Open in new tab

Ruby Functions Practice

Lesson Duration: 120 minutes

Learning Objectives

- Practice creating functions
- Practice creating objects, assigning variables, calling functions
- Practice testing, write expectation, then write solution

Practicing Functions

Today we have seen a lot of stuff. Classes, objects, variables, methods…

These are the key building blocks of OO programming languages. Don't expect it all to sink in right away, like any craft we need to practice to allow it to settle in out minds.

This exercise is to practice the theory we have learned.

> `send folder with ruby_functions_practice.rb` and `ruby_functions_practice_spec.rb`

Notice that `ruby_functions_practice` is empty. This is where we will need to write our functions. `ruby_functions_practice_spec.rb` is the file we will run.

It is using `require_relative`, it is pulling in the file so all the functions defined will be availble in our file. Simple require is used for outside libraries, gems, add-ons, while `require_relative` is for files in our own directory structure.

In here, we have a bunch of "tests". We are using a little test framework called Minitest. It allows us to test the outcome of running our functions - which is good news, we don't need to manually eyeball everything to see if our code is working.

```
#terminal

ruby  specs/ruby_functions_spec.rb
```

You can see that we have 10 failing tests. Let's pass the first one.

```
#ruby_functions_spec.rb
#Example
def test_return_10()
  return_10_result = return_10()
  assert_equal( 10, return_10_result )
end
```

```
#terminal

ruby  specs/ruby_functions_spec.rb
```

We have an expected failure - perfect. What does the error say. Okay the method is not defined. Let's define it.

> fail, fail again, fail better

```
#ruby_functions_practice.rb
def return_10
end
```

```
#terminal

ruby  specs/ruby_functions_spec.rb
```

Okay we now have the method but looking at our text we can see it is not giving us what we expect. Let's fix that

```
#ruby_functions_practice.rb

def return_10
  return 10
end
```

Now our test should pass.

```
#terminal

ruby  specs/ruby_functions_spec.rb
```

Now that our test passes, we should make a commit to git. Go and make the rest of the tests pass, making a git commit each time.

Note

Not a race, take your time. Try what you think will work, if it doesn't look at error. Try and think about what our Rubybot is doing. Before you call an instructor try and think about how will pose your question in terms of variable, object, classes, functions. Why does this not work? Will not be as valuable question as I created an instance of this class, and called this method on it, I expect this to happen, but instead saw this.

Published with GitHub Pages