operators

Code

```
def compute_pythag(a, b):  
    c_squared = a*a + b*b  
    c = square_root(c_squared)  
    return c  
  
def main():  
    side_1 = input_float("Enter side one:")  
    side_2 = input_float("Enter side two:")  
    hypotenuse = compute_pythag(side_1, side_2)  
    print(hypotenuse)
```

Output

```
❏  
Enter side one: 3  
Enter side two: 4
```

Memory

Variables

side_1

side_2

Objects

3.0
float

4.0
float

Operators

Code

```
def compute_pythag(a, b):  
    c_squared = a*a + b*b  
    c = square_root(c_squared)  
    return c  
  
def main():  
    side_1 = input_float("Enter side one:")  
    side_2 = input_float("Enter side two:")  
    hypotenuse = compute_pythag(side_1, side_2)  
    print(hypotenuse)
```

Output

```
❏  
Enter side one: 3  
Enter side two: 4
```

Memory

Variables

side_1

a

side_2

b

Objects

3.0
float

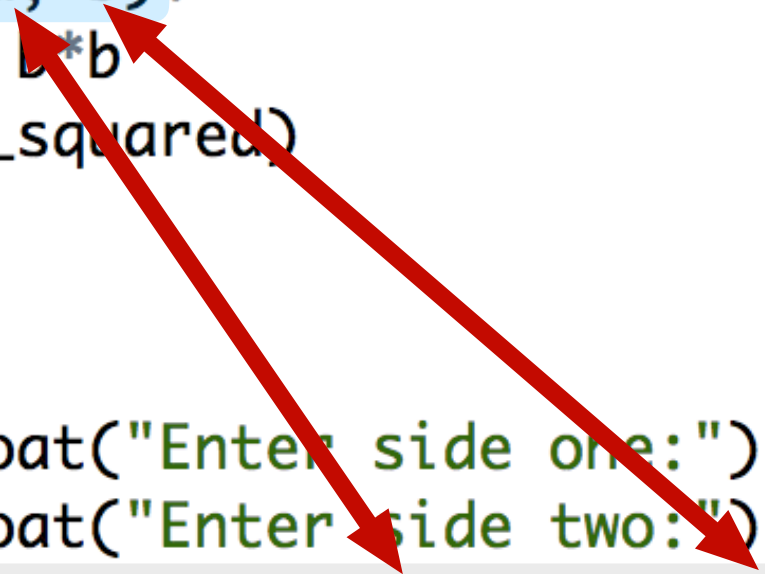
4.0
float

25.0
float

Operators

Code

```
def compute_pythag(a, b):  
    c_squared = a*a + b*b  
    c = square_root(c_squared)  
    return c  
  
def main():  
    side_1 = input_float("Enter side one:")  
    side_2 = input_float("Enter side two:")  
    hypotenuse = compute_pythag(side_1, side_2)  
    print(hypotenuse)
```



Memory

Variables

side_1

a

side_2

b

Objects

3.0

float

4.0

float

25.0

float

Output

```
❏  
Enter side one: 3  
Enter side two: 4
```

Operators

Code

```
def compute_pythag(a, b):  
    c_squared = a*a + b*b  
    c = square_root(c_squared)  
    return c  
  
def main():  
    side_1 = input_float("Enter side one:")  
    side_2 = input_float("Enter side two:")  
    hypotenuse = compute_pythag(side_1, side_2)  
    print(hypotenuse)
```

Output

```
❏  
Enter side one: 3  
Enter side two: 4
```

Memory

Variables

side_1

a

side_2

b

Objects

3.0

float

4.0

float

25.0

float

Operators

Code

```
def compute_pythag(a, b):  
    c_squared = a*a + b*b  
    c = square_root(c_squared)  
    return c  
  
def main():  
    side_1 = input_float("Enter side one:")  
    side_2 = input_float("Enter side two:")  
    hypotenuse = compute_pythag(side_1, side_2)  
    print(hypotenuse)
```

Output

```
❏  
Enter side one: 3  
Enter side two: 4
```

Memory

Variables

side_1

a

side_2

b

c_squared

Objects

3.0

float

4.0

float

25.0

float

Operators

Code

```
def compute_pythag(a, b):  
    c_squared = a*a + b*b  
    c = square_root(c_squared)  
    return c  
  
def main():  
    side_1 = input_float("Enter side one:")  
    side_2 = input_float("Enter side two:")  
    hypotenuse = compute_pythag(side_1, side_2)  
    print(hypotenuse)
```

Output

```
Enter side one: 3  
Enter side two: 4
```

Memory

Variables

side_1

a

side_2

b

c_squared

Objects

3.0

float

4.0

float

25.0

float

5.0

float

Operators

Code

```
def compute_pythag(a, b):  
    c_squared = a*a + b*b  
    c = square_root(c_squared)  
    return c  
  
def main():  
    side_1 = input_float("Enter side one:")  
    side_2 = input_float("Enter side two:")  
    hypotenuse = compute_pythag(side_1, side_2)  
    print(hypotenuse)
```

Output

```
Enter side one: 3  
Enter side two: 4
```

Memory

Variables

side_1

a

side_2

b

c_squared

c

Objects

3.0
float

4.0
float

25.0
float

5.0
float

Operators

Code

```
def compute_pythag(a, b):  
    c_squared = a*a + b*b  
    c = square_root(c_squared)  
    return c  
  
def main():  
    side_1 = input_float("Enter side one:")  
    side_2 = input_float("Enter side two:")  
    hypotenuse = compute_pythag(side_1, side_2)  
    print(hypotenuse)
```

Output

```
Enter side one: 3  
Enter side two: 4
```

Memory

Variables

side_1

a

side_2

b

c_squared

c

Objects

3.0
float

4.0
float

25.0
float

5.0
float

Operators

Code

```
def compute_pythag(a, b):  
    c_squared = a*a + b*b  
    c = square_root(c_squared)  
    return c  
  
def main():  
    side_1 = input_float("Enter side one:")  
    side_2 = input_float("Enter side two:")  
    hypotenuse = compute_pythag(side_1, side_2)  
    print(hypotenuse)
```

Output

```
Enter side one: 3  
Enter side two: 4
```

Memory

Variables

side_1

a

side_2

b

c_squared

c

Objects

3.0

float

4.0

float

25.0

float

5.0

float

Operators

Code

```
def compute_pythag(a, b):  
    c_squared = a*a + b*b  
    c = square_root(c_squared)  
    return c  
  
def main():  
    side_1 = input_float("Enter side one:")  
    side_2 = input_float("Enter side two:")  
    hypotenuse = compute_pythag(side_1, side_2)  
    print(hypotenuse)
```

Output

```
Enter side one: 3  
Enter side two: 4
```

Memory

Variables

side_1

a

side_2

b

c_squared

c

hypotenuse

Objects

3.0

float

4.0

float

25.0

float

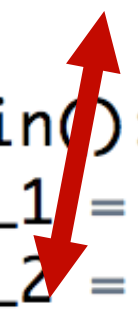
5.0

float

Operators

Code

```
def compute_pythag(a, b):  
    c_squared = a*a + b*b  
    c = square_root(c_squared)  
    return c  
  
def main():  
    side_1 = input_float("Enter side one:")  
    side_2 = input_float("Enter side two:")  
    hypotenuse = compute_pythag(side_1, side_2)  
    print(hypotenuse)
```



Output

```
Enter side one: 3  
Enter side two: 4
```

Memory

Variables

side_1

a

side_2

b

c_squared

c

hypotenuse

Objects

3.0

float

4.0

float

25.0

float

5.0

float

Operators

Code

```
def compute_pythag(a, b):  
    c_squared = a*a + b*b  
    c = square_root(c_squared)  
    return c  
  
def main():  
    side_1 = input_float("Enter side one:")  
    side_2 = input_float("Enter side two:")  
    hypotenuse = compute_pythag(side_1, side_2)  
    print(hypotenuse)
```

Output

```
Enter side one: 3  
Enter side two: 4
```

Memory

Variables

side_1

side_2

hypotenuse

Objects

3.0
float

4.0
float

25.0
float

5.0
float

Operators

Code

```
def compute_pythag(a, b):  
    c_squared = a*a + b*b  
    c = square_root(c_squared)  
    return c  
  
def main():  
    side_1 = input_float("Enter side one:")  
    side_2 = input_float("Enter side two:")  
    hypotenuse = compute_pythag(side_1, side_2)  
    print(hypotenuse)
```

Output

```
Enter side one: 3  
Enter side two: 4  
5.0
```

Memory

Variables

side_1

side_2

hypotenuse

Objects

3.0
float

4.0
float

25.0
float

5.0
float

Function Breakdown

```
def compute_pythag(a, b):  
    c_squared = a*a + b*b  
    c = square_root(c_squared)  
    return c
```

Function Breakdown

```
def compute_pythag(a, b):  
    c_squared = a*a + b*b  
    c = square_root(c_squared)  
    return c
```

Function Breakdown

Parameters

These are just variables

```
def compute_pythag(a, b):  
    c_squared = a*a + b*b  
    c = square_root(c_squared)  
    return c
```

Function Breakdown

```
hyp = compute_pythag(side_1, side_2)
```

is like...

```
a = side_1  
b = side_2
```

```
def compute_pythag(a, b):  
    c_squared = a*a + b*b  
    c = square_root(c_squared)  
    return c
```

Function Breakdown

```
def compute_pythag(a, b):  
    c_squared = a*a + b*b  
    c = square_root(c_squared)  
    return c
```


Function Breakdown

```
def compute_pythag(a, b):  
    c_squared = a*a + b*b  
    c = square_root(c_squared)  
    return c
```

Function Breakdown

```
hyp = compute_pythag(side_1, side_2)
```

is like...

```
hyp = c
```

```
def compute_pythag(a, b):  
    c_squared = a*a + b*b  
    c = square_root(c_squared)  
    return c
```

Function Breakdown

```
def compute_pythag(a, b):  
    c_squared = a*a + b*b  
    c = square_root(c_squared)  
    return c
```

Return

The object the function gives back

Today's Exercises

Calculator

Conversion

Bilheteria

Today's Exercises

Calculator

Conversion

Bilheteria
