Computer Programming (with Python) Week 3

Matt Harrison

Solution to Problem

School

```
question = "What is your school?"
school = raw_input(question)
print "You go to " + school
```

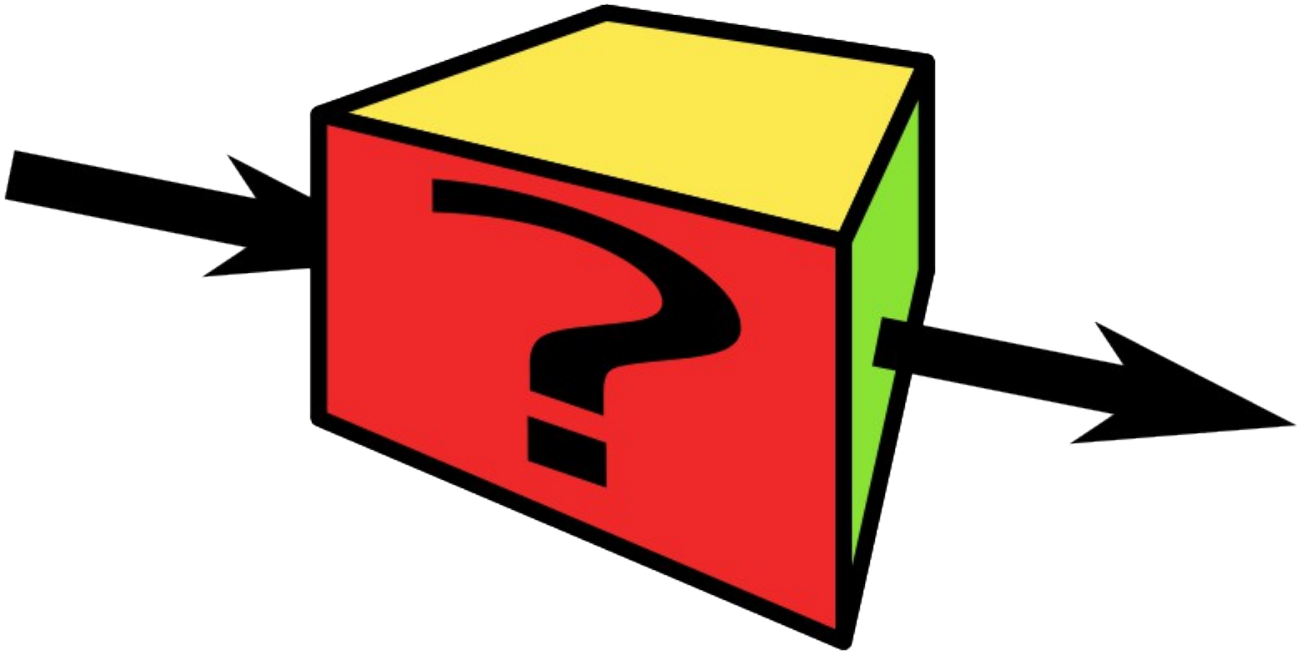
Lessons

- I need to be more descriptive

Functions

Functions take input, apply an operation and return output

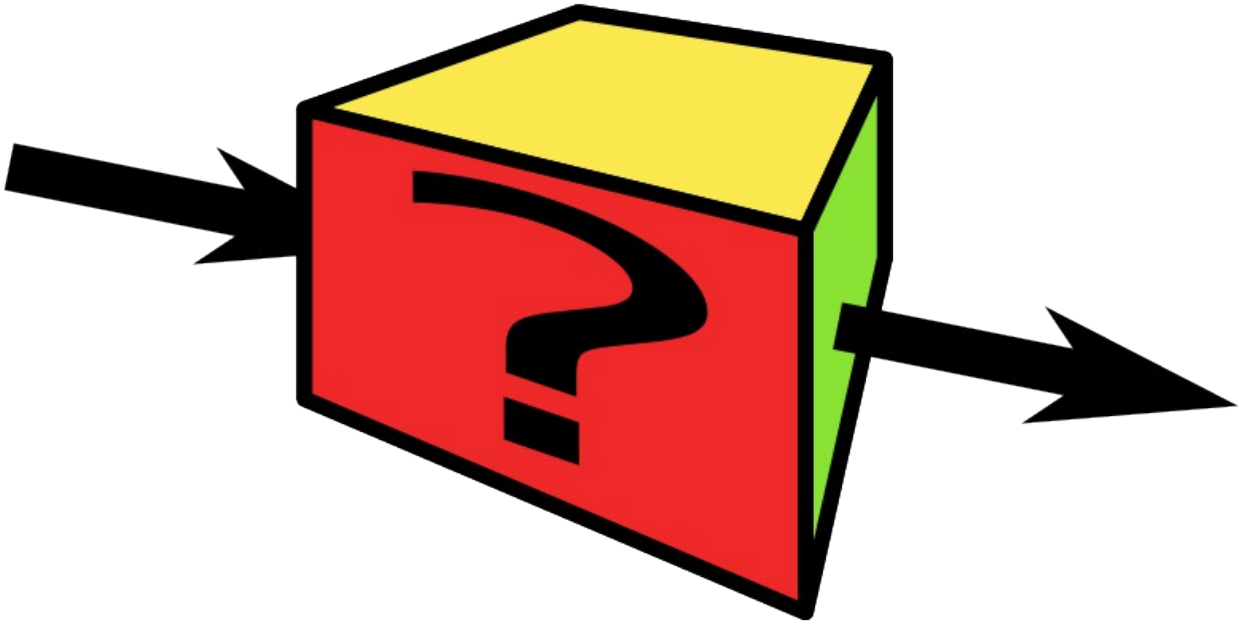
2



6

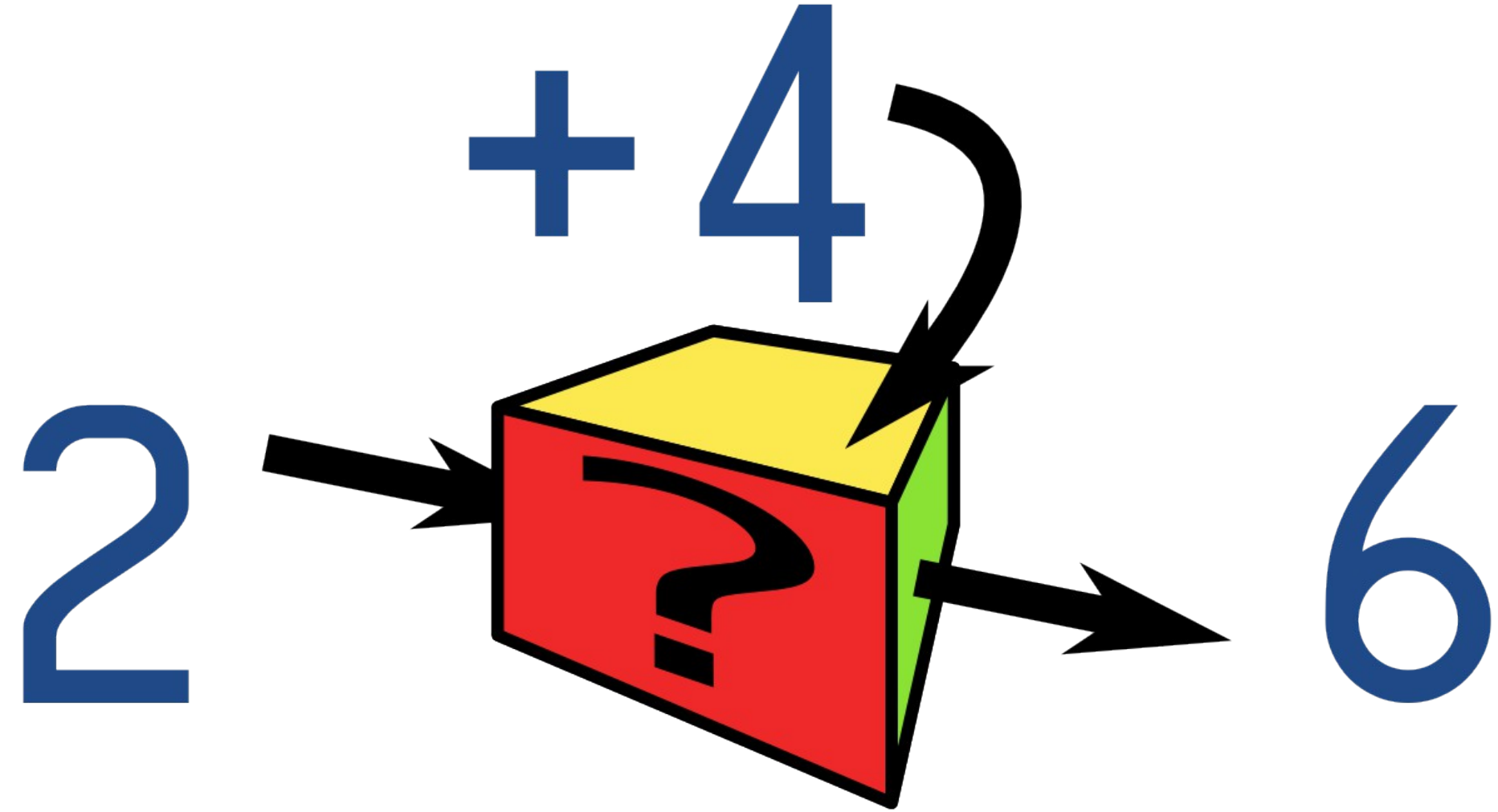
What does ? do?

1



5

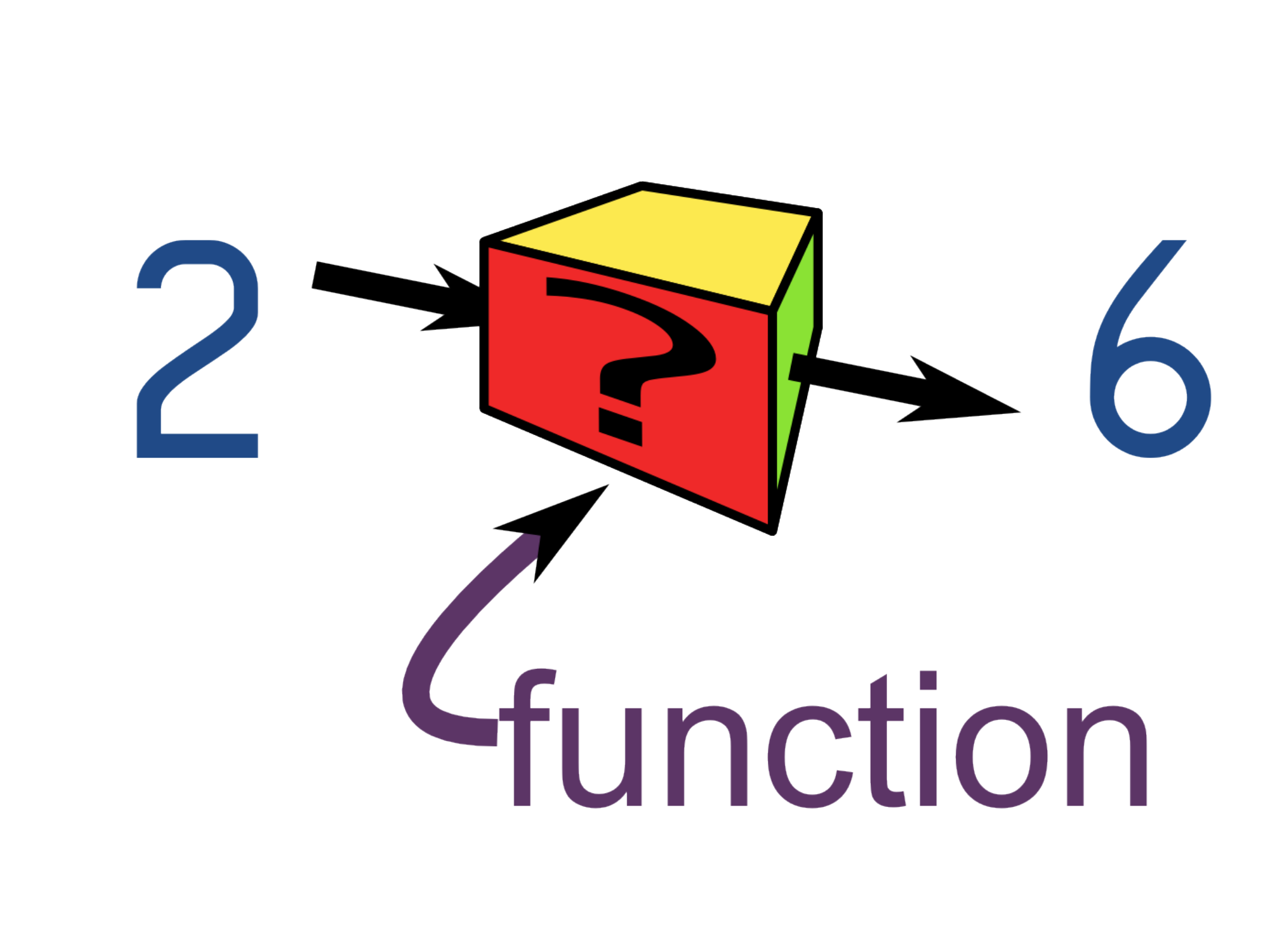
What does ? do?


+4
2
?
6

Three important parts

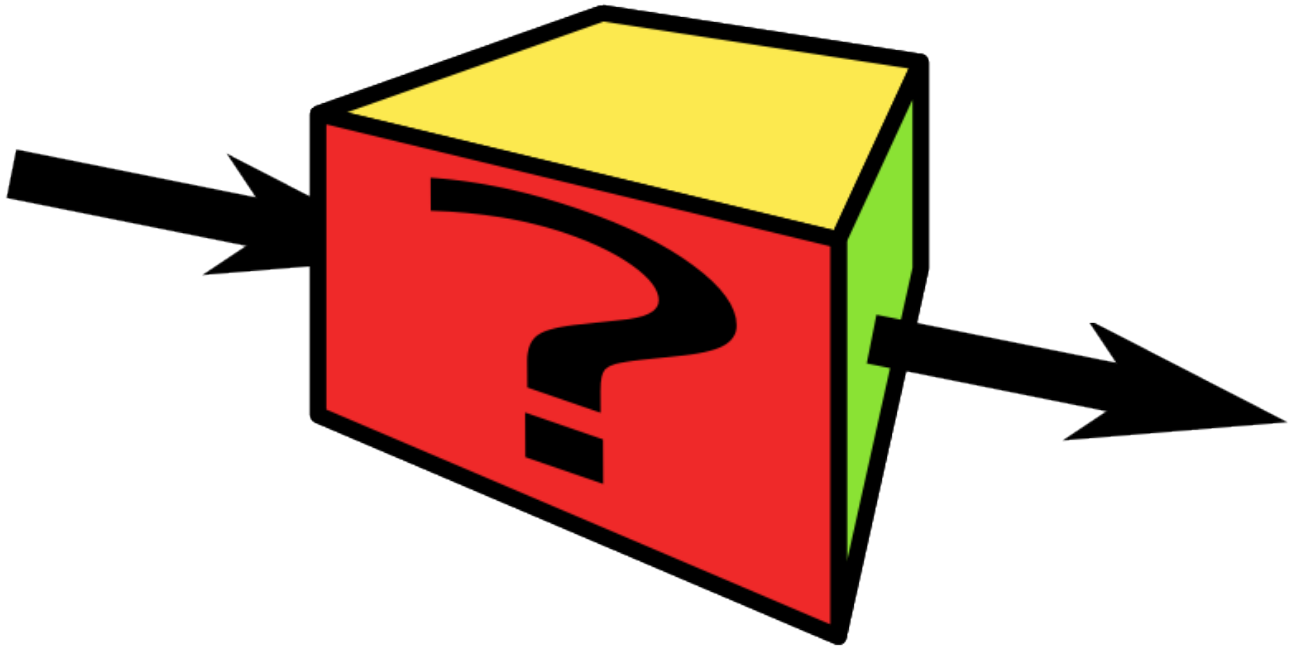
Parts

- Function
- Input
- Output


2
?
6
function

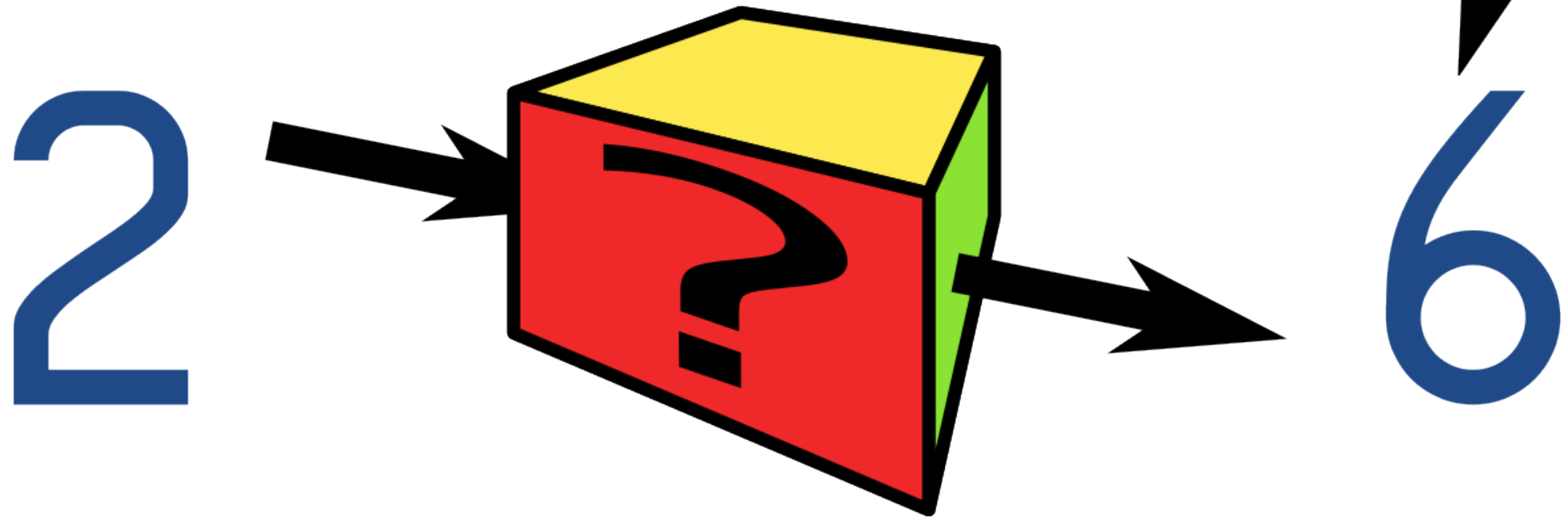
input

2



6

output



Python version

```
def add_4(input):
    answer = input + 4
    return answer
```

or

```
def add_4(num):
    return num + 4
```

Calling functions

```
def add_4(num):
    answer = num + 4
    return answer

seven = add_4(3)
print seven
```

Quiz

Name function, input and output

```
def add_4(num):
    answer = num + 4
    return answer
```

Colons!

colons (:) go with *indentation* in Python

Python version

Find colon and indentation

```
def add_4(num):
    answer = num + 4
    return answer
```

Functions

- Start with `def`
- then function name
- then `(`
- then *input variables*
- then `)`
- then `:` (and return)
  - then indent logic
  - then `return` answer

`def`

`def` means *define* a function

function names

similar to variable names:

- lower case
- `_` (underscore multiply words)
- can't start with numbers
- should be verbs (ie **`add_two`**, **`load_data`**)

Input

If you have more than one input, separate with commas (,)

```
def add(x, y, z):
    return x + y + z
```

Input 2

Sometimes you don't have input

```
def get_name():
    return "Matt"
```

Whitespace

All of function *logic* is indented (4 spaces).

```
def process(x, y):
    a = x + y
    b = x - y
    c = x * y
    return a + b + c
```

`return`

`return` tells the program what the *output* is

return 2

Don't have to have a `return`

```
def print_name():
    print "Matt"
```

Calling Functions

```
output = function_name(input)
```

Have we seen any *functions*?

YES!

- `int`
- `str`
- `raw_input`

Assignment

Write a function that:

- takes a number
- subtracts 5 from that number
- returns the result

Extra Assignment

Write a function that:

- takes degrees Celsius (Metric)
- returns degrees Fahrenheit (US)

(C to F multiply by 9 divide by 5 add 32)

Conditions

New data *type*

- int
- float
- string
- boolean

Boolean

```
a = True
b = False
```

`if` statement

```
age = 10
if age > 18:
    print "OLD!"
```

`if` statement

- Start with **`if`**
- then *condition*
- then **`:`** (and return)
  - then indent logic

Conditionals

Syntax	Meaning
`>`	Greater than
`<`	Less than
`>=`	Greater than or equal
`<=`	Less than or equal
`==`	Equal to
`!=`	Not equal to

Conditionals evaluate to Booleans

```
>>> print 1 != 3
True
>>> print "matt" ==
"Fred"
False
```

Examples

```
name = "matt"
if name == 'matt':
    print "Cool!"

cash = 0.3
if cash < 1.0:
    print "too bad"
```

elif

```
grade = 80
letter = "F"
if grade > 90:
    letter = "A"
elif grade > 80:
    letter = "B"
elif grade > 70:
    letter = "C"
```

elif

```
grade = 80
letter = "F"
if grade > 90:
    letter = "A"
elif grade > 80:
    letter = "B"
elif grade > 70:
    letter = "C"
```

grade = ? (note the indentation)

else

```
name = 'matt'
if name == "matt":
    print "same"
else:
    print "different"
```

function with if

```
def is_matt(name):
    result = False
    if name == "matt":
        result = True
    elif name == "Matt":
        result = True
    else:
        result = False
    return result
```

Assignment

Write a function that:

- takes a number
- returns:
  - `"G"` if greater than 1000000000
  - `"M"` if greater than 1000000
  - `"K"` if greater than 1000

credits

- http://www.flickr.com/photos/marcp_dmoz/