

From HTML to PostGIS

Michał Okulewicz

Wydział Matematyki i Nauk Informatycznych
Politechnika Warszawska

Lecture plan

- 1 JavaScript
 - Introduction
 - DOM and Events

JavaScript

- Purpose and syntax of JavaScript
- JavaScript based web applications (DOM and Events)
 - Creating and changing the contents of the website
 - Event handler properties (*element.onsth = handler;*)
 - Document Object Model Level 2 Events
(add/removeEventListener)
 - Creating and changing the styles on the website
 - style property
 - using classes
- Selected additional topics
 - Utilizing canvas
 - jQuery
 - In-context library example: GoogleMaps API

Introduction

- JavaScript is a programming language primarily used to enrich the functionality of web sites by adding dynamic interaction with the user
- JavaScript is an interpreted language with interpreter built in the most of the browsers
- **JavaScript is the reason why we even do have the Single Page Applications**
- JavaScript processing engines are available also outside the browser (see: [node.js](https://nodejs.org/))
- Side note: The original JavaScript was actually one of the early dialects of an ECMAScript standard

Versions of ECMAScript and their support

Version	Support
ES5 (2009)	All major browsers
ES6 / ES2015	Edge 17 / FF 60 / CH 68
ES2016	CH 68
ES2017	CH 68
ES2018	CH 68

[source](#)

Further reading

- [ES5 specification](#) and [W3Schools walk-through](#)
- [ES6 specification](#) and [W3Schools walk-through](#)
- **Differences between ES5 and ES6** (note to self: click it!)
- [Document Object Model specification](#)
- [DOM Level 2 Events specification](#)

The simplest example

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="content-type"
        content="text/html; charset=utf-8" />
  <script type="text/javascript">
    console.info('Hello!');
  </script>
</head>
<body>
  <p>Check the console to see results.</p>
</body>
</html>
```

Analyze types and syntax details

Run `ip_4_js_fin_playground.html...`

Check the following:

- Numbers and strings
- Booleans (see: [John 18,38](#))
- Loops and exceptions (play with debugger)
- Objects definitions

Analyze types and syntax details

Run `ip_4_js_fin_playground.html...`

Check the following:

- Numbers and strings
- Booleans (see: [John 18,38](#))
- Loops and exceptions (play with debugger)
- Objects definitions

Analyze types and syntax details

Run `ip_4_js_fin_playground.html...`

Check the following:

- Numbers and strings
- Booleans (see: [John 18,38](#))
- Loops and exceptions (play with debugger)
- Objects definitions

Analyze types and syntax details

Run `ip_4_js_fin_playground.html...`

Check the following:

- Numbers and strings
- Booleans (see: [John 18,38](#))
- Loops and exceptions (play with debugger)
- Objects definitions

Analyze types and syntax details

Run `ip_4_js_fin_playground.html...`

Check the following:

- Numbers and strings
- Booleans (see: [John 18,38](#))
- Loops and exceptions (play with debugger)
- Objects definitions

Not so stupid example

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="content-type"
        content="text/html; charset=utf-8" />
  <script type="text/javascript">
    document.addEventListener("DOMContentLoaded", function (event) {
      var p = document.createElement('p');
      var text = document.createTextNode('DOM fully loaded and parsed');
      p.appendChild(text);
      document.body.appendChild(p);
    });
  </script>
</head>
<body>
  <p>This time no need to check the console.</p>
</body>
</html>
```

Not so stupid example explained

- This one operates on a tree-like structure of an HTML document
- It waits till the document is ready
- It creates a paragraph and text node in memory
- It constructs a paragraph with contents
- It adds paragraph to the body of the document

Typical JS use cases

- Content manipulation
- Data validation

Table generation

Run ip_4_js_partII_playground.html...

Table generator

Rows:

Columns:

(0,0)	(0,1)	(0,2)
(1,0)	(1,1)	(1,2)

(note to self: remember to talk about **closure**)

Form validation

Run `ip_4_js_partII_playground.html...`

Sample HTML form

Two words with capital letters input:

Descending ordered sequence:

Send form

This time consider using classes instead of inline style changes (note to self: remember JS hooks).