

Jasmine - 3

Thursday, 13 May 2021

22:20

Matchers

It is just a function

Implements a boolean comparison between the actual value and the expected value

```
expect(/* actual value */).toBe(/* expected value */);
```

-> boolean comparison -> true / false

-> actual value with the expected value (calculator.total === 5) -> true / false

- A matcher is responsible for reporting to jasmine if the expectation is true or false -> passing or failing spec

toBe Matcher (type check -> ===)

```
describe('calculator.js', function() {  
  it('should initialize the total', function() {  
    const calculator = new Calculator();  
    expect(calculator.total).toBe(0); // pass  
    expect(calculator.total).toBe("0"); // fail  
  });  
});
```

toEqual Matcher (=== -> strict comparison)

Deep equality comparison:

- Equal keys and Equal values

```
{ someKey: 'some value' }
```

```
{ someKey: 'some value' }
```

=> true

```
{ someKey: 'some value' }
```

```
{ someKey2: 'some value' }
```

```
{ someKey: some value }
```

or

```
{ someKey: 'some value3' }
```

```
{ someKey: 'some value' }
```

=> false

toBeTruthy(), toBeFalsy() Matchers

Truthy: A truthy value is a value that translates to true when evaluated in a boolean context

Falsy: A value that translates to false when evaluated in a boolean context

```
if (condition /* a boolean context */) {}
```

toBeDefined, toBeUndefined

```
** undefined **
```

Global property

Primitive -> Data that is not an object and has no methods

A value that is automatically assigned to variables that have been just declared

```
// asymmetric matchers!
```

```
// Not equal in each side!
```