

# Functions in Python

You kinda know this already

# You've used functions before!

```
color("blue")
```

```
forward(100)
```

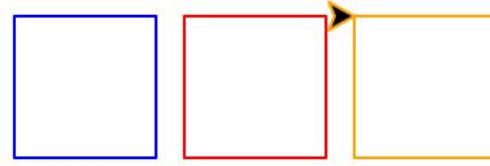
```
right(90)
```

```
print("This is a function")
```

# What is a function?



A little section of code  
inside your program that  
does a specific job.



```
for i in range(4):
    pencolor("blue")
    forward(50)
    right(90)
```

```
penup()
forward(60)
pendown()
```

```
for i in range(4):
    pencolor("red")
    forward(50)
    right(90)
```

```
penup()
forward(60)
pendown()
```

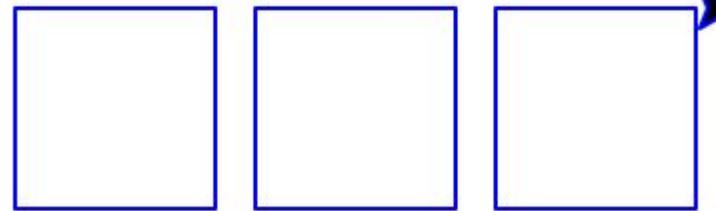
```
for i in range(4):
    pencolor("orange")
    forward(50)
    right(90)
```



Just do a `for` loop  
inside another `for`  
loop!

How do I change  
the color?

```
for i in range(3):  
    for i in range(4):
```



```
forward(50)  
pendown()
```



# We would use a function



A little section of code that draws squares of different colors.



## A Peek at a Real-Life Function Example



peek

Find the Best Things to Do

Choose from Thousands of Activities in 20+ Destinations

PICK A PLACE

LET'S GO

Sign Up

For Tour Operators

Hand-picked **tours** and **activities** for when you're at **home** or **traveling**

[www.peek.com](http://www.peek.com)

The Best Selection

Instant Booking

A Dose of

girls who  
CODE



**peek**

**Redwood Forest Train**  
75-min steam engine ride in forest

Watch enormous redwood trees pass by as you ride this historic steam train route from Roaring Camp in Felton to the top of Bear Mountain.

**Book — \$26**



peek

Private Chinatown Food Tour  
3-hr private guided walking tour and tastings  
Your group will love strolling through Chicago's Chinatown, learning about the area and cuisine as you taste delicious regional specialties.

Book — \$150



No problem!



Our designers put a drop down menu on our front page. Now we need the actual information to show up when someone chooses a city. Make sure to include the picture and price.

peek

Sign Up

For Your Operators >

## Things to Do This Week in [Chicago](#) ▾

[SEE MORE IN CHICAGO](#)

## Explore Perfect Days

Tastemakers such as Tory Burch and Wolfgang Puck share their favorite ways to spend 24 hours in a city they love. Get insider tips on the best things to do, places to see, and restaurants to visit.



girls who  
code

peek

"fun for locals" to find  
experiences you'll love.

price guarantee.

Sign Up exp  
- 50

For Tour Operators ›

memories!

## Things to Do This Week in [Chicago](#) ▾

🔍

- USA
- Boston
- Chicago
- Washington D.C.
- Los Angeles
- Las Vegas
- Miami

Tastemakers such as Tory Burch and Wolfgang Puck share their favorite ways to spend 24 hours in a city they love. Get insider tips on the best things to do, places to see, and restaurants to visit.

Sign Up

For Tour Operators ›



## Things to Do This Week in **San Francisco** ▾



Golden Gate Seaplane Tour  
San Francisco Bay



Urban Hiking Tours  
Castro



Historic Train Tour  
Santa Cruz



GoCar Tours in San Francisco  
Multiple SF Locations



Scenic Dining Cruises  
Embarcadero

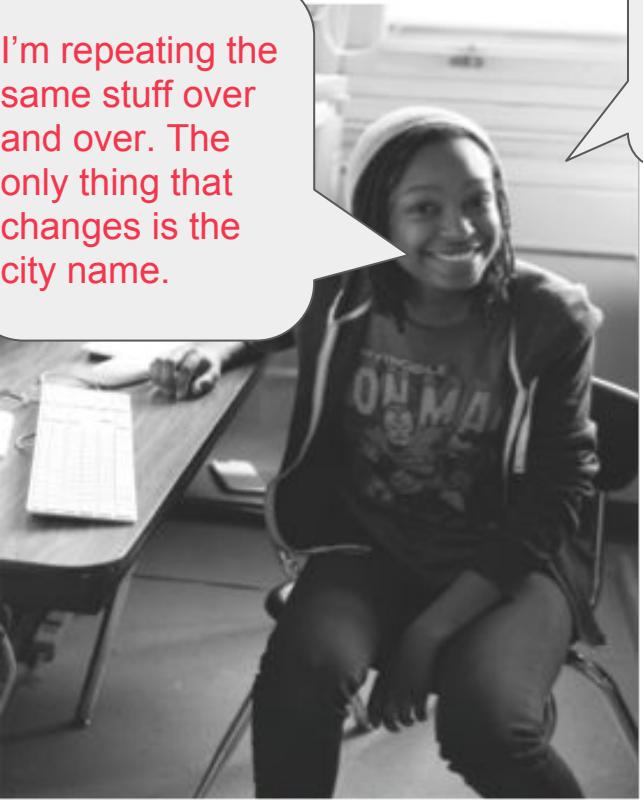


Victorian Houses Tour  
Pacific Heights / Cow Hollow

[www.peek.com/activity-p3024474-urban\\_hiking\\_tours\\_san\\_francisco/](http://www.peek.com/activity-p3024474-urban_hiking_tours_san_francisco/)

girls who [REDACTED]

```
if user_choice == "New York":  
    tours = get_tours_for("New York")  
    for each tour get the price and picture and name  
    show each tour name, picture and price  
elif: user_choice == "San Francisco":  
    tours = get_tours_for("San Francisco")  
    for each tour get the price and picture and name  
    show each tour name, picture and price  
elif: user_choice == "Chicago":  
    tours = get_tours_for("Chicago")  
    for each tour get the price and picture and name  
    show each tour name, picture and price  
elif: user_choice == "Boston":  
    tours = get_tours_for("Boston")  
    for each tour get the price and picture and name  
    show each tour name, picture and price  
elif: user_choice == "Washington D.C.":  
    tours = get_tours_for("Washington D.C.")  
    for each tour get the price and picture and name  
    show each tour name, picture and price  
elif: user_choice == "Los Angeles":  
    tours = get_tours_for("Los Angeles")  
    for each tour get the price and picture and name  
    show each tour name, picture and price
```



I'm repeating the same stuff over and over. The only thing that changes is the city name.

I could just use a function!

```
def show_tours(city_name):
    tours = get_tours_for(city_name)
    for each tour get the price and picture and name
    show each tour name, picture and price

show_tours("New York")
show_tours("Boston")
show_tours("San Francisco")
show_tours("Washington D.C")
```

Sign Up

For Tour Operators >



## Things to Do This Week in **Boston** ▾



Sign Up

For Tour Operators >

peek

## Things to Do This Week in [Miami](#) ▾



Indoor Rock Climbing Classes  
Three Lakes

From  
\$55



Seaplane Tours  
Key Biscayne

From  
\$160



Jet Ski Tours  
Miami Beach

From  
\$70



Local Flavors Food Tours  
South Beach

From  
\$55



Zoo Miami  
Larry and Penny Thompson Memorial Park

From  
\$17



ZWF Wildlife Encounters  
Miami

From  
\$134

# Let's Write Our Own Functions!



# How to write a function

```
def greet(name):  
    print("Hi, " + name)
```

# A template for Python functions

```
def name_of_function(parameters...):  
    body  
  
    return
```

# How to use a function, AKA calling it



```
def greet(name):  
    print("Hi " + name)
```

```
greet("Jane")
```

```
greet("Jill")
```

```
greet("Julie")
```

# Template for calling a function

```
name_of_function(parameters...)
```

# Things to watch out for when calling a function

Order

Parameters

Spelling

# Order

```
greet("Jane")
```

```
def greet(name):  
    print("Hi " + name)
```

```
Traceback (most recent call last):  
  File "<pyshell#20>", line 1, in <module>  
    greet("Jane")  
NameError: name 'greet' is not defined
```

If you call a function before defining it , I don't know what to do.



# Parameters

```
def greet(name):  
    print("Hi " + name)  
  
>>> greet()  
Traceback (most recent call last):  
  File "<pyshell#2>", line 1, in <module>  
    greet()  
TypeError: greet() missing 1 required positional argument: 'name'  
>>> greet("Jane", "Jill")  
Traceback (most recent call last):  
  File "<pyshell#3>", line 1, in <module>  
    greet("Jane", "Jill")  
TypeError: greet() takes 1 positional argument but 2 were given
```

Please call  
functions with the  
right number of  
parameters!



# Spelling

```
def greet(name):  
    print("Hi " + name)
```

```
>>> gree("Julie")  
Traceback (most recent call last):  
  File "<pyshell#4>", line 1, in <module>  
    gree("Julie")  
NameError: name 'gree' is not defined
```

If you misspell something, I won't know what you're talking about.



A photograph of two young women sitting at a desk, looking down at a laptop screen together. The woman on the left has long blonde hair tied back and is wearing a white t-shirt. The woman on the right has dark hair in a bun and is wearing a floral top. They are both smiling and appear to be engaged in a shared activity on the laptop.

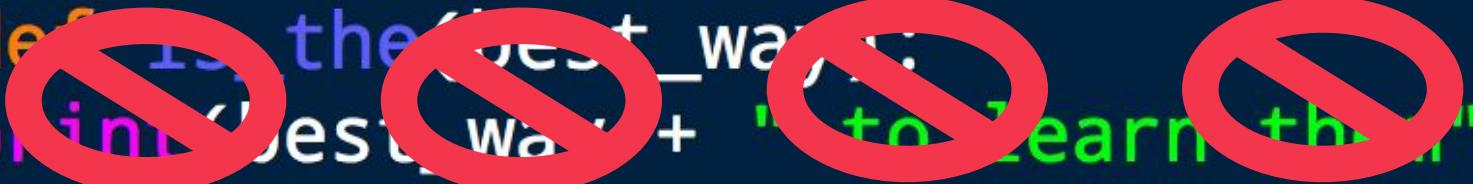
**Let's Review!!**

# Which, if any, of these functions have a mistake in the way they are written?

```
def this_review(will):  
    print(will + " hopefully be helpful")
```

```
def going_over(something):  
    print(something + " more than once")
```

```
def is_the_best_way(best_ways):  
    print(best_ways + " to Learn things")
```



```
def repetition(helps):  
    print(helps + " you 'get' it")
```

# Which, if any, of these functions have a mistake in the way they are written?

```
writing_a_function_can_be():
    print(car_be + "!" )
```

```
def but_you_have(to_know):
    print(to_know + "how to write it")
```

```
def once_you(have_that):
    print(have_that + "figured out")
```

```
def your_apps(will_be):
    print(will_be + " easier to maintain")
```

Which, if any, of these functions have a mistake in the way they are written?

```
def there_is(a_lot):  
    print(a_lot + " to remember")
```

```
def but_once(you_get):  
    print(you_get + " it, its easy")
```

```
def at_least(writing):  
    print(writing + " them is easy")
```

```
def it_takes_some_time_to_practice_to_master():  
    print("It takes some time to practice to master")
```

# Which, if any, of these functions have a mistake in the way they are written?

```
def as_you(write_more):  
    print(write_more + " functions")
```

```
def yourself(you_will,  
            print(you_will + " be able to tell")
```



```
def right_away(when):  
    print(when + " something is missing")
```

```
def even_the(smallest):  
    print(smallest + " detail")
```

# Using a Function: Which of these are proper calls to the function?

```
def combined_age(my_age, your_age):  
    return my_age + your_age
```

```
combined_age(15, 14)
```

```
NameError: name 'combined_ag' is not defined
```

```
combined_age()
```

```
NameError: name 'combined_ag' is not defined
```

```
combined_age(30,55)
```

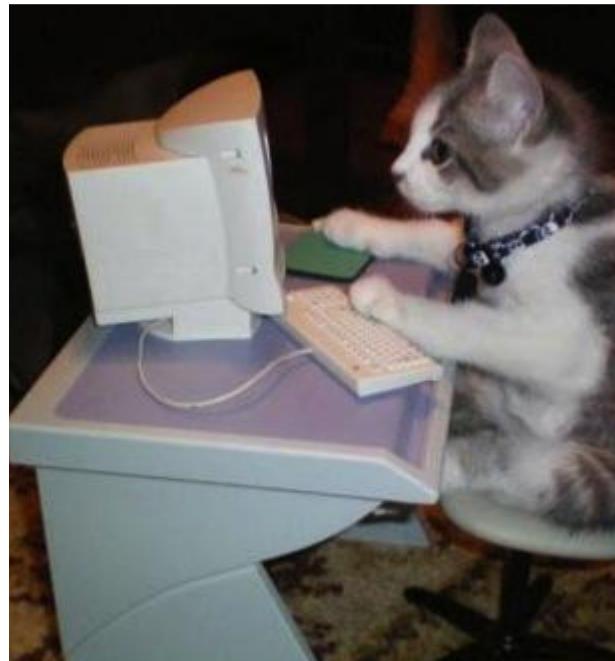
```
combined_age(16, 17)
```

```
NameError: name 'age_difference' is not defined
```

# Let's Go Deeper into Functions!



# Remember what functions are



A little section of code  
inside your program that  
does a specific job.

# Answer a question

```
def combined_age(my_age, your_age):  
    return my_age + your_age
```

combined\_age(15, 14)

What is our combined\_age  
if my\_age is 15 and  
your\_age is 14?

29

It **returns** the answer

Without the return statement, the function will do the math but won't give the answer



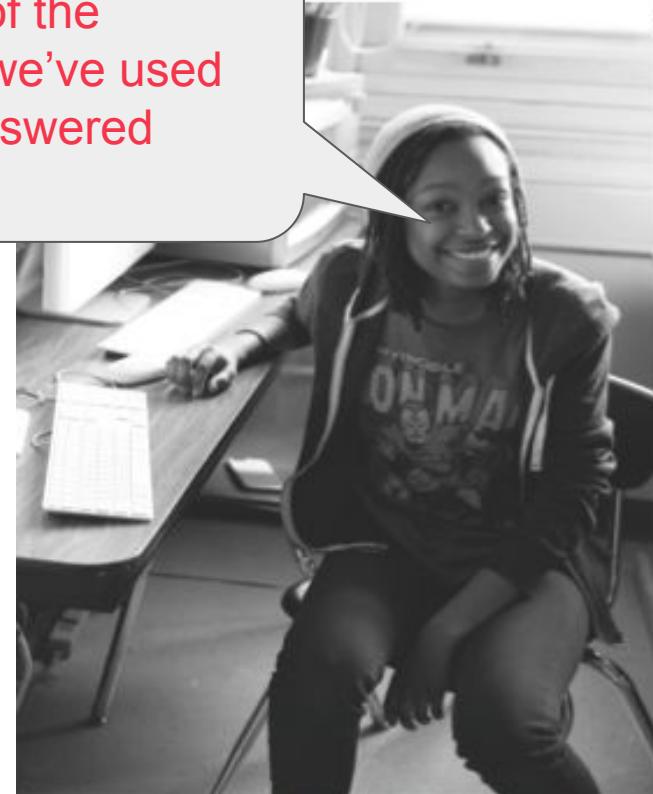
```
def combined_age(my_age, your_age):  
    my_age + your_age
```

If the job you want the function to do is “Answer a question” it must have a **return** statement.

# The Other Job of Functions

But most of the functions we've used haven't answered questions.

Carry out a command, aka do what you tell them to do



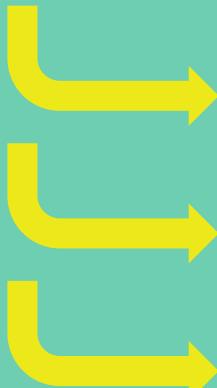
# Do What You Say

- `color("blue")` → Make the color of the turtle blue
- `right(90)` → Turn the turtle right by 90 degrees
- `forward(100)` → Move the turtle forward 100 pixels
- `print("This is a function")` → Print the string “This is a function” to the shell

# Built-in Functions

```
print("This will be printed to the shell")
```

```
input("Write something: ")
```



Write something:

something

```
return 'something'
```



So functions can either answer a question or do what I tell them.

## Things to keep in mind:

Is this function answering a question?

Is it following a command?

Or they can do both. And to answer a question it has to have the **return** statement.

Is it doing both?



## What about parameters?

```
def name_of_function(parameters...):  
    body
```

Something the function needs to know so that  
it can do it's job

# Some functions don't need parameters

`clear()`



Clears the screen

`pendown()`



Puts the turtle's pen down so that it can draw

`penup()`



Picks up the turtle's pen so that it can move without leaving a mark.

# How do you know when to use parameters?

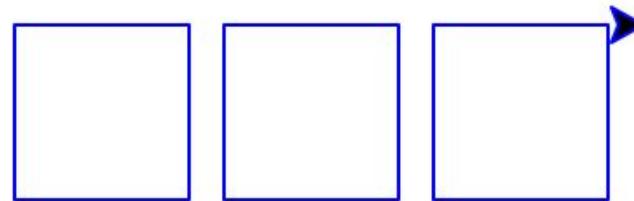
Does it do the same thing every time?

`pendown()`

Is it a little different each time?

`forward(100)`

# Remember this?



```
for i in range(3):
    for i in range(4):
        pencolor("blue")
        forward(50)
        right(90)
    penup()
    forward(60)
    pendown()
```

# ONE LAST THING ABOUT PARAMETERS

**Parameters  
are little  
variables  
inside a  
function**

**They are  
weird for  
3 reasons**

# Parameters only exist inside their functions

```
def combined_age(my_age, your_age):  
    return my_age + your_age
```

```
combined_age(15, 14)
```

```
print(my_age)
```

```
NameError: name 'my_age' is not defined
```

# Parameters override outside variables with the same name.

```
my_age = 100  
  
def combined_age(my_age, your_age):  
    return my_age + your_age  
  
combined_age(15, 14) → 29  
  
print(my_age)
```

**BAD**  
**THING**

You can use outside variables inside a function as if they were parameters. (The names must be different!)

```
my_friends_age = 17
```

```
def combined_age(my_age, your_age,  
    return my_age + your_age + my_friends_age
```

```
combined_age(15, 14)
```



46

A photograph of two young women sitting at a desk, looking down at a laptop screen together. The woman on the left has long blonde hair tied back in a ponytail and is wearing a white t-shirt. The woman on the right has dark hair in a bun and is wearing a floral top. They are both smiling and appear to be engaged in a task on the laptop.

**One more review!**

# Function Jobs

```
def parallel_lines():
    def position():
        x_coordinate = xcor()
        y_coordinate = ycor()
        return (x_coordinate, y_coordinate)
        forward(50)
        right(90)
        pendown()
        forward(200)
```

## Function Jobs 2 (Trickier)

```
def get_full_name(first_name, last_name):  
    return first_name + " " + last_name
```

```
full_name = get_full_name("Grace", "Hopper")
```

```
def print_full_name(first_name, last_name):  
    print(first_name + " " + last_name)
```

```
printed_name = print_full_name("Ruzwana", "Bashir")
```

# Playing with Parameters

```
my_favorite_color = "tomato"  
width = 50  
length = 100
```

```
def rectangle(width, length):  
    pendown()  
    pencolor(my_favorite_color)  
    for i in range(2):  
        forward(width)  
        right(90)  
        forward(length)  
        right(90)
```

```
rectangle(100, 200)
```

# Playing with Parameters 2 (Trickier)

```
my_favorite_color = "green"
width = 10
length = 50
penup()
color(my_favorite_color)
pencolor(my_favorite_color)
for i in range(5):
    forward(width)
    right(90)
    forward(length)
    right(90)
    rectangle(10, 50, "green")
```

# This was a lot!!



Remember to call your functions!