JAVASCRIPT LANGUAGE & TOOLS

THE MAGIC OF DYNAMIC WEB PAGES

HUY TRONG NGUYEN

KMS TECHNOLOGY

2015 AUG 18

```
<div ng-app="project">
  <h2>JavaScript Projects</h2>
  <div ng-view></div>
```

```
</style> <!--
Ugly Hack due to
jsFiddle issue:
http://goo.gl/BU
```

```
angular.module('SharedServices', [])
    .config(function ($httpProvider) {

$httpProvider.responseInterceptors.push('myHttpInterceptor');
        var spinnerFunction = function (data, headersGetter) {
            // todo start the spinner here

            alert('start spinner');
            return data;
        };

$httpProvider.defaults.transformRequest.push(spinnerFunction);
    })
```

JavaScript Projects

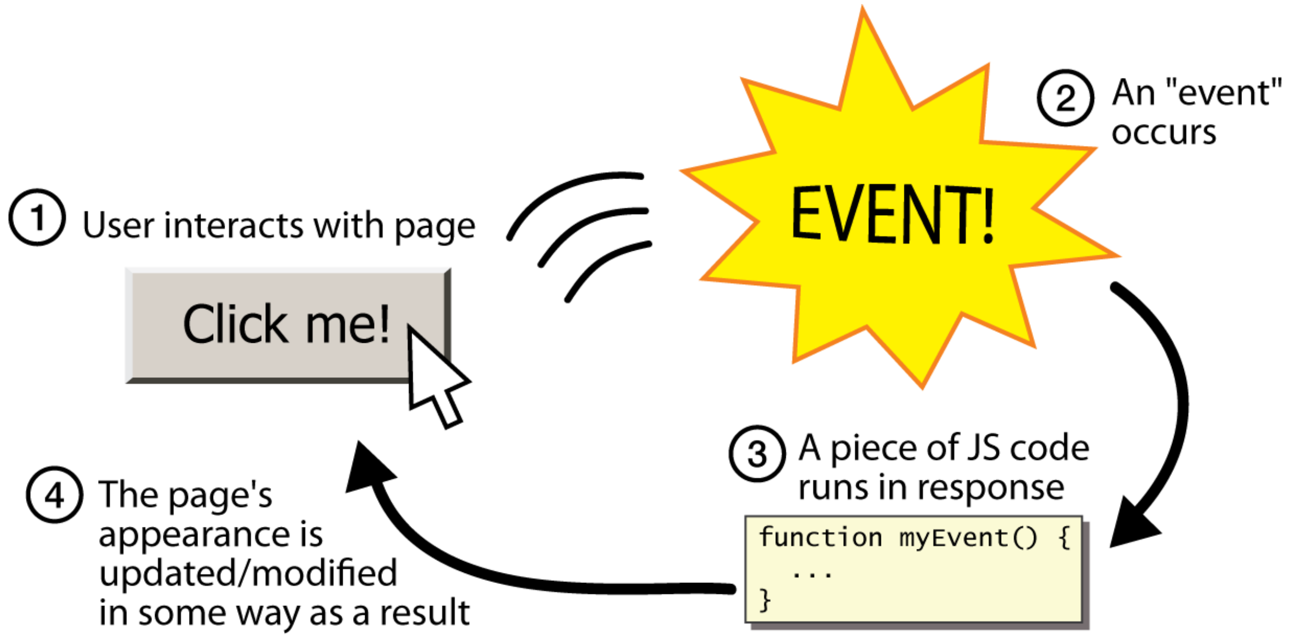
Search

Project	Description
';fa	';l'fa~'
123	123
1234	123
534	
asdasd	qewqweqweqwe
ggg	grgrtgt44
jQuery	javascrip library

WHAT'S COMING NEXT?

- Introduction to JavaScript development
- Functions & Function Expressions
- Nested Functions & Closures
- Tools for JavaScript development
- Assignment

EVENT-DRIVEN PROGRAMMING



1 User interacts with page
Click me!
2 An "event" occurs
EVENT!
3 A piece of JS code runs in response
4 The page's appearance is updated/modified in some way as a result

```
function myEvent() {
   ...
}
```

- JavaScript is a front-end scripting language
- Client-side, mobile and server-side technology
- Simple and flexible
- Powerful to manipulate the DOM

KMS JavaScript Coding Guidance & Best Practices

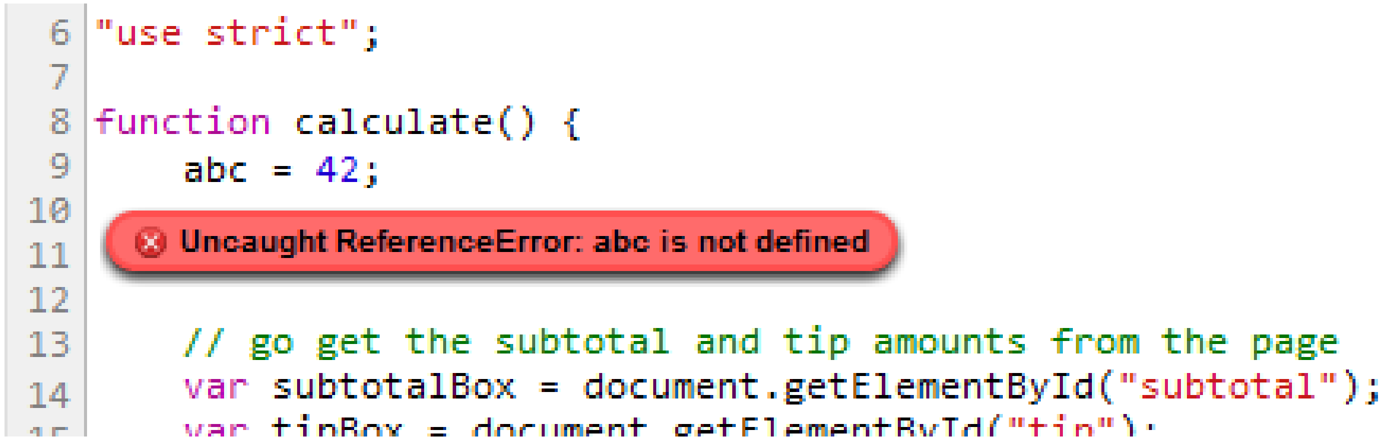
https://github.com/kms-technology/javascript

FORMATTING BLOCKS IN JAVASCRIPT

- Put { at the end of the block and } alone on a line under the corresponding parent block
- Indent the block contents by a soft [Tab]
  - Soft [tab] = 2 spaces
- Example:

```
if (some condition) {
··// Block contents indented by a soft [Tab]
}
```

STRICT MODE



```
"use strict";

function calculate() {
    abc = 42;

Uncaught ReferenceError: abc is not defined

    // go get the subtotal and tip amounts from the page
    var subtotalBox = document.getElementById("subtotal");
    var tipBox = document.getElementById("tip");
```

- Some of the characteristics of Strict mode:
  - Prevents variable declaration without var
  - Converts silent errors to exceptions
    - Trying to change the value of document
    - Deleting the prototype of an object

undefined AND null VALUES

- In JavaScript there is a special value `undefined`
  - It means the variable has not been defined (no such variable in the current context)
- `undefined` is different than `null`
  - `null` means that an object exists and is empty

```
var x;
console.log(x); // undefined
x = document.getElementById('id-not-existing');
console.log(x); // null
```

EQUALITY: == != === !==

- Most logical operators automatically convert types. These are all true:
  - 5 < '7'
  - 42 == 42.0
  - '5.0' == 5
- The === and !== are strict equality tests; checks both type and value:
  - '5.0' === 5 is false

Functions in JavaScript

FUNCTION OBJECT

- Functions are one of the most powerful features in JavaScript
  - And one of the most important
- First-class functions in JavaScript
  - They can be assigned to variables or properties, passed as arguments and returned by other functions
- JavaScript does not support function overloading

arguments OBJECT

- Every function have a special object called `arguments`
  - It holds information about the function and all the parameters passed to the function
- The arguments object is not an array
- If in need to iterate it, better parse it to an array:

```
function printArguments() {
  var args = [].slice.apply(arguments);
  for(var i in args) {
    console.log(args[i]);
  }
}
printArguments(1, 2, 3, 4); //1, 2, 3, 4
```

FUNCTION EXPRESSION

- Function expressions are created using the function literal
  - They are available where they are defined
    - And cannot be used beforehand
  - Can be invoked immediately
- The name of function expressions is optional
  - If the name is missing the function is anonymous

```
var printMsg = function (msg){
    console.log('Message: ' + msg);
}
printMsg('Hello');
```

IMMEDIATELY INVOKED FUNCTION EXPRESSIONS (IIFE)

- In JavaScript, functions expressions can be invoked immediately after they are defined
  - Can be anonymous
  - Create a function scope
  - Don't pollute the global scope
  - Handle objects with the same identifier

```
(function(){
  var result = 5;
})();
console.log(result);//ReferenceError
```

Variable Scope

GLOBAL SCOPE

- The global scope is the scope of the web page
  - Or the Node.js app
- Objects belong to the global scope if:
  - They are define outside of a function scope
  - They are defined without var
    - Fixable with `'use strict'`

```
function arrJoin(arr, separator) {
  separator = separator || '';
  arr = arr || [];
  arrString = '';
  for (var i = 0; i < arr.length; i += 1) {
    arrString += arr[i];
    if (i < arr.length - 1) arrString += separator;
  }
  return arrString;
}
```

arr, separator and i belong to the scope of arrJoin

arrString and arrJoin belong to the global scope

FUNCTION SCOPE

- JavaScript does not have a block scope like other programming languages (C#, Java, C++)
  - { and } does not create a scope!
- Yet, JavaScript has a function scope
  - Function expressions create scope
  - Function declarations create scope

```
if(true) { var result = 5; }
console.log(result); //logs 5
```

```
function logResult(){ var result = 5; }
if(true) logResult();
console.log(result); //ReferenceError
```

Simple Modules

NESTED FUNCTIONS

- Functions can be declared everywhere in the JavaScript code
  - Even inside another function:

```
function x(){
  function y() { /* solves international problems  */ }
}
```

  - Inner functions are available only inside their parent scope
    - i.e. y() can be called only from inside x()
  - Remark: every time x() is invoked, a new y() is created!

CLOSURES

- Closures are a special kind of structure
  - They combine a function and the context of this function

```
function outer(x){
  function inner(y){
    return x + ' ' + y;
  }
  return inner;
}
```

inner() forms a closure.
It holds a reference to x

In the context of f1,
x has value 5

```
var f1 = outer(5);
console.log(f1(7)); //outputs 5 7
```

In the context of f2, x
has value "Peter"

```
var f2 = outer('Peter');
console.log(f2('Petrov'));  //outputs Peter Petrov
```

MODULES

- A module is the result that is returned from an IIFE
  - Allows hidden data
  - Evades the polluting of the global scope

```
var getNextId = (function () {
  //lastId is available only inside the IIFE
  var lastId = 0;
  return function () {
    return lastId += 1;
  }
}());
console.log(getNextId());     //prints 1
console.log(getNextId());     //prints 2
console.log(lastId); //throws ReferenceError
```

Object-oriented Design

OBJECT-ORIENTED PROGRAMMING

- OOP means that the application/program is constructed as a set of objects
  - Each object has its purpose
  - Each object can hold other objects
- JavaScript is prototype-oriented language
  - Uses prototypes to define hierarchies
    - Does not have definition for class or constructor
    - ECMAScript 6 introduces classes

CREATING OBJECTS

- When using a function as an object constructor it is executed when called with new
  - Each of the instances is independent

```
function Person(name, age){
  this.name = name;
  this.age = age;
}
var person1 = new Person('George', 23);
console.log(person1.name);
//logs: George
var person2 = new Person('Maria', 18);
console.log(person2.age);
//logs: 18
```

this IN FUNCTION SCOPE

- When executed over a function, without the new operator
  - this refers to the parent scope

```
function Person(name) {
  this.name = name;
  this.getName = function getPersonName() {
    return this.name;
  }
}
var p = new Person("Gosho");
var getName = p.getName;
console.log(p.getName()); //Gosho
console.log(getName()); //undefined
```

BETTER METHOD ATTACHMENT

- Instead of attaching the methods to this in the constructor

```
function Person(name,age){
  //…
  this.sayHello = function(){
    //…
  }
}
```

- Attach them to the prototype of the constructor

```
function Person(name,age){
  //…
}
Person.prototype.sayHello = function() {
  //…
};
```

Object.defineProperty()

- `Object.defineProperty(obj, p, dscrptr)` defines property `p` on object `obj`
  - Example:

```
Object.defineProperty(Person.prototype, 'name', {
  get: function () {
    return this._name;
  },
  set: function (name) {
    if (!validateName(name)) {
      throw new Error('Name is invalid');
    }
    this._name = name;
  }
});
```

```
//calls the setter
p.name = 'Jane Doe';
//calls the getter
console.log(p.name);
```

CONSTRUCTORS WITH MODULES

- Function constructors can be put inside a module
  - Introduces a better abstraction of the code
  - Allows to hide constants and functions
- JavaScript has first-class functions, so they can be easily returned by a module

```
var Person = (function () {
  function Person(name) {
    //…
  }
  Person.prototype.walk = function (distance){ /*...*/ };
  return Person;
})();
```

Tools for JavaScript Development

JAVASCRIPT DEVELOPMENT TOOLS

- All a developer needs for coding JavaScript is a text editor
  - Notepad++
  - Sublime Text 2 / Sublime Text 3
  - Atom.io
  - TextMate
  - JetBrains WebStorm
  - Visual Studio Community 2013

DEBUGGING JAVASCRIPT

- Sadly, no intelligent way of debugging client-side JavaScript
  - The only way to debug JavaScript is through the browser
- Fortunately all browsers have their own debugging tool/plugin that makes it easier
  - Firefox has Firebug
  - Chrome and Opera have Web developer
  - Internet Explorer has F12
  - Node.js has Node Inspector

JAVASCRIPT UTILITIES

- JavaScript Code Quality Tools
  - http://www.jshint.com/
  - http://www.jslint.com/
- JavaScript performance tester
  - http://jsperf.com

JAVASCRIPT RESOURCES

- Mozilla Development Network (MDN)
  - https://developer.mozilla.org
  - Mostly used for the presentations
- Books:
  - https://s3.amazonaws.com/dailyjs/files/build-a-javascript-framework.pdf
  - http://addyosmani.com/resources/essentialjsdesignpatterns/book/
  - https://github.com/kms-technology/javascript

PROJECT TOOLS

- NPM & Bower
  - Install Node.js packages or client libraries
- Grunt/Gulp
  - Tasks runner
  - Create different tasks for build/development/test cases
- Yeoman
  - Scaffolding of applications
  - One-line-of-code to create a project template with views/routes/modules/etc…

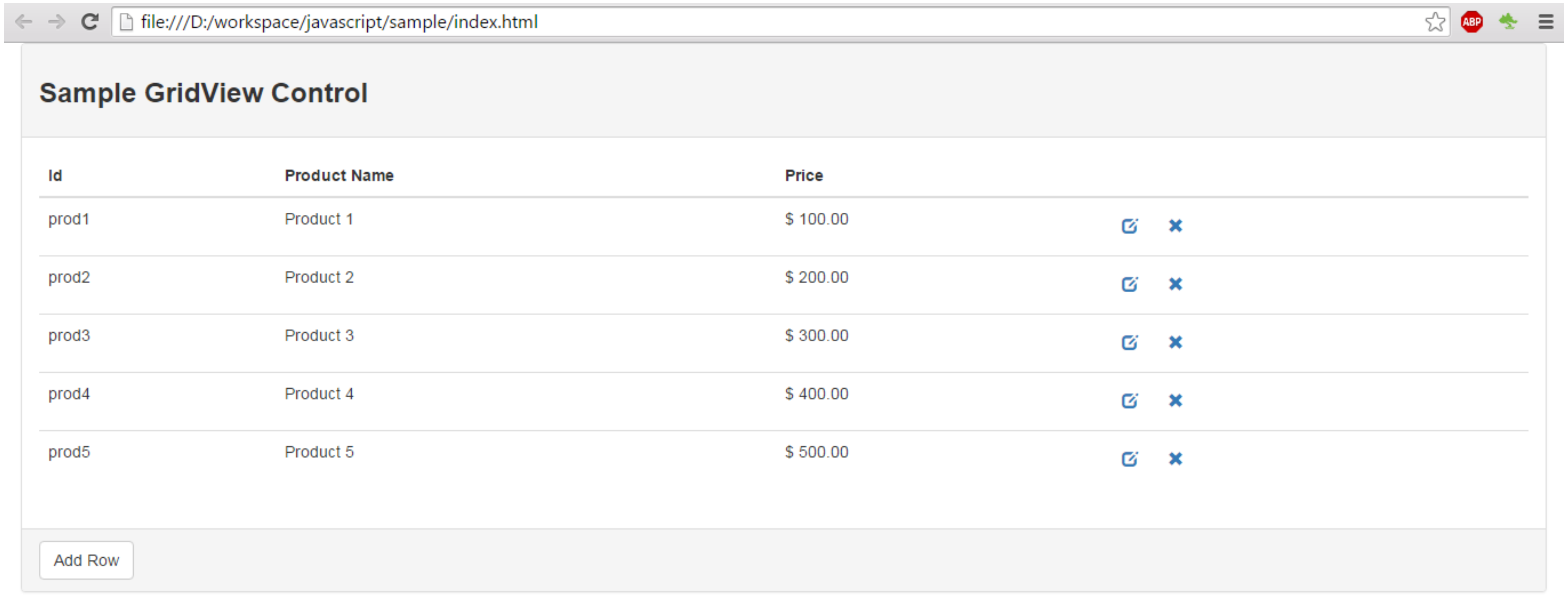
WHAT IS A JAVASCRIPT LIBRARIES/FRAMEWORK

- A JS library is pre-written code that aims to facilitate and /or jump start development, especially AJAX based tasks
- Prominent JSL
  - jQuery
  - lodash
  - Modernizr
  - BackboneJS
  - AngularJS
  - ReactJS

ASSIGNMENT

- Implement a GridView control
  - Data can be loaded from an array
  - A header row can be configured
  - Data value can be formated
  - Row can be added/removed dynamically
  - Each row can be edited

ASSIGNMENT


file:///D:/workspace/javascript/sample/index.html

Sample GridView Control

Id	Product Name	Price	
prod1	Product 1	$ 100.00	
prod2	Product 2	$ 200.00	
prod3	Product 3	$ 300.00	
prod4	Product 4	$ 400.00	
prod5	Product 5	$ 500.00	

HOME PAGE

```
var grid = new GridView({
  columns: [{
    dataIndex: 'id',
    header: 'Id'
  }, {
    dataIndex: 'name',
    header: 'Product Name'
  }, {
    dataIndex: 'price',
    header: 'Price',
    display: function(value) {
      return '$ ' + (value / 100).toFixed(2);
    }
  }],
  renderTo: document.getElementById('ui-view')
});
```

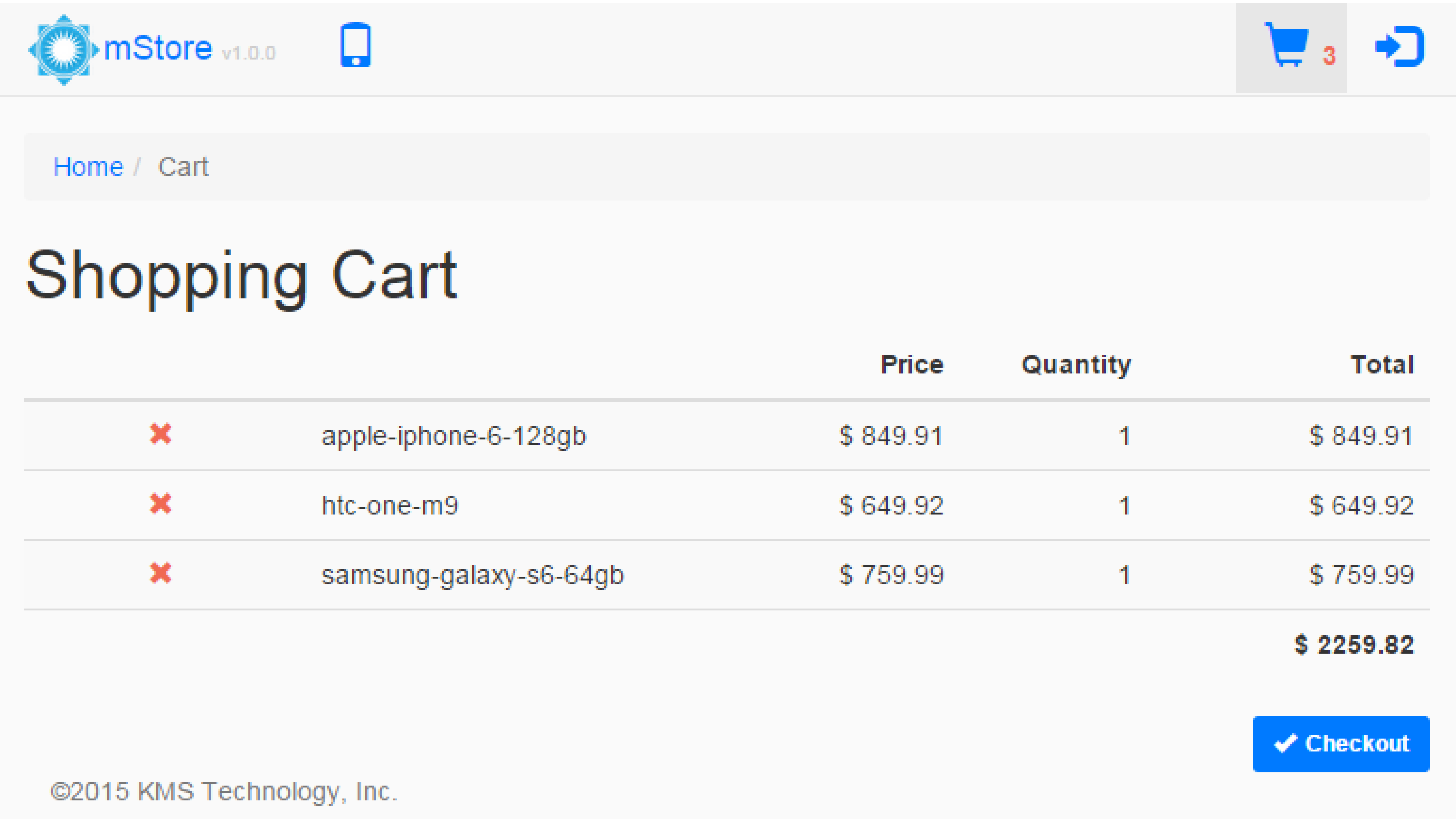
GRIDVIEW

```
var GridView = (function() {
 'use strict';

  function GridView(options) {
    this.options = options || {};
  }

  GridView.prototype.render = function() {
  }
  GridView.prototype.loadData = function(data) {
  }
  GridView.prototype.addRow = function(dataRow) {
  }
  ...

  return GridView;
})();
```

DOM

```
window.Dom || (function(window) {
  'use strict';

  window.Dom = {
    createElement: function(tag, attributes, children) {
    },
    render: function(node, parent) {
    }
  }
})(window);
```

CART



mStore v1.0.0

3

Home / Cart

Shopping Cart

		Price	Quantity	Total
✖	apple-iphone-6-128gb	$ 849.91	1	$ 849.91
✖	htc-one-m9	$ 649.92	1	$ 649.92
✖	samsung-galaxy-s6-64gb	$ 759.99	1	$ 759.99
				$ 2259.82

©2015 KMS Technology, Inc.

GRADING TABLE

Content	Grade
Implement a GridView component/module	30
Apply GridView to real application	30
GridView should be flexible configuration	10
JavaScript Coding Styleguide	10
Enhance GridView component (*)	10
Integrated with server side (*)	10

THANK YOU