Unit Testing w/Data Structures

Playing with the cool kids

Cool Kids?

The CS majors

When you graduate from Prime, you're likely to encounter coworkers that have backgrounds in Computer Science. Those people will have had an entire course about data structures and having some idea of what they are talking about will be super useful.

What is a data structure?

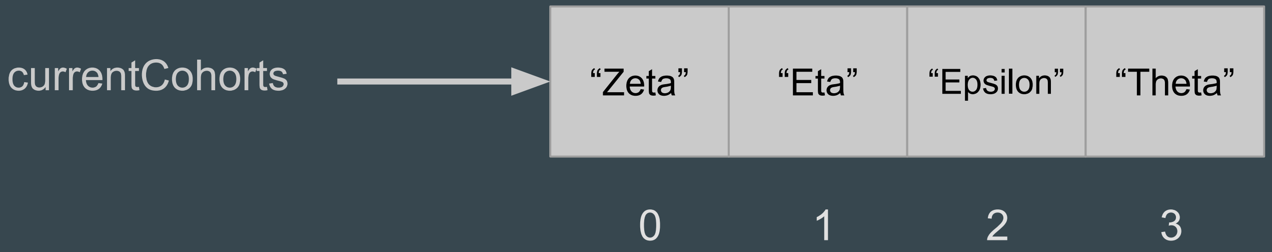
- A design for efficiently storing and accessing data
- An array is an example of a data structure you already know

Data Structure Intro

Data Structure: Array

Stores a sequence of values of the same data type (usually, not in JavaScript)

```
var currentCohorts = [“Zeta”, “Eta”, “Epsilon”, “Theta”];
```



currentCohorts

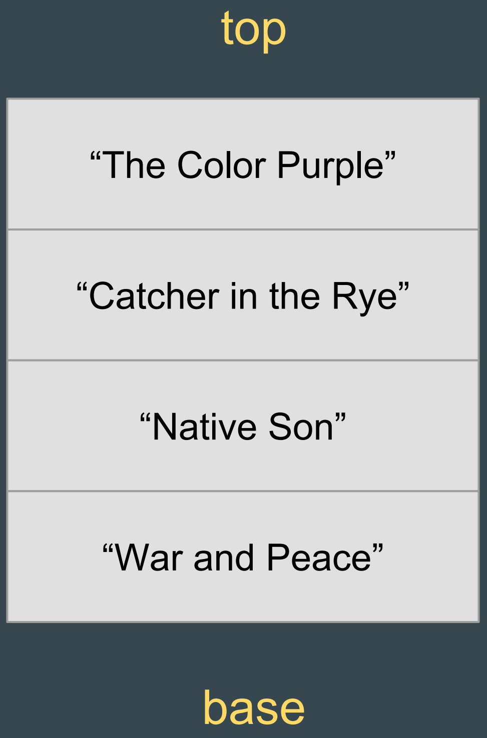
“Zeta”	“Eta”	“Epsilon”	“Theta”
0	1	2	3

Array Capabilities

- Add element (`push, unshift`)
- Remove element (`pop, shift`)
- Any many more...

Data Structure: Stack

- Ordered collection of items
- LIFO (Last In First Out) behavior



top

"The Color Purple"

"Catcher in the Rye"

"Native Son"

"War and Peace"

base

Stack Capabilities

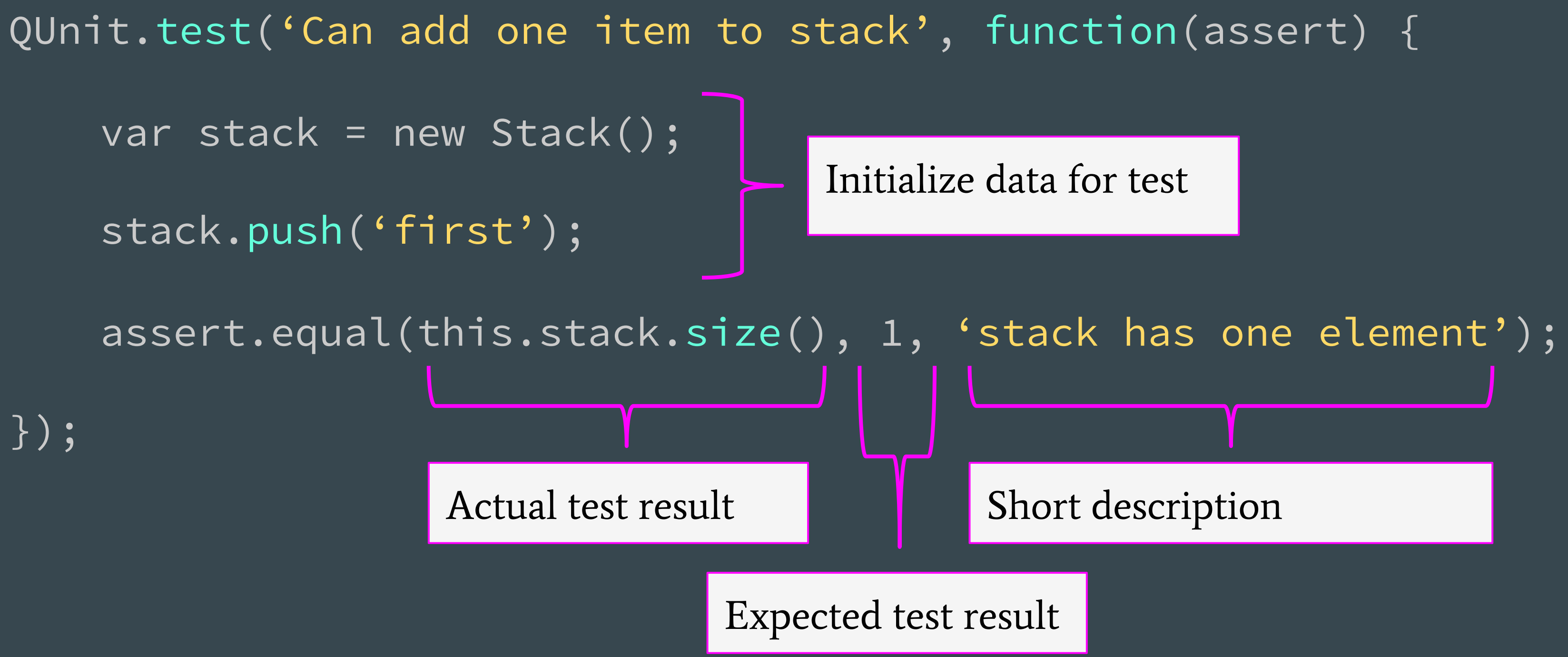
method	**what it does**
push(element(s))	Add one or more new items to top of stack.
pop()	Removes top item from stack. Returns removed element.
peek()	Returns top element of stack without removing it.
isEmpty()	Return true if stack has no elements. False if has 1+ elements.
clear()	Removes all elements of stack.
size()	Returns how many elements are in stack.

Unit Testing

What is a unit test?

- It’s a method for testing small parts of your code
- Those small parts are often distinct functions
    - Thus, it also encourages modularity (which is good)

QUnit Test Example



```
QUnit.test(‘Can add one item to stack’, function(assert) {

    var stack = new Stack();

    stack.push(‘first’);

    assert.equal(this.stack.size(), 1, ‘stack has one element’);

});
```

Initialize data for test

Actual test result

Short description

Expected test result

Assignment Prep

Data Structure: Queue

- Similar to Stack, but uses FIFO
  - First In First Out (FIFO)

Queue Capabilities

method	**what it does**
enqueue(element(s))	Add 1+ elements to back of queue
dequeue()	Removes first element from queue. Returns element.
front()	Returns first element without removing it.
isEmpty()	Return true if queue has no elements. False if has 1+ elements.
size()	Returns how many elements are in queue.

Resources

QUnit

- Website
- API documentation
- Cookbook - has examples

Other

- jsDoc

Some Other Data Structures

- Linked Lists
- Sets
- Trees
- Graphs
- Dictionaries

Sources

“Learning JavaScript Data Structures and Algorithms” by Loiane Groner [book]