

Code Challenge – Front End

Version 1.0.1

March 16, 2016

Overview:

The objective of this challenge to assess a candidate's proficiency in Front End technologies and development process.

- a. Please complete the challenge as best of your experience and skills.
- b. Follow the instructions. Solution should meet all challenge requirements.
- c. Feel free to elaborate in your own way and get creative.
- d. Code with keeping readability, performance, testability, and best practices in the mind.
- e. Bonus points for thoughtful consideration of the edge cases and self- identifying your usage of design patterns and progressive Web standards.
- f. Follow **SOLID** [https://en.wikipedia.org/wiki/SOLID_\(object-oriented_design\)](https://en.wikipedia.org/wiki/SOLID_(object-oriented_design)) principle of object oriented programming and design.

Submission:

- a. Create a new repo **egen-fe-challenge** on your github.com account and commit all your code to it. Make sure all IDE and OS related files are in **.gitignore**.
- b. Once completed, send the link of the repo to the recruiter.

Calculator App

1. Write a **Calculator** definition that matches this test spec:

```
describe( "Calculator", function() {  
  var calculator;  
  beforeEach( function(){  
    calculator = new Calculator();  
  });  
  
  it("adds 1 and 2", function(){  
    expect( calculator.add( 1, 2 ) ).to.equal( 3 );  
  });  
  
  it("subtracts 2 from 9", function(){  
    expect( calculator.subtract( 9, 2 ) ).to.equal( 7 );  
  });  
  
  it("multiplies 4 and 3", function(){  
    expect( calculator.multiply( 4, 3 ) ).to.equal( 12 );  
  });  
  
  it("divides 10 by 2", function(){  
    expect( calculator.divide( 10, 2 ) ).to.equal( 5 );  
  });  
  
  it("does not divide by 0", function(){  
    expect( calculator.divide( 5, 0 ) ).to.equal( NaN );  
  });  
});
```

2. Write a **ScientificCalculator** definition that matches this test spec:

```
describe( "ScientificCalculator", function(){
  var calculator;
  beforeEach( function(){
    calculator = new ScientificCalculator();
  });

  it("extends Calculator", function(){
    expect( calculator ).to.be.instanceOf( Calculator );
    expect( calculator ).to.be.instanceOf( ScientificCalculator );
  });

  it("returns the sine of PI / 2", function(){
    expect( calculator.sin( Math.PI / 2 ) ).to.equal( 1 );
  });

  it("returns the cosine of PI", function(){
    expect( calculator.cos( Math.PI ) ).to.equal( -1 );
  });

  it("returns the tangent of 0", function(){
    expect( calculator.tan( 0 ) ).to.equal( 0 );
  });

  it("returns the logarithm of 1", function(){
    expect( calculator.log( 1 ) ).to.equal( 0 );
  });
});
```