

Specification Testing with HSpec

HSpec¹ is a testing library that offers an expressive way to write tests.

Here's a quick example of how an HSpec test might look like, copied from the above link:

```
main :: IO ()
main = hspec $ do
  describe "Prelude.read" $ do
    it "can_parse_integers" $ do
      read "10" `shouldBe` (10 :: Int)

    it "can_parse_floating-point_numbers" $ do
      read "2.5" `shouldBe` (2.5 :: Float)
```

Our start point is the function hspec:

```
hspec :: Spec -> IO ()
```

It takes as input a **specification**, which is in effect a tree of tests with descriptions, and returns an IO () action that runs the tests.

In the example above, the argument to the hspec function is everything that follows the first dollar sign, namely:

```
do
  describe "Prelude.read" $ do
    it "can_parse_integers" $ do
      read "10" `shouldBe` (10 :: Int)

    it "can_parse_floating-point_numbers" $ do
      read "2.5" `shouldBe` (2.5 :: Float)
```

This uses the do notation that we are familiar with from IO, but it is now used to express a value of type Spec.

TODO

¹<https://hspec.github.io/>