

# Web Application Development

---

## „Extended“ Javascript and CSS

**Ing. Michal Radecký, Ph.D.**  
[www.cs.vsb.cz/radecky](http://www.cs.vsb.cz/radecky)



## Why „extended“ JavaScript

- Easy development and implementation
- Better compatibility and multiplatform operation
- Mature development concepts
- Not necessary to reinvent the wheel
  
- Available documentation
  - <http://devdocs.io>
- Support within develop tools
  - VS Code
  - <http://www.jsfiddle.net/>



# 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

# jQuery

- Upcoming version4 (beta)
  - Delete outdated code(IE11+)
  - Performance and compatibility improvements
  - Modularization
  - Emphasis on asynchronous programming
  - ECMAScript 6+

A leaner, faster, and better adapted library to modern web development standards, while maintaining backward compatibility for developers who choose to use jQuery.

```
<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(){
```

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

```
$("#panel").slideDown("slow");
```

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>
```

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

```
</head>
```

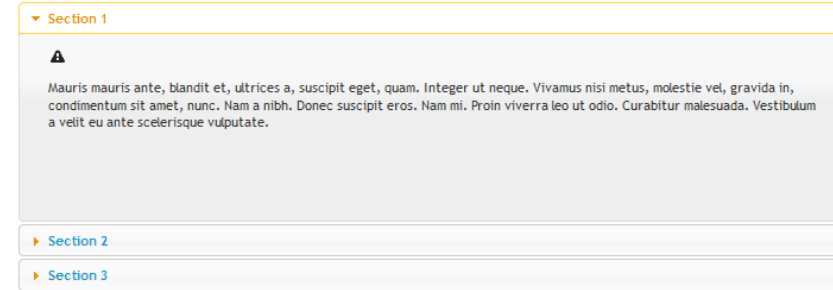
```
$("#pane")
```

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

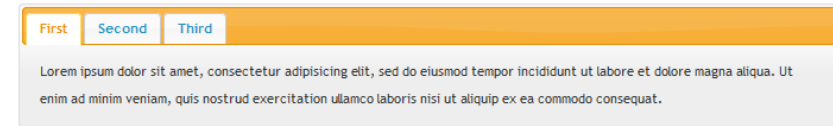
# jQuery UI

- Extension of jQuery
- Interactive components of user interface, focused on complex web applications
- A huge set of control elements (incremental library)
- Templates for visualization
- Not supported in JQuery 4.0

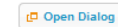
## Accordion



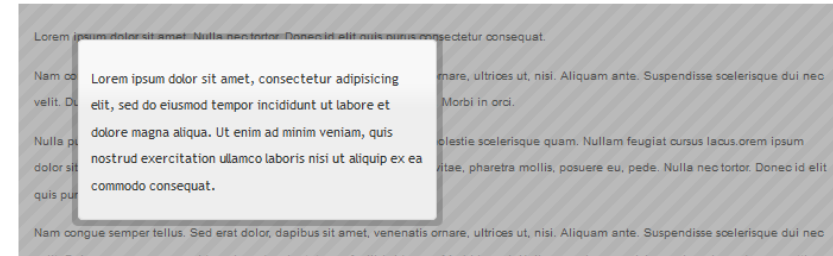
## Tabs



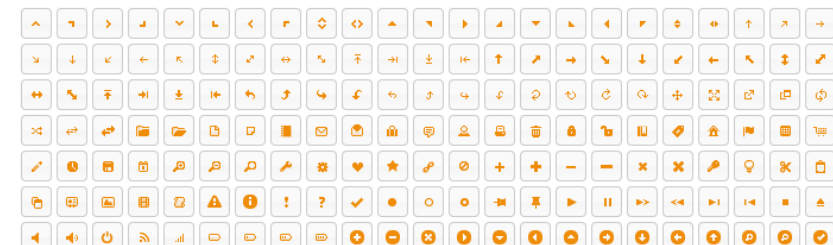
## Dialog



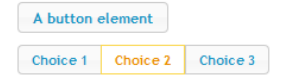
## Overlay and Shadow Classes



## Framework Icons (content color preview)



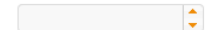
## Button



## Autocomplete



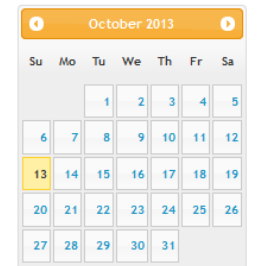
## Spinner



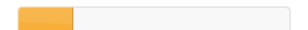
## Slider



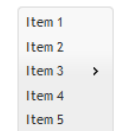
## Datepicker



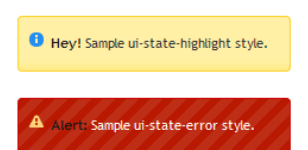
## Progressbar



## Menu

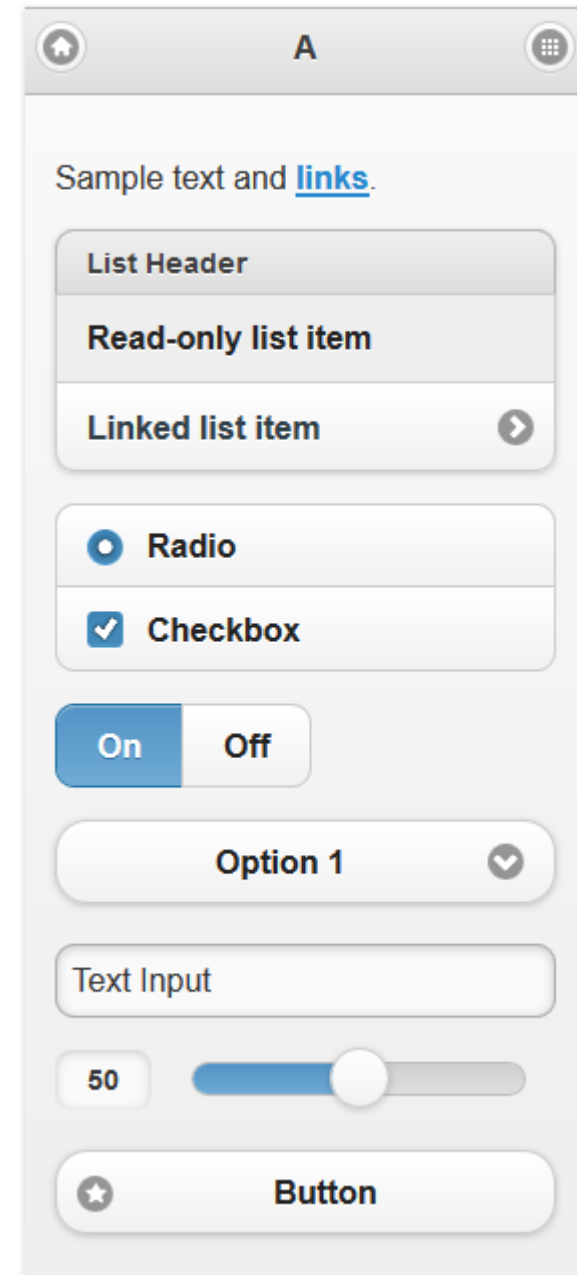


## Highlight / Error



# 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
- Deprecated, will no longer be supported





# 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 objected-oriented 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+ (updating)
- 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 (direction) - polymorphism
- Duck-typing is applied – the interface is determining
- And more else... (destructuring assignment, const, for .. of, ...)
- Asynchronous paradigm
- JSX support

# What can IDE for TypeScript offer

- 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 split into two panes. The left pane shows the source code with comments, and the right pane shows the compiled JavaScript output. The Solution Explorer on the right shows a variable 'p' selected.

```
// 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>

typescript  
Hledaný výraz

kotlin  
Vyhledávací dotaz

dart  
Vyhledávací dotaz

coffescript  
Vyhledávací dotaz



Celosvětově

2004–současnost

Programování

Vyhledávání na webu

### Zájem v průběhu času



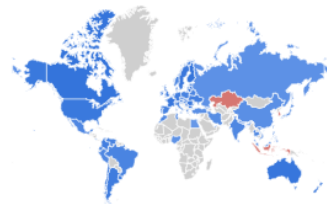
### Porovnání podle oblasti

Oblast



typescript kotlin dart coffescript

Řazení: Zájem o termín typescript



1	Izrael	typescript, kotlin, dart
2	Belgie	typescript, kotlin, dart
3	Dánsko	typescript, kotlin, dart
4	Kanada	typescript, kotlin, dart
5	Finsko	typescript, kotlin, dart



## Why „extended“ CSS

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



# Less

- CSS preprocesor – 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 environments
- Features
  - Variables
  - Mixins
  - Nesting of elements
  - Math operations
  - Functions
  - Imports

<https://sass-lang.com/>

# LESS vs. SASS

## { Comparison }

	LESS (@)	SASS (\$)
Variables	<code>@plainRed: #ff0000; @softBlue: #bce7f3;</code>	<code>\$plainRed: #ff0000; \$softBlue: #bce7f3;</code>
Mixins	<code>.fluidBox{   width: 50%;   box-sizing: border-box; }</code>	<code>@mixin fluidBox{   width: 50%;   box-sizing: border-box; }</code>
Parametric Mixins	<code>.rounded(@radius: 5px){   -webkit-border-radius: @radius;   -moz-border-radius: @radius;   border-radius: @radius; }</code>	<code>@mixin rounded(\$radius: 5px){   -webkit-border-radius: \$radius;   -moz-border-radius: \$radius;   border-radius: \$radius; }</code>
Functions	<code>lighten(#ff0000, 10%); Saturate(#ff0000, 20%);</code>	<code>lighten(#ff0000, 10%); Saturate(#ff0000, 20%);</code>
Operators	<code>#header{   width: (@headerW - 50) * 2; }</code>	<code>#header{   width: (\$headerW - 50) * 2; }</code>
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](http://www.stylus-lang.com)

## PostCSS

- More transformer than preprocessor
- Applying various rules and overrides to existing CSS
- Based on JavaScript
- [www.postcss.org](http://www.postcss.org)

## Styled-components, Compass, ...