

1. The return values of the expressions are as follows:
 - a. `[].class` returns Array
 - b. `3 / 2` returns 1
 - c. `3.0 / 2.0` returns 1.5
 - d. `[].nil?` returns false
 - e. `def h; "Hello World"; end` returns nil
2. My favorite feature of Ruby is that I can use plain English, and it will be translated into code. I have always wanted to do that.
3. Personally, I believe that writing test cases for code that does not exist is always a bad idea, because you may have some idea of how the code will work, but it will turn out that the code cannot function that way.
4. I think it hinders development. If I could have simply written the test cases in coding language, and not had to worry about if one test case returned before another, I could have done it in much less time. Also, the code for the actual application is only 4 lines, but the test cases are at least 10, and I feel like it will almost always be this case, where the testing takes much longer than the actual coding.
5. The advantage of using more complex language is that you can more accurately describe what the test is testing, but the drawback is that you must then define what your language means in terms of the code.
- 6.