1.
```
[].class => Array
3 / 2 => 1
3.0/2.0 => 1.5
[].nil => NoMethodError: undefined method 'nil' for []:Array
def h... => nil
```

2.

I like how arrays can be sliced and negative indexed like in Python.
It makes things very brief.

3.

I can imagine some cases where BDD is of use. If arranging or executing
a test case is overly complicated, then BDD makes things clearer. However,
the setting-up of BDD, overhead in development, and my lack of seeing such
test cases makes it hard to see the benefits.

4.

I see that BDD isn't so hard to write. I can see it being helpful for
getting the top level feature requirements.

5.

The use of English makes the cases able to be interpreted immediately. It
states simply what state the program needs to be in, what happens, and then
what should result. However, misuse of language may lead to some ambiguity.