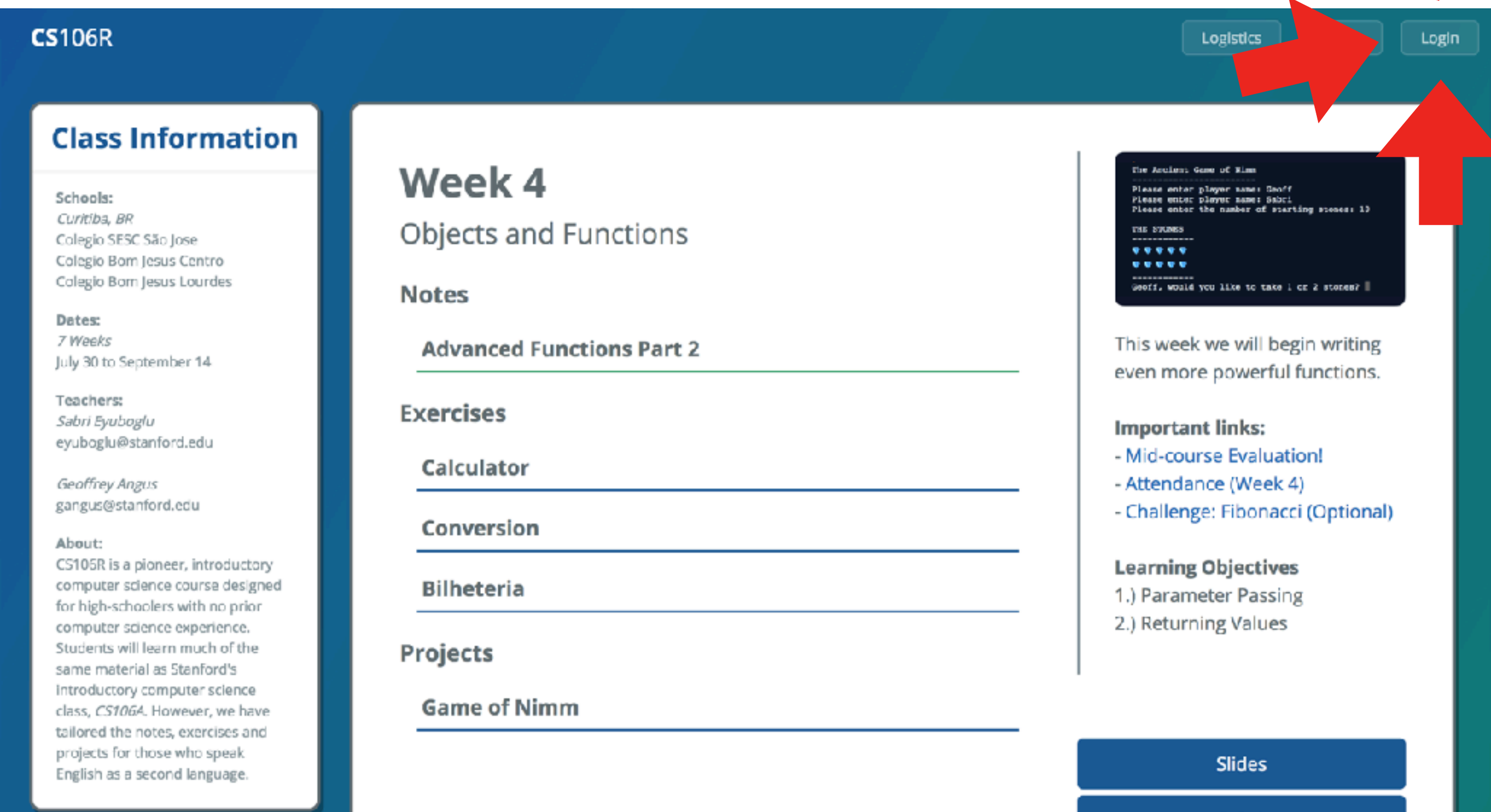
geoffreyangus.github.io/CS106R/



Week 4

CS106R

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

Last week on CS106R...

Last week on CS106R...

Objects

Variables

Operators

`5` `"Hello, World!"`

`favorite_singer`

`+` `-` `*` `/` `>` `==` `and`

Objects

4 Basic Object Classes

string

Sequences of characters – text

Example

```
"Hello, World!"
```

int

Integers – whole numbers

Examples

```
5   3450   0   -17   1
```

float

Fractional numbers

Examples

```
-5.0   0.174   3.14
```

bool

True or false

Examples

```
True   False
```

Variables

Label Objects with **Variables**

```
favorite_singer = "Beyonce"
```

variable — label operator — string object

Input Functions

```
favorite_singer = input_string("Who is your favorite singer?")
```

Output

```
>
Who is your favorite singer?
```

Input Functions

`input_string(`*`message`*`)` Returns a `string` input by user.

`input_int(`*`message`*`)` Returns a `int` input by user.

`input_float(`*`message`*`)` Returns a `float` input by user.

`input_bool(`*`message`*`)` Returns a `bool` input by user.

Operators

```
number = 5 * 3
print(number)
```

Output

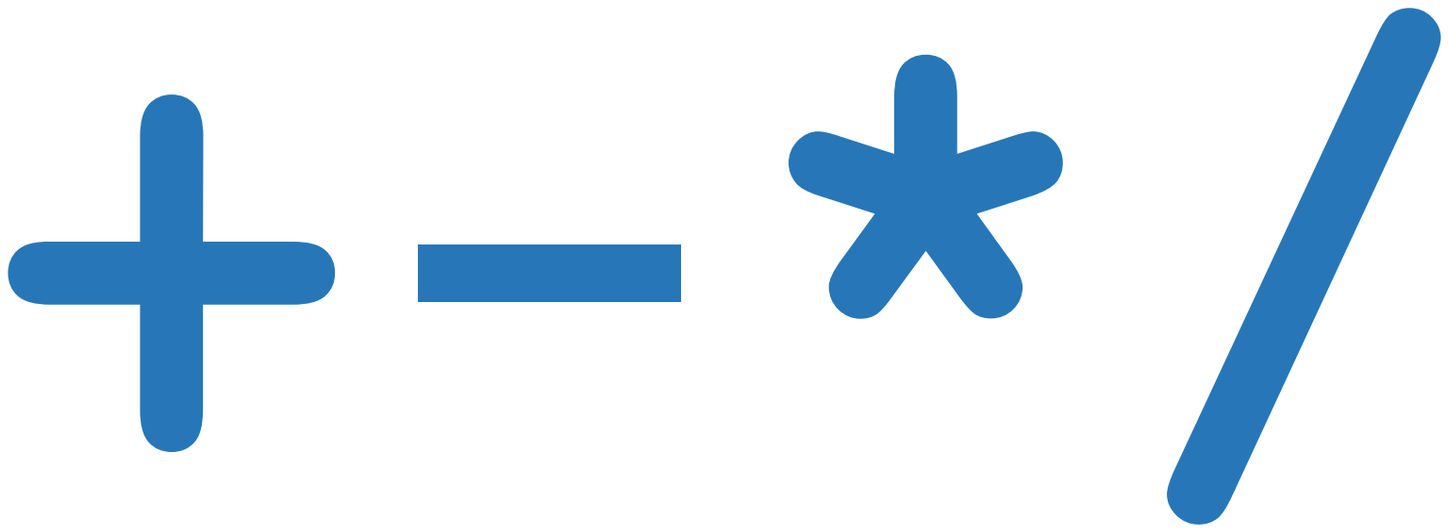
Operators

```
number = 15
print(number)
```

Output

```
15
```

Operators



Operators

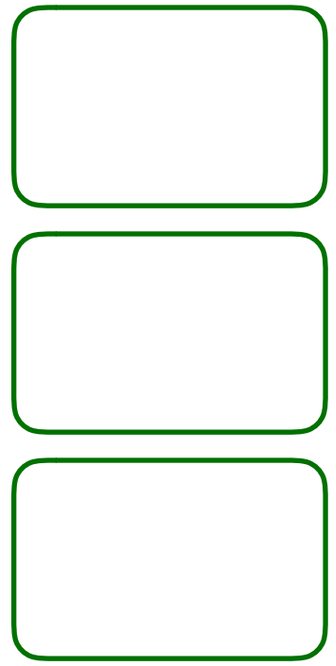
Code

```
def main():
    num_1 = input_int("Enter first int:")
    num_2 = input_int("Enter second int:")
    sum = num_1 + num_2
    print(sum)
```

Memory

Variables

Object



Output

Arithmetic Operators

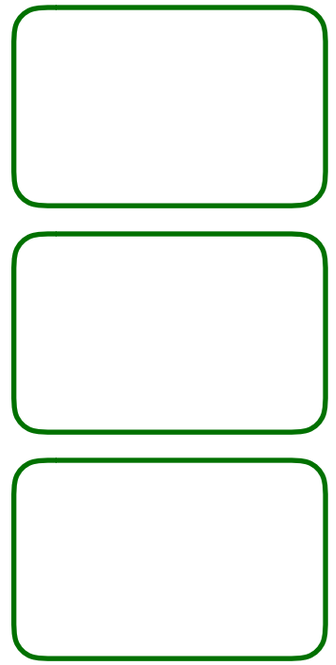
Code

```
def main():
    num_1 = input_int("Enter first int:")
    num_2 = input_int("Enter second int:")
    sum = num_1 + num_2
    print(sum)
```

Memory

Variables

Object



Output

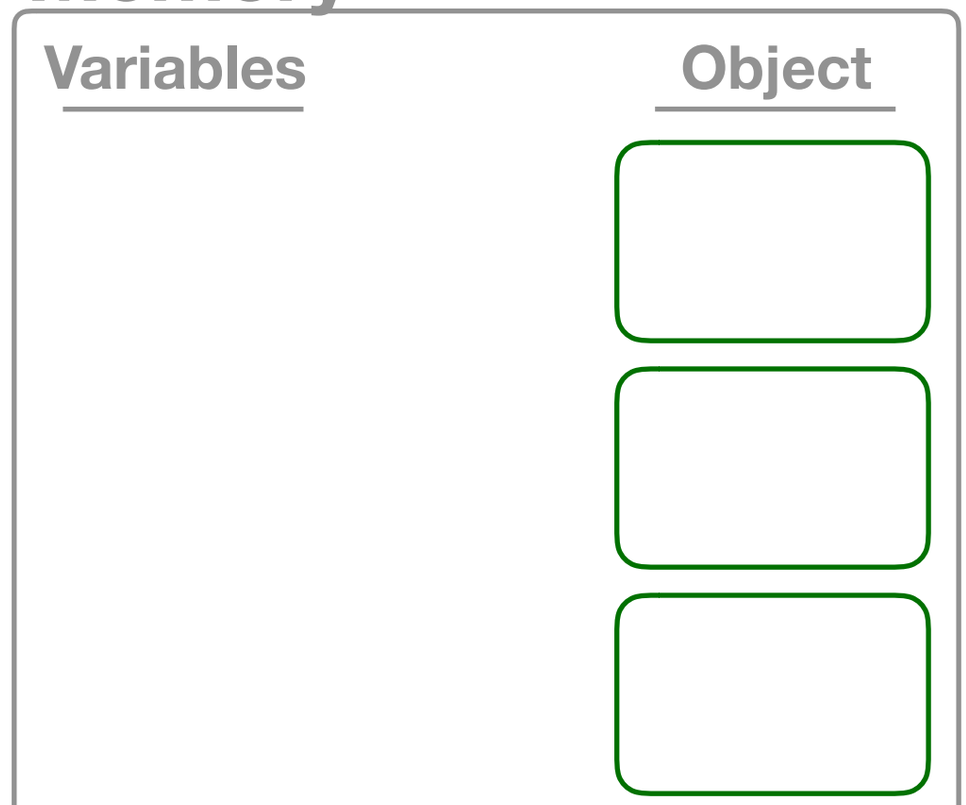
```
>
Enter first int:
```

Arithmetic Operators

Code

```
def main():
    num_1 = input_int("Enter first int:")
    num_2 = input_int("Enter second int:")
    sum = num_1 + num_2
    print(sum)
```

Memory



Output

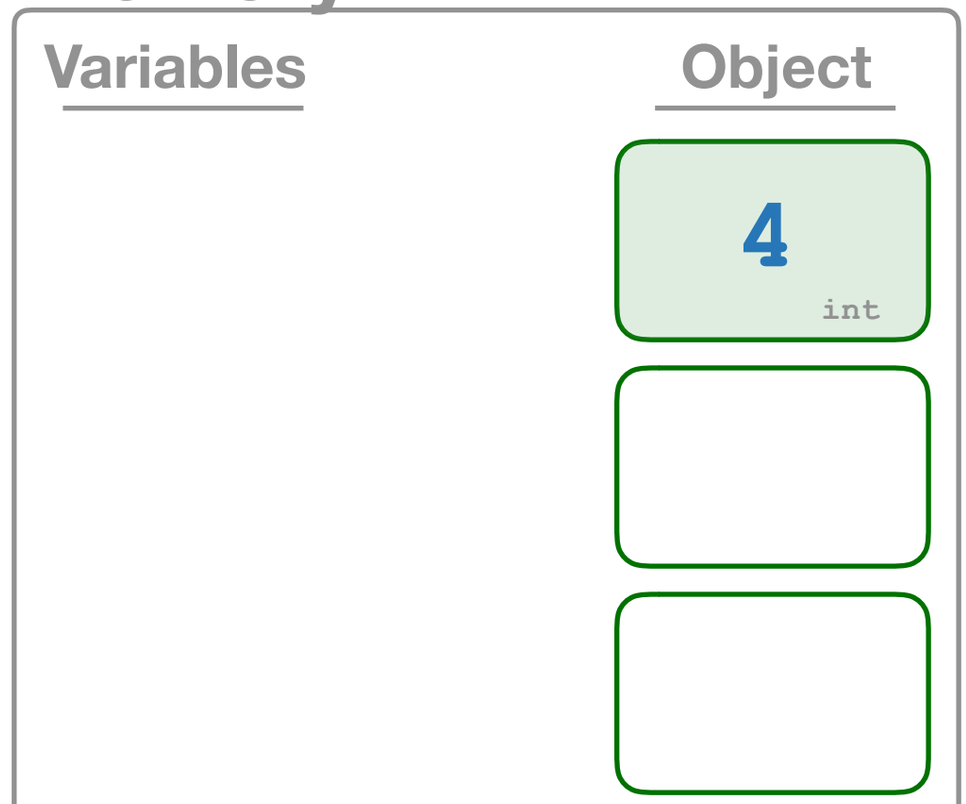
```
>
Enter first int: 4
```

Arithmetic Operators

Code

```
def main():
    num_1 = input_int("Enter first int:")
    num_2 = input_int("Enter second int:")
    sum = num_1 + num_2
    print(sum)
```

Memory



Output

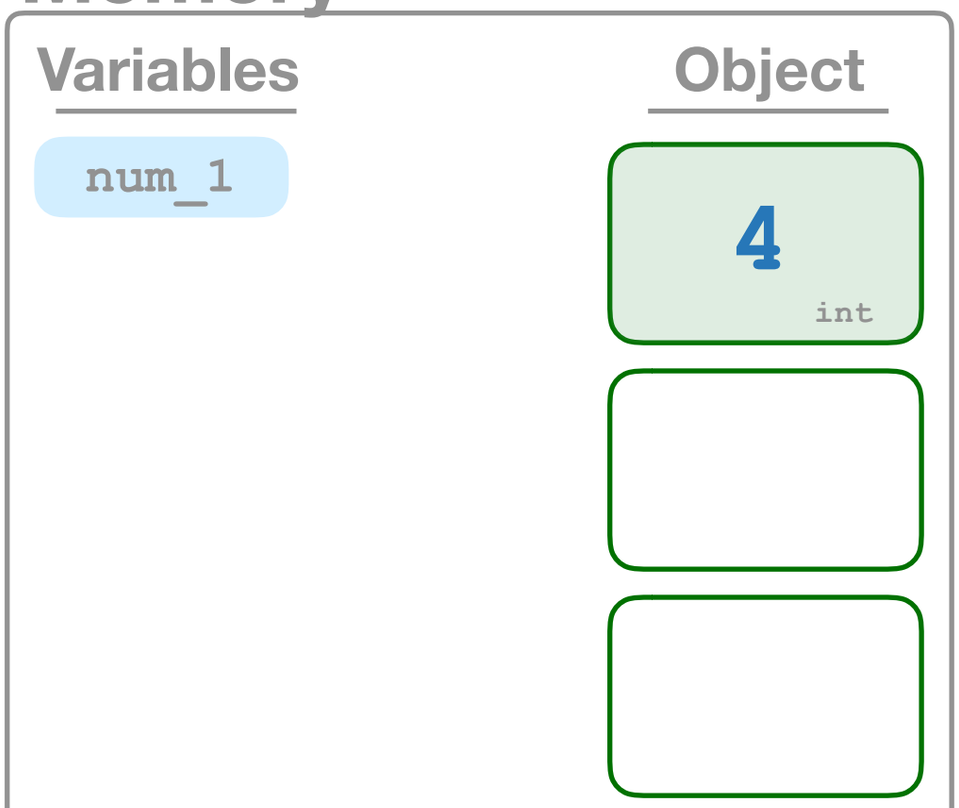
```
>
Enter first int: 4
```

Arithmetic Operators

Code

```
def main():
    num_1 = input_int("Enter first int:")
    num_2 = input_int("Enter second int:")
    sum = num_1 + num_2
    print(sum)
```

Memory



Output

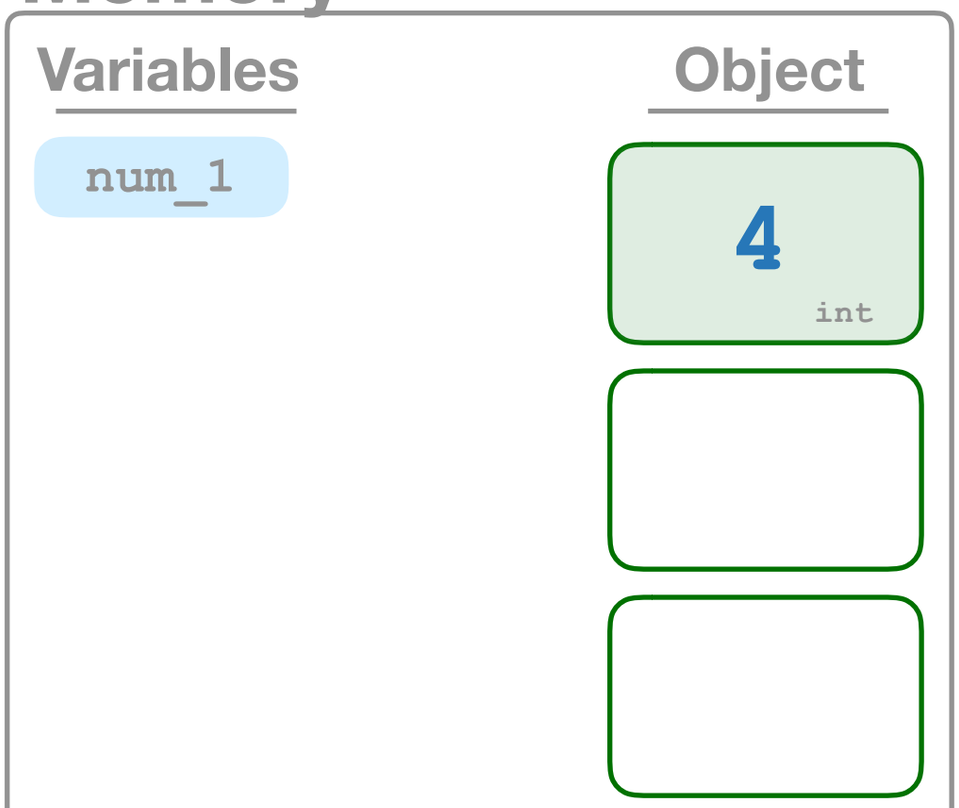
```
>
Enter first int: 4
```

Arithmetic Operators

Code

```
def main():
    num_1 = input_int("Enter first int:")
    num_2 = input_int("Enter second int:")
    sum = num_1 + num_2
    print(sum)
```

Memory



Output

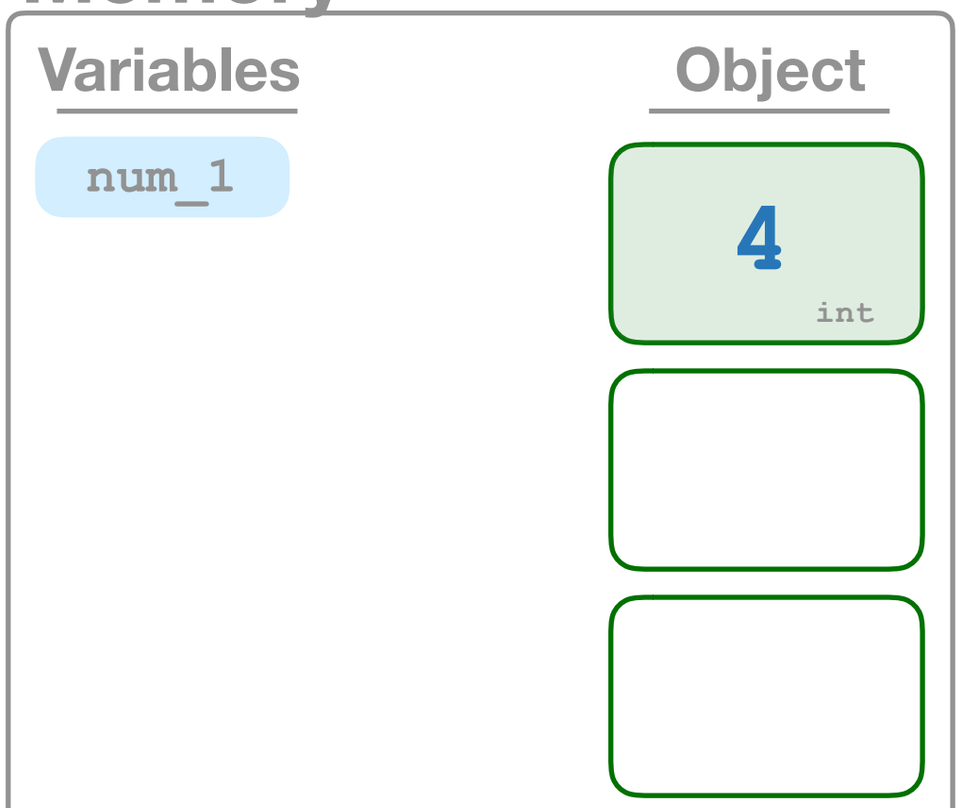
```
>
Enter first int: 4
Enter second int:
```

Arithmetic Operators

Code

```
def main():
    num_1 = input_int("Enter first int:")
    num_2 = input_int("Enter second int:")
    sum = num_1 + num_2
    print(sum)
```

Memory



Output

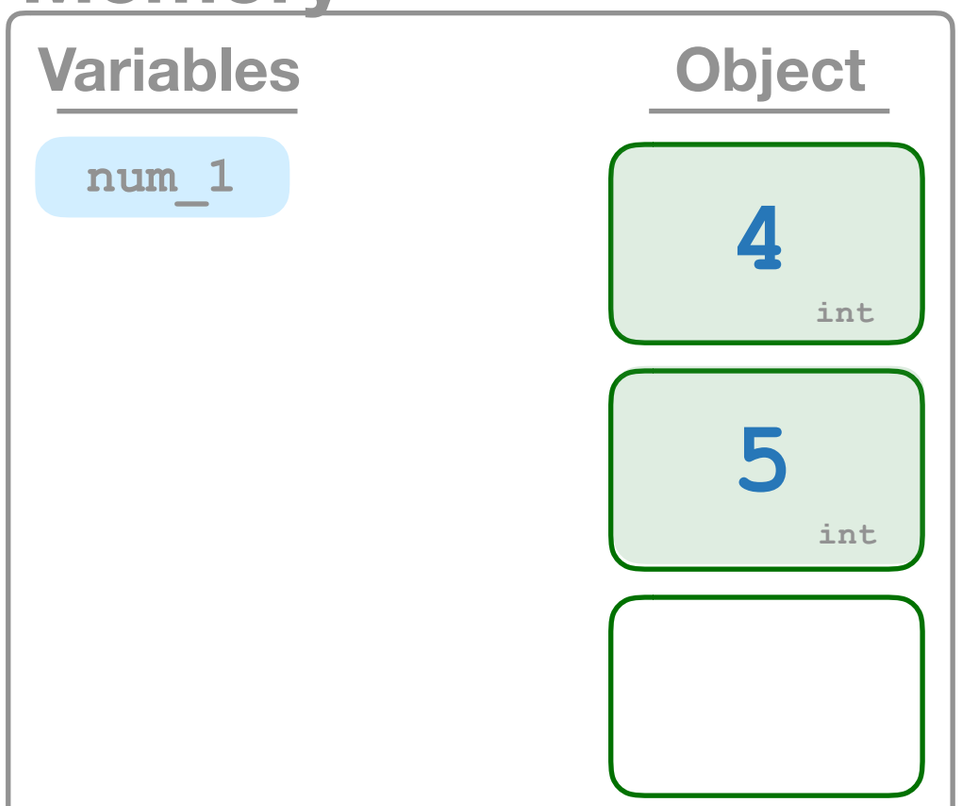
```
>
Enter first int: 4
Enter second int: 5
```

Arithmetic Operators

Code

```
def main():
    num_1 = input_int("Enter first int:")
    num_2 = input_int("Enter second int:")
    sum = num_1 + num_2
    print(sum)
```

Memory



Output

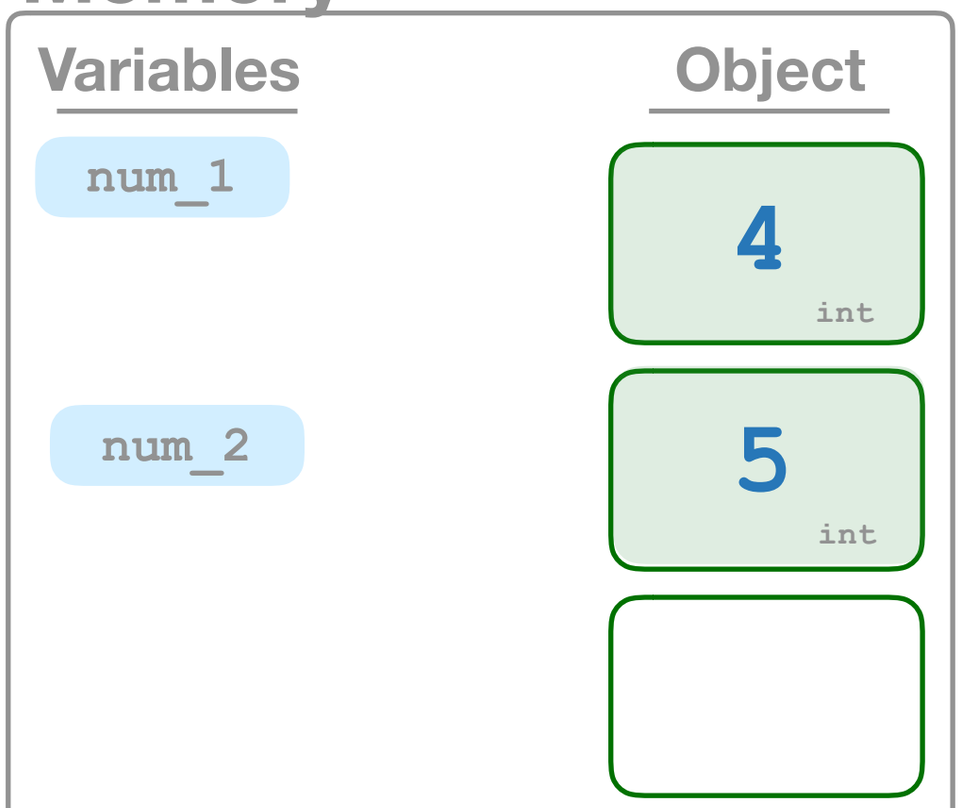
```
>
Enter first int: 4
Enter second int: 5
```

Arithmetic Operators

Code

```
def main():
    num_1 = input_int("Enter first int:")
    num_2 = input_int("Enter second int:")
    sum = num_1 + num_2
    print(sum)
```

Memory



Output

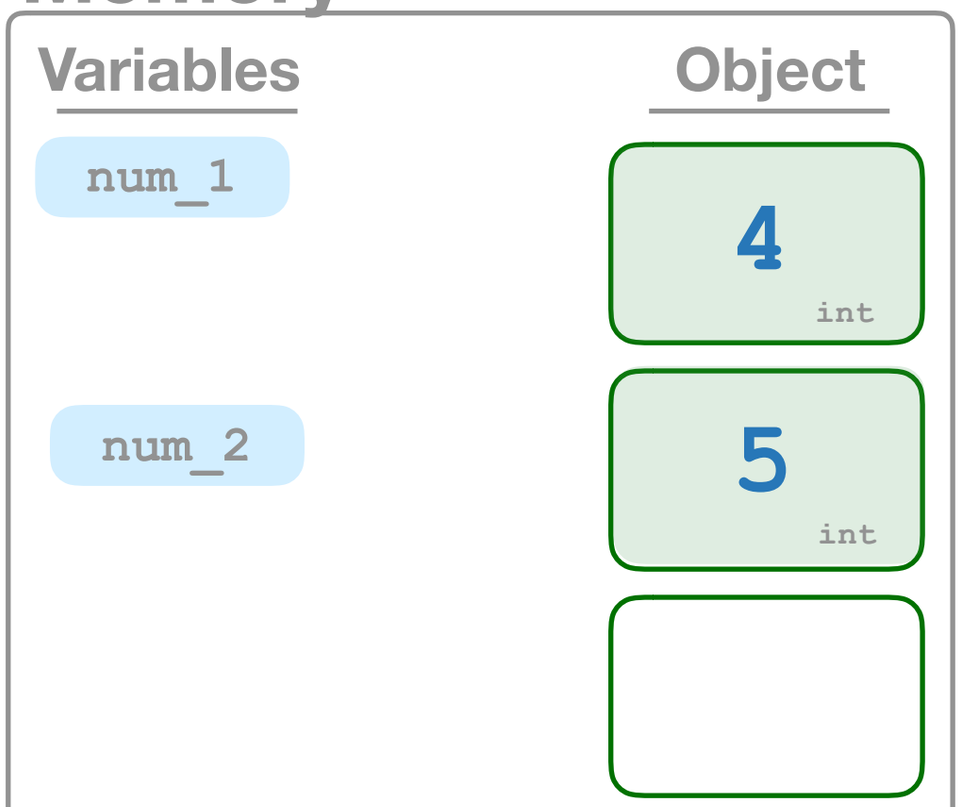
```
>
Enter first int: 4
Enter second int: 5
```

Arithmetic Operators

Code

```
def main():
    num_1 = input_int("Enter first int:")
    num_2 = input_int("Enter second int:")
    sum = num_1 + num_2
    print(sum)
```

Memory



Output

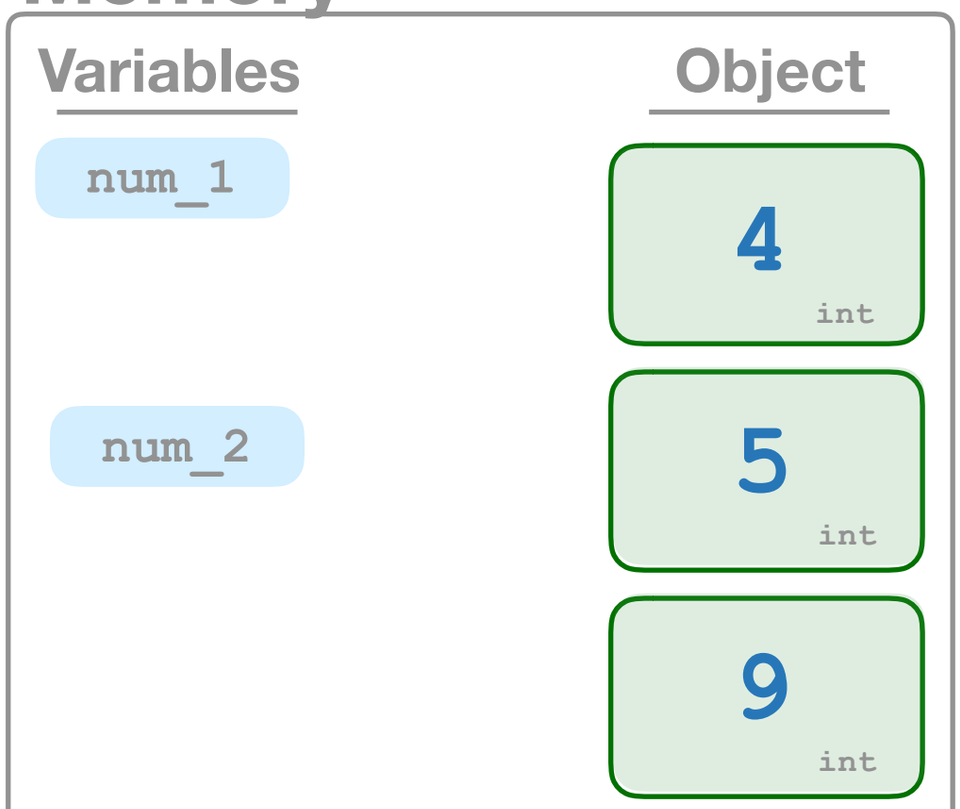
```
>
Enter first int: 4
Enter second int: 5
```

Arithmetic Operators

Code

```
def main():
    num_1 = input_int("Enter first int:")
    num_2 = input_int("Enter second int:")
    sum = num_1 + num_2
    print(sum)
```

Memory



Output

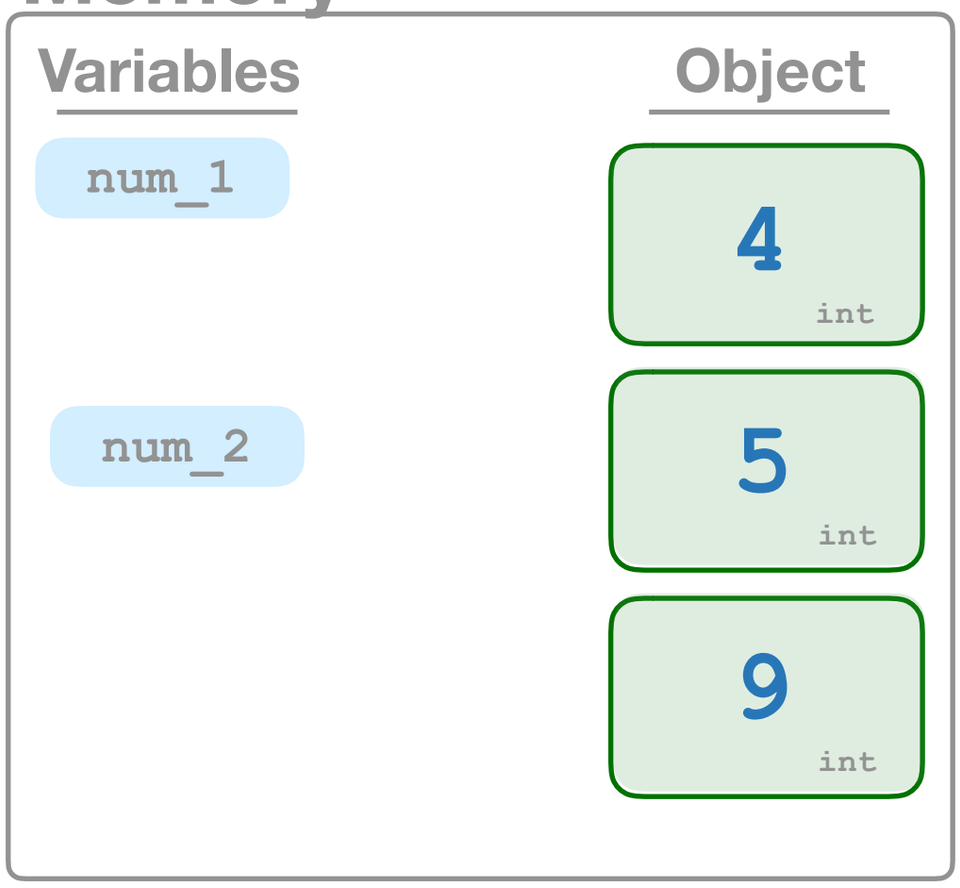
```
>
Enter first int: 4
Enter second int: 5
```

Arithmetic Operators

Code

```
def main():
    num_1 = input_int("Enter first int:")
    num_2 = input_int("Enter second int:")
    sum = num_1 + num_2
    print(sum)
```

Memory



Output

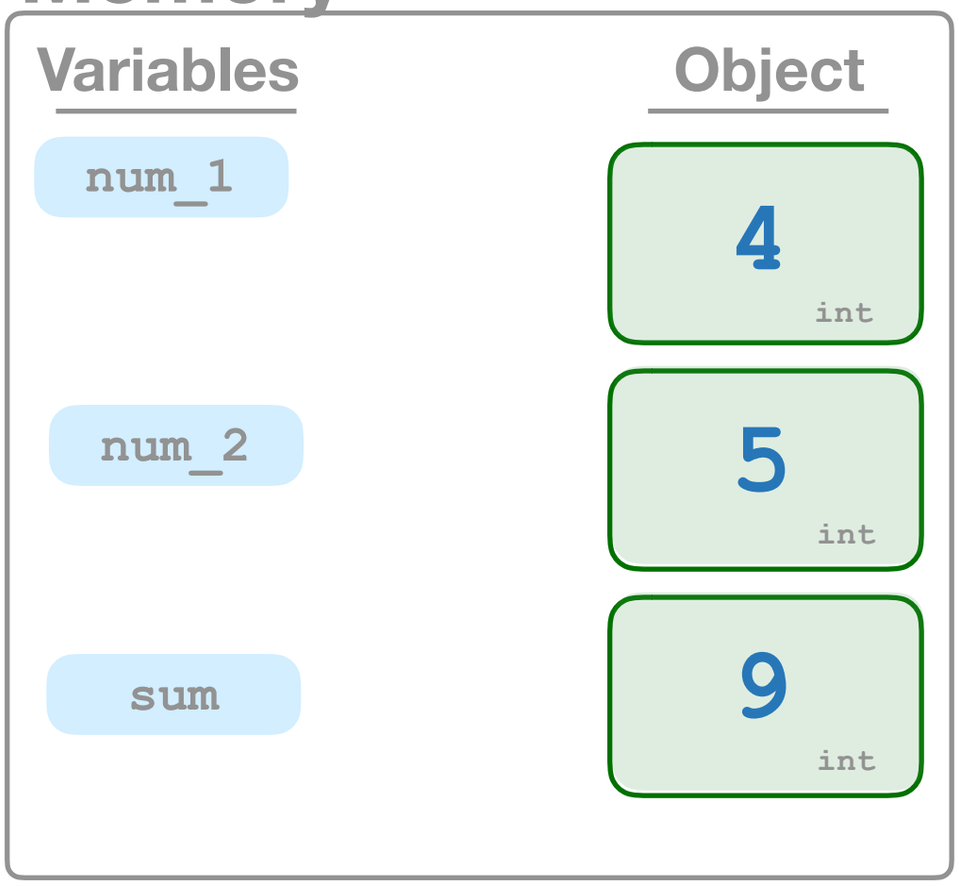
```
>
Enter first int: 4
Enter second int: 5
```

Arithmetic Operators

Code

```
def main():
    num_1 = input_int("Enter first int:")
    num_2 = input_int("Enter second int:")
    sum = num_1 + num_2
    print(sum)
```

Memory



Output

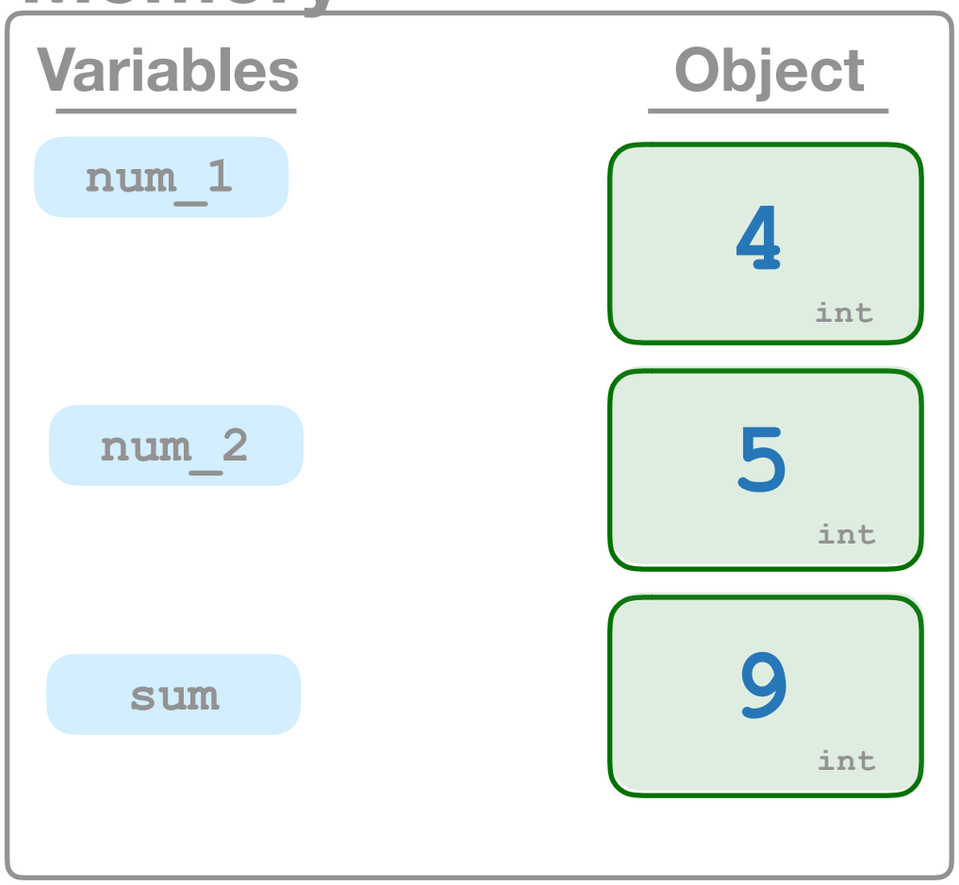
```
>
Enter first int: 4
Enter second int: 5
```

Arithmetic Operators

Code

```
def main():
    num_1 = input_int("Enter first int:")
    num_2 = input_int("Enter second int:")
    sum = num_1 + num_2
    print(sum)
```

Memory



Output

```
>
Enter first int: 4
Enter second int: 5
9
```

Operators

Operators

Code

```
def main():
  favorite_food = input_string("Favorite food?")
  if favorite_food == "Pão de Queijo":
    print("Me too!")
  else:
    print("Have you tried Pão de Queijo though?")
```

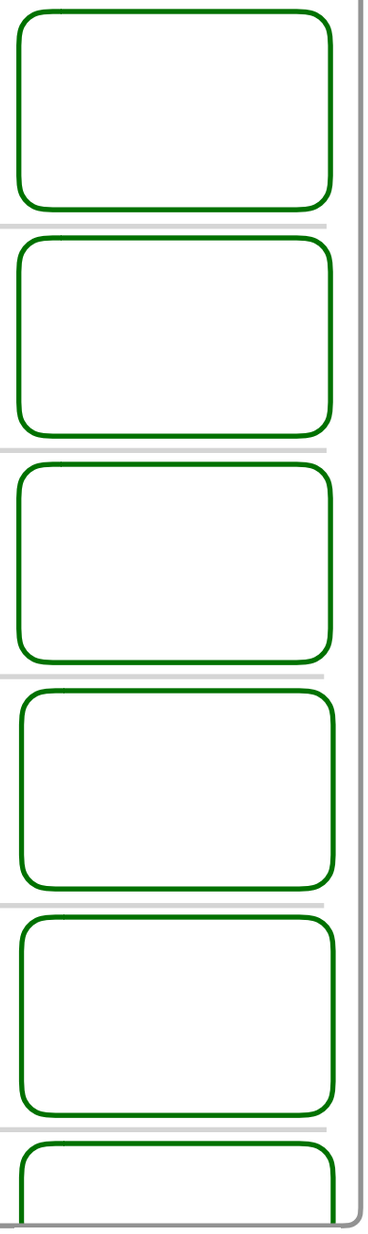
Output

```
>
```

Memory

Variables

Objects



Operators

Code

```
def main():
  favorite_food = input_string("Favorite food?")
  if favorite_food == "Pão de Queijo":
    print("Me too!")
  else:
    print("Have you tried Pão de Queijo though?")
```

Output

```
>
Favorite food?
```

Memory

Variables

Objects

Operators

Code

```
def main():
  favorite_food = input_string("Favorite food?")
  if favorite_food == "Pão de Queijo":
    print("Me too!")
  else:
    print("Have you tried Pão de Queijo though?")
```

Output

```
Favorite food? Pão de Queijo
```

Memory

Variables

Objects

Operators

Code

```
def main():
  favorite_food = input_string("Favorite food?")
  if favorite_food == "Pão de Queijo":
    print("Me too!")
  else:
    print("Have you tried Pão de Queijo though?")
```

Output

```
>
Favorite food? Pão de Queijo
```

Memory

Variables | Objects

Operators

Code

```
def main():
    favorite_food = input_string("Favorite food?")
    if favorite_food == "Pão de Queijo":
        print("Me too!")
    else:
        print("Have you tried Pão de Queijo though?")
```

Output

```
Favorite food? Pão de Queijo
```

Memory

Variables | Objects

"Favorite food?" String

"Pão de Queijo" String

Operators

Code

```
def main():
    favorite_food = input_string("Favorite food?")
    if favorite_food == "Pão de Queijo":
        print("Me too!")
    else:
        print("Have you tried Pão de Queijo though?")
```

Output

```
>
Favorite food? Pão de Queijo
```

Memory

Operators

Code

```
def main():
  favorite_food = input_string("Favorite food?")
  if favorite_food == "Pão de Queijo":
    print("Me too!")
  else:
    print("Have you tried Pão de Queijo though?")
```

Output

```
>
Favorite food? Pão de Queijo
```

Memory

Operators

Code

```
def main():
    favorite_food = input_string("Favorite food?")
    if favorite_food == "Pão de Queijo":
        print("Me too!")
    else:
        print("Have you tried Pão de Queijo though?")
```

Output

```
>
Favorite food? Pão de Queijo
```

Memory

Operators

Code

```
def main():
    favorite_food = input_string("Favorite food?")
    if favorite_food == "Pão de Queijo":
        print("Me too!")
    else:
        print("Have you tried Pão de Queijo though?")
```

Output

```
>
Favorite food? Pão de Queijo
```

Memory

Operators

Code

```
def main():
    favorite_food = input_string("Favorite food?")
    if favorite_food == "Pão de Queijo":
        print("Me too!")
    else:
        print("Have you tried Pão de Queijo though?")
```

Output

```
>
Favorite food? Pão de Queijo
```

Memory

Operators

Code

```
def main():
    favorite_food = input_string("Favorite food?")
    if favorite_food == "Pão de Queijo":
        print("Me too!")
    else:
        print("Have you tried Pão de Queijo though?")
```

Output

```
>
Favorite food? Pão de Queijo
Me too!
```

Memory

Operators

Operators

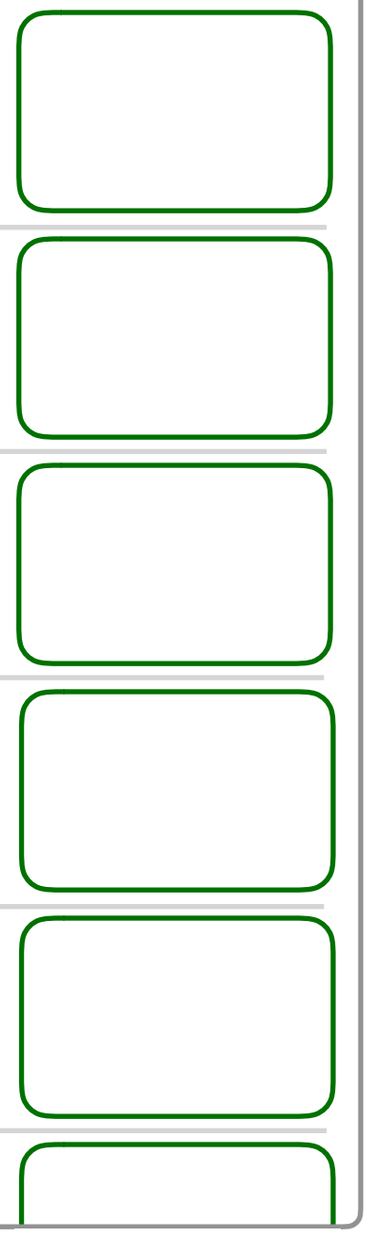
Code

```
def main():
    age = input_int("How old are you?")
    if age >= 18:
        print("You can vote!")
    else:
        wait_time = 18 - age
        print("You can't vote!")
        print(wait_time)
```

Output

Memory

Variables | Objects



Operators

Code

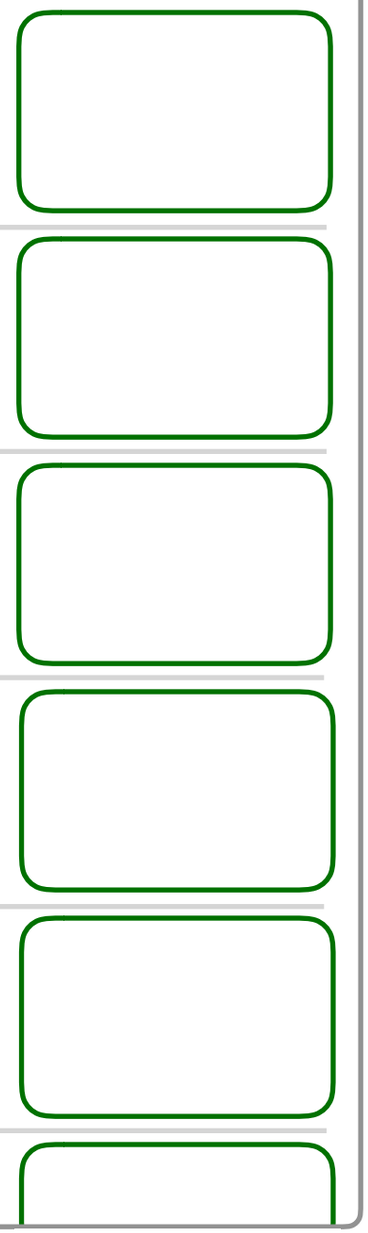
```
def main():
    age = input_int("How old are you?")
    if age >= 18:
        print("You can vote!")
    else:
        wait_time = 18 - age
        print("You can't vote!")
        print(wait_time)
```

Output

```
How old are you?
```

Memory

Variables | Objects



Operators

Code

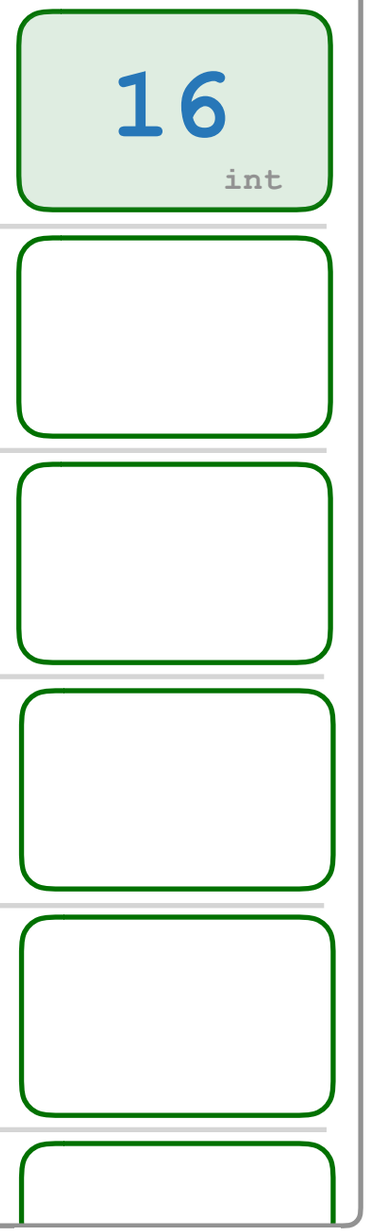
```
def main():
    age = input_int("How old are you?")
    if age >= 18:
        print("You can vote!")
    else:
        wait_time = 18 - age
        print("You can't vote!")
        print(wait_time)
```

Output

```
How old are you? 16
```

Memory

Variables | Objects



Operators

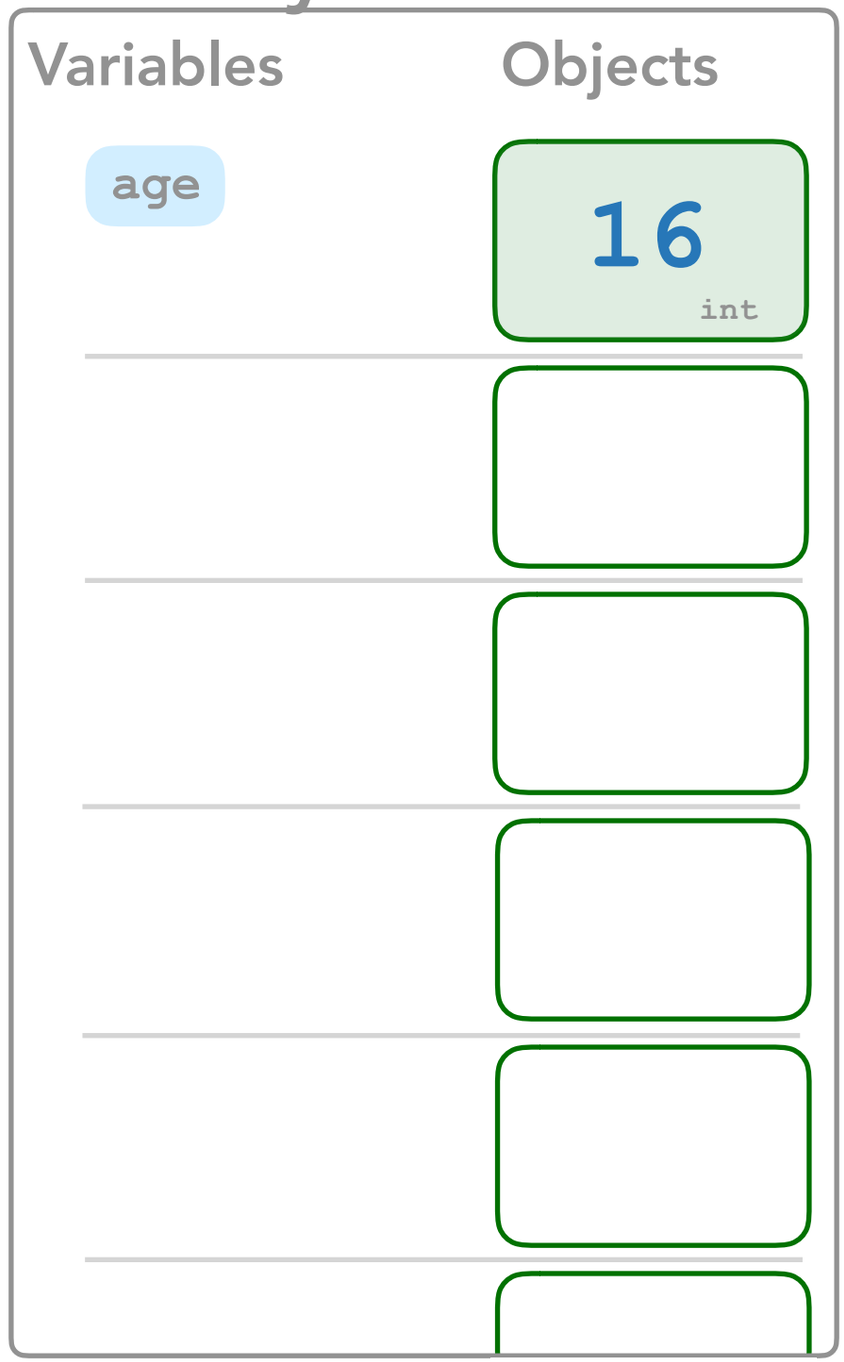
Code

```
def main():
    age = input_int("How old are you?")
    if age >= 18:
        print("You can vote!")
    else:
        wait_time = 18 - age
        print("You can't vote!")
        print(wait_time)
```

Output

```
How old are you? 16
```

Memory



Operators

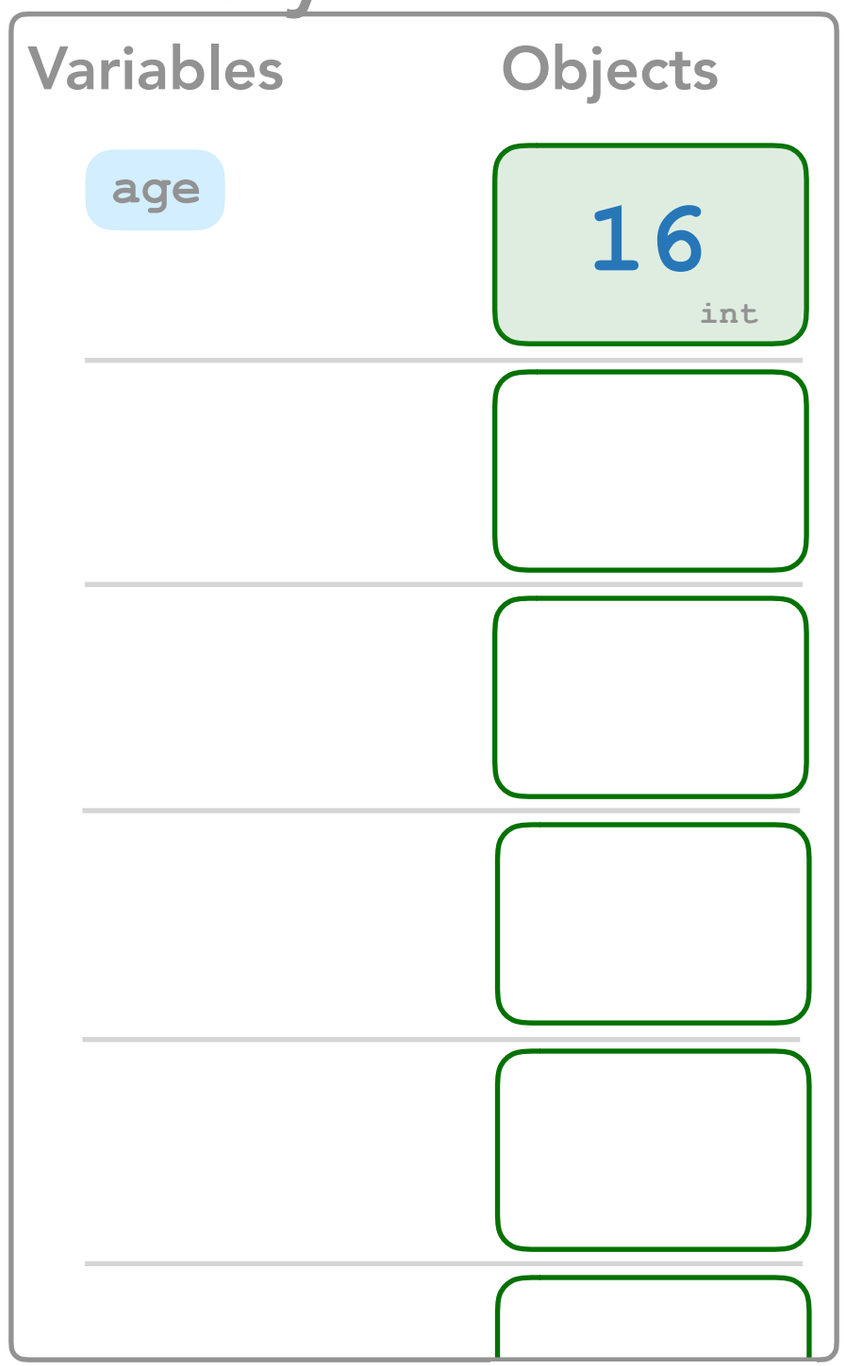
Code

```
def main():
    age = input_int("How old are you?")
    if age >= 18:
        print("You can vote!")
    else:
        wait_time = 18 - age
        print("You can't vote!")
        print(wait_time)
```

Output

```
How old are you? 16
```

Memory



Operators

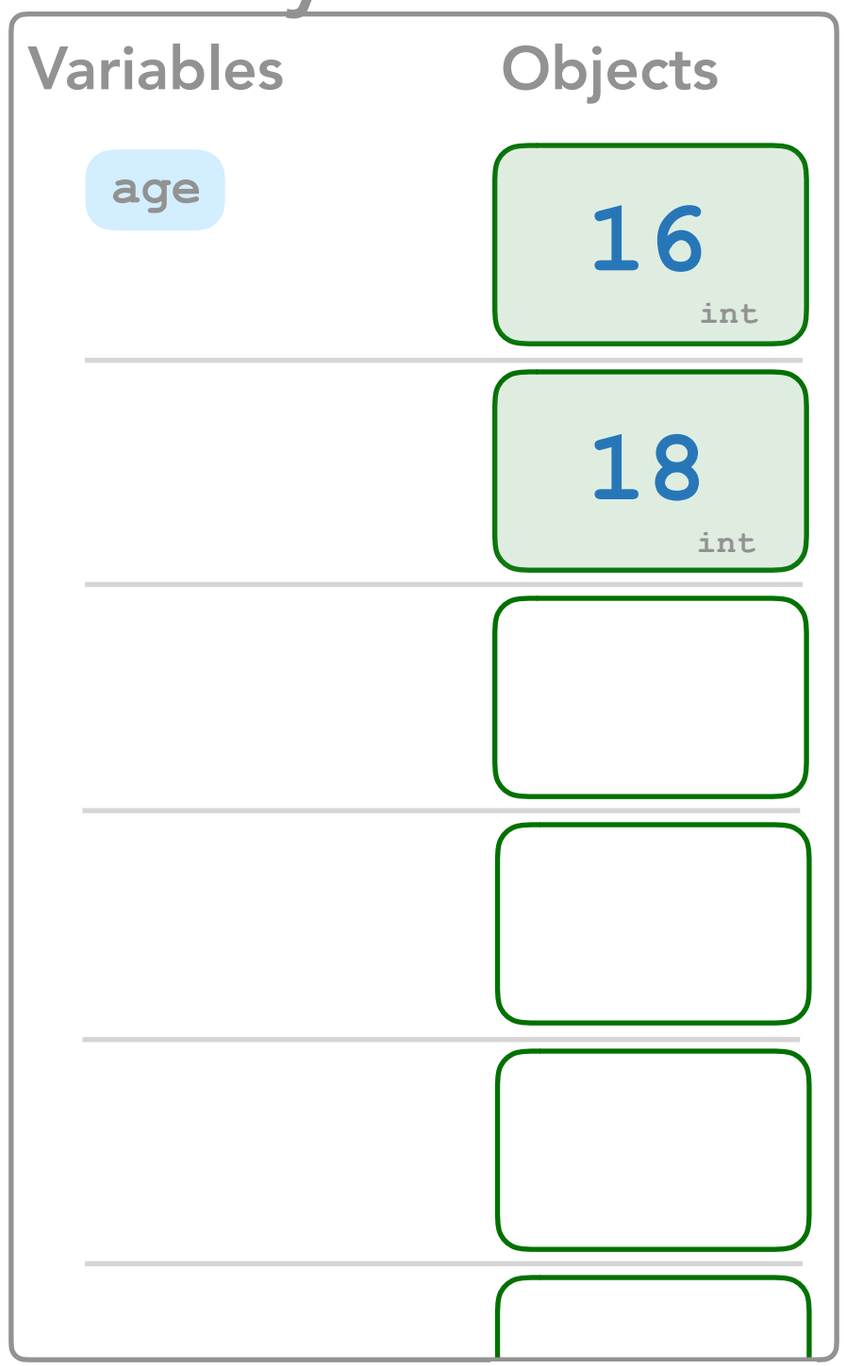
Code

```
def main():
    age = input_int("How old are you?")
    if age >= 18:
        print("You can vote!")
    else:
        wait_time = 18 - age
        print("You can't vote!")
        print(wait_time)
```

Output

```
How old are you? 16
```

Memory



Operators

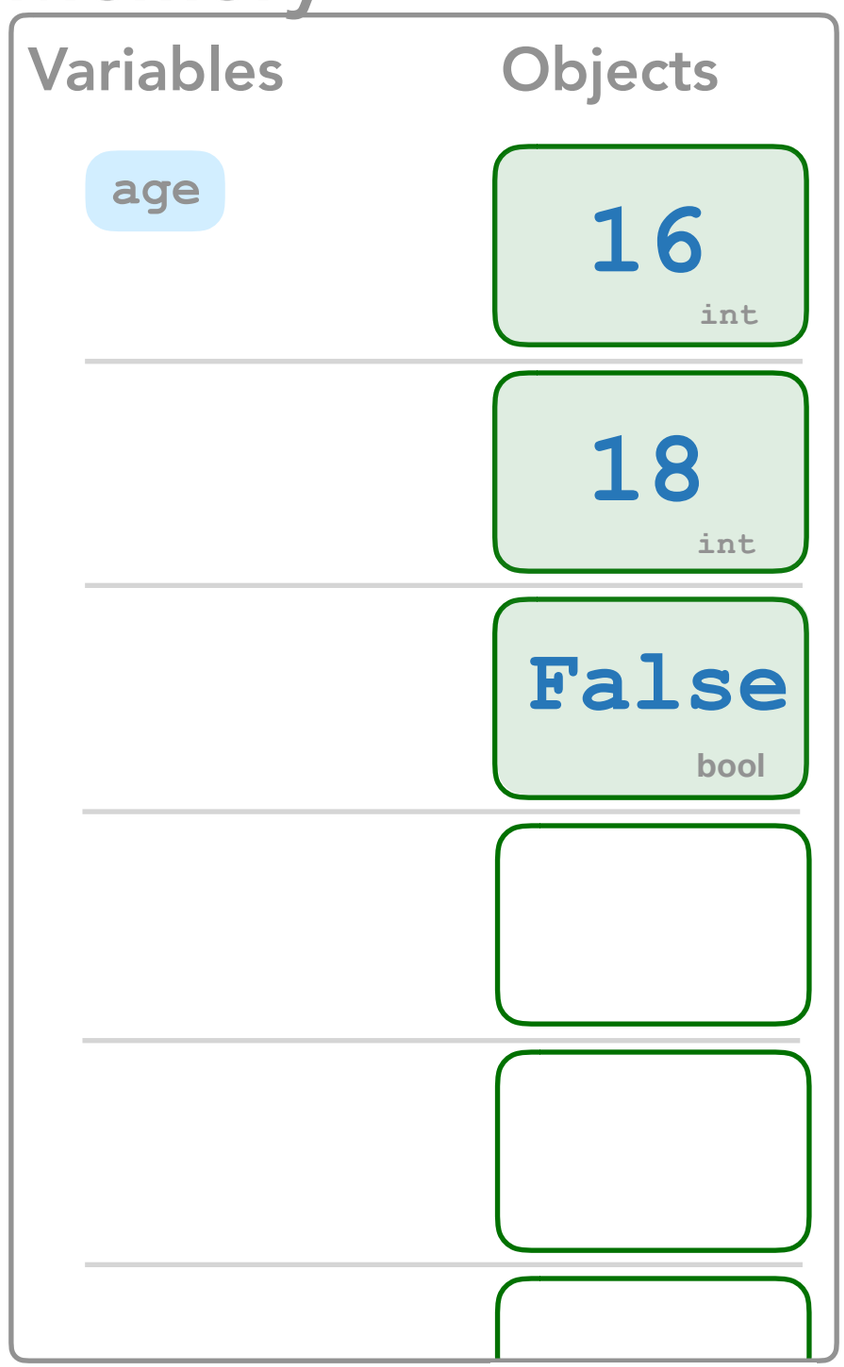
Code

```
def main():
    age = input_int("How old are you?")
    if age >= 18:
        print("You can vote!")
    else:
        wait_time = 18 - age
        print("You can't vote!")
        print(wait_time)
```

Output

```
How old are you? 16
```

Memory



Operators

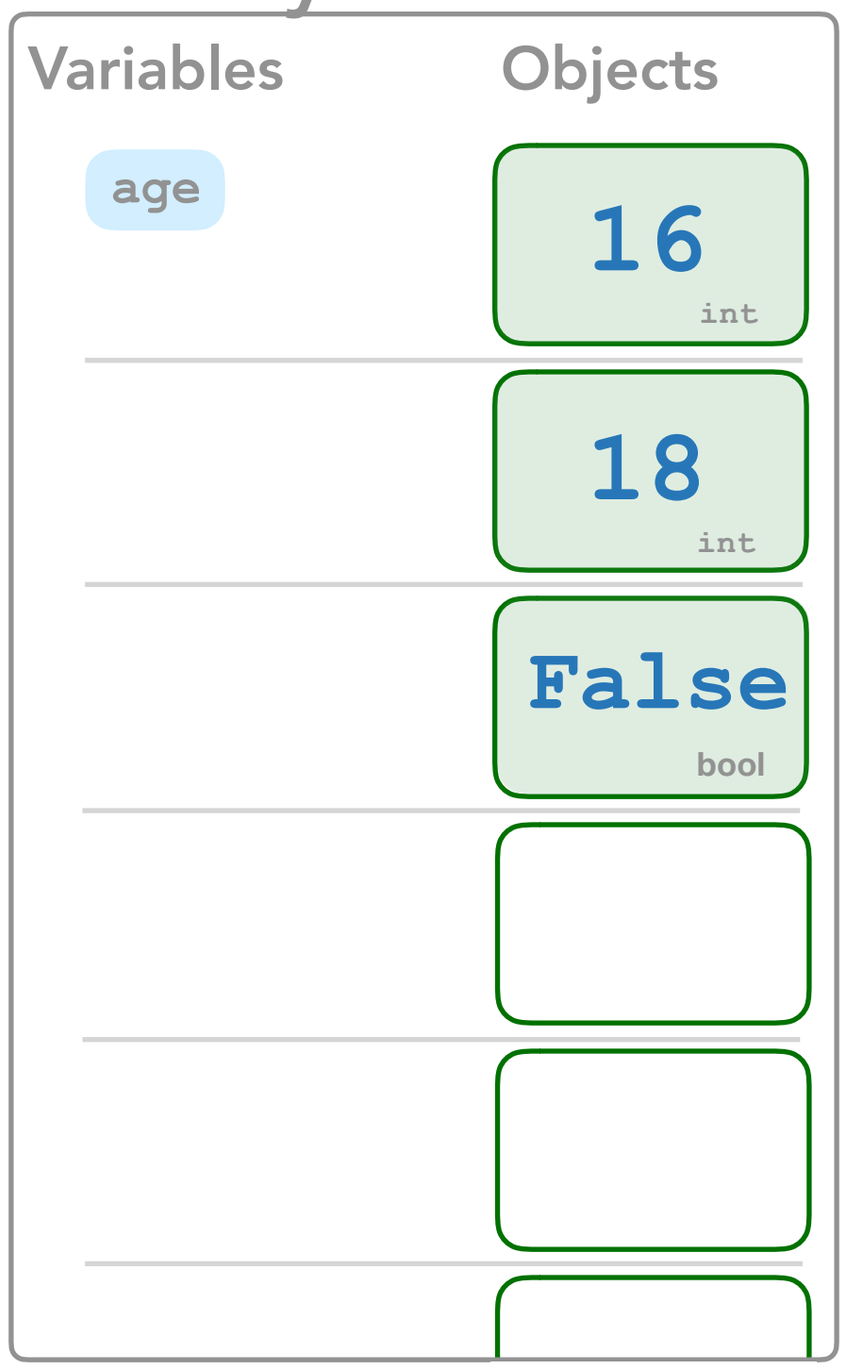
Code

```
def main():
    age = input_int("How old are you?")
    if age >= 18:
        print("You can vote!")
    else:
        wait_time = 18 - age
        print("You can't vote!")
        print(wait_time)
```

Output

```
How old are you? 16
```

Memory



Operators

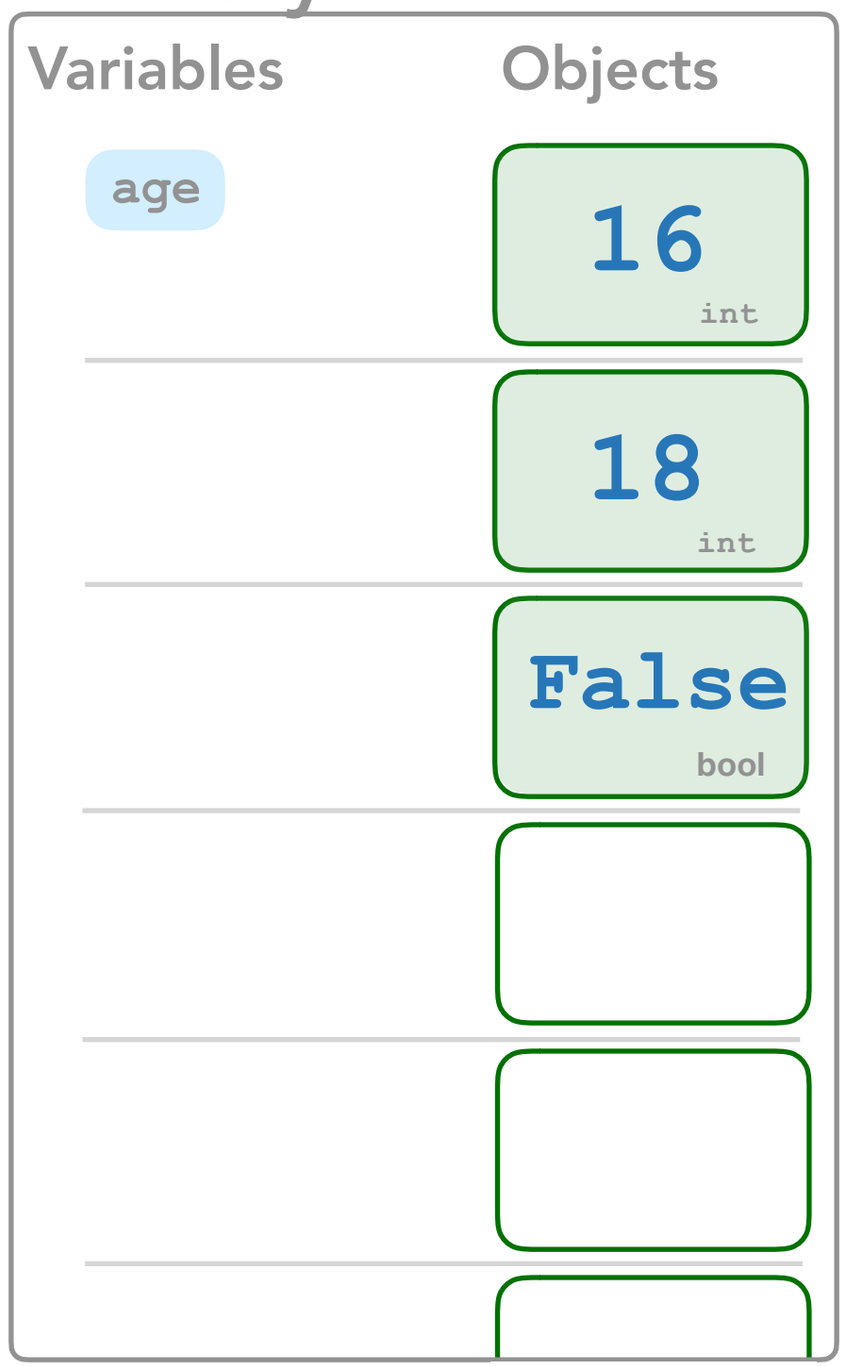
Code

```
def main():
    age = input_int("How old are you?")
    if age >= 18:
        print("You can vote!")
    else:
        wait_time = 18 - age
        print("You can't vote!")
        print(wait_time)
```

Output

```
How old are you? 16
```

Memory



Operators

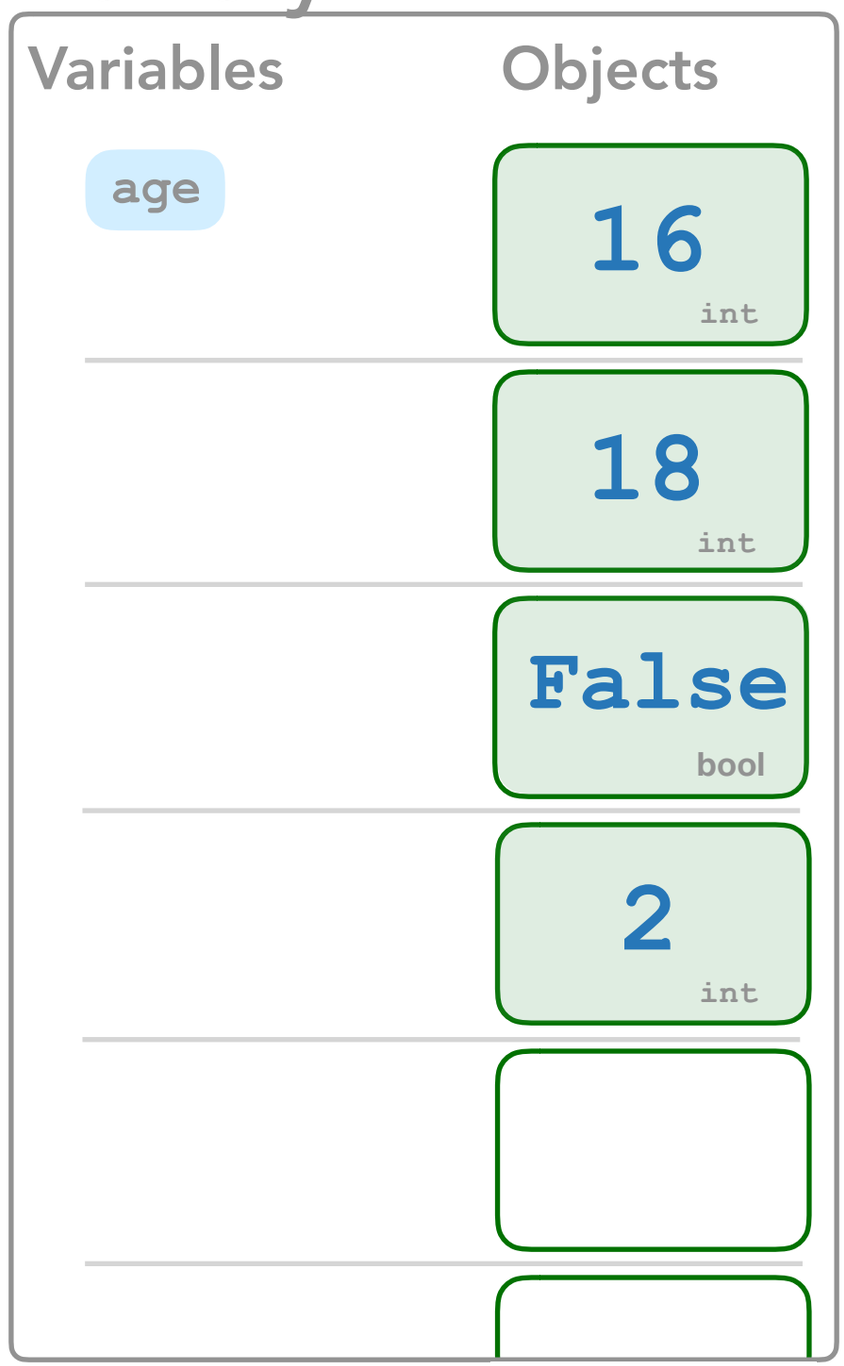
Code

```
def main():
    age = input_int("How old are you?")
    if age >= 18:
        print("You can vote!")
    else:
        wait_time = 18 - age
        print("You can't vote!")
        print(wait_time)
```

Output

```
How old are you? 16
```

Memory



Operators

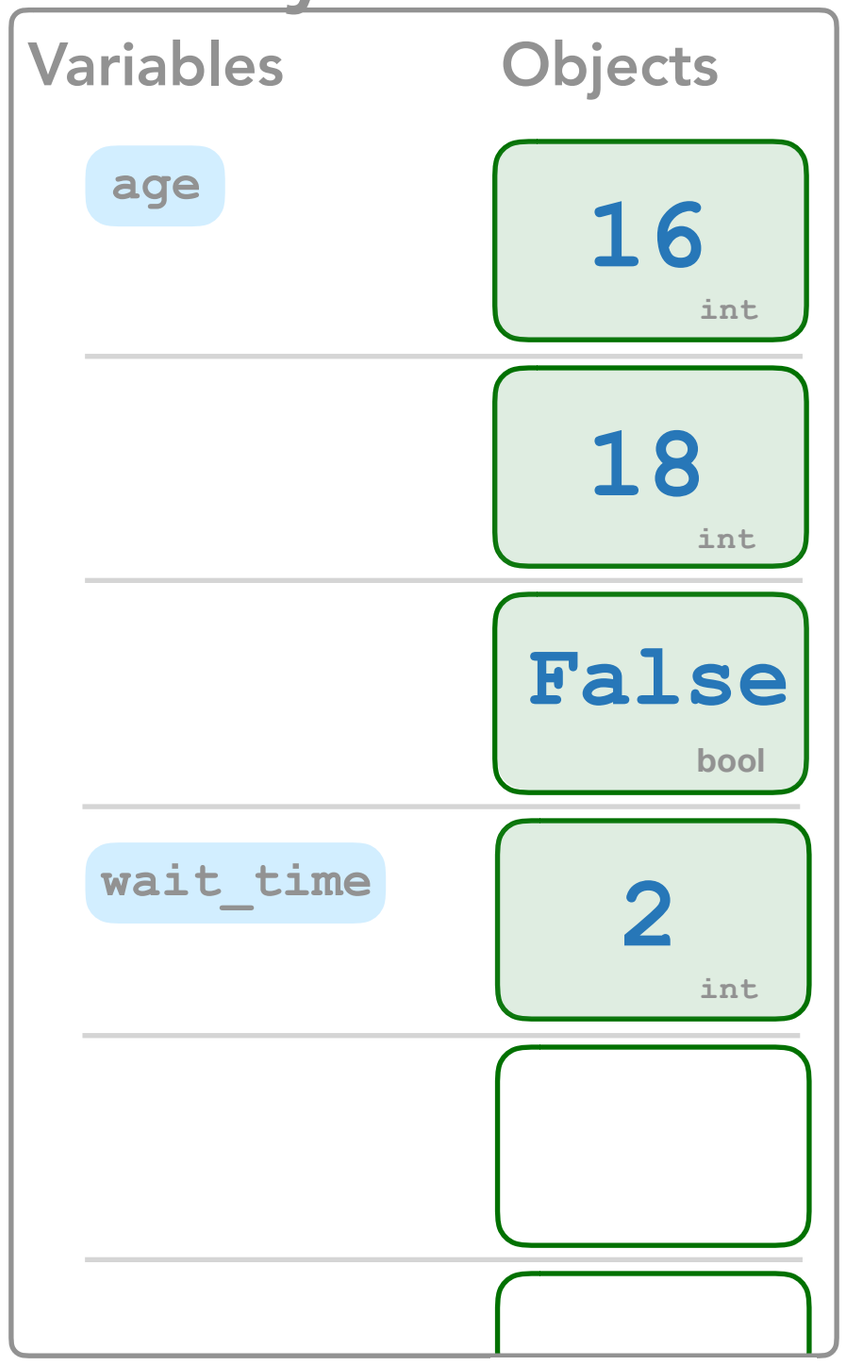
Code

```
def main():
    age = input_int("How old are you?")
    if age >= 18:
        print("You can vote!")
    else:
        wait_time = 18 - age
        print("You can't vote!")
        print(wait_time)
```

Output

```
How old are you? 16
```

Memory



Operators

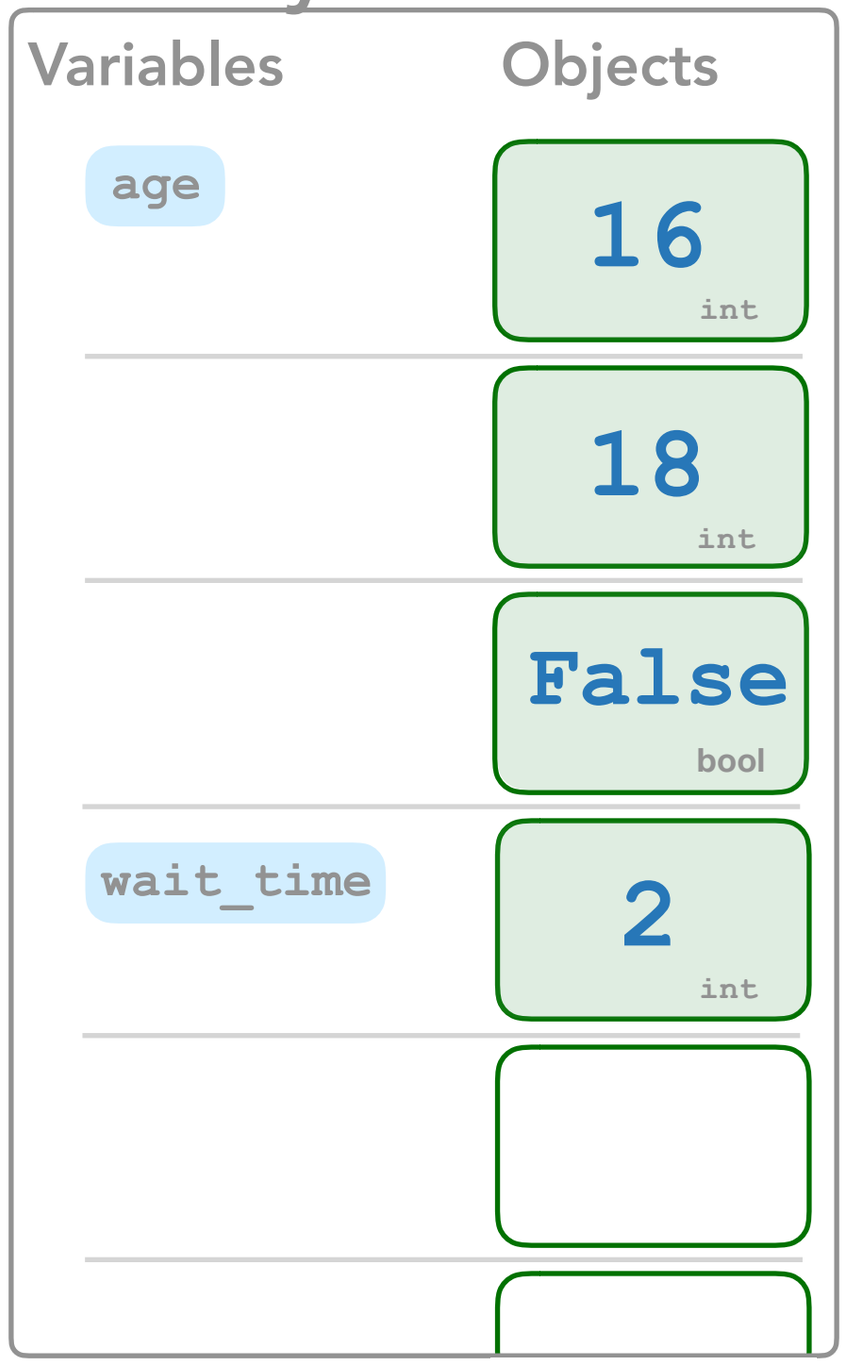
Code

```
def main():
    age = input_int("How old are you?")
    if age >= 18:
        print("You can vote!")
    else:
        wait_time = 18 - age
        print("You can't vote!")
        print(wait_time)
```

Output

```
How old are you? 16
You can't vote!
```

Memory



Operators

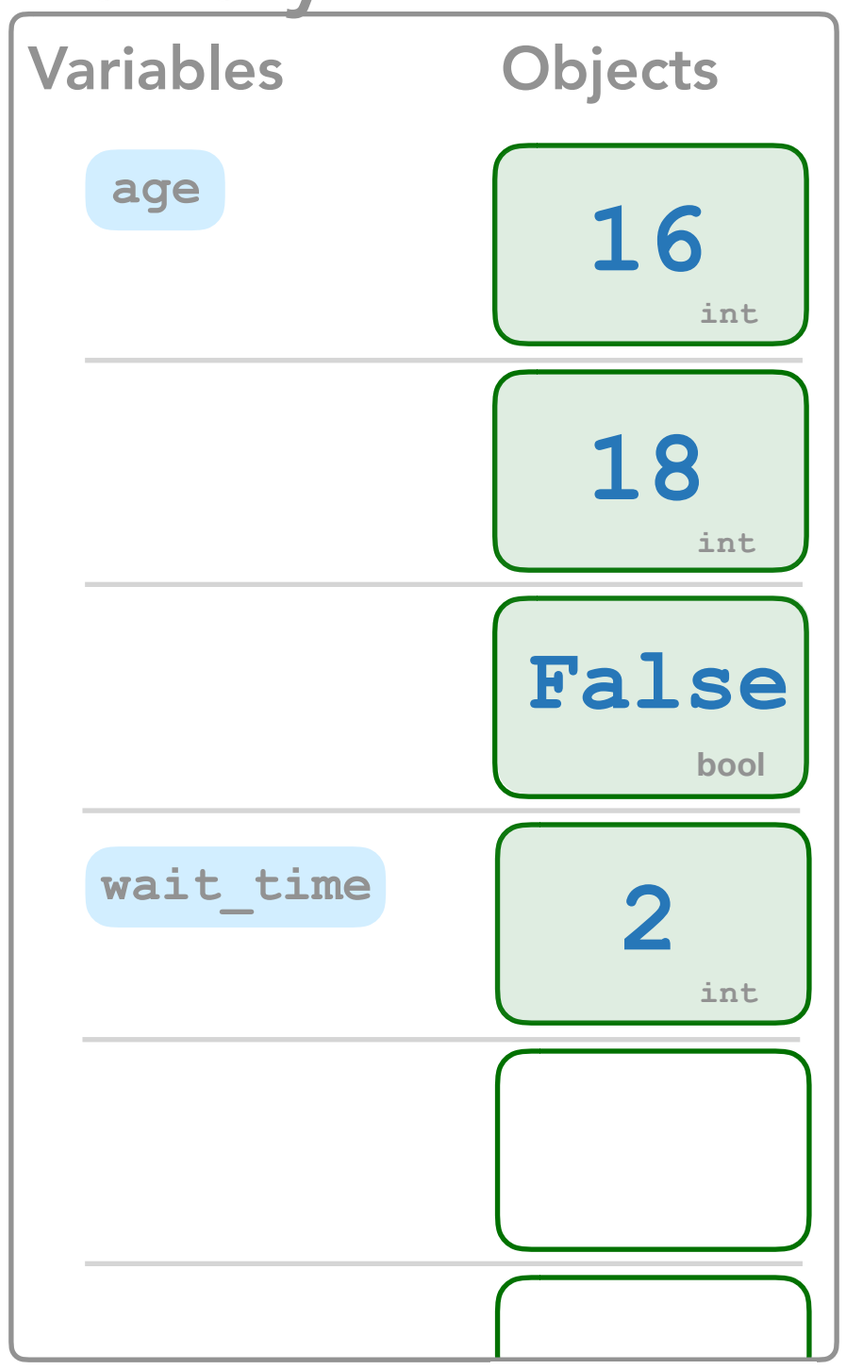
Code

```
def main():
    age = input_int("How old are you?")
    if age >= 18:
        print("You can vote!")
    else:
        wait_time = 18 - age
        print("You can't vote!")
        print(wait_time)
```

Output

```
How old are you? 16
You can't vote!
2
```

Memory



Operators

```
not False == True
```

```
True and False == False
```

```
True or False == True
```

Calculator

Conversion

Bilheteria

Example: Calculator 2.0

Conversion Functions

`str(object)`

Example

`str(5)` returns a `string` `"5"`

`int(object)`

Example

`int("5")` returns an `int` `5`

`float(object)`

Example

`float("5.3")` returns an `float` `5.3`

`bool(object)`

Example

`float("True")` returns an `bool` `True`

This week on CS106R...

This week on CS106R...

Writing **Functions** with **Objects**

So far you've written functions like this:

```
def turn_left():
    turn_right()
    turn_right()
    turn_right()
```

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

How do we write functions that:

receive objects

and

return objects

How do we write functions that:

receive objects

and

return objects

Example: Pythagoras 2.0

IMPORTANT IDEA

Every **function** has its own variables

Variables cannot exist across **functions**

Scope

Code

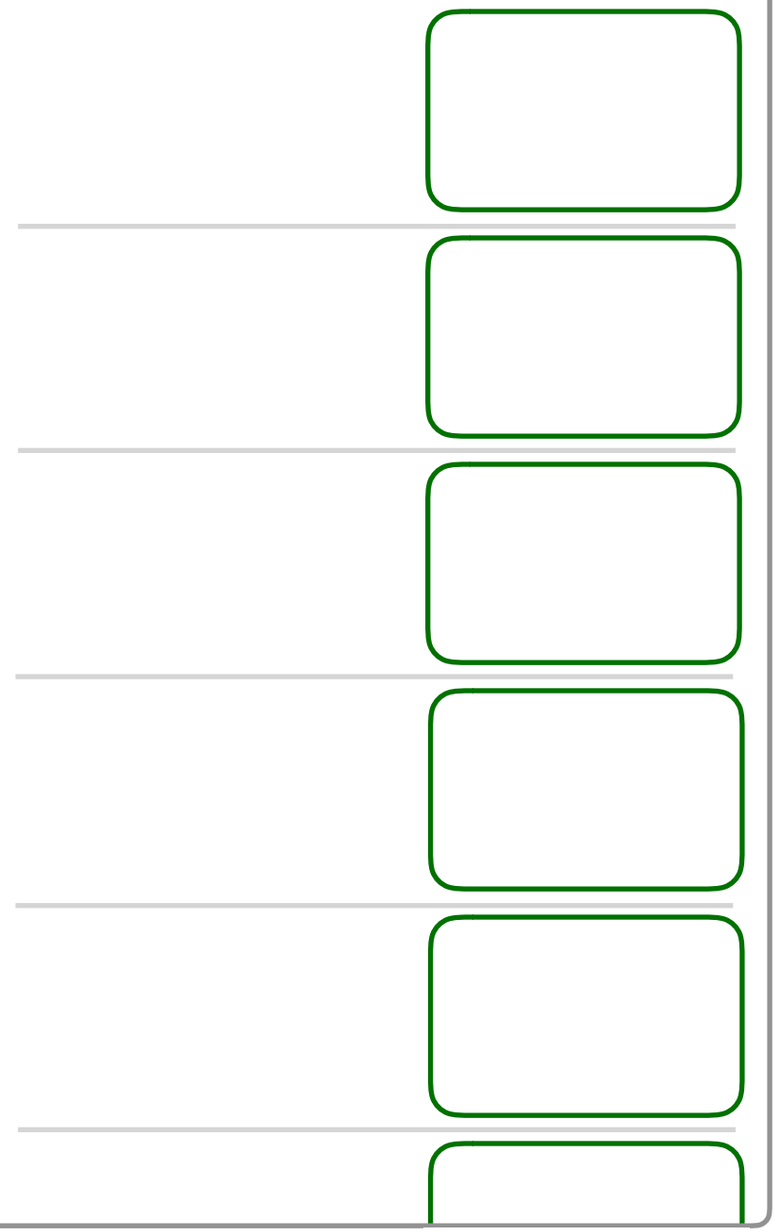
```
def compute_pythag(a, b):
  c_squared = a*a + b*b
  c = square_root(c_squared)
  return c

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

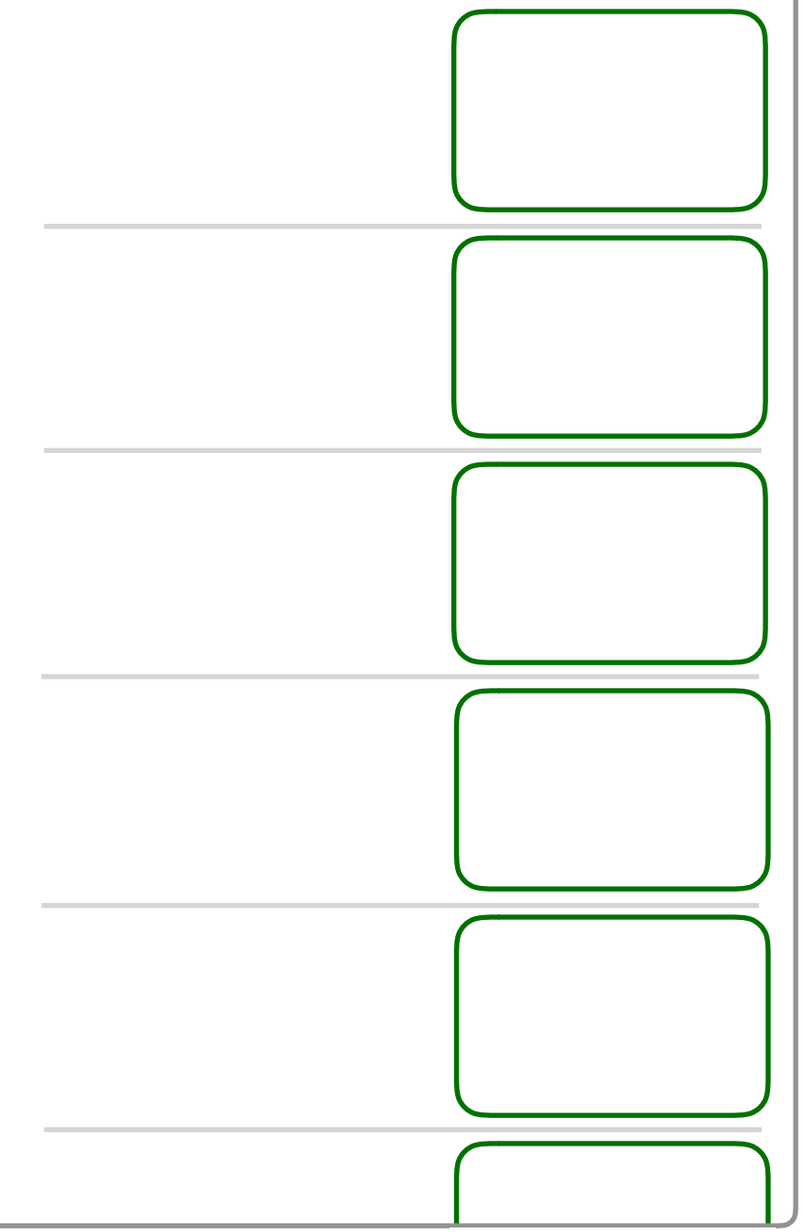
```
def compute_pythag(a, b):
  c_squared = a*a + b*b
  c = square_root(c_squared)
  return c

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

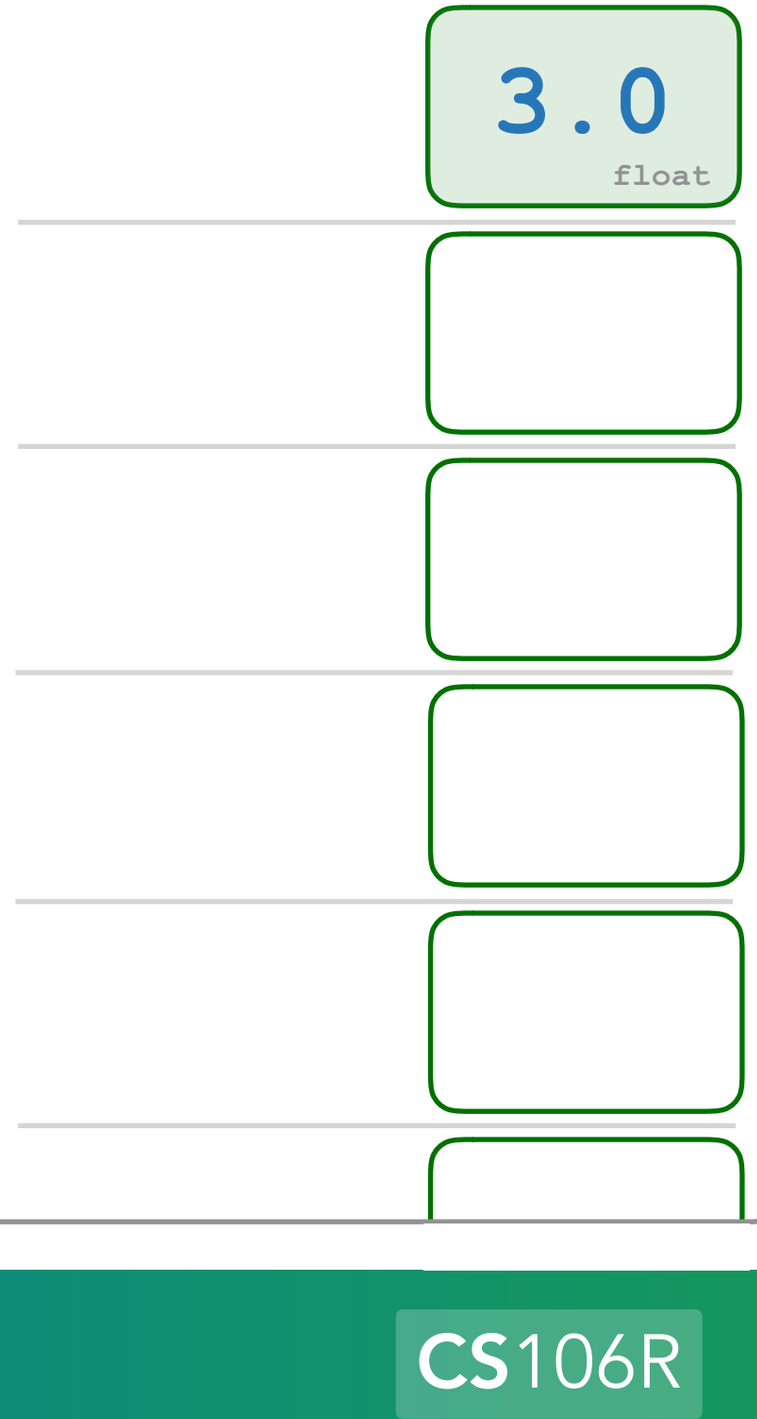
```
def compute_pythag(a, b):
  c_squared = a*a + b*b
  c = square_root(c_squared)
  return c

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



Operators

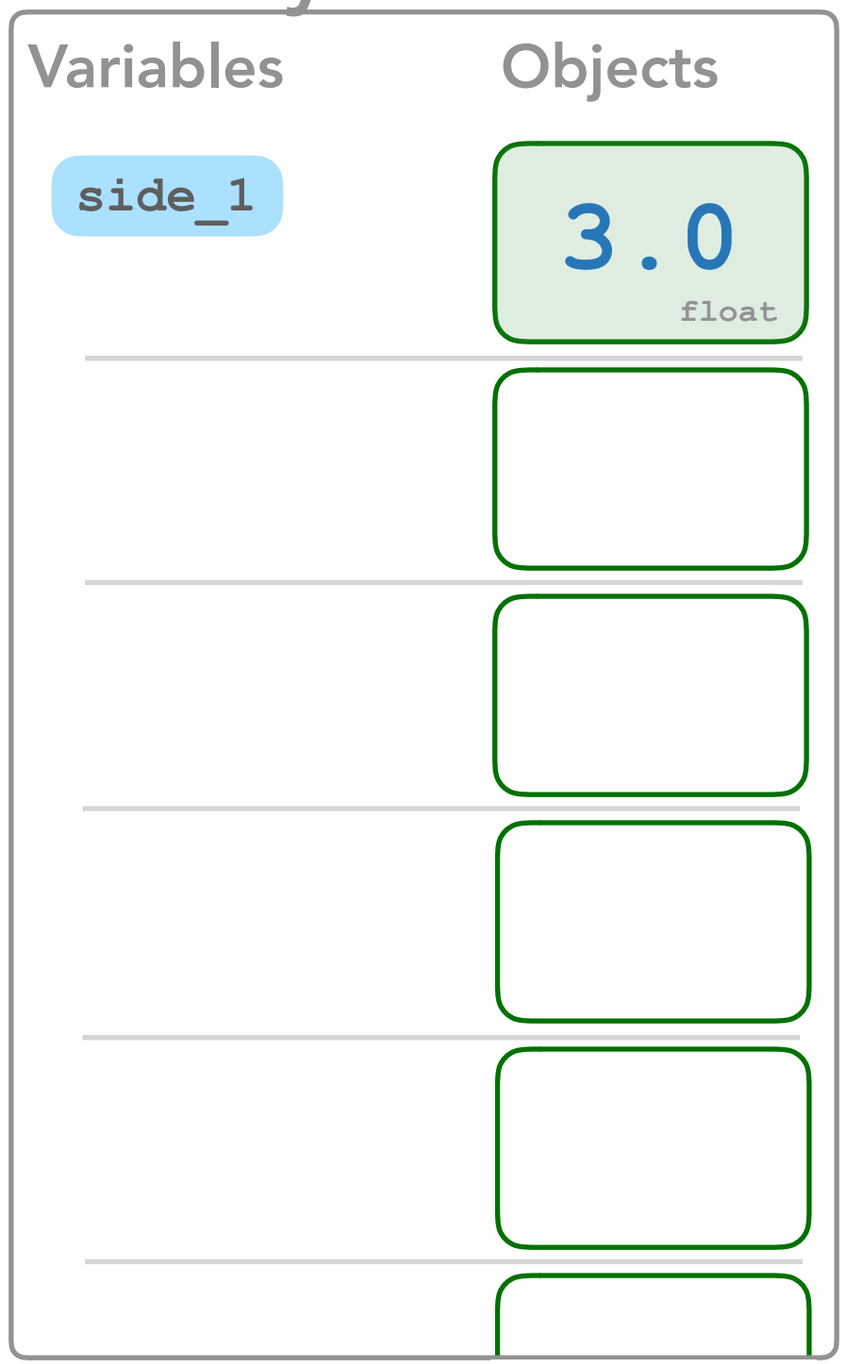
Code

```
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



Operators

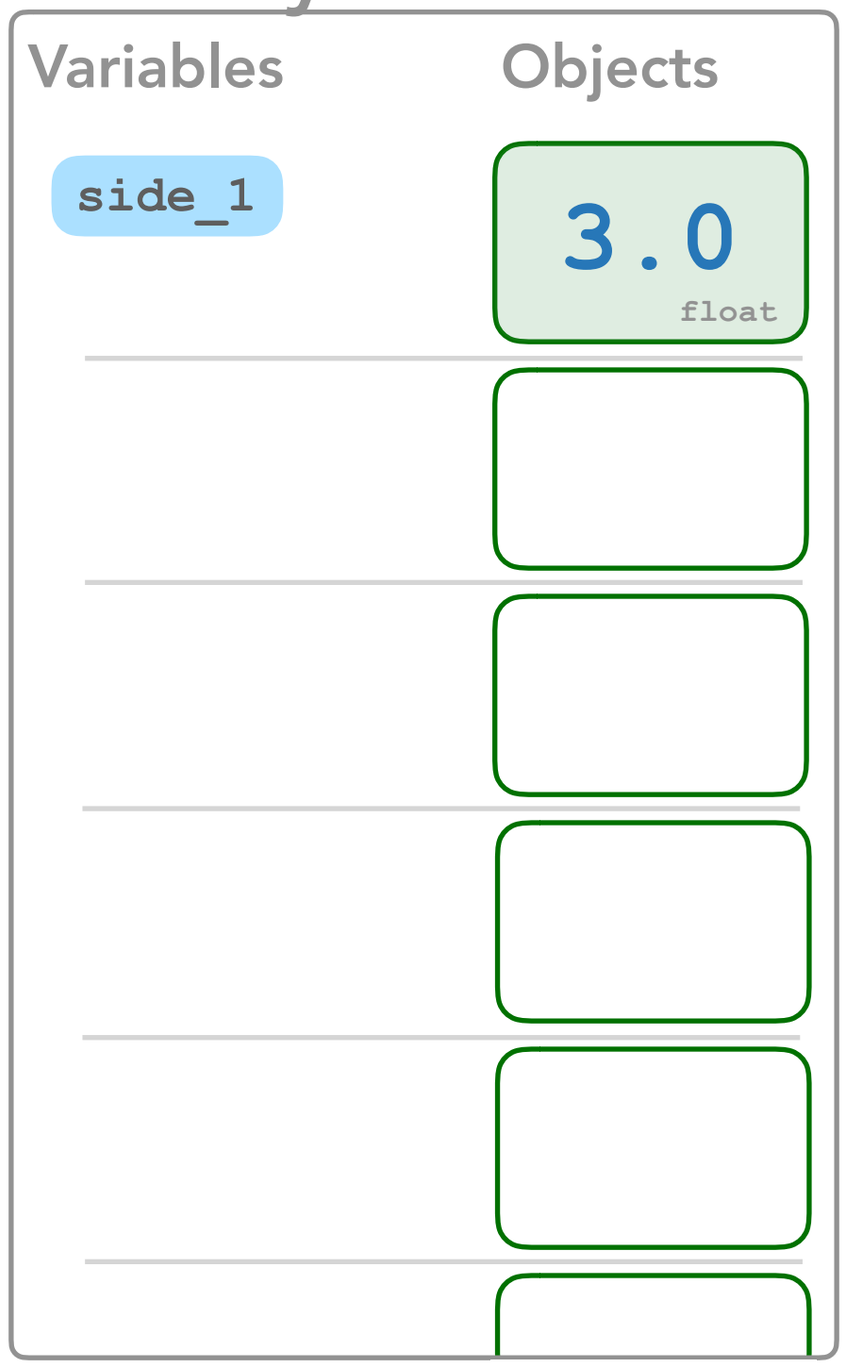
Code

```
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



Operators

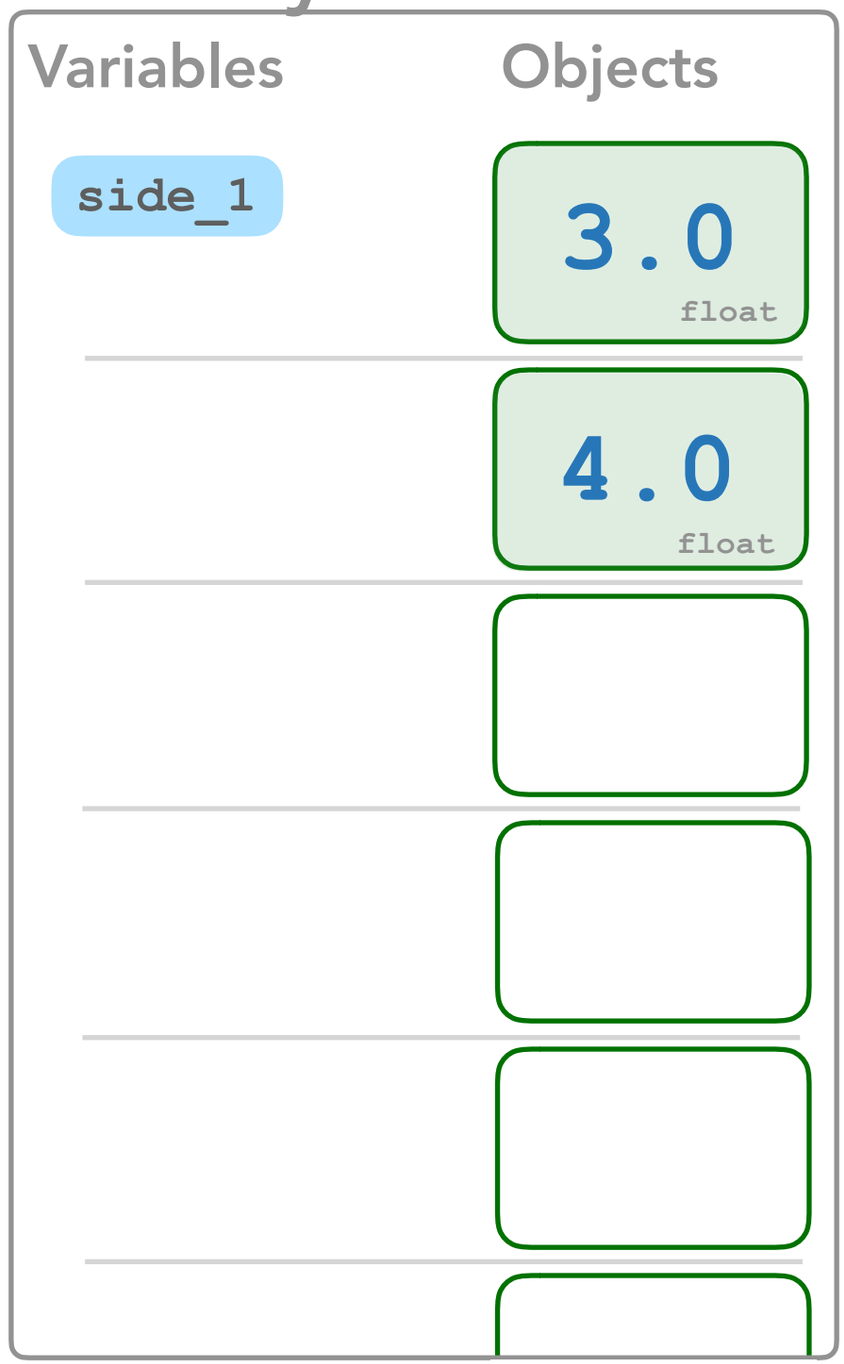
Code

```
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



Operators

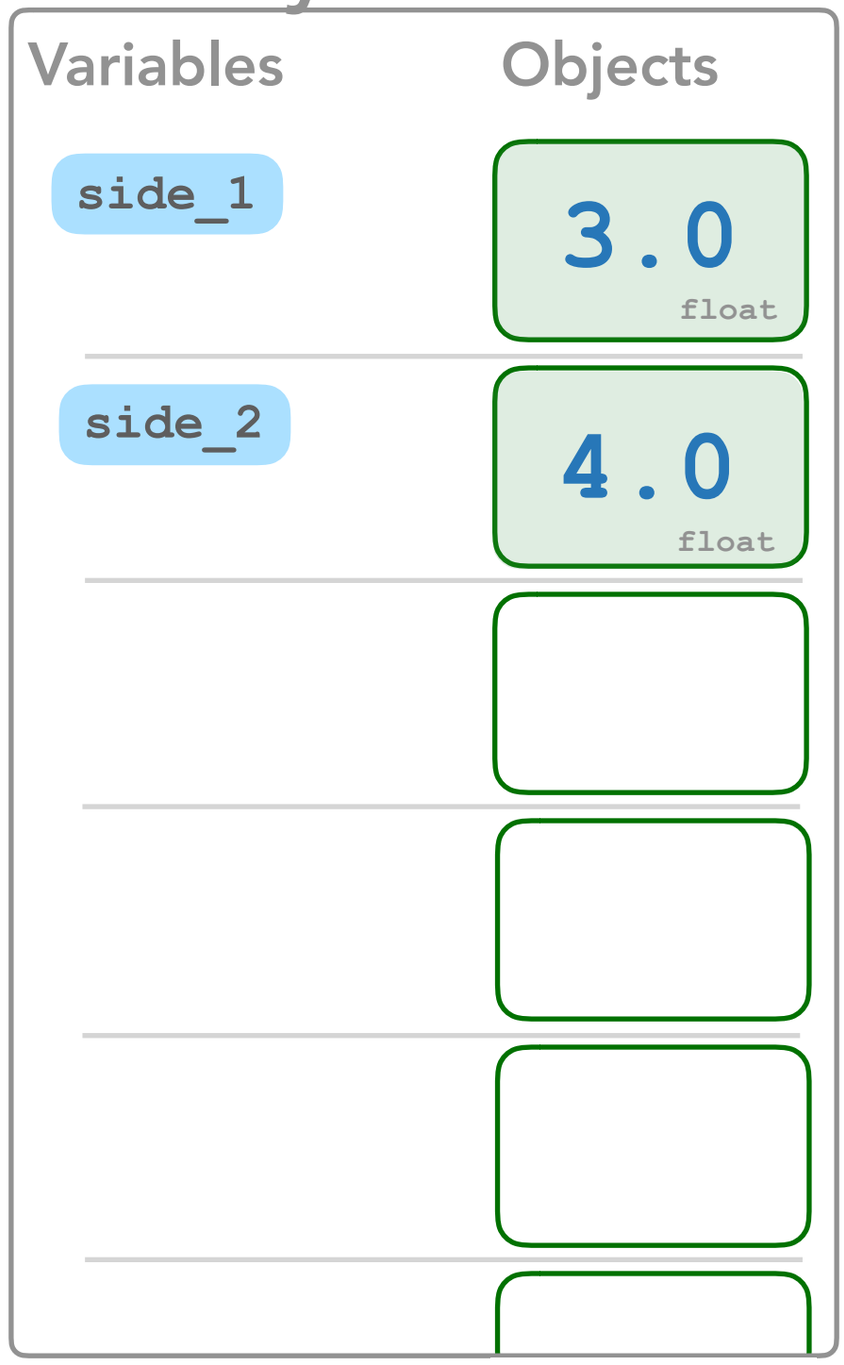
Code

```
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



Operators

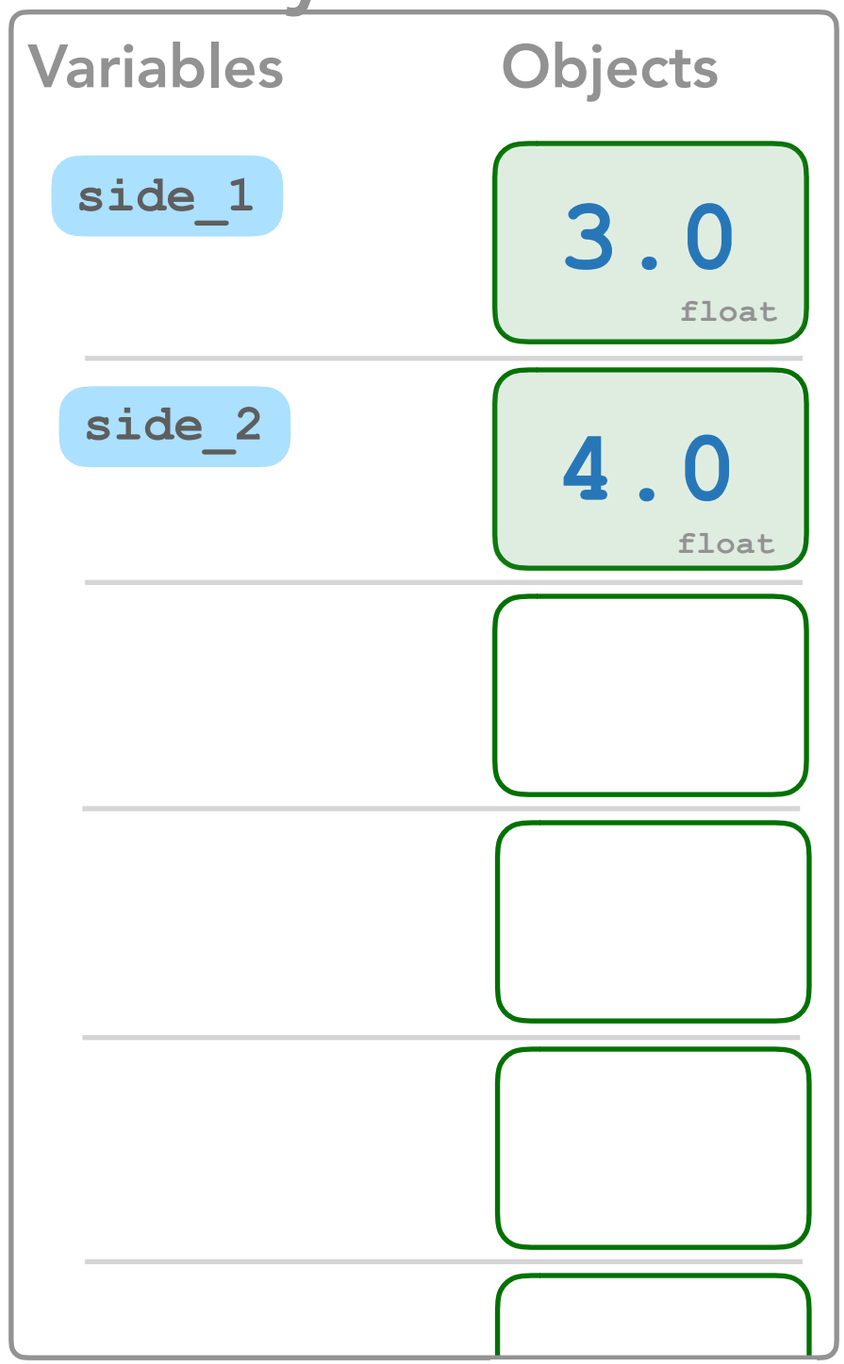
Code

```
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



Operators

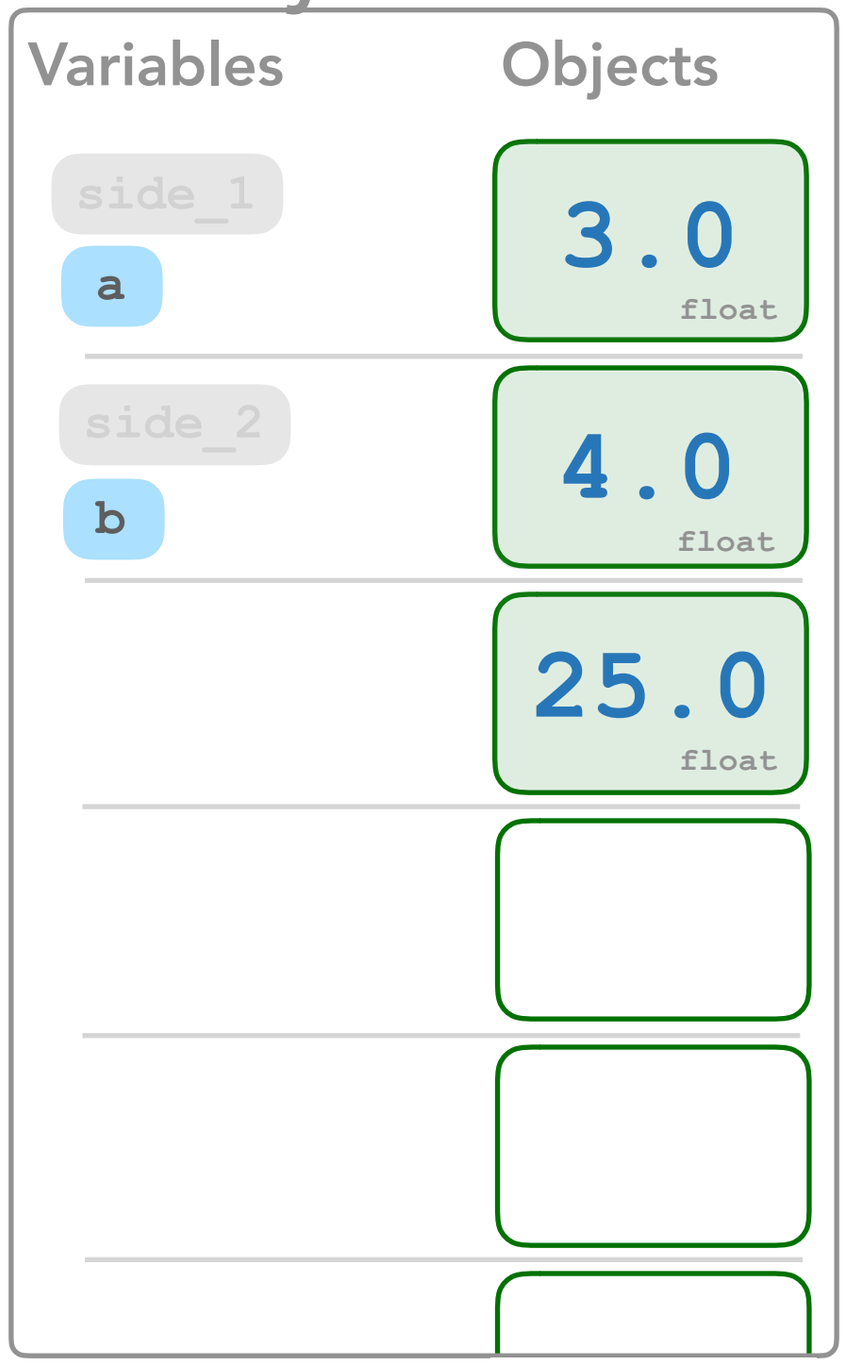
Code

```
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



Operators

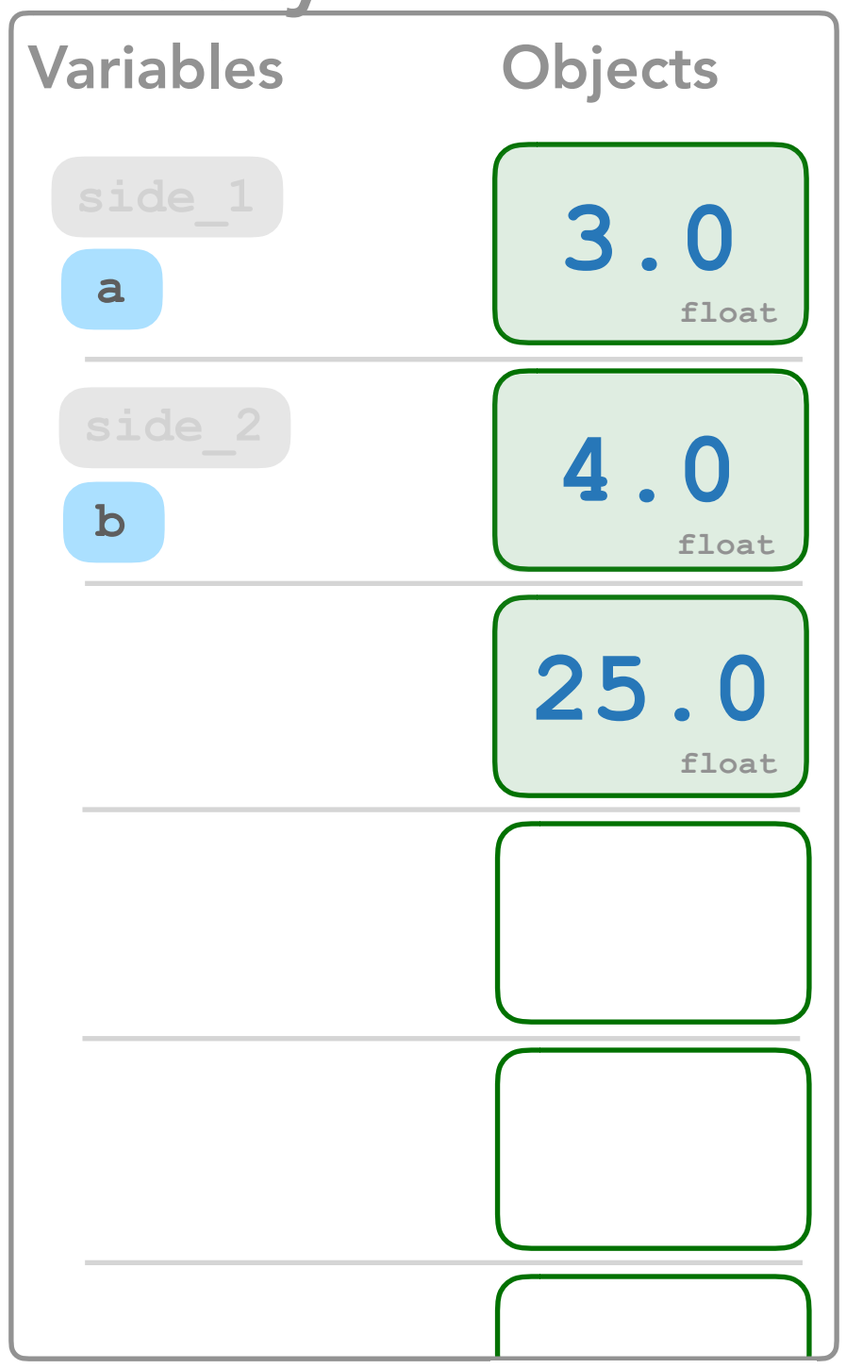
Code

```
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



Operators

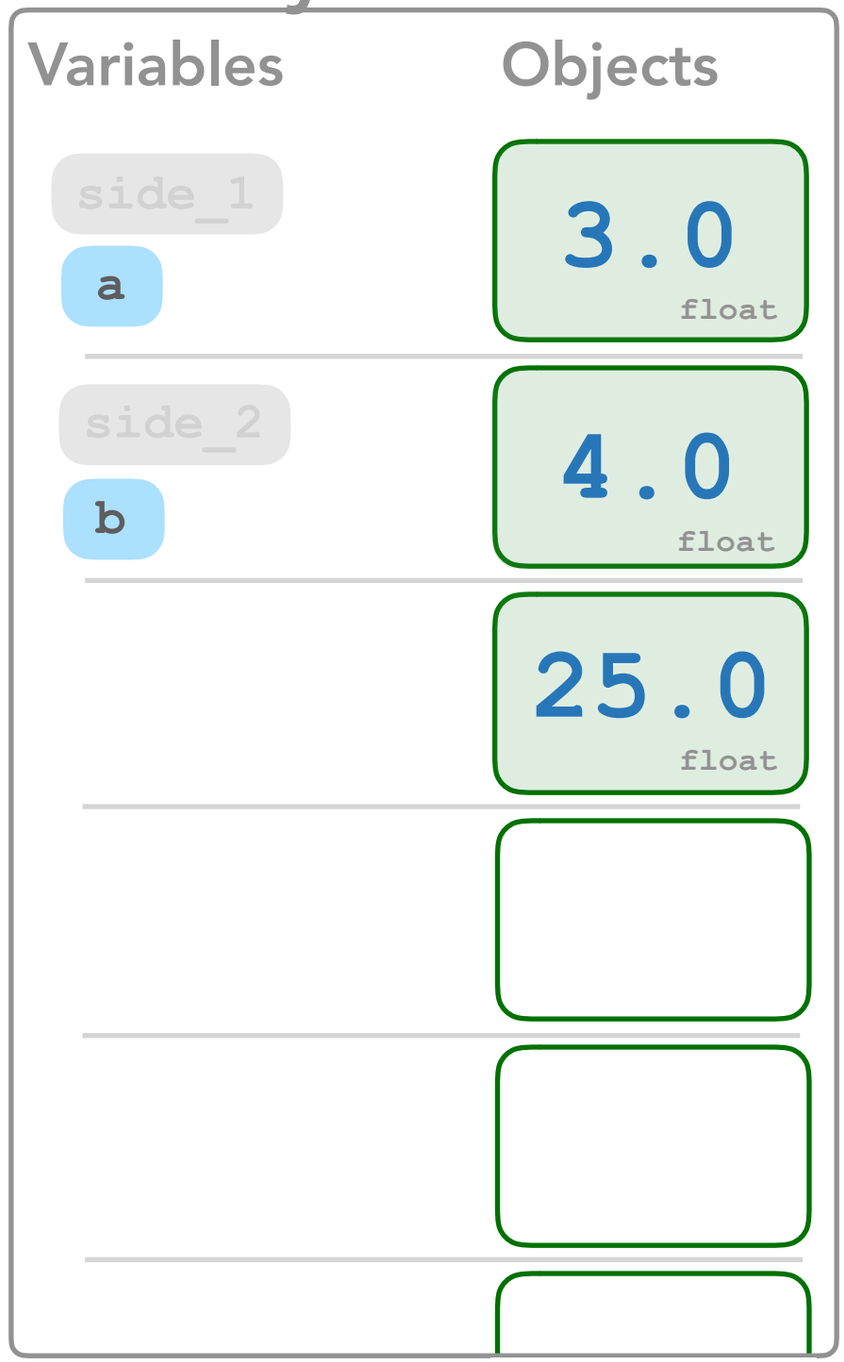
Code

```
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



Operators

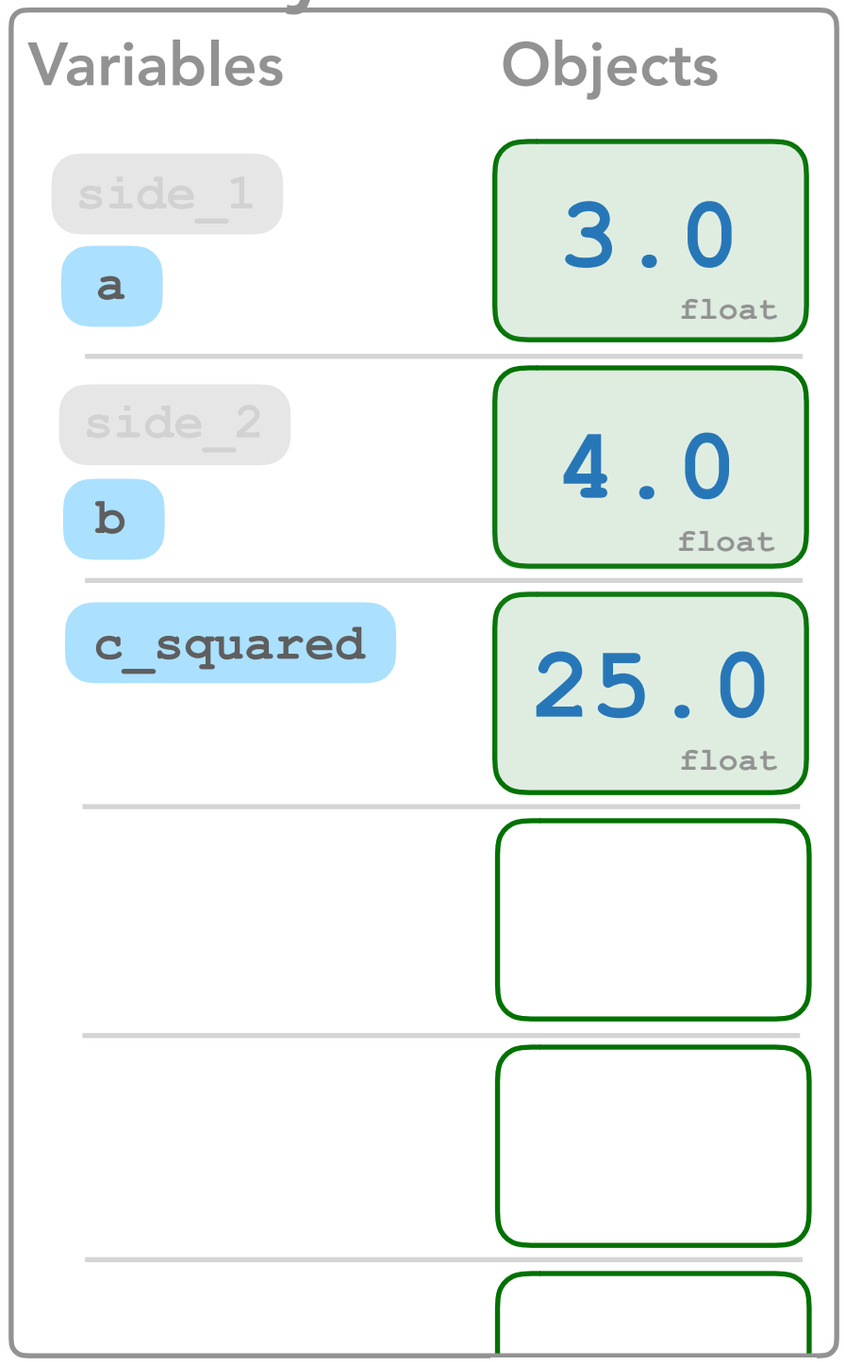
Code

```
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



Operators

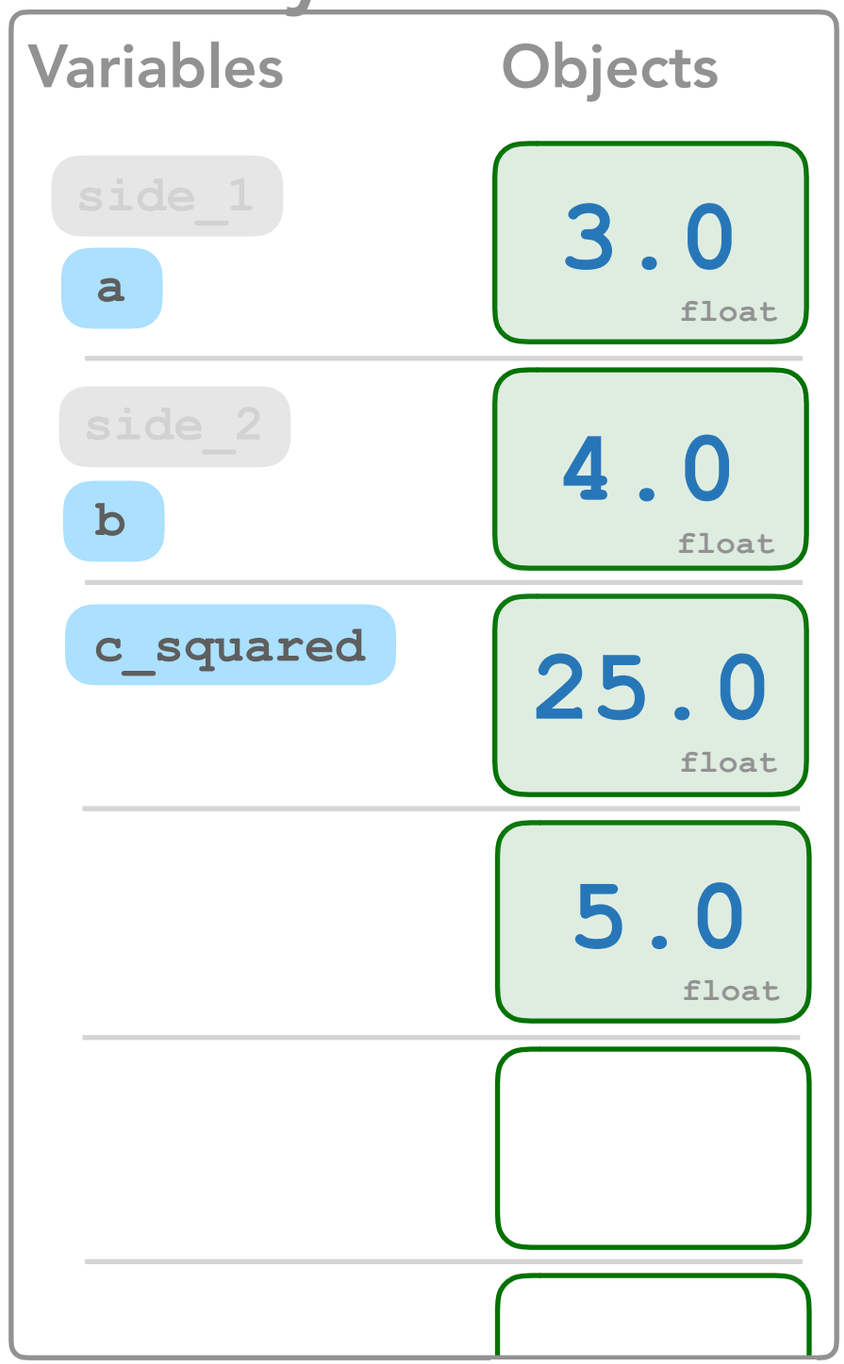
Code

```
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



Operators

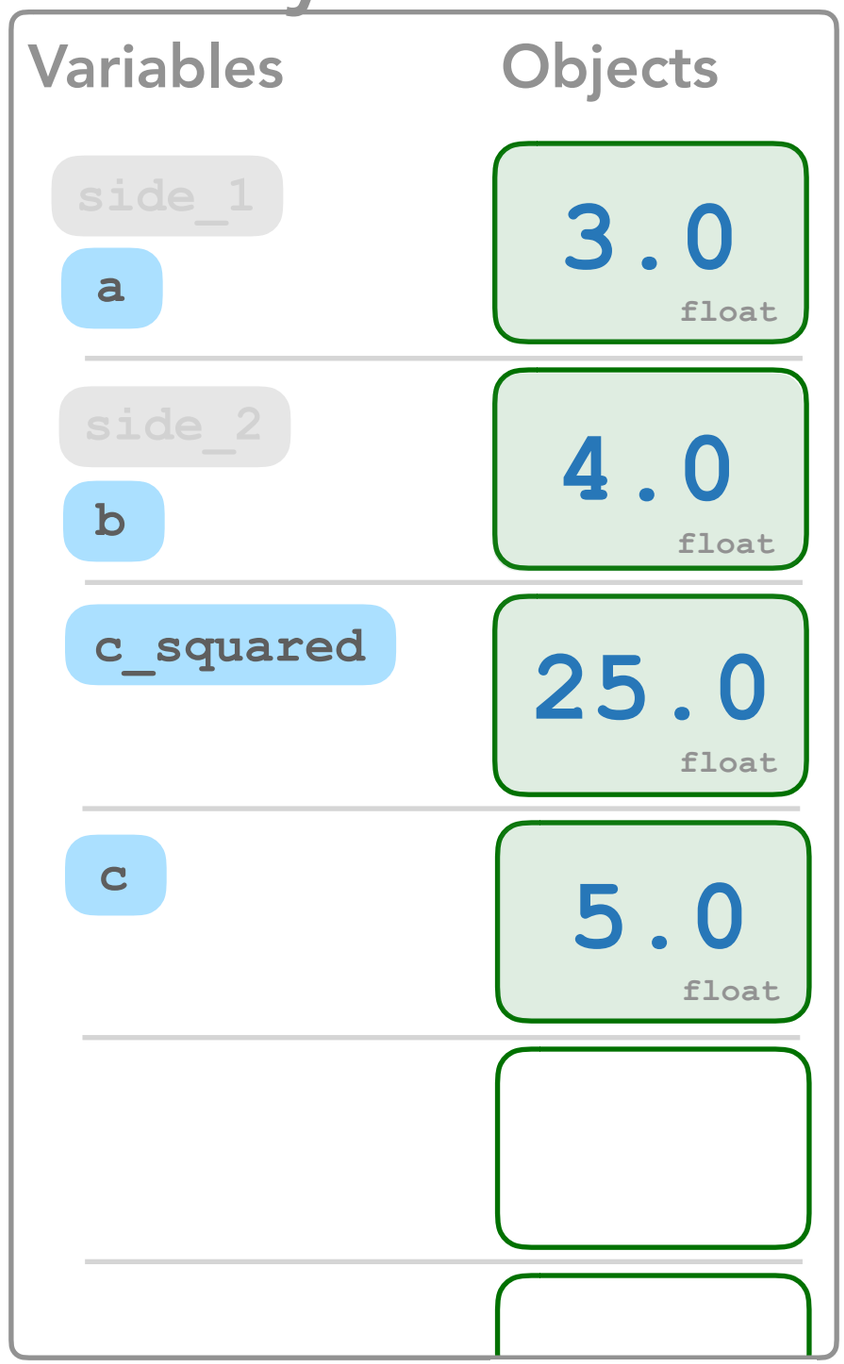
Code

```
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



Operators

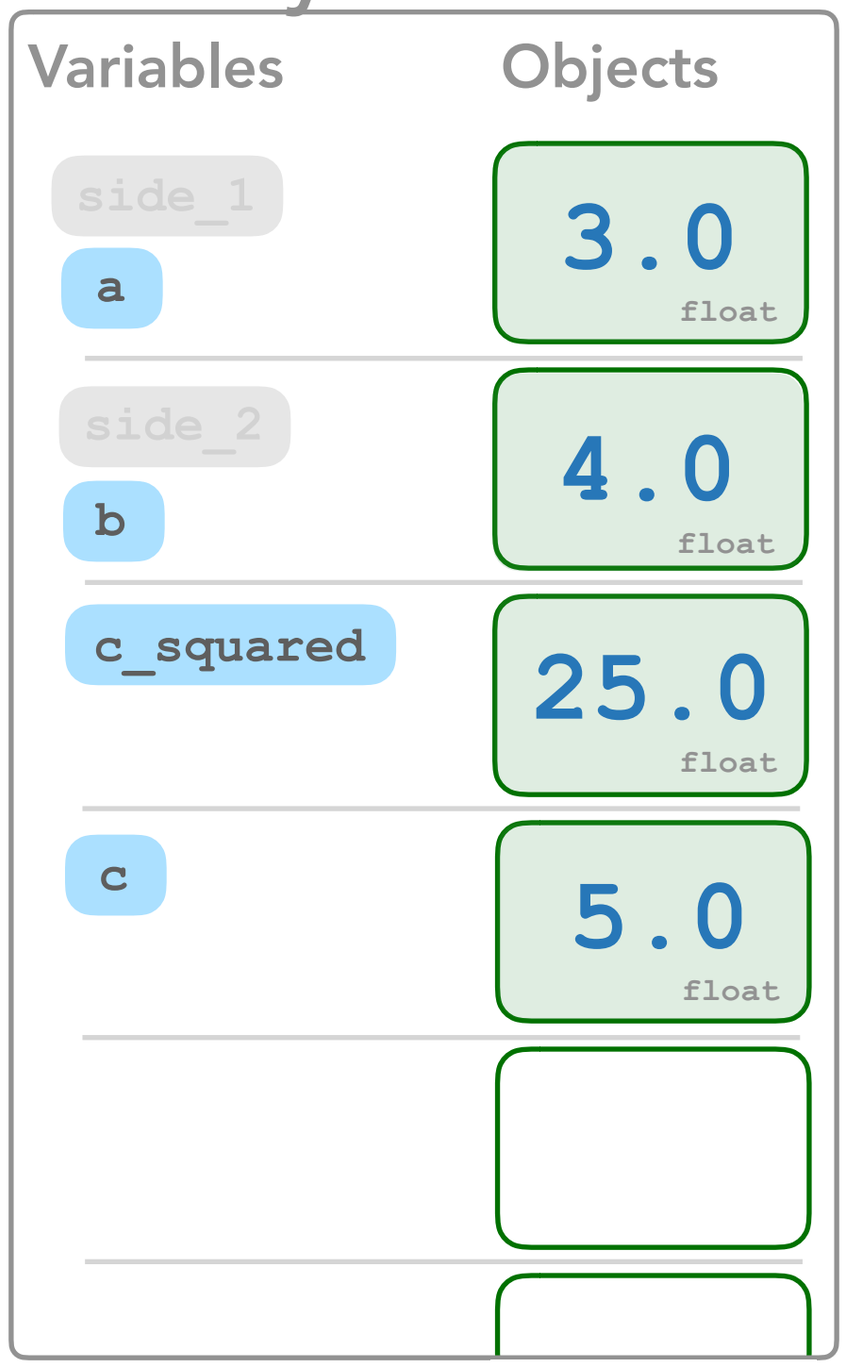
Code

```
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



Operators

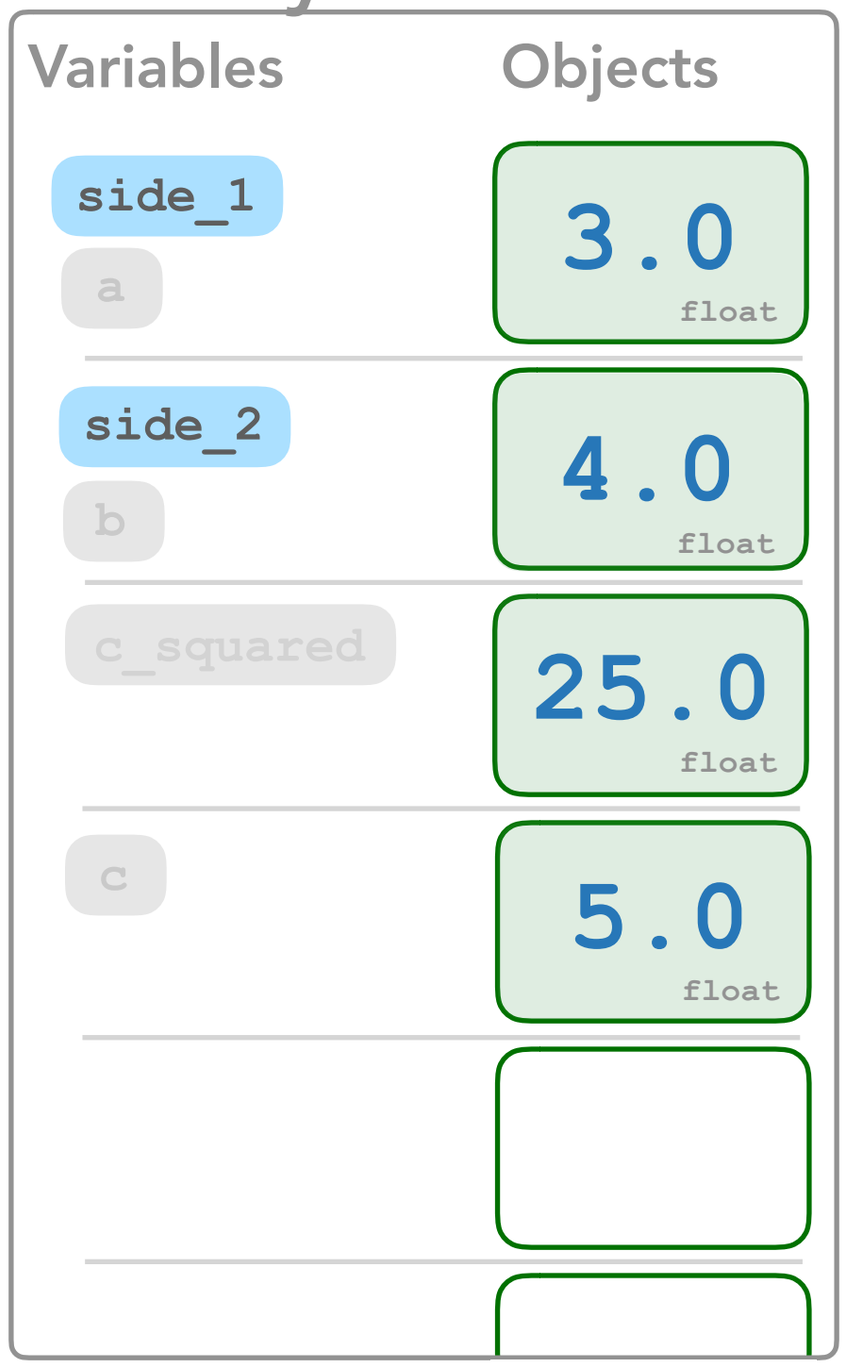
Code

```
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



Operators

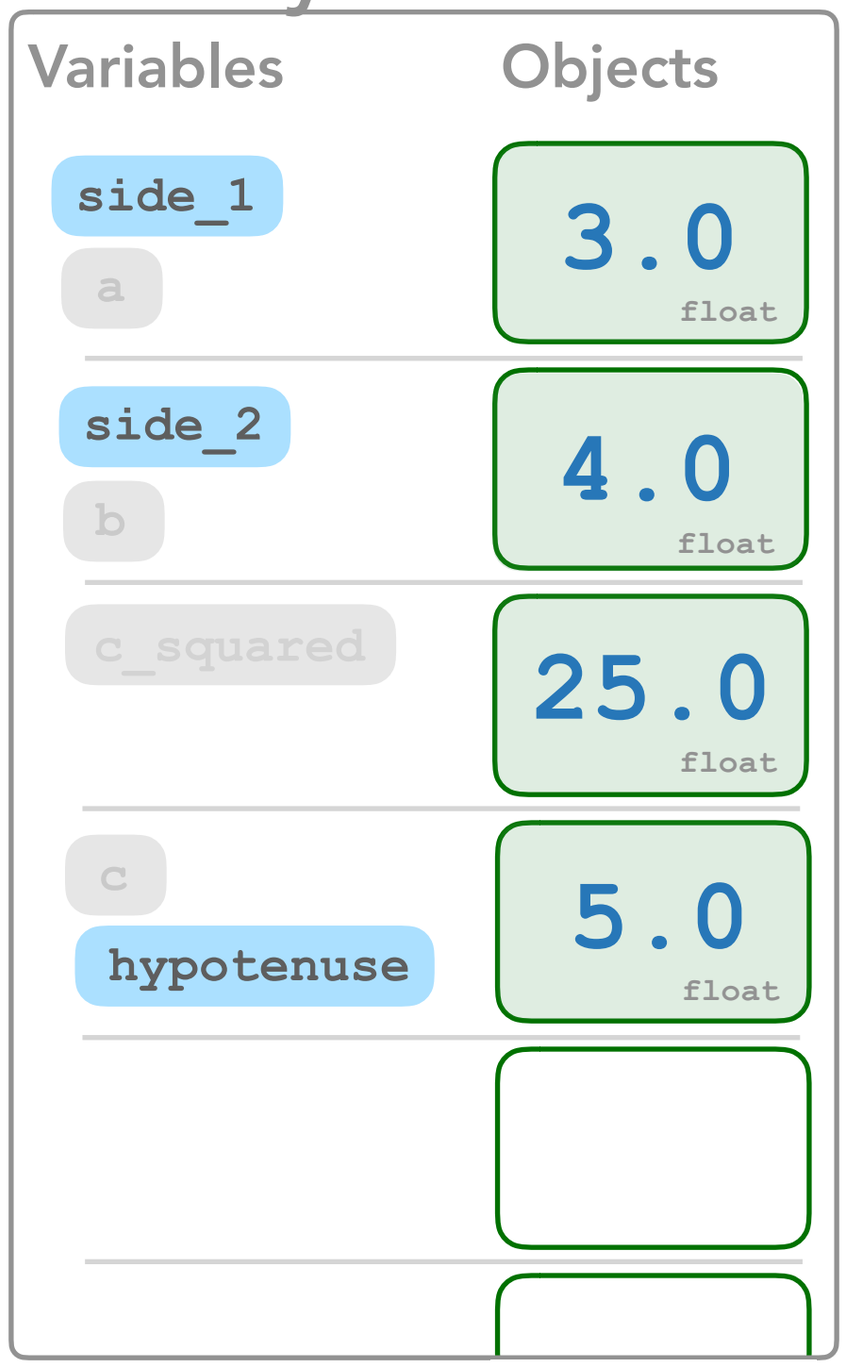
Code

```
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



Operators

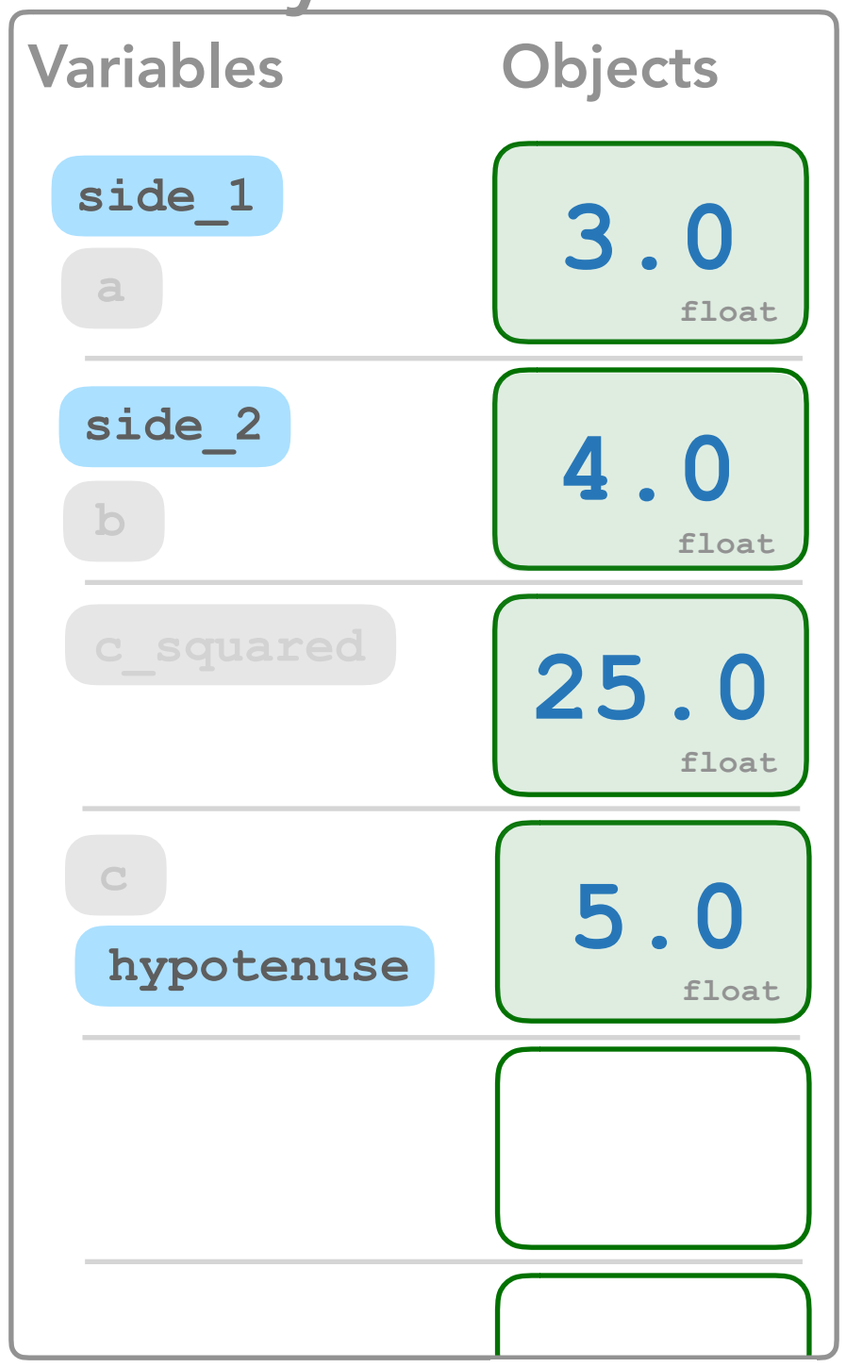
Code

```
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



Operators

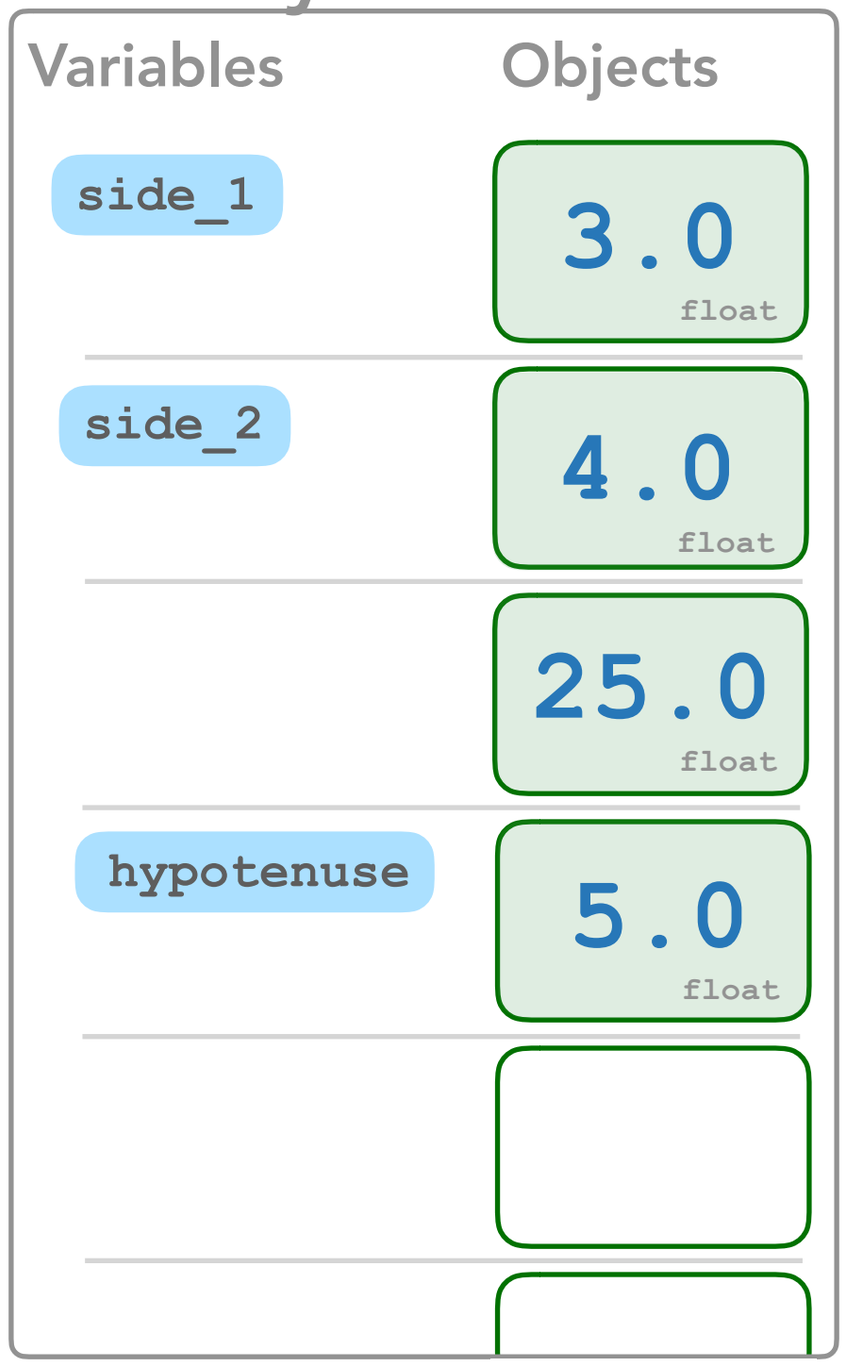
Code

```
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



Operators

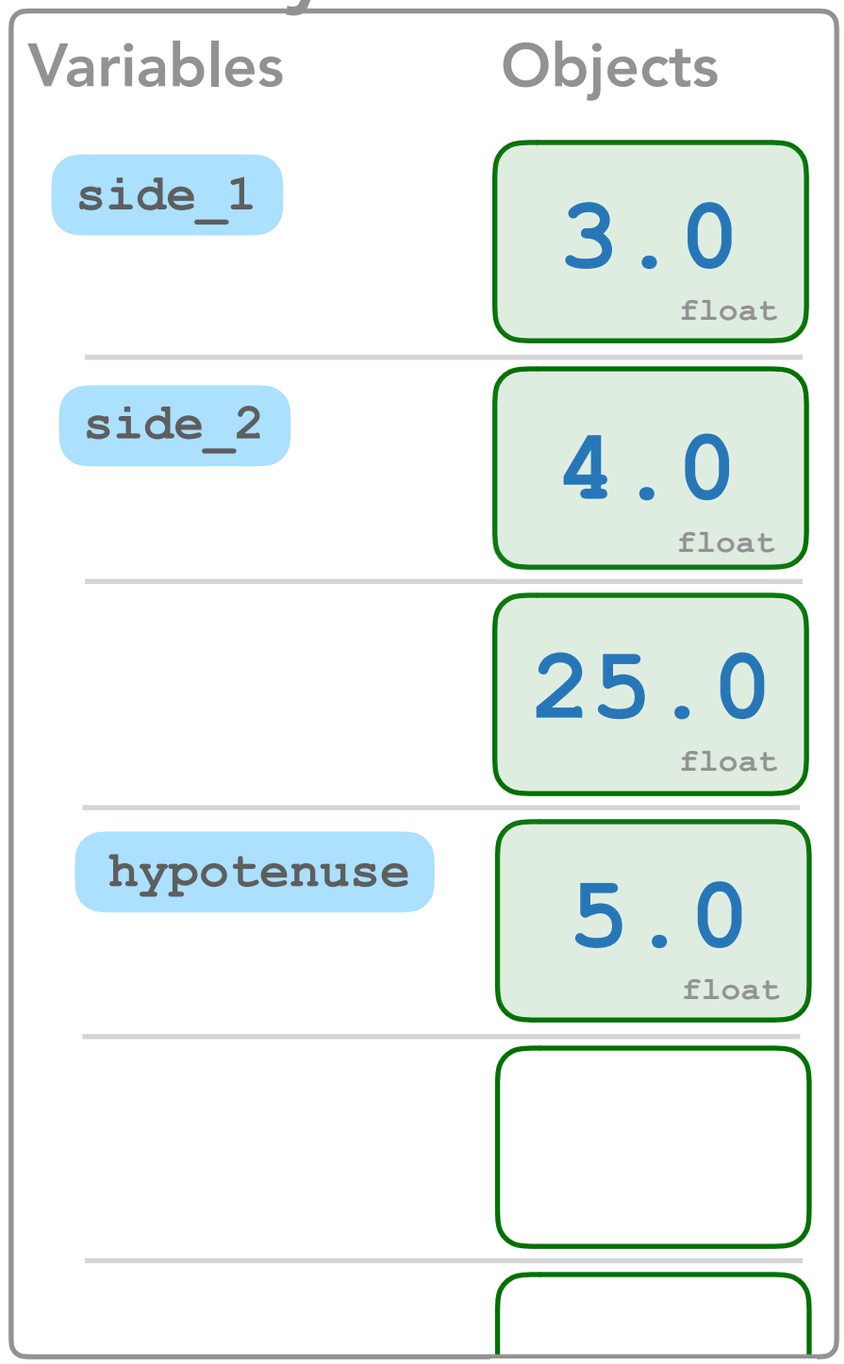
Code

```
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



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
```

Return

The object the function

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

Calculator

Conversion

Bilheteria

Calculator

Conversion

Bilheteria

WORK ON PROJECTS!!