BDD To The Bone

Using Behave and Selenium to Test-Drive Web Applications


I HAVE NO IDEA
WHAT I'M DOING
quickmeme.com


YES
THIS IS DOG

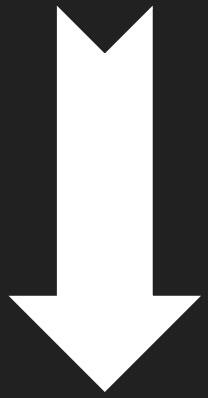
URL Shorteners?


goo.gl
url shortener


bitly


TinyURL

https://docs.google.com/presentation/d/1iDdRQQSx7QGzVaOWlG1YizvqFQM1J5VhQENGuR13iHs/



https://goo.gl/MyruRa

Requirements #1 - #3

When a user enters in a URL, they are provided a shortened URL.

When a user navigates to a shortened URL, they are redirected to the original URL. This should be as fast as possible.

There is a way to see how many people have been redirected through this URL.

Requirement 4

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Requirement 5

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

~~~~~~~~~~

.

.

.

.

.

.

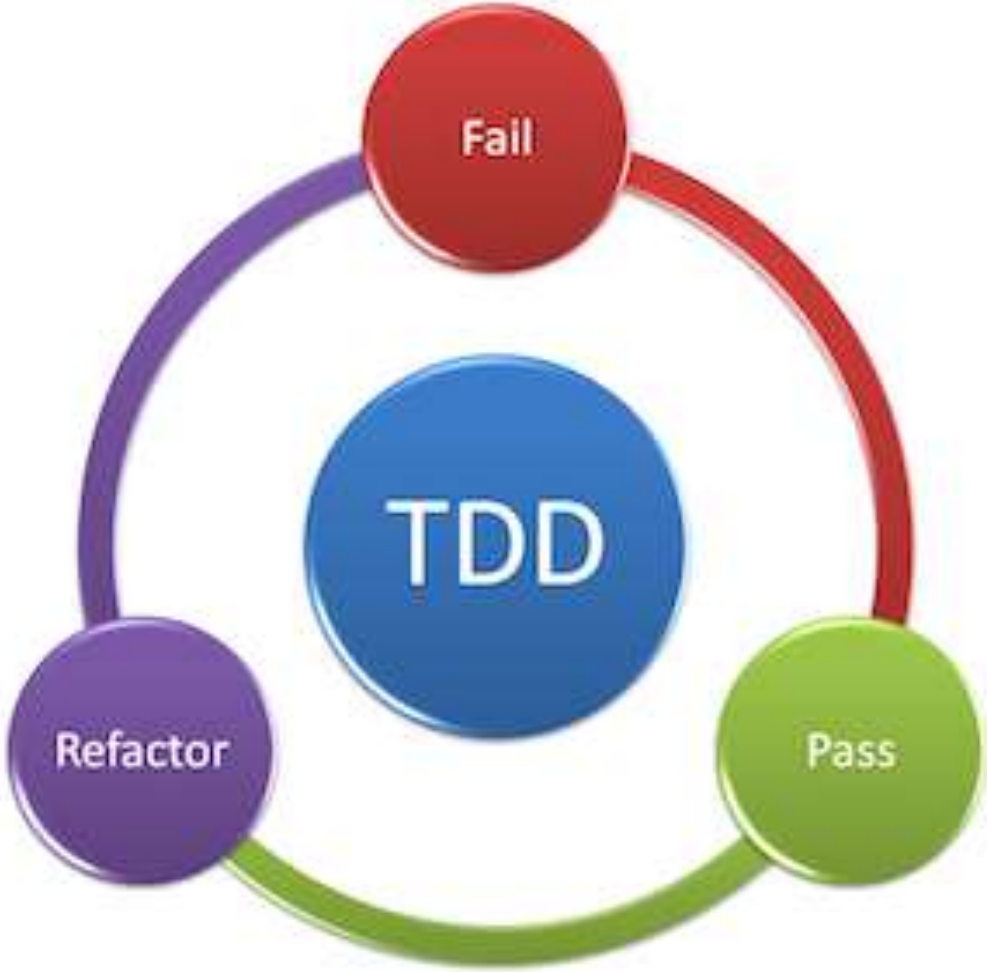
Requirement 571

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

How do you want to do this?


Fail
TDD
Refactor
Pass

That's great!

But...

Requirements #1 - #3

When a user enters in a URL, they are provided a shortened URL.

When a user navigates to a shortened URL, they are redirected to the original URL. This should be as fast as possible.

There is a way to see how many people have been redirected through this URL.


AM I OUT OF TOUCH WITH THE CUSTOMER?
NO, IT'S TDD THAT'S WRONG
memegenerator.net


FALSE
TDD IS ACTUALLY PRETTY GREAT
imgflip.com

Tests are meant to answer a question (they can't prove there are no bugs)

Do I have confidence that I can ship my code?

Does my code do what I want it to do?

How does my code work with 10,000 users?

Can my code run for weeks on end?

Does my code do what the customer wants?

Do I have confidence that I can ship my code?

Does my code do what I want it to do?

How does my code work with 10,000 users?

Can my code run for weeks on end?

Does my code do what the customer wants?

UNIT TESTS

Do I have confidence that I can ship my code?

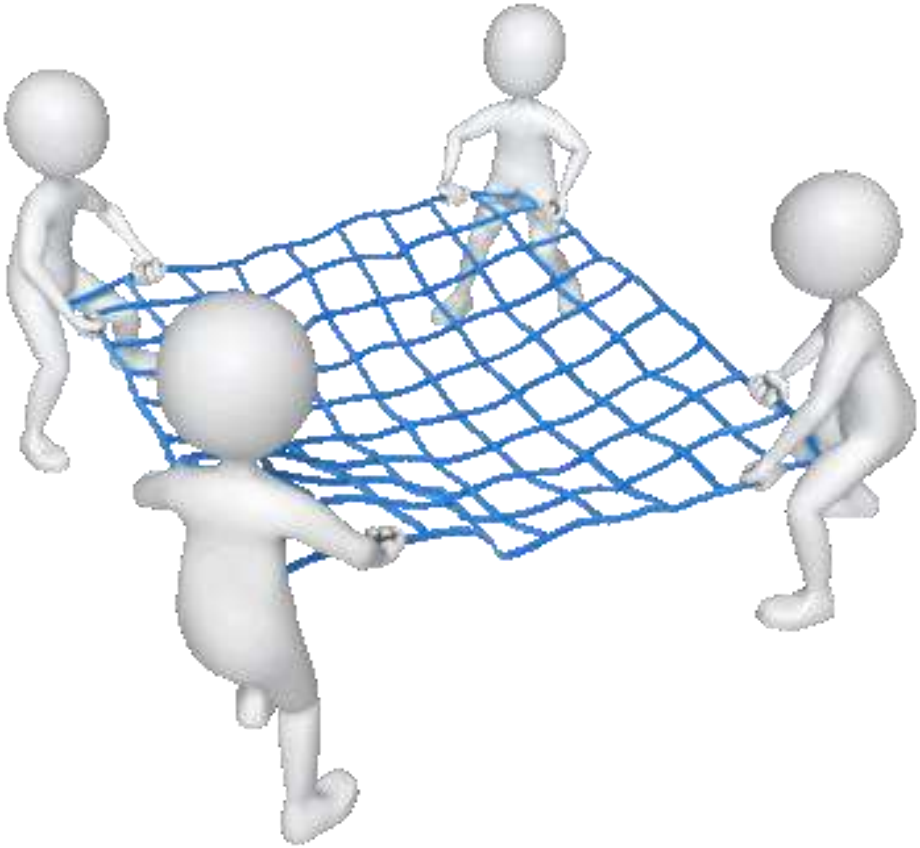
Does my code do what I want it to do?

How does my code work with 10,000 users?

Can my code run for weeks on end?

Does my code do what the customer wants?

Acceptance Testing



Problems?

I still have to figure out exactly what the customer wants

The customer is going to change their mind

I have to trace back
to requirements

Executable Specifications

I want to have tests written as close to plain English requirements as possible

I want to run these tests often to make sure I always am giving the customer what they want

Gherkin

Gherkin Example

Feature: Our service makes short URLs out of long URLs

All URLs will start with http://patl.ly:8080/ and end in a number representing the lookup index

Scenario: Shortening a URL

Given a url http://www.python.org

When we shorten it through our service

Then we should receive a shortened URL

Gherkin Example

Feature: Our service makes short URLs out of long URLs

All URLs will start with http://patl.ly:8080/ and end in a number representing the lookup index

Scenario: Shortening a URL

Given a url http://www.python.org

When we shorten it through our service

Then we should receive a shortened URL

Gherkin Example

Feature: Our service makes short URLs out of long URLs

All URLs will start with http://patl.ly:8080/ and end in a number representing the lookup index

Scenario: Shortening a URL

Given a url http://www.python.org

When we shorten it through our service

Then we should receive a shortened URL

Gherkin Example

Feature: Our service makes short URLs out of long URLs

All URLs will start with http://patl.ly:8080/ and end in a number representing the lookup index

Scenario: Shortening a URL

Given a url http://www.python.org

When we shorten it through our service

Then we should receive a shortened URL

Gherkin Example

```
Feature: Our service makes short URLs out of long URLs

        All URLs will start with http://patl.ly:8080/ and end in a number

        representing the lookup index

  Scenario: Shortening a URL

    Given a url http://www.python.org

    When we shorten it through our service

    Then we should receive a shortened URL
```

Gherkin Example

Feature: Our service makes short URLs out of long URLs

All URLs will start with http://patl.ly:8080/ and end in a number representing the lookup index

Scenario: Shortening a URL

Given a url http://www.python.org

When we shorten it through our service

Then we should receive a shortened URL

Let's look at some features

So far, this is just looking like a test case

Behave

Behave is a Python library used to hook up Python code to Gherkin statements

Gherkin Features become executable specifications

Drive the requirements conversation up front

Let's write some steps

So how do we implement these steps


PAT.LY
Enter a URL, and we'll give you a new one that's much shorter!
Enter URL:
Get me a short link!
URL
Redirects
http://google.com
3
Image Credits

HTTP!

urllib

requests

These are HTTP libraries, not JS libraries

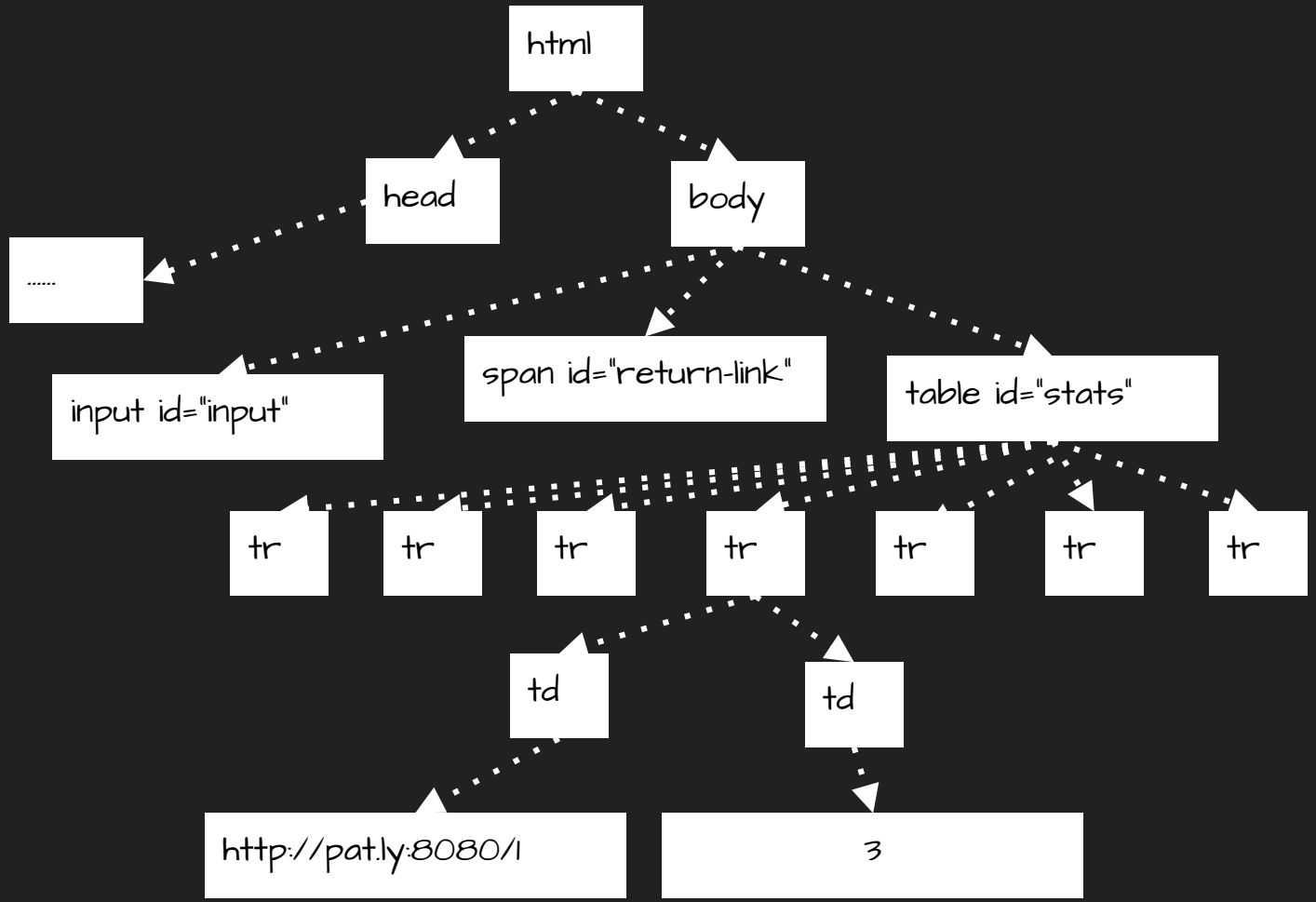
Even if they did do Javascript, do they do it the same way as a browser?

How do you test clicks, and mouse hovers, and all other front-end things?

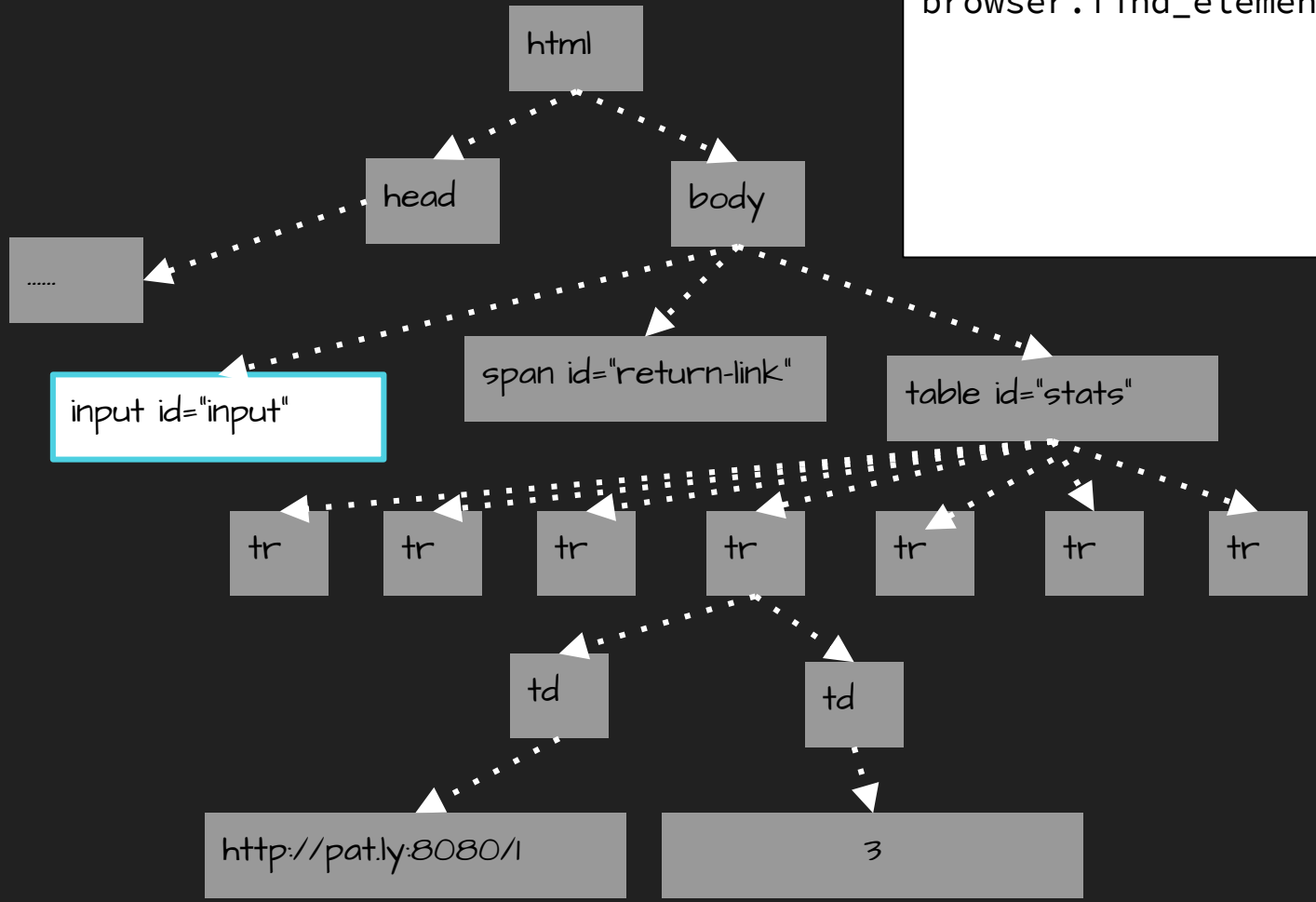
It's your responsibility to deliver a solution, not just a piece of code

selenium

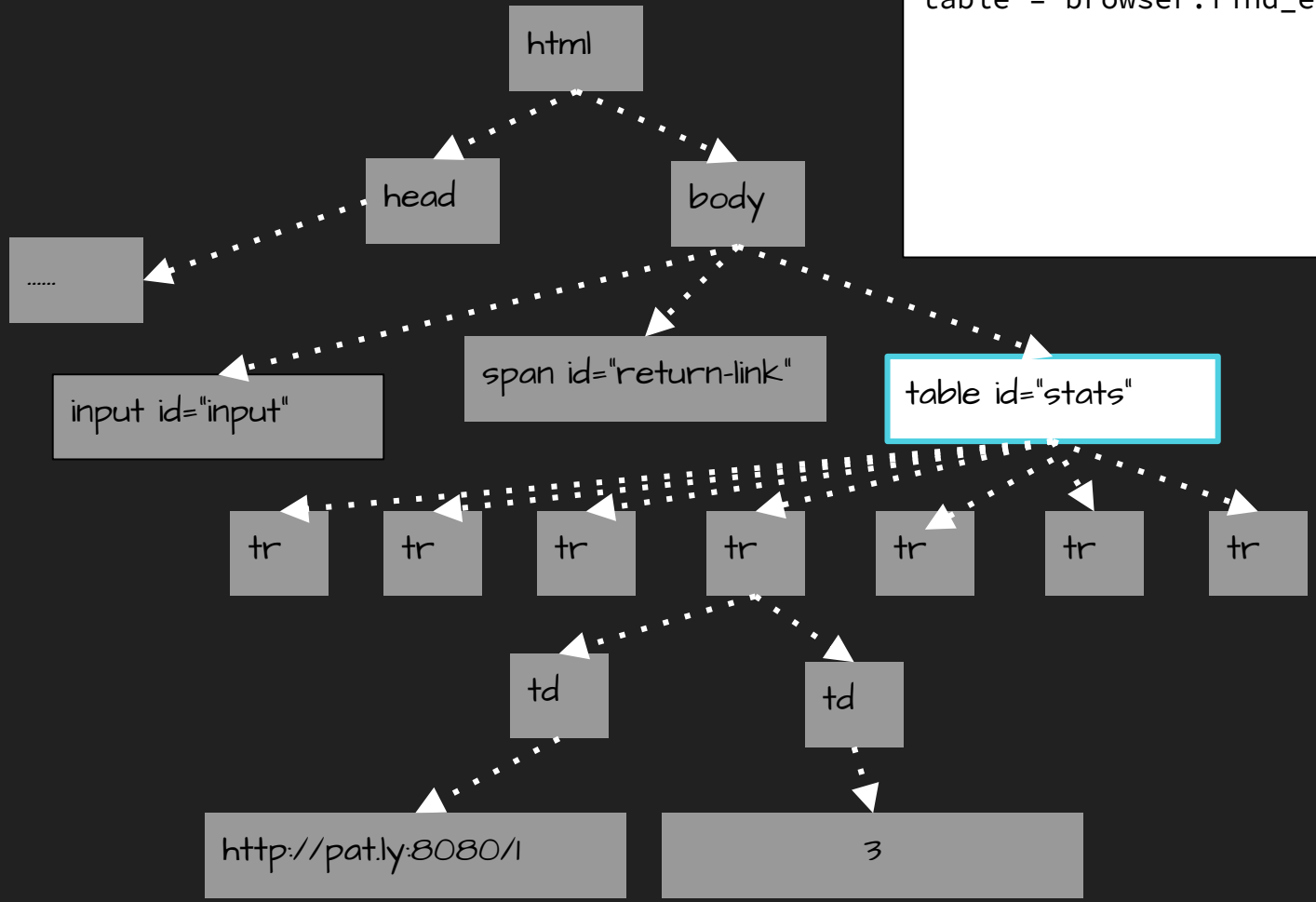
Selenium lets you control web browsers through Python


html
head
body
.....
input id="input"
span id="return-link"
table id="stats"
tr
tr
tr
tr
tr
tr
tr
td
td
http://pat.ly:8080/1
3

```
browser.find_element_by_id(“input”)
```



```
table = browser.find_element_by_id(“stats”)
```



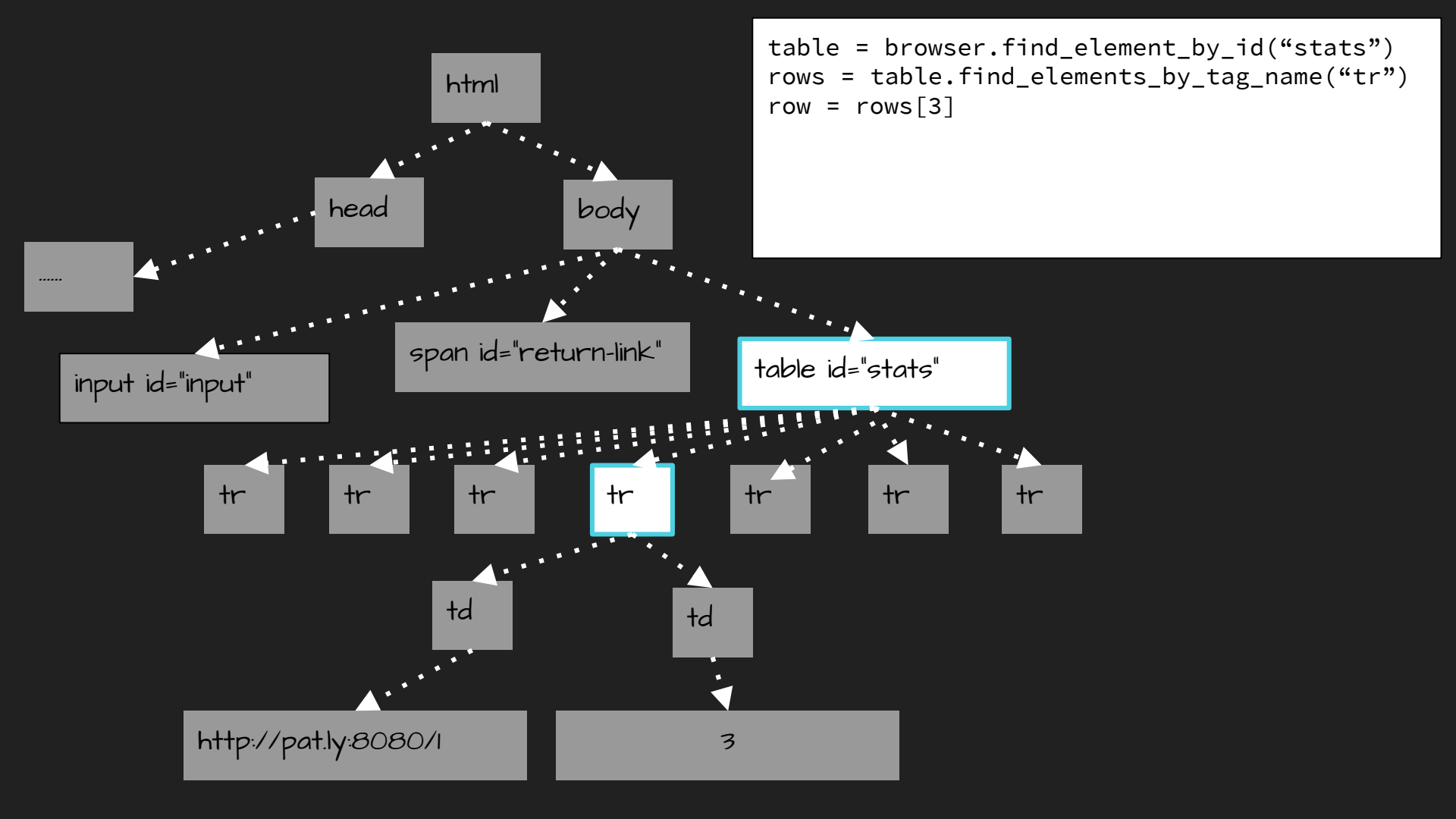
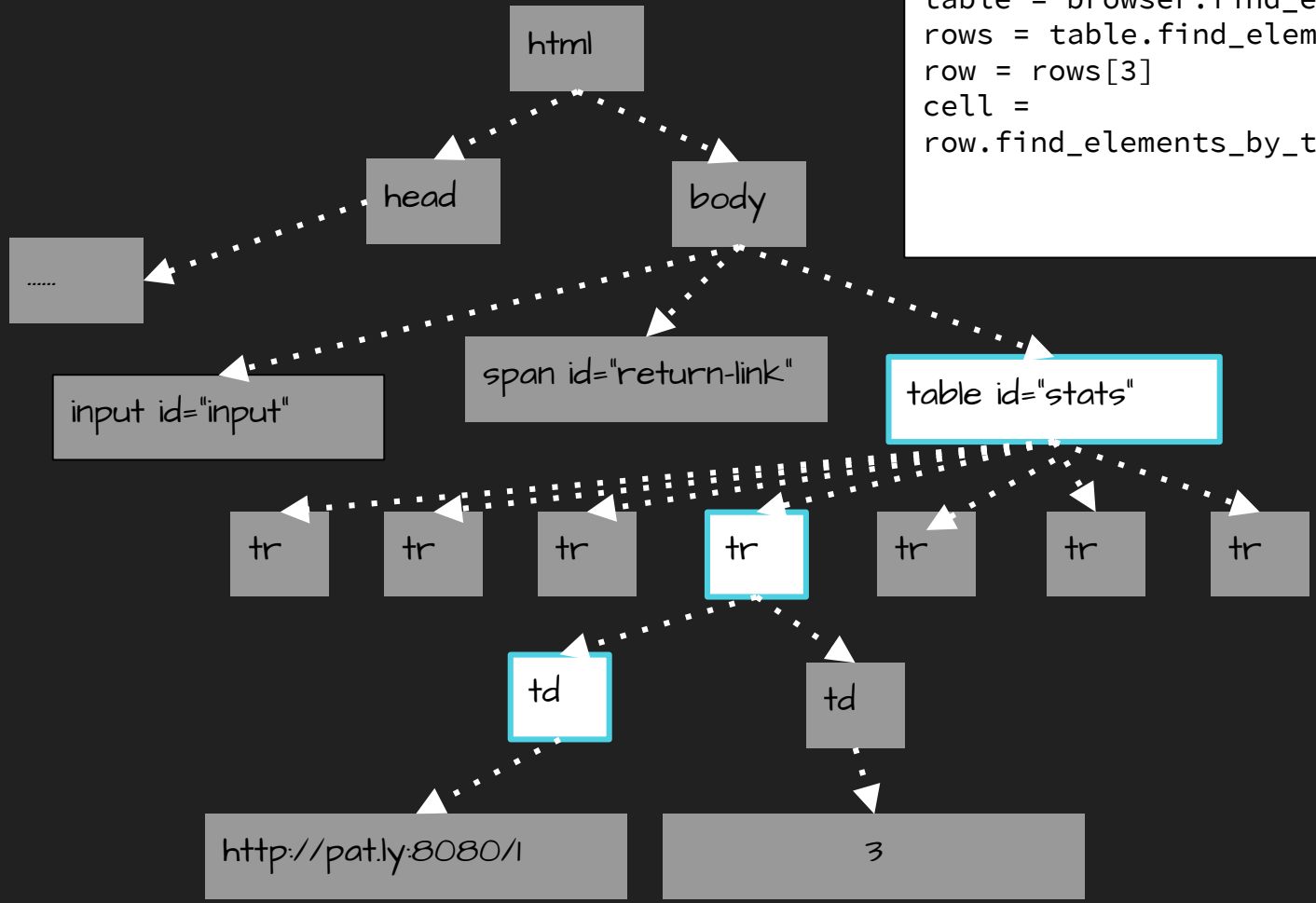
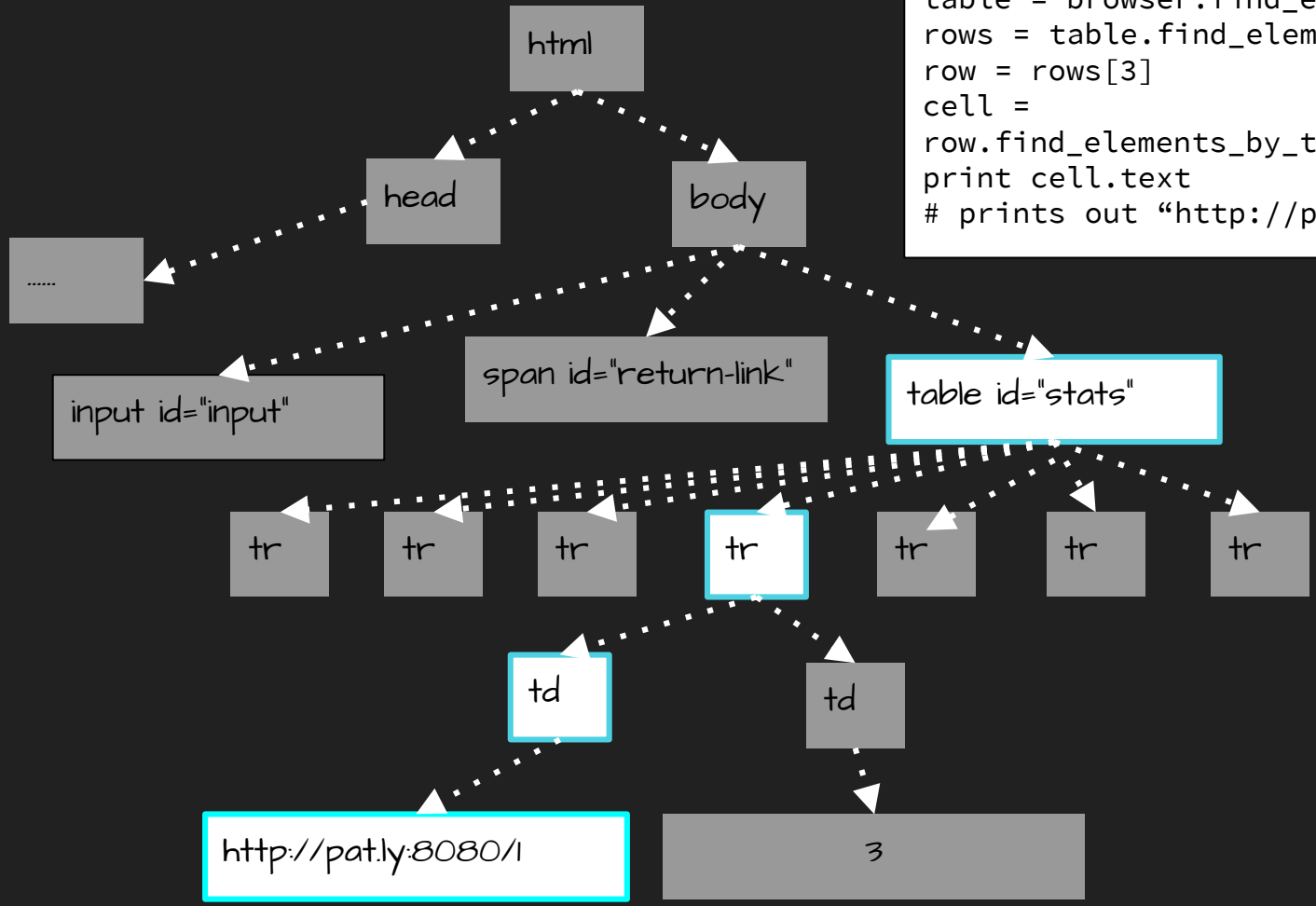


table = browser.find_element_by_id(“stats”)
rows = table.find_elements_by_tag_name(“tr”)
row = rows[3]
html
head
body
.....
input id="input"
span id="return-link"
table id="stats"
tr
tr
tr
tr
tr
tr
tr
td
td
http://pat.ly:8080/1
3




```
table = browser.find_element_by_id(“stats”)
rows = table.find_elements_by_tag_name(“tr”)
row = rows[3]
cell =
row.find_elements_by_tag_name(“td”)[0]
```




```
table = browser.find_element_by_id(“stats”)
rows = table.find_elements_by_tag_name(“tr”)
row = rows[3]
cell =
row.find_elements_by_tag_name(“td”)[0]
print cell.text
# prints out “http://pat.ly:8080/1”
```

Example time!

Find elements by:

Name

ID

Class Name

Tag Name

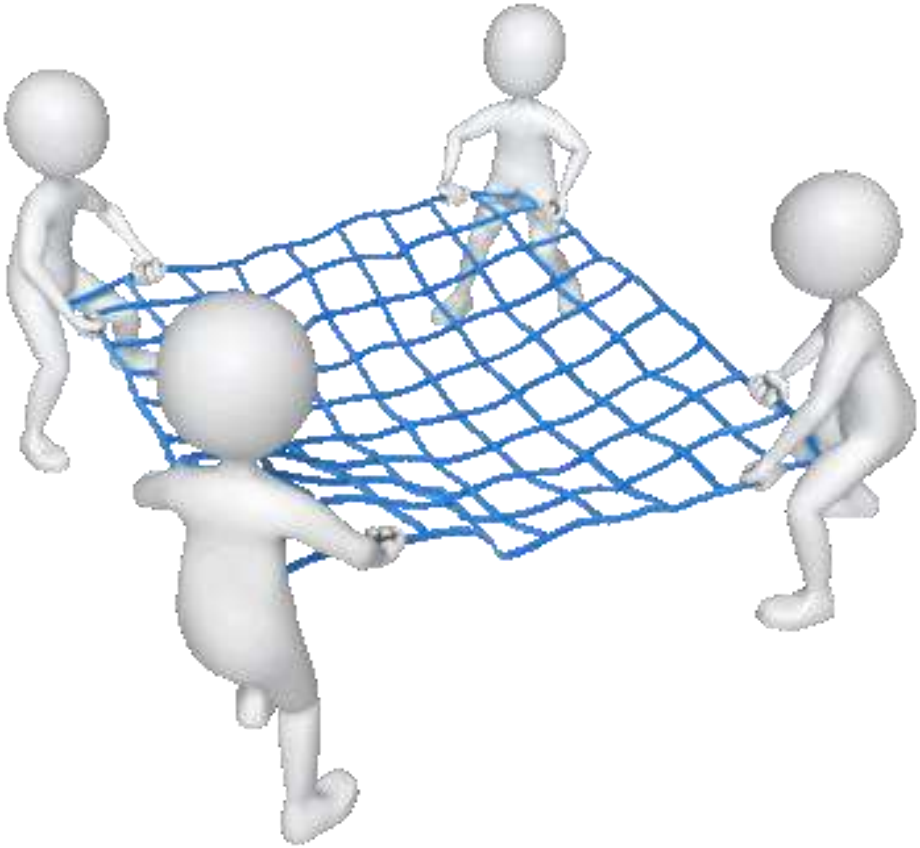
Link Text

Partial Link Text

Interact with the page:

- Clicking
- Sending input
- Modify cookies
- Move the mouse
- Screenshotting
- Executing Arbitrary Javascript

Let's look at the tests again



Other Behave features

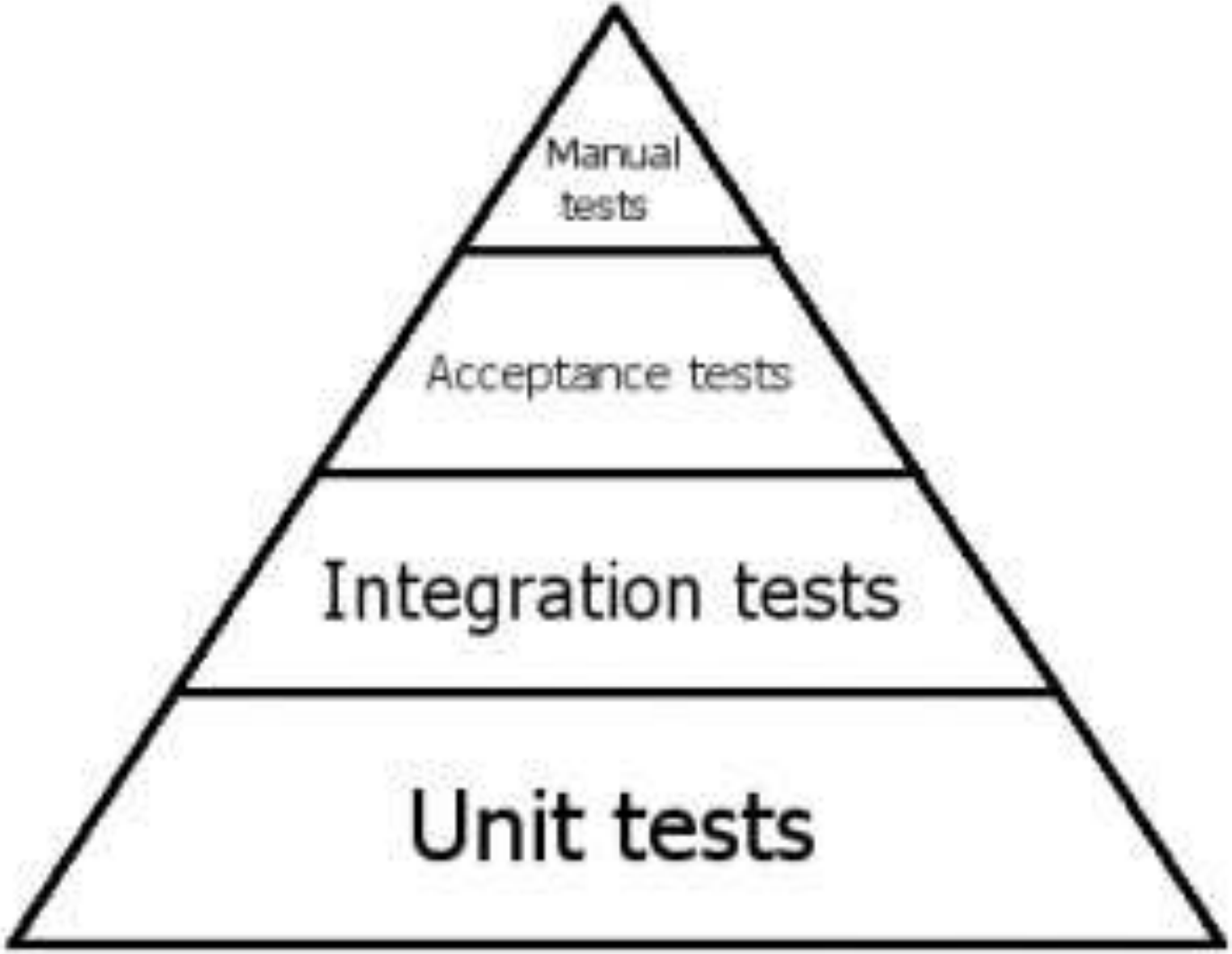
Scenario outlines

Customizable environment file

Regex matching


GIVES PYTHON TALK
ON COOL LIBRARIES
DOESN'T TALK ABOUT
WHEN THINGS GO WRONG
imgflip.com

Selenium (rather the web browser) is slow


Manual tests
Acceptance tests
Integration tests
Unit tests

Acceptance Tests are unwieldy if you have a lot

If nobody reads them, you lose some value

The UI is typically one of the more fragile pieces

But...

There is still value to be had

Driving the conversations with customers is critical

So....

Don't let your requirements go out of date

Keep talking with your customers

Build your safety net of acceptance tests

BDD +
Gherkin +
Behave +
Selenium +
Python =.....

Giving your customers what they really want

Project at

https://github.com/pviafore/BddToTheBone

Follow me:

@PatViaforever