

Development of Internet Applications

jQuery, TypeScript, LESS

Ing. Michal Radecký, Ph.D.

www.cs.vsb.cz/radecky

Why „extended“ JavaScript

- Easy development and implementation
- Better compatibility and multiplatform operation
- Mature development concepts
- Available documentation
 - <http://devdocs.io>
- Support within develop tools
 - VS Code
 - <http://www.jsfiddle.net/>

JavaScript frameworks

- They are JavaScript libraries which help with development of applications and make the work easier.
- The developer can be more focused on solving of problems, not on the optimization and debugging of the code for all web browsers.
- They are based on pure JavaScript and extend the objects, methods, etc. (by usage of prototype)
- Usually, there is a huge set of plugins that implements common features and functions (DOM modification, AJAX, photgallery, etc.)
- It also brings new concepts in development, operation and UI



jQuery

- Javascript library
- Current version: 3.7.1 (1.4.1 – IE 6,7,8, Migrate Plugin)
- Size: 250kB
- Basic syntax: function \$ or jQuery
 - Prototyping of the native Window object
- Main focus
 - Manipulation with DOM
 - Events
 - Animations
 - Communication (AJAX, JSON)
- Huge utilization of anonymous and nested functions
- Plugins and extensions

```
<head>
```

```
<script type="text/javascript" src="jquery.js"></script>
```

Include the jquery file

```
<script type="text/javascript">
```

```
$(document).ready(function(){
```

The "ready event" (Binds a function to be executed whenever the DOM is ready)

This part can be written in an external .js file.

```
$(".button").click(function(){  
    $("#panel").slideDown("slow");  
});
```

Where do you want to bind the function?
It can be CSS class, ID, Selectors (ie. DIV, H1, A, P, LI...)

This function will be triggered when an element with class="button" is clicked

What would like to do with #panel?
In this case, slide it down with "slow" speed.

```
});
```

```
</script>
```

```
</head>
```

Where do you want to apply this function?
In this case, it is the element with id="panel"

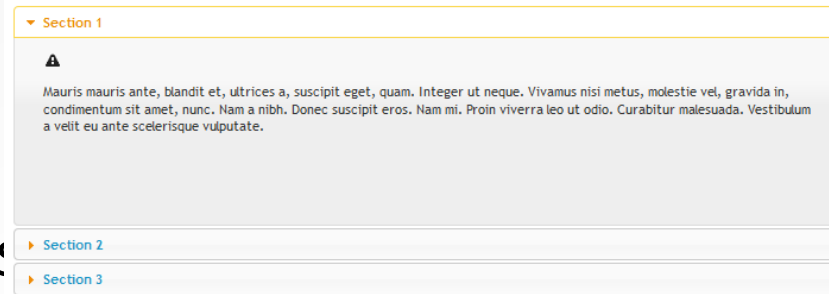
```
$("#panel")
```

The quotation marks can be either single or double.
ie. ("class") or (.class)

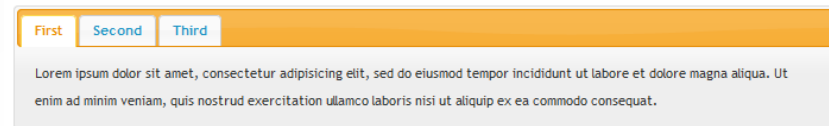
jQuery UI

- Extension of jQuery
- Interactive components of user interface, focused on complex web applications
- A huge set of control elements (incremental library)
- Templates for visualization

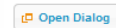
Accordion



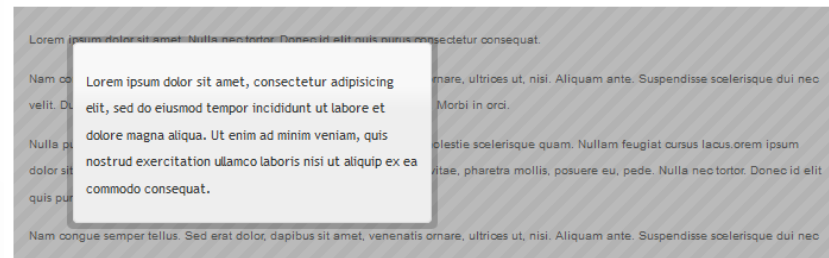
Tabs



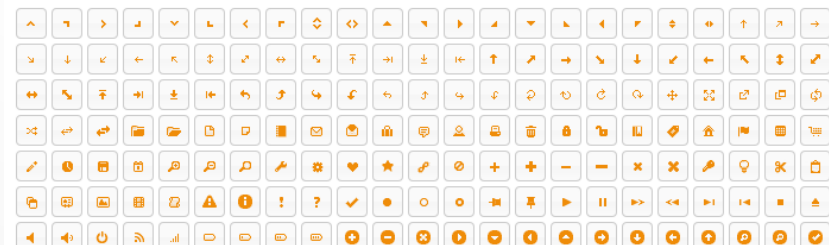
Dialog



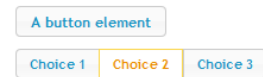
Overlay and Shadow Classes



Framework Icons (content color preview)



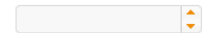
Button



Autocomplete



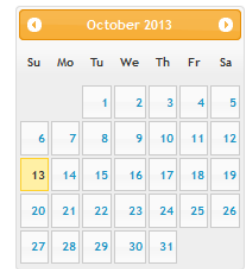
Spinner



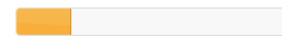
Slider



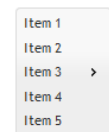
Datepicker



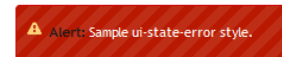
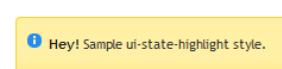
Progressbar



Menu

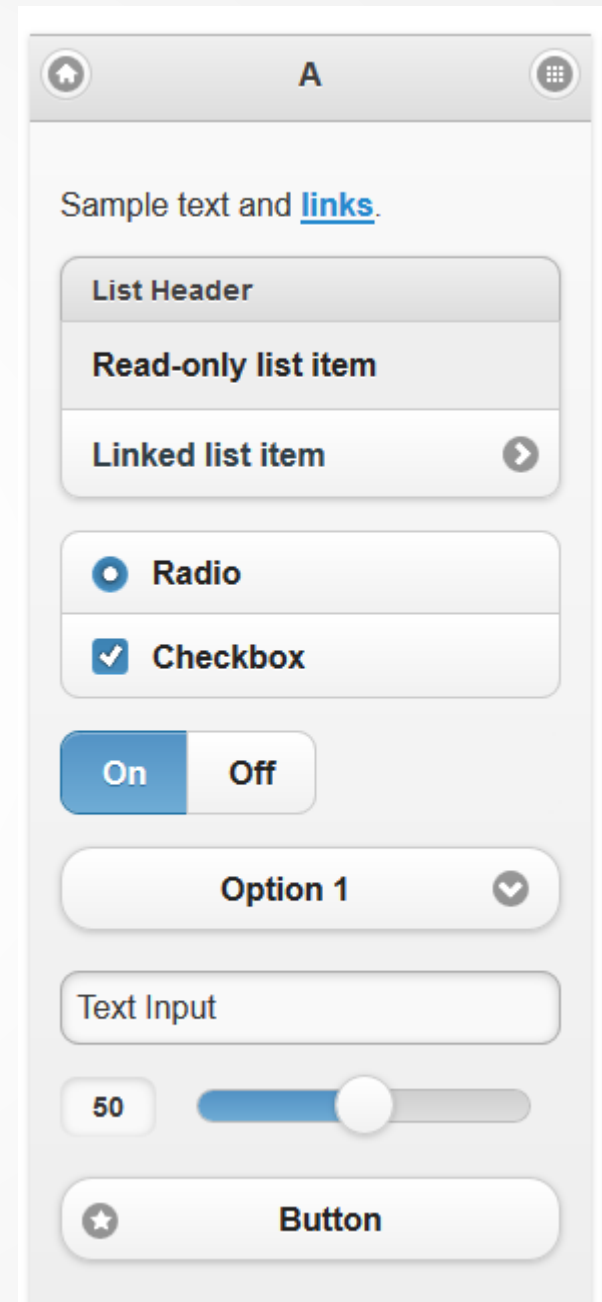


Highlight / Error



jQuery Mobile

- Extension of jQuery
- Based on jQuery UI
- Optimization of size and functionality on mobile devices
- Web application with universal look-and-feel across platforms
- Templates for visualization



Other UI frameworks

- ReactJS - Material UI, Redux, Semantic UI
- Angular.js
- Vue.js
- UIKit
- Foundation
- Bootstrap

- React Native
- Ionic
- Framework 7
- Apache Cordova

TypeScript

TypeScript

- The programming language by Microsoft – transpiler to JavaScript
- Compiler/transpiler is necessary (integrated in VS)
- Many other IDE with TypeScript support (i.e. VS Code, WebStorm, Atom, Sublime Text, or Eclipse).
- „Every JavaScript code is TypeScript code as well“
- The output is always JavaScript code, however it is more effective and easier way of development (VS Code)
- Optimization, minification, etc.
- There is no influence on performance during operation – it is still pure JavaScript
- Syntax and constructions based on ECMAScript 6+
- Integration and utilization within different environments and approaches

What can TypeScript offer

- Static data types
- Classes and inheritance
- Modules/namespaces
- Interfaces
- Generic data types
- Covariation and contravariation - polymorphism
- Duck-typing is applied – the interface is determining
- And more else... (destructuring assignment, const, for .. of, ...)
- Asynchronous paradigm

What can IDE offer thanks to TypeScript

- IntelliSense for own code, javascript libraries and DOM
- Highlighting of warnings and errors
- Refactoring
- Go To Definition and Find All References
- Repository of definition libraries- <https://github.com/borisyankov/DefinitelyTyped>
- <https://code.visualstudio.com/docs/typescript/typescript-compiling>

TypeScript

The screenshot shows the Microsoft Visual Studio IDE with a TypeScript file named 'file1.ts'. The code is organized into several sections: an interface, a module, a class, and local variables. The interface 'IPoint' has a 'getDist()' method. The 'Shapes' module contains a 'Point' class that implements 'IPoint'. The 'Point' class has a constructor and a 'getDist()' method. The local variables section declares 'p' as an 'IPoint', 'dist' as a number, and 'ost' as a string.

```
// Interface
interface IPoint {
  getDist(): number;
}

// Module
module Shapes {

  // Class
  export class Point implements IPoint {
    // Constructor
    constructor (public x: number, public y: number) { }

    // Instance member
    getDist() { return Math.sqrt(this.x * this.x + this.y * this.y); }

    // Static member
    static origin = new Point(0, 0);
  }
}

// Local variables
var p: IPoint = new Shapes.Point(3, 4);
var dist = p.getDist();
var ost = "asfd";
```

```
var Shapes;
(function (Shapes) {
  var Point = (function () {
    function Point(x, y) {
      this.x = x;
      this.y = y;
    }
    Point.prototype.getDist = function () {
      return Math.sqrt(this.x * this.x + this.y * this.y);
    };
    Point.origin = new Point(0, 0);
    return Point;
  })();
  Shapes.Point = Point;
})(Shapes || (Shapes = {}));

var p = new Shapes.Point(3, 4);
var dist = p.getDist();
var ost = "asfd";
```

Other „transpilers“

- Opal – input language is Ruby
- Kotlin/JS – relation to Java, Swift (Objective-C), Java VM
- PureScript – inspired by Haskell (functional programming language)
- CoffeeScript – inspired by Ruby, Python, Haskell
- Dart – inspired by C, multiplatform compilation (JS, WebAssembly, Dart Native)
- <https://hackernoon.com/10-more-typescript-alternatives>

Dart
Programming langua...

Kotlin
Programming langua...

TypeScript
High-level programm...

CoffeeScript
Programming langua...



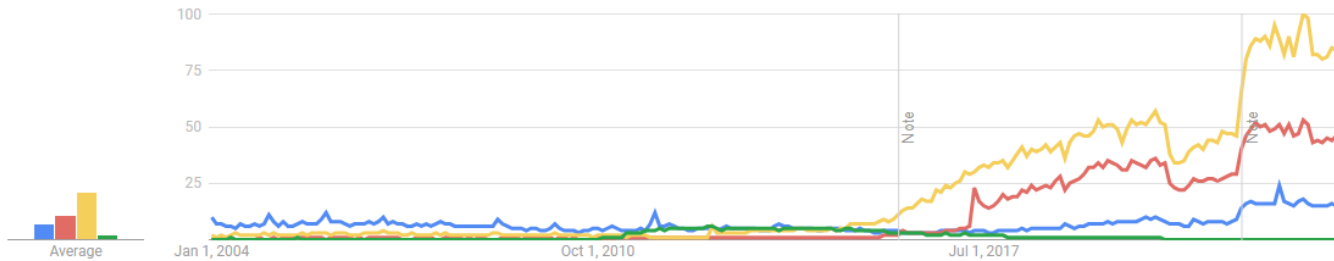
Worldwide ▾

2004 - present ▾

All categories ▾

Web Search ▾

Interest over time ?



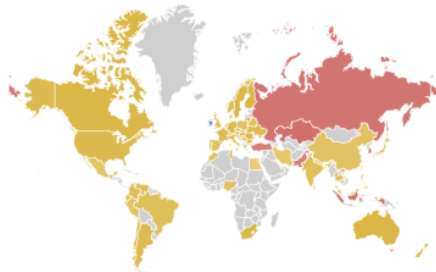
Compared breakdown by region

Region ▾



Dart **Kotlin** **TypeScript** **CoffeeScript**

Sort: Interest for Dart ▾



1	Ireland	
2	Egypt	
3	Nepal	
4	Malaysia	
5	Bangladesh	

Why „extended“ CSS

- Easy development and implementation
- Mature development concepts
- Better compatibility and multiplatform operation
- Easy to learn
- Support within develop tools



Sass

Less

- CSS preprocessor – from special syntax to output in pure CSS
- „CSS code is still LESS code“
- Improving development and sustainability
- Compilation can be performed in several environments
- Features
 - Variables
 - Mixins
 - Nesting of elements
 - Math operations
 - Functions
 - Imports
- <http://lesscss.org/>

SASS/SCSS

- CSS preprocessor
- Originally syntax based on "indentation"
- Now based on CSS syntax (SCSS) - CSS is SASS
- Support for compilation in different languages
- Features
 - Variables
 - Mixins
 - Nesting of elements
 - Math operations
 - Functions
 - Imports
- <https://sass-lang.com/>

LESS vs. SASS

{ Comparison }

	LESS (@)	SASS (\$)
Variables	<pre>@plainRed: #ff0000; @softBlue: #bce7f3;</pre>	<pre>\$plainRed: #ff0000; \$softBlue: #bce7f3;</pre>
Mixins	<pre>.fluidBox{ width: 50%; box-sizing: border-box; }</pre>	<pre>@mixin fluidBox{ width: 50%; box-sizing: border-box; }</pre>
Parametric Mixins	<pre>.rounded(@radius: 5px){ -webkit-border-radius: @radius; -moz-border-radius: @radius; border-radius: @radius; }</pre>	<pre>@mixin rounded(\$radius: 5px){ -webkit-border-radius: \$radius; -moz-border-radius: \$radius; border-radius: \$radius; }</pre>
Functions	<pre>lighten(#ff0000, 10%); Saturate(#ff0000, 20%);</pre>	<pre>lighten(#ff0000, 10%); Saturate(#ff0000, 20%);</pre>
Operators	<pre>#header{ width: (@headerW - 50) * 2; }</pre>	<pre>#header{ width: (\$headerW - 50) * 2; }</pre>
Frameworks	LESSHat, LESS ELEMENTS	COMPASS
Language Base	Javascript (originally Ruby)	Ruby

Other preprocessors

- Stylus
 - Syntax based on original SASS, „indentation“
 - Relation to node.js family
 - www.stylus-lang.com
- PostCSS
 - More transformer than preprocessor
 - Applying various rules and overrides to existing CSS
 - Based on JavaScript
 - www.postcss.org
- styled-components, Compass, ...