

Find type of object:

## YOU TRY IT!

- In your console, find the type of:
  - 1234
  - 8.99
  - 9.0
  - True
  - False

**ANSWER:**

```
>>> type(1234)
<class 'int'>
>>> type(8.99)
<class 'float'>
>>> type(9.0)
<class 'float'>
>>> type(True)
<class 'bool'>
>>> type(False)
<class 'bool'>
```

**Type Conversion:**

## YOU TRY IT!

- In your console, find the type of:
  - float(123)
  - round(7.9)
  - float(round(7.2))
  - int(7.2)
  - int(7.9)

**ANSWER:**

```
>>> float(123)
123.0
>>> round(7.9)
8
>>> float(round(7.2))
7.0
>>> int(7.2)
7
>>> int(7.9)
7
```

### Expressions:

## YOU TRY IT!

- In your console, find the values of the following expressions:

- `(13-4) / (12*12)`
- `type(4*3)`
- `type(4.0*3)`
- `int(1/2)`

### ANSWER:

```
>>> (13-4)/(12*12)
0.0625
>>> type(4*3)
<class 'int'>
>>> type(4.0*3)
<class 'float'>
>>> int(1/2)
0
```

### Variables:

## YOU TRY IT!

- Which of these are allowed in Python? Type them in the console to check.

- `x = 6`
- `6 = x`
- `x*y = 3+4`
- `xy = 3+4`

### ANSWER:

```
>>> x=6
>>> x
6
>>> 6=x
File "<stdin>", line 1
SyntaxError: can't assign to literal
>>> x*y=3+4
File "<stdin>", line 1
SyntaxError: can't assign to operator
>>> xy=3+4
>>> xy
7
```

## Execution of code

# YOU TRY IT!

- These 3 lines are executed in order. What are the values of `meters` and `feet` variables at each line in the code?

```
meters = 100
feet = 3.2808 * meters
meters = 200
```

### ANSWER:

Using Python tutor

Step 1:

Python 3.6  
[known limitations](#)

```
1 meters = 100
2 feet = 3.2808 * meters
3 meters = 200
```

[Edit this code](#)

→ line that just executed  
→ next line to execute

<< First < Prev Next > Last >>

Step 1 of 3

Frames: Objects

Step 2:

Python 3.6  
[known limitations](#)

```
1 meters = 100
2 feet = 3.2808 * meters
3 meters = 200
```

[Edit this code](#)

→ line that just executed  
→ next line to execute

<< First < Prev Next > Last >>

Step 2 of 3

Frames: Objects

Global frame	
meters	100

Step 3:

Python 3.6  
[known limitations](#)

```
1 meters = 100
2 feet = 3.2808 * meters
3 meters = 200
```

[Edit this code](#)

→ line that just executed  
→ next line to execute

<< First < Prev Next > Last >>

Step 3 of 3

Frames: Objects

Global frame	
meters	100
feet	328.08

Step 4:

Python 3.6  
[known limitations](#)

```
1 meters = 100
2 feet = 3.2808 * meters
→ 3 meters = 200
```

[Edit this code](#)

→ line that just executed  
→ next line to execute

<< First < Prev Next > Last >>

Done running (3 steps)

Frames

Global frame	
meters	200
feet	328.08

Objects

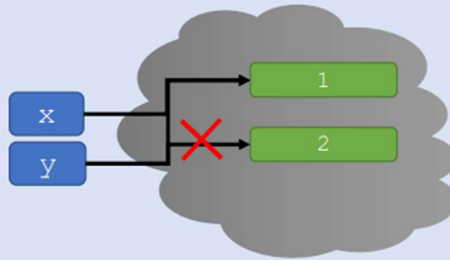
Swapping:

## YOU TRY IT!

- Swap values of x and y without binding the numbers directly. Debug (aka fix) this code.

```
x = 1
y = 2

y = x
x = y
```



ANSWER:

Step 1:

Python 3.6  
[known limitations](#)

```
→ 1 x = 1
→ 2 y = 2
3
4 y = x
5 x = y
```

[Edit this code](#)

→ line that just executed  
→ next line to execute

<< First < Prev Next > Last >>

Step 2 of 4

Frames

Global frame	
x	1

Objects

Step 2:

Python 3.6  
[known limitations](#)

1 x = 1

→ 2 y = 2

3

→ 4 y = x

5 x = y

[Edit this code](#)

→ line that just executed  
→ next line to execute

<< First < Prev Next > Last >>

Step 3 of 4

Frames

Objects

Global frame

x 1

y 2

Step 3:

Python 3.6  
[known limitations](#)

1 x = 1

2 y = 2

3

→ 4 y = x

→ 5 x = y

[Edit this code](#)

→ line that just executed  
→ next line to execute

<< First < Prev Next > Last >>

Step 4 of 4

Frames

Objects

Global frame

x 1

y 1

Step 4:

Python 3.6  
[known limitations](#)

1 x = 1

2 y = 2

3

4 y = x

→ 5 x = y

[Edit this code](#)

→ line that just executed  
→ next line to execute

<< First < Prev Next > Last >>

Done running (4 steps)

Frames

Objects

Global frame

x 1

y 1