Repetition

Testning

Mocha

Mocha är ett testramverk för Node.Js


```
npm install --save-dev mocha
```

Den förser oss med de vanliga delarna som behövs för att testa koden.

Mocha

```
var assert = require('assert');
describe('Array', function () {
  describe('#indexOf()', function () {
    it('should return -1 when the value is not present', function () {
      assert.equal([1, 2, 3].indexOf(4), -1);
    });
  });
});
```

Describe = beskriver test case, så att vi lättare ska förstå vad som testas

It = talar om vad vi förväntar oss av testet och sedan kör den själva testkoden

Assert = Antagande, detta är den viktigaste delen av test frameworks

https://mochajs.org/#installation

Chai

Chai är en Assert library som hjälper oss att definiera hur tester ska köras. Koden blir snyggare och lättare att förstå när vi använder Chai.

npm install chai

Den har en samling med specifika Asserts man kan använda i sitt testande.

https://www.chaijs.com/

Should

Chai should låter oss skapa frågor till objekter eller variabeln vi hanterar.

```
chai.should();

name.should.be.a('string');
name.should.equal('Pelle');
name.should.have.lengthOf(5);
person.should.have.property('name');
```

https://www.chaijs.com/

Should alternativ

Istället för

```
var assert = require('assert');
var name='Pelle'

describe('Name', function () {
  describe('length', function () {
    it('should return 5 when the value is Pelle', function () {
      assert.equal(name.length, 5);
    });
  });
});
```

kan vi skriva

```
name.should.have.lengthOf(4);
```

Assert, Expect, Should

Vad är skillnaden?

En assert är som den klassiska assert från Nunit (bland annat), den tar emot parametrar som ska kontrolleras och en parameter som talar om vad man förväntar sig i textform, så att testresultater blir läsbart.

```
var doctors2005={actors:{'Eccleston','Tennant','Smith','Capaldi','Whittaker'}}

assert.LengthOf(doctors2005.actors,5,'Five actors so far');
```

Assert, **Expect**, Should

Expect använder sig av kopplade funktioner för att göra det enklare för kodaren att förstå testet

```
var doctors2005={actors:{'Eccleston','Tennant','Smith','Capaldi','Whittaker'}}

expect(doctors2005).to.have.property('actors').with.lengthOf(5);
```

Assert, Expect, **Should**

Should fungerar som Expect men man börjar inte med testobjektet först. Istället skriver man förutsättningar först och sedan lägger man in objektet.

```
var doctors2005={actors:{'Eccleston','Tennant','Smith','Capaldi','Whittaker'}}

doctors2005.should.have.property('actors').with.lengthOf(5);
```

Skillnaden?

```
var doctors2005={actors:{'Eccleston','Tennant','Smith','Capaldi','Whittaker'}}

assert.LengthOf(doctors2005.actors,5,'Five actors so far')

expect(doctors2005).to.have.property('actors').with.lengthOf(5)

doctors2005.should.have.property('actors').with.lengthOf(5)
```

Vad är bäst?

- Ingen och alla…
- Använd den du trivs bäst med,
  eller den som används av andra i projektet
  eller den man kommit överens om i gruppen att använda
- Håll dig till samma sort i samma projekt

Hur kör man testerna

- I package config lägger du till detta

```
"scripts": {
    "test": "mocha ./test",
    "start": "node index.js"
  },
```

```
npm run test
```

Mer läsning

- https://mochajs.org/
- https://www.chaijs.com/