# Numbers, Variables, And More

# Write Your Review

Click on a star to change your rating 1 - 5, where 5 = great! and 1 = really bad

**Your Review:**

Your Reivew

999 Characters remaining

# Your Info:

**Name:**

Name

**Email:**

Email

**I Agree to the Terms blah blah blah**

Submit

# Basic Data Types

Strings

Integers

Booleans

Floats

# Data Types

| | | |
|---|---|---|
| Strings | Integers | Frozenset |
| Booleans | Floats | Bytes |
| Dictionary | Complex | Range |
| List | Tuple | Set |

# Integers

- Whole numbers only
- Positive or Negative
- No Decimal Points!

# Integers

- Whole numbers only
- Positive or Negative
- No Decimal Points!

9

# Integers

- Whole numbers only
- Positive or Negative
- No Decimal Points!

9

378

# Integers

- Whole numbers only
- Positive or Negative
- No Decimal Points!

| | | |
|---|---|---|
| 9 | 378 | -21 |

# Integers

- Whole numbers only
- Positive or Negative
- No Decimal Points!

| | | |
|---|---|---|
| 9 | 378 | -21 |

# Floats

- Written With a Decimal Point
- Positive or Negative

# Integers

- Whole numbers only
- Positive or Negative
- No Decimal Points!

9

378

-21

# Floats

- Written With a Decimal Point
- Positive or Negative

1.5

# Integers

- Whole numbers only
- Positive or Negative
- No Decimal Points!

9

378

-21

# Floats

- Written With a Decimal Point
- Positive or Negative

1.5

9.99

# Integers

- Whole numbers only
- Positive or Negative
- No Decimal Points!

| | | |
|---|---|---|
| 9 | 378 | -21 |

# Floats

- Written With a Decimal Point
- Positive or Negative

| | | |
|---|---|---|
| 1.5 | 9.99 | -2.0 |

# INT

1234987234321

+8

0

378

1_000_000_000

-99

-8363247238498

1782

# FLOAT

1.3333333

0.5

+8.1

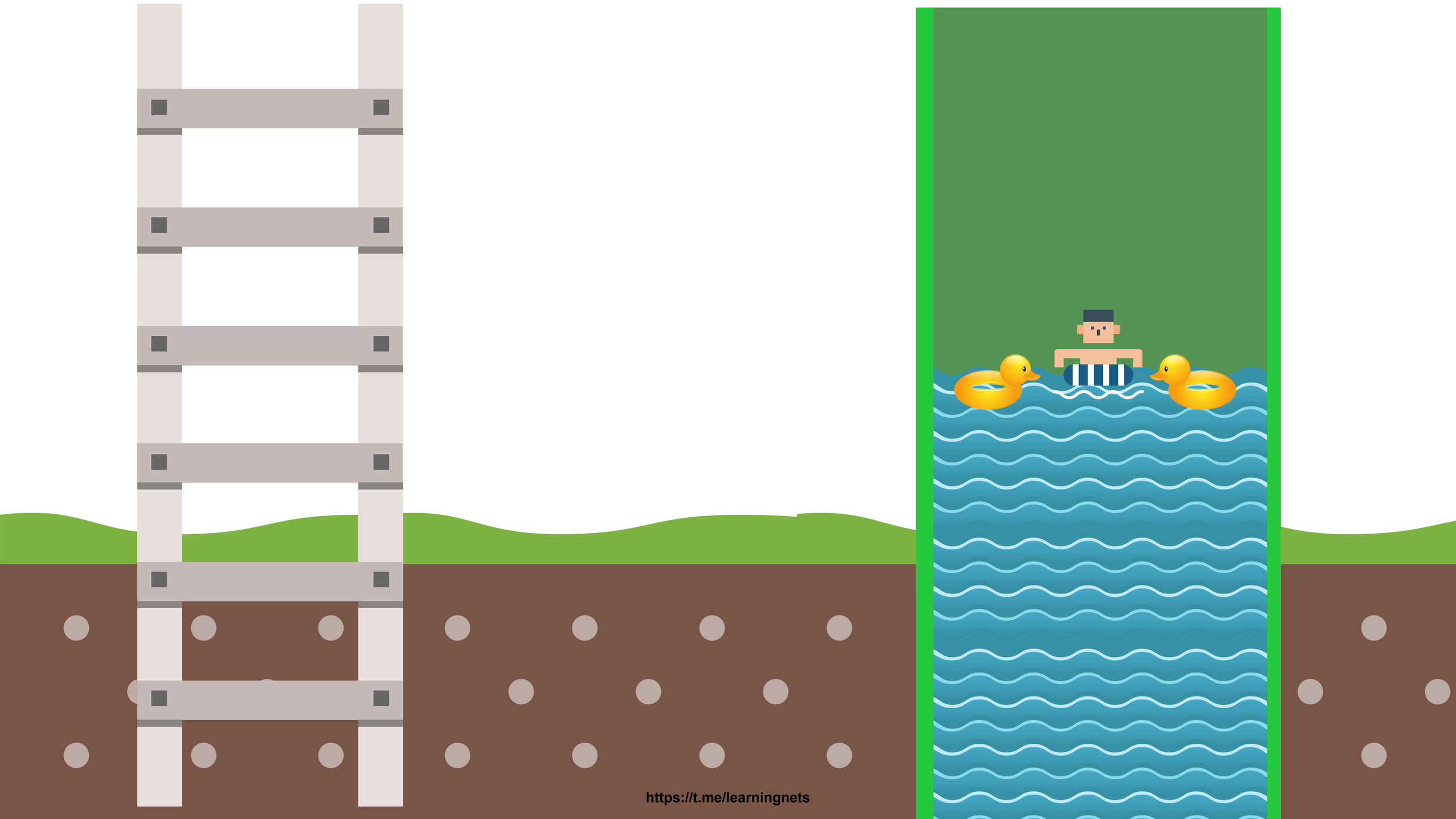
-.2

0.0

0.8372984732894733

1.234987234321e12

04.0



# Ints

1_000

777

1,000

0777

# Floats

1_000.55

1.234e12

1,000.55

1.234 e12

# Operators

Operators are special characters in Python that perform operations on value(s). Below are some of the most common:

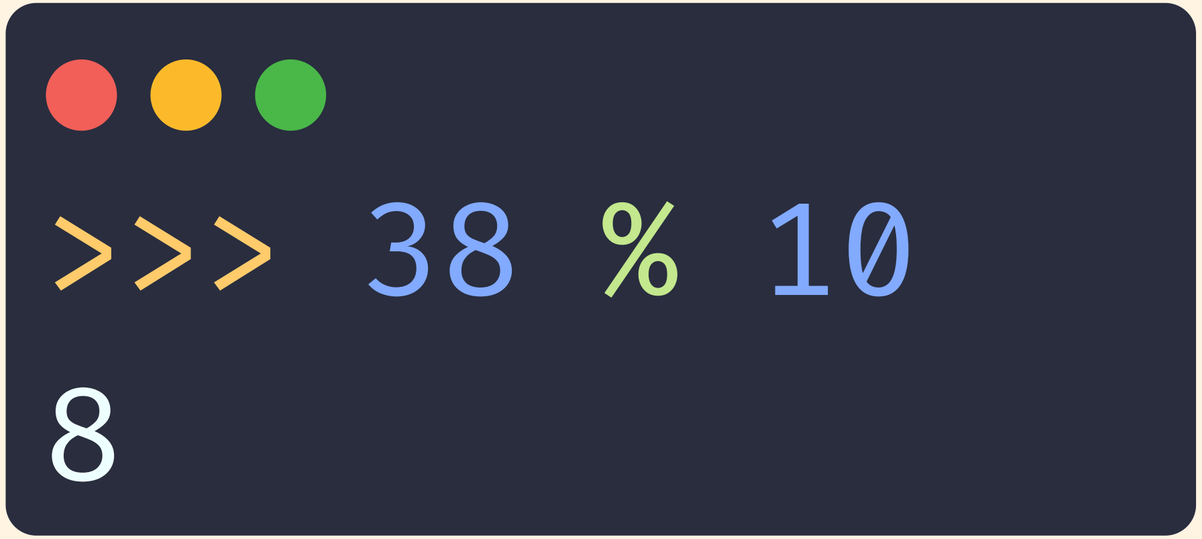
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| + | * | > | <= | and | is | = | *= |
| - | / | < | == | or | in | += | /= |
| ** | % | >= | != | not | != | -= | \|= |

# Order of Operations

| | |
|---|---|
| Parentheses | ( ) |
| Multiplication and Division | * / // |
| Addition and Subtraction | + - |

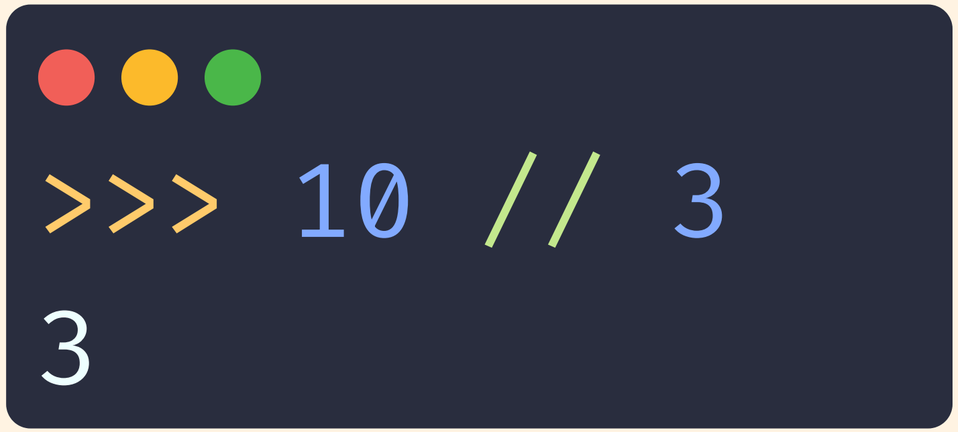
| | |
|---|---|
| Integer Division | // |
| Exponentiation | ** |
| Modulo | % |

# Modulo




10 goes into 38...
3 times, with a remainder of 8

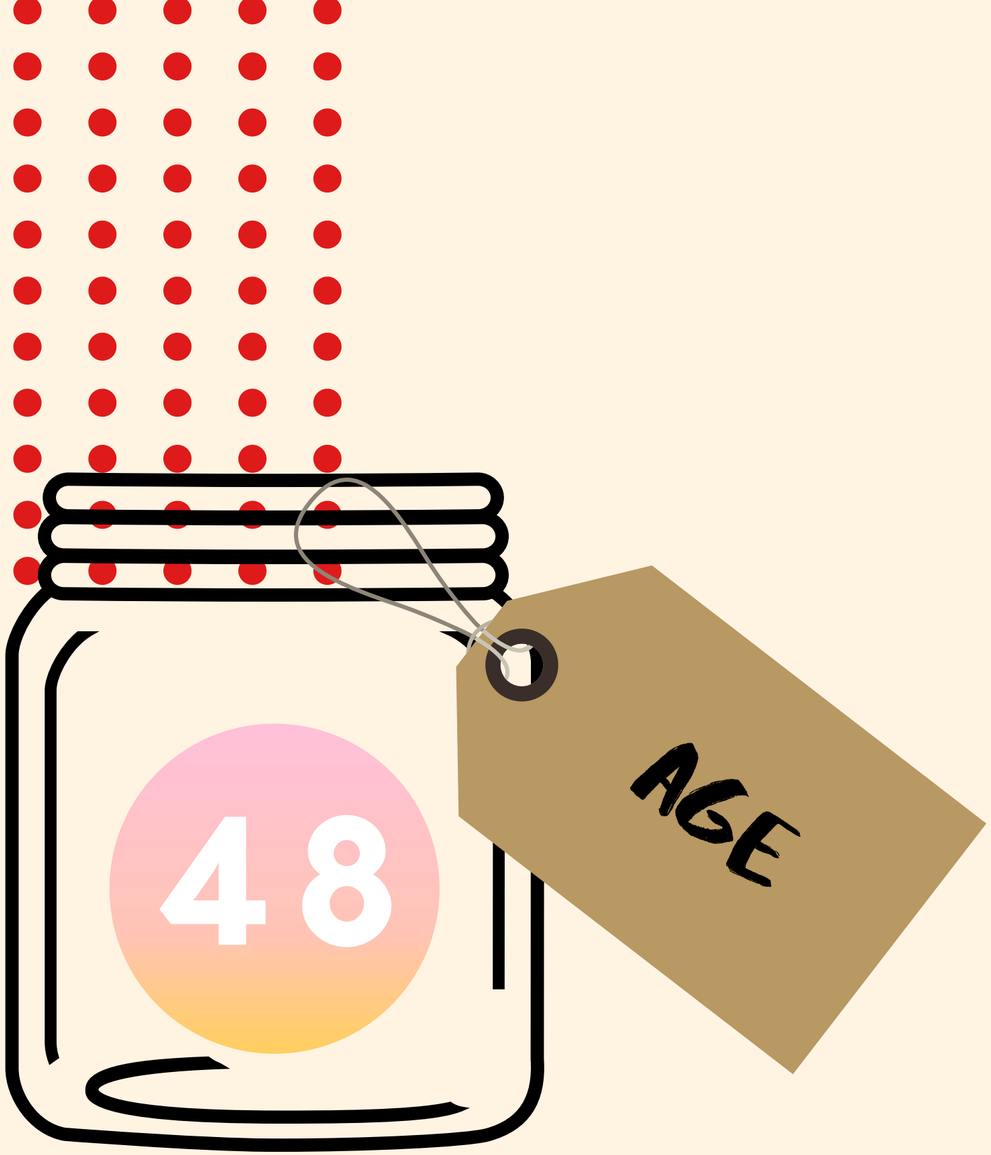
# Integer Division




# Comments

```
# this never runs
```

Python will ignore any lines starting with the # symbol

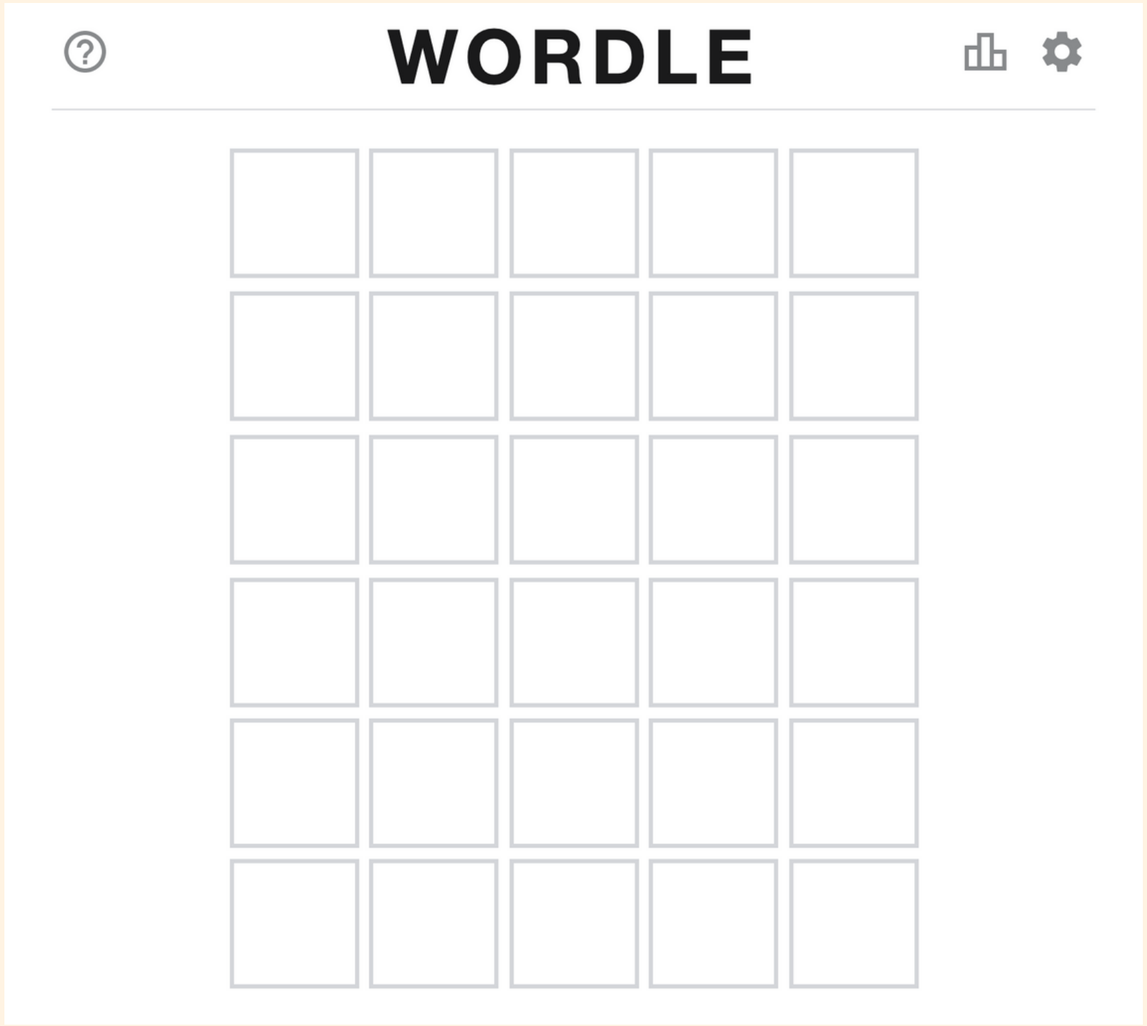



# Variables

## VARIABLES ARE LIKE LABELS FOR VALUES

We can store a value and give it a name so that we can:

- Refer back to it later
- Use that value to do...stuff
- Change it later on




# Potential Variables

num_guesses → 0

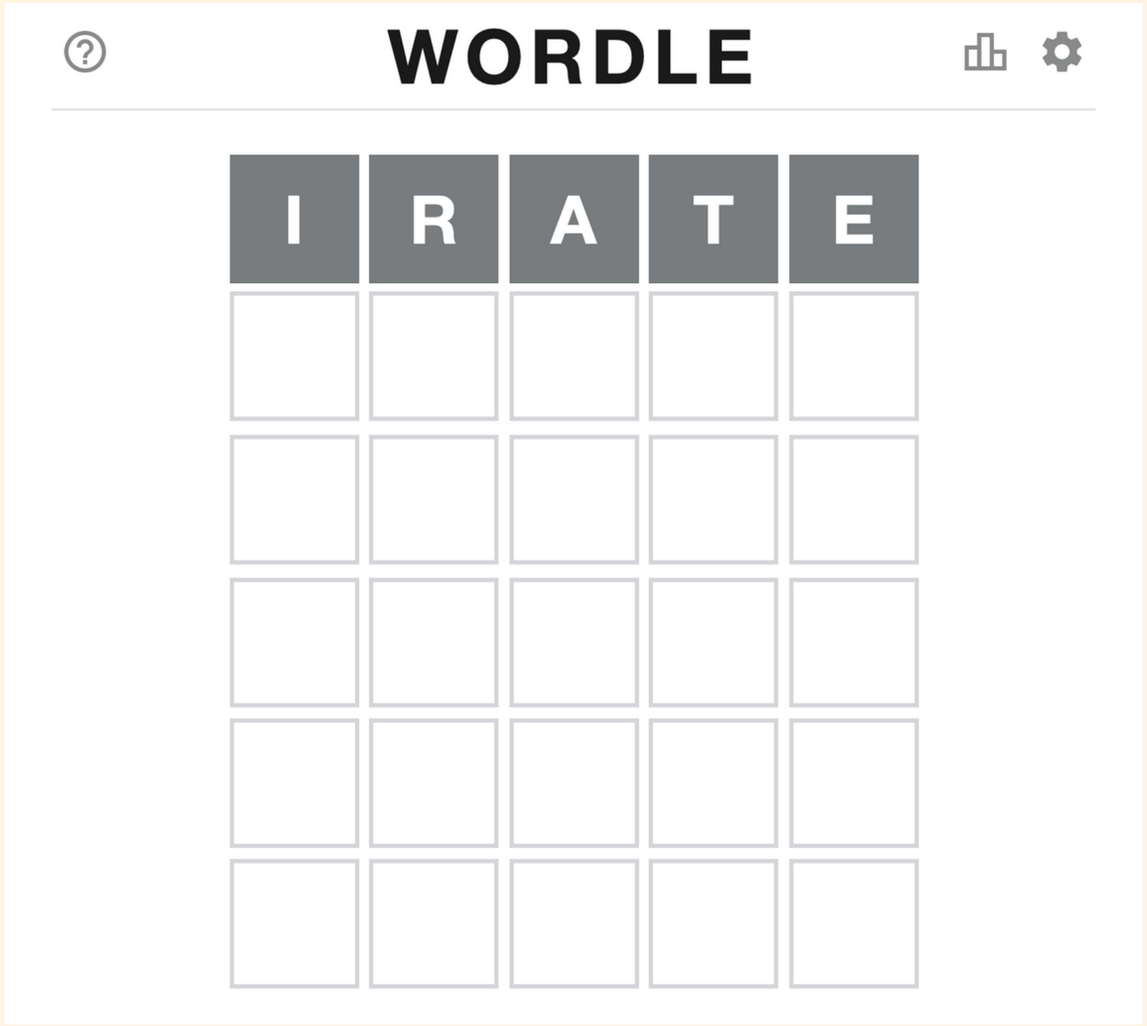
target_word → "'knoll"

max_guesses → 6

guesses →

correct_letters →

game_over → False




# Potential Variables

num_guesses → 1

target_word → "'knoll"

max_guesses → 6

guesses → ""irate"

correct_letters →

game_over → False


# Potential Variables

num_guesses → 2

target_word → ""knoll"

max_guesses → 6

guesses → ""irate", "sound"

correct_letters → "o", "n"

game_over → False


# Potential Variables

num_guesses → 3

target_word → "knoll"

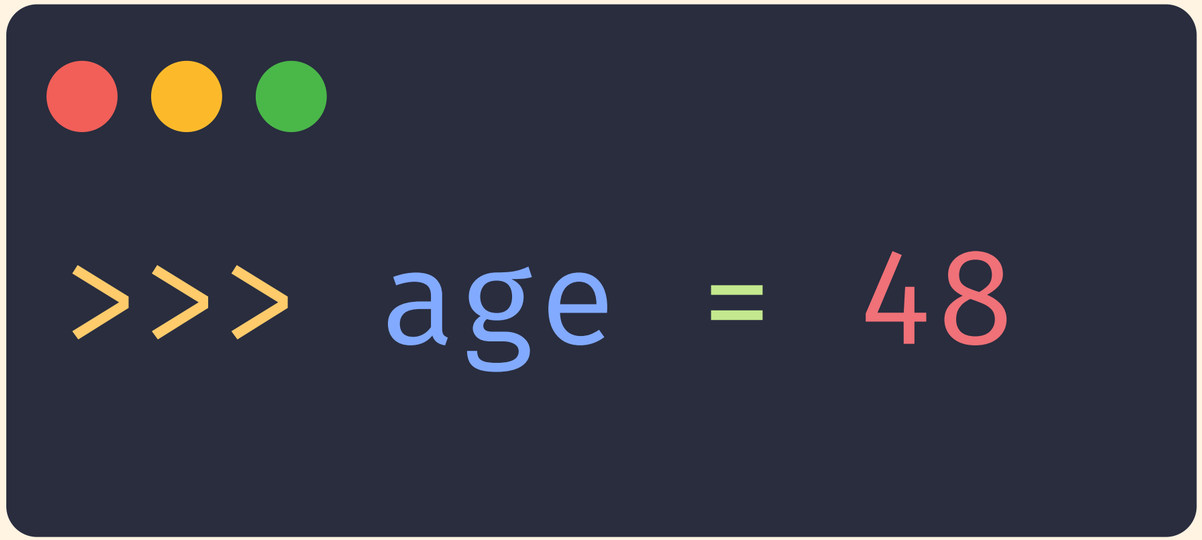
max_guesses → 6

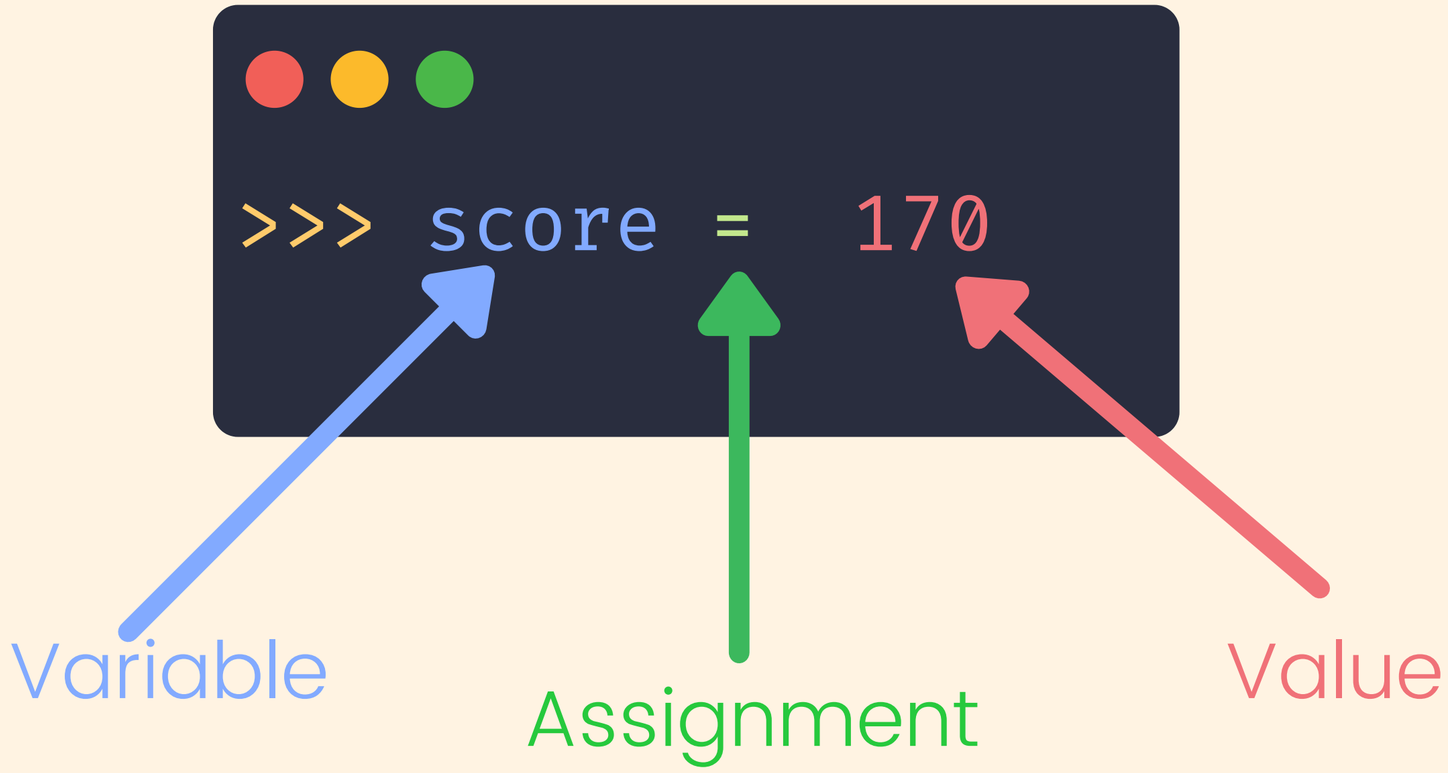
guesses → ""irate", "sound", "knoll"

correct_letters → "o", "n", "k", "l"

game_over → True

# Variables






variable123

first_name

player_1

123variable

first name

False

def

I

O

x

FirstName

FIRSTNAME

# Python Keywords

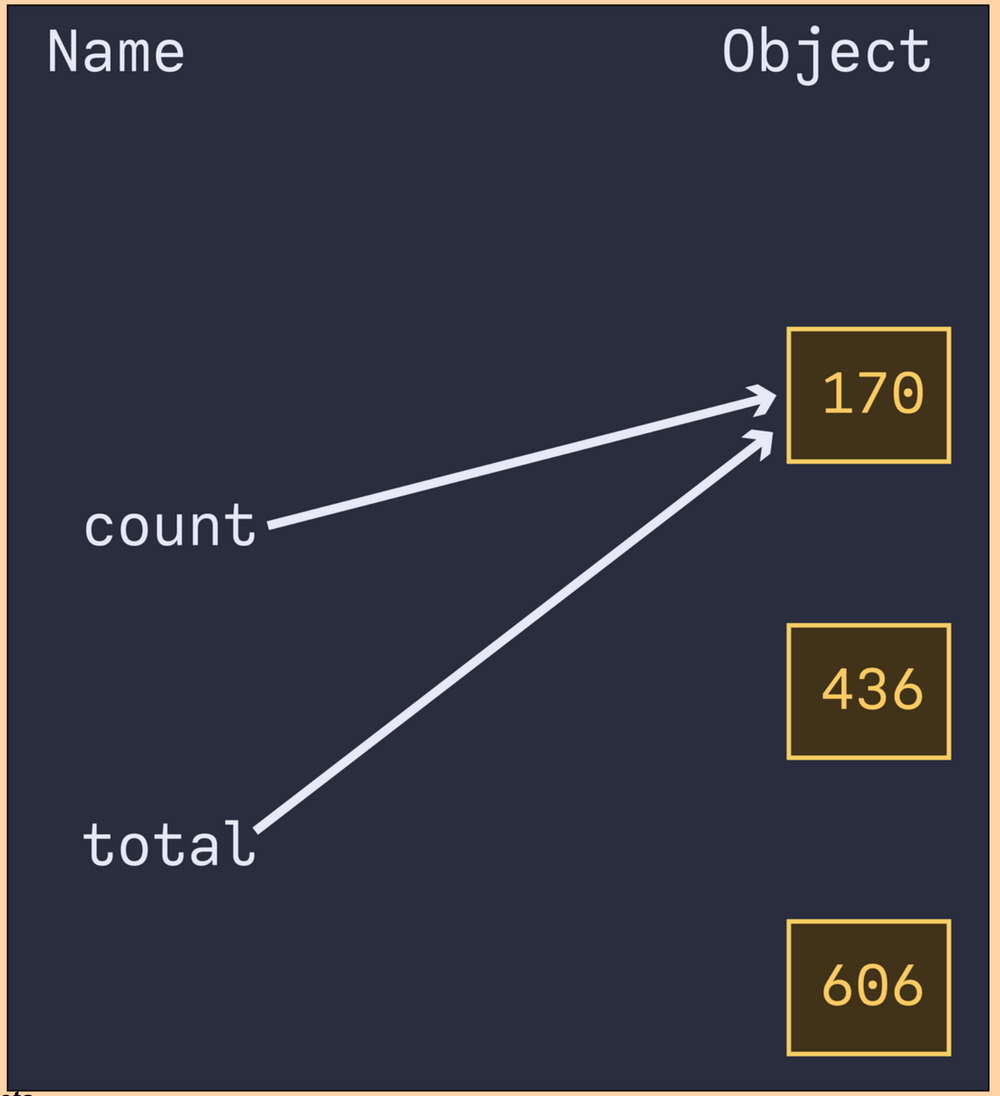
```
>>> help("keywords")
```

| | | | | |
|---|---|---|---|---|
| False | await | else | import | pass |
| None | break | except | in | raise |
| True | class | finally | is | return |
| and | continue | for | lambda | try |
| as | def | from | nonlocal | while |
| assert | del | global | not | with |
| async | elif | if | or | yield |

```
False   raise   in          not       global
True    try     is          with      del
None    except  as          or        assert
for     import  and         class
while   from    continue    def
if      pass    yield       finally
else    break   await       lambda
elif    return  async       nonlocal
```

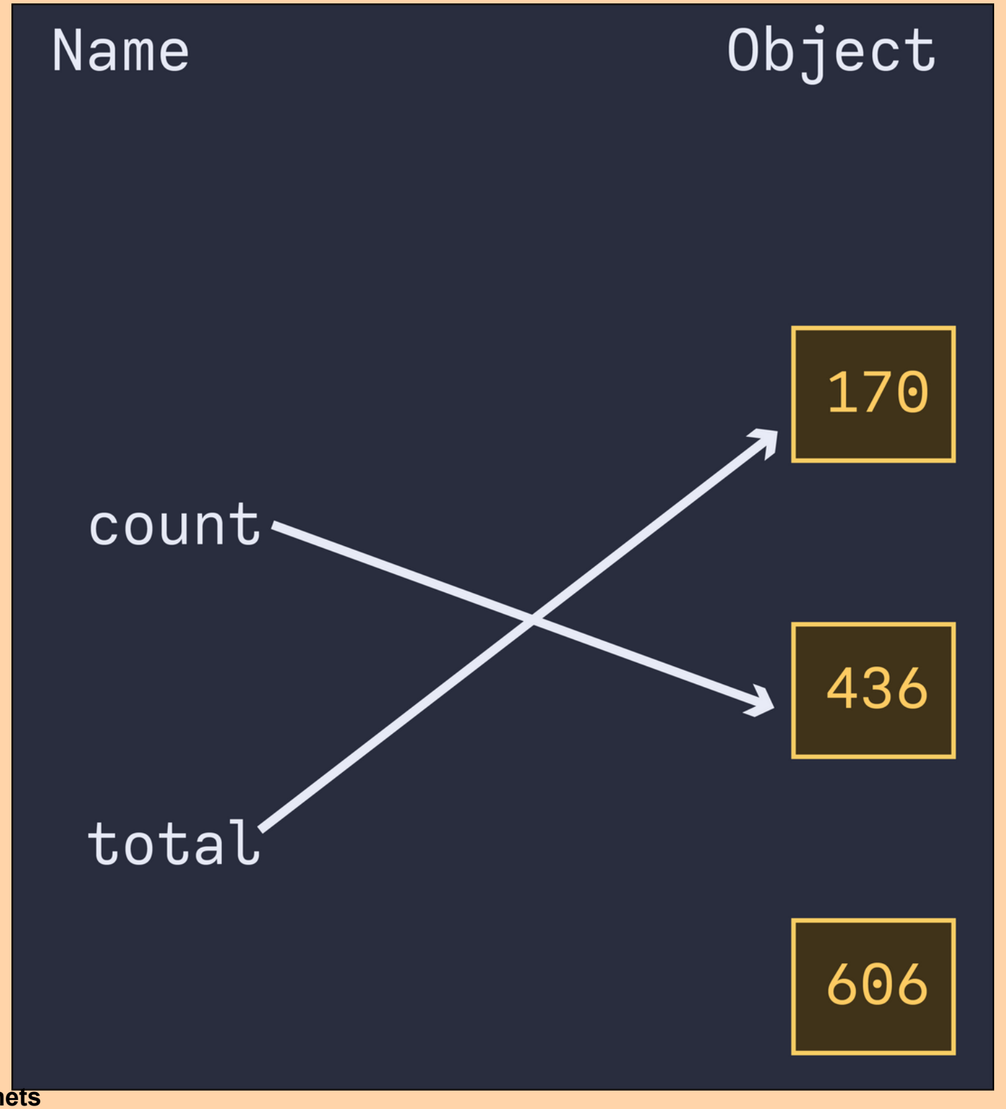
# How Variables Work

```
>>> count = 170

>>> total = 170
```



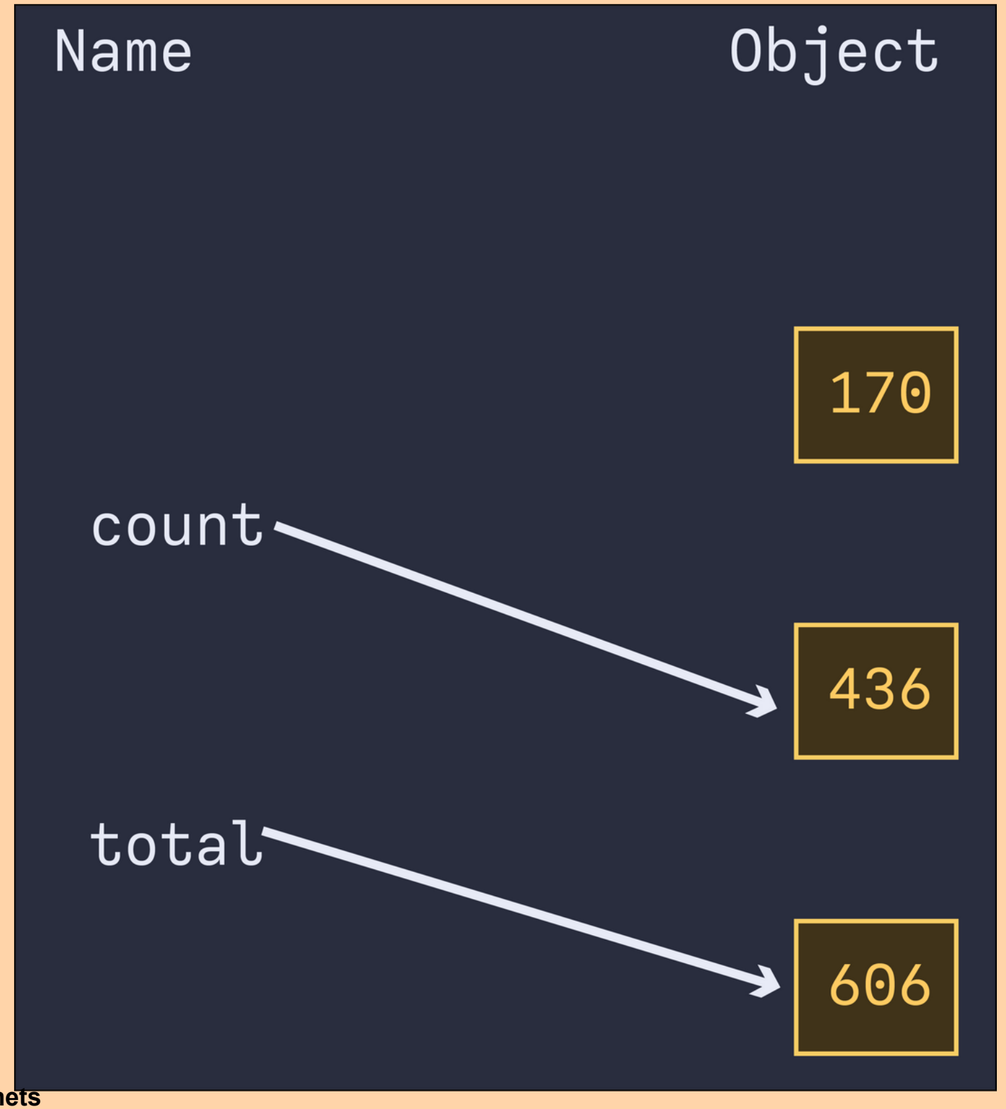
# How Variables Work

```
>>> count = 170

>>> total = 170

>>> count = 436
```



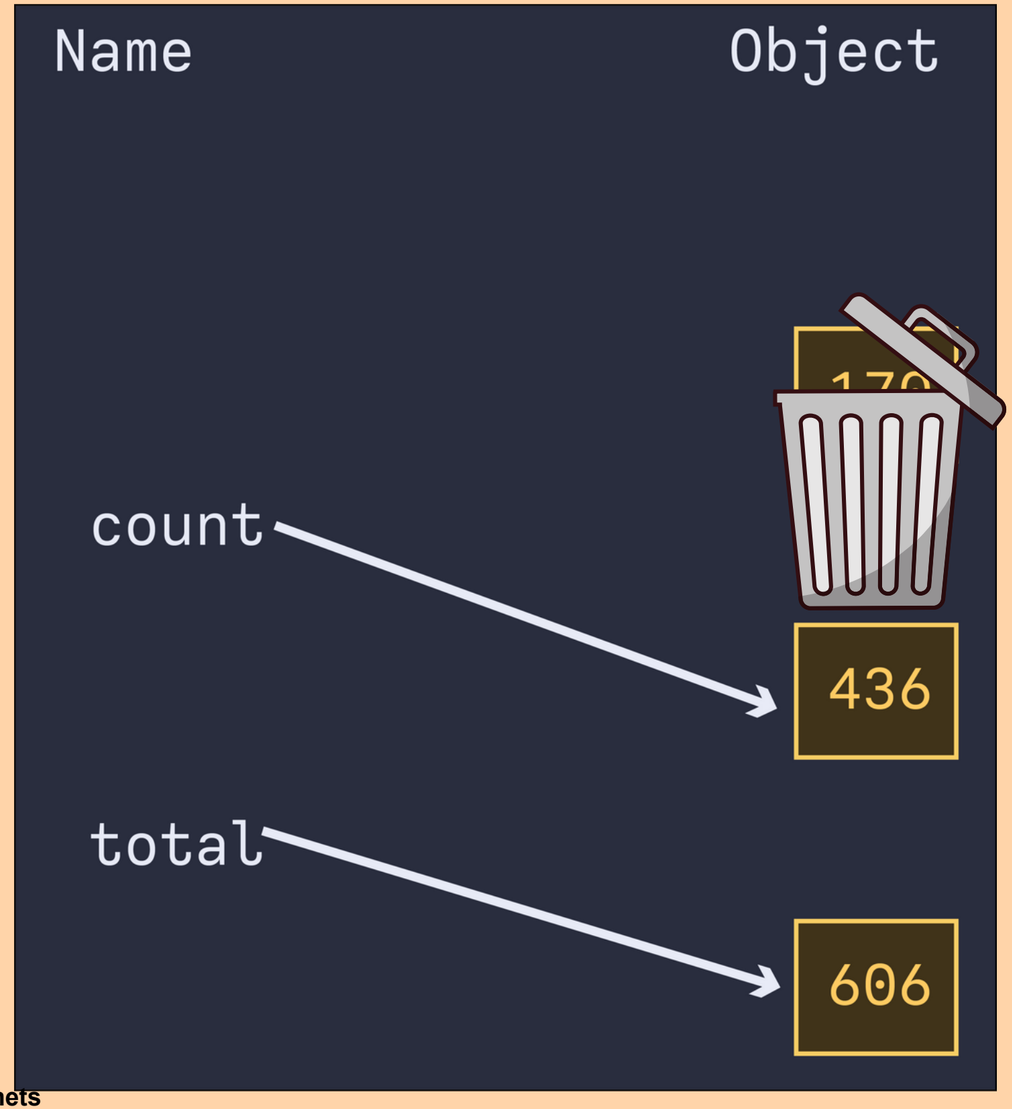
# How Variables Work

```
>>> count = 170
>>> total = 170
>>> count = 436
>>> total = 606
```



# How Variables Work

```
>>> count = 170
>>> total = 170
>>> count = 436
>>> total = 606
```



# Assignment Operators

`+=`

`-=`

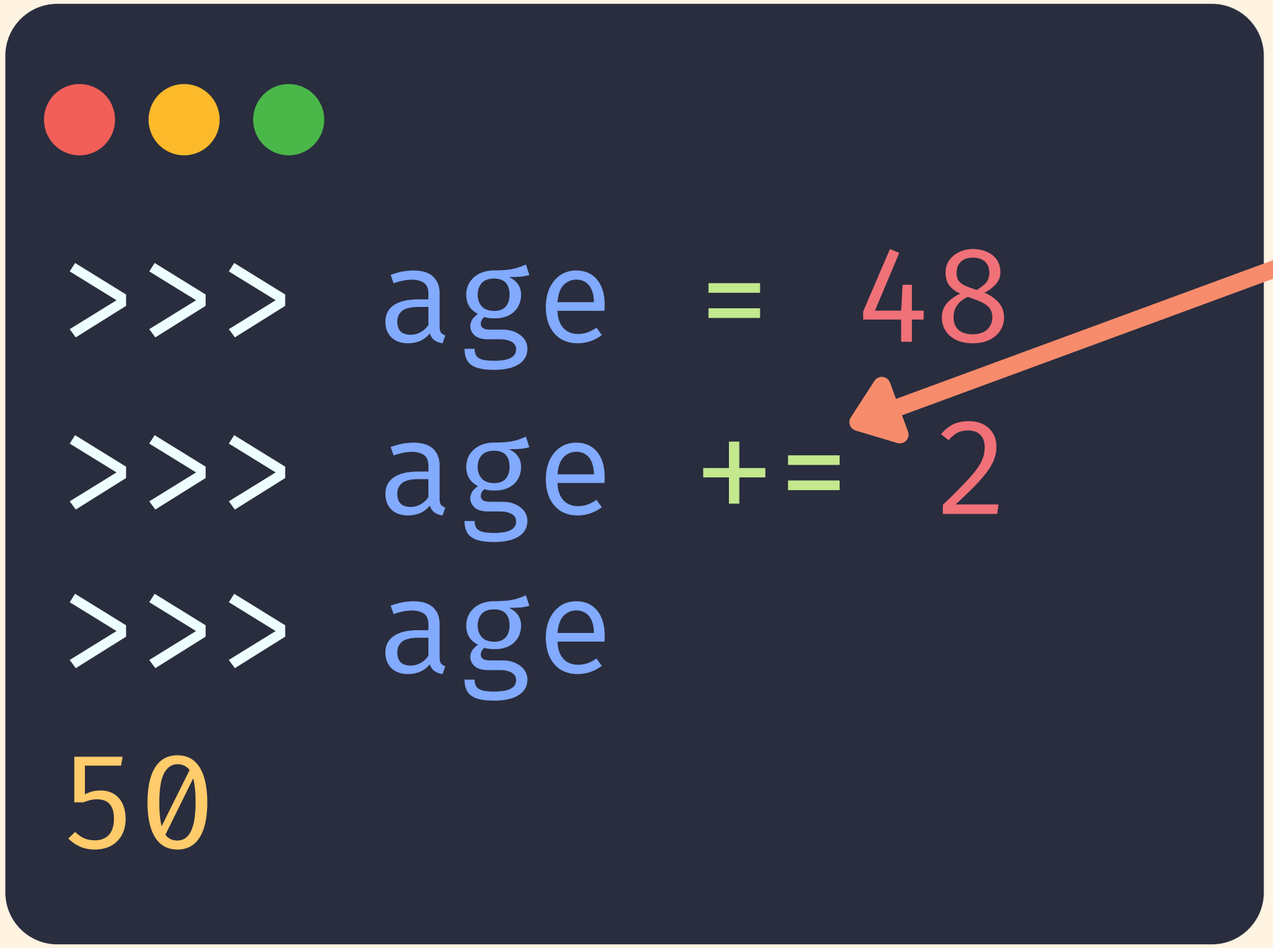
`*=`

`/=`

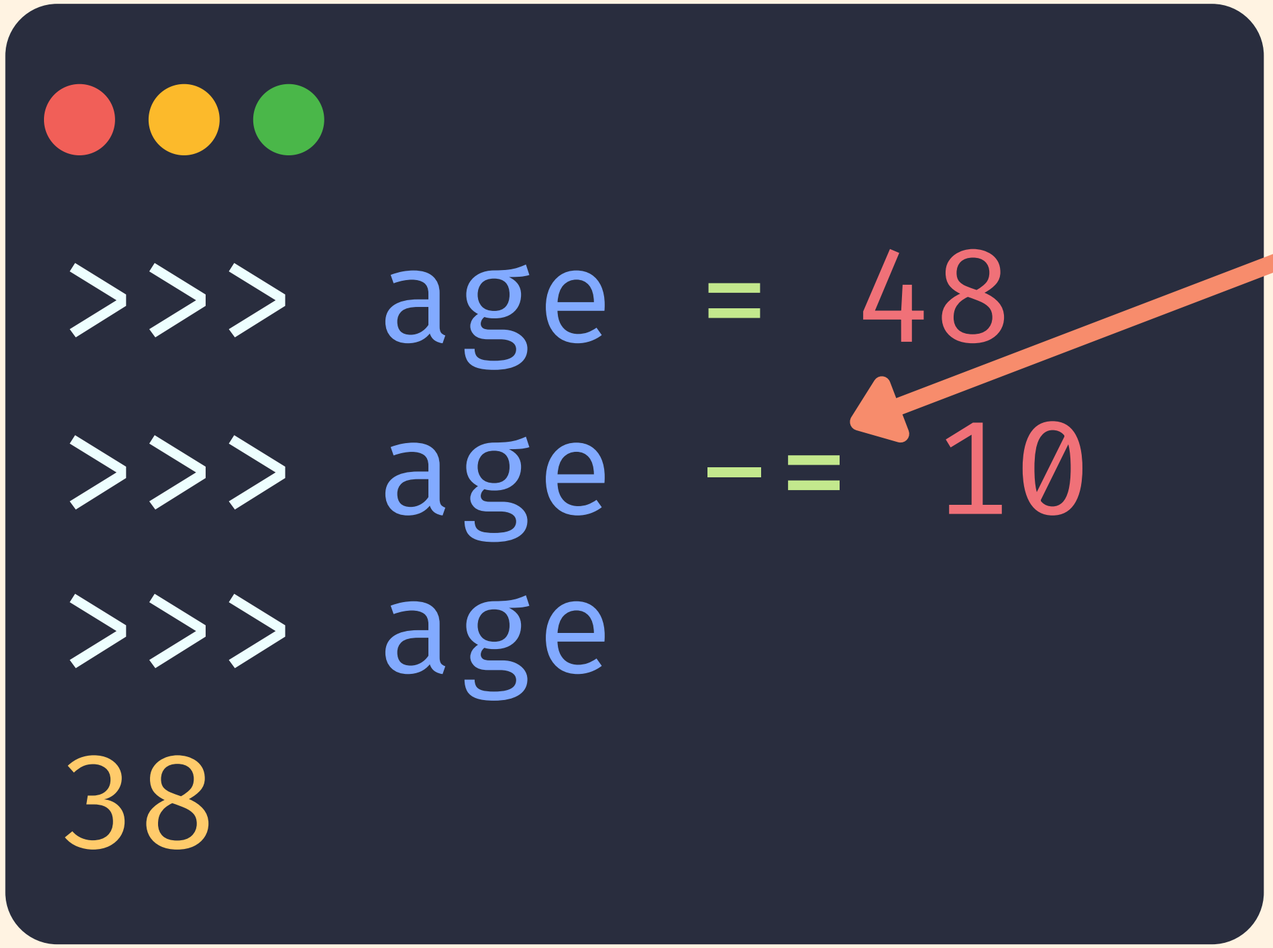
`//=`

`**=`

`%=`




Update age to be its current value (48) plus 2




Update age to be its current value (48) minus 10