

EXERCISE: TESTING SUCCESS

- ▶ Let's test when `parse()` does *not* throw
- ▶ We'll give it *valid input* and *check the return value*
- ▶ `assert.doesNotThrow()` is a thing. *Do not be tempted!*
 - ▶ A successful assertion made on a return value implies the function does not throw!
- ▶ **Add an assertion passing `["- -foo"]` to `parse()` with no second `opts` parameter**
 - ▶ Hint: Use `assert.deepStrictEqual()`
 - ▶ Hint: Refer to [table of expected behavior](#)

ANSWER: TESTING SUCCESS

```
test/bargs.spec.js
```

```
// continued
```

```
assert.deepStrictEqual(  
  parse(['--foo']),  
  {_: [], foo: true}  
);
```

UNDERSTANDING BARGS VIA EXAMPLE

Input Array	Output JS Object	Notes
<code>["--foo"]</code>	<code>{foo: true}</code>	
<code>["--foo", "--bar"]</code>	<code>{foo: true, bar: true}</code>	
<code>["--foo", "-b"]</code>	<code>{foo: true, b: true}</code>	
<code>["---foo", "-bar"]</code>	<code>{foo: true, bar: true}</code>	
<code>["--foo=baz"]</code>	<code>{foo: "baz"}</code>	
<code>["--foo", "baz"]</code>	<code>{foo: "baz"}</code>	<i>needs options object</i>
<code>["baz"]</code>	<code>{_: "baz"}</code>	
<code>["--foo", "baz"]</code>	<code>{_: "baz", foo: true}</code>	
<code>["--", "--foo"]</code>	<code>{_: "--foo"}</code>	
<code>["--foo", "a", "--foo", "b"]</code>	<code>{foo: ["a", "b"]}</code>	<i>needs options object</i>
<code>["--foo", "---foo"]</code>	<code>{foo: true}</code>	