The Start of Frontend

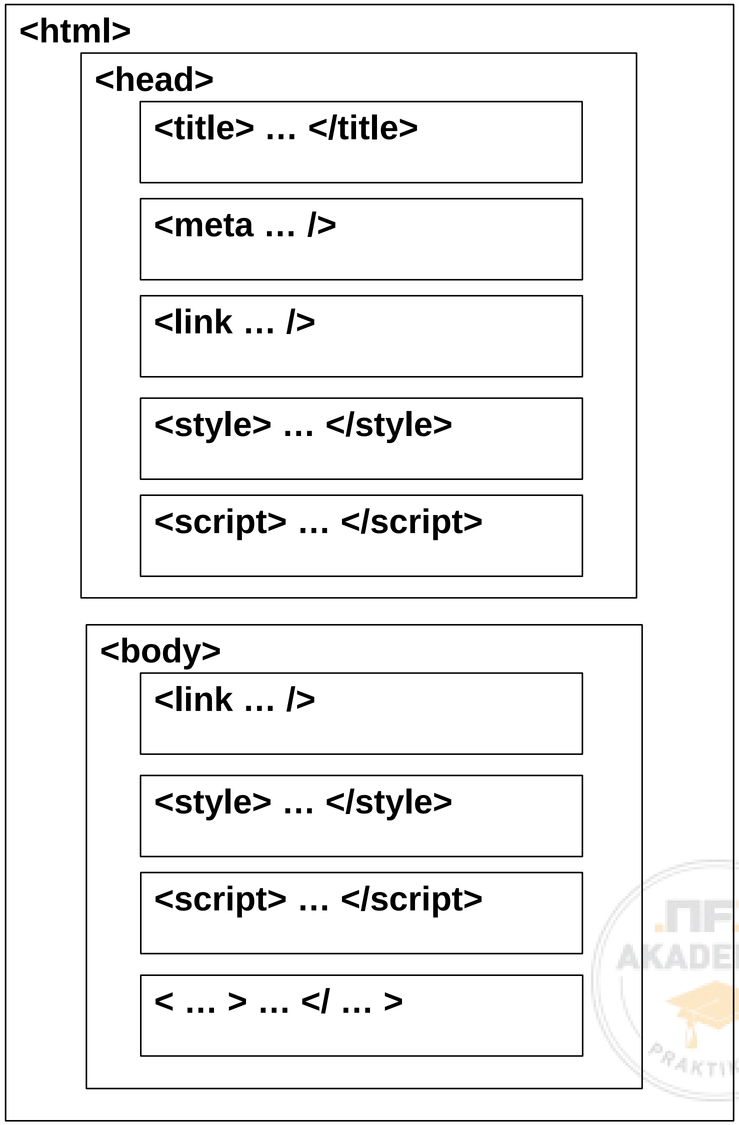
Martynas Barzda, Software Engineer
martynas.barzda@nfq.lt

HTML (Hyper Text Markup Language)

- Created in '90 at CERN for documents sharing purposes by Tim Barnes-Lee
- Specifications maintained by World Wide Web Consortium (W3C) since 1996
- HTML5 standardized in late 2014 and it became recommendation
- XHTML is extended HTML4 with XML 1.0 specification
- Uses .html or .htm for filename extension, MIME: text/html

The HTML Concept

- html
  - head
    - title
    - meta
    - link
    - style
    - script
  - body
    - link
    - style
    - script
    - …



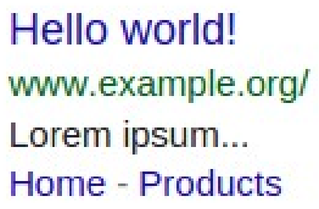
The Markup

```
<!DOCTYPE html>
<html lang=“en“>
    <head>
        <title>Hello world!</title>
    </head>
    <body>
        <p>Hello world!</p>
    </body>
</html>
```

Semantic Markup

- Search-engine spiders are able to rate significance of pieces of text they found in markup, the semantic structures applied to bring out the meaning of published text
- Good semantic improves accessibility and screen reader or audio browser can correct ascertain the structure of document

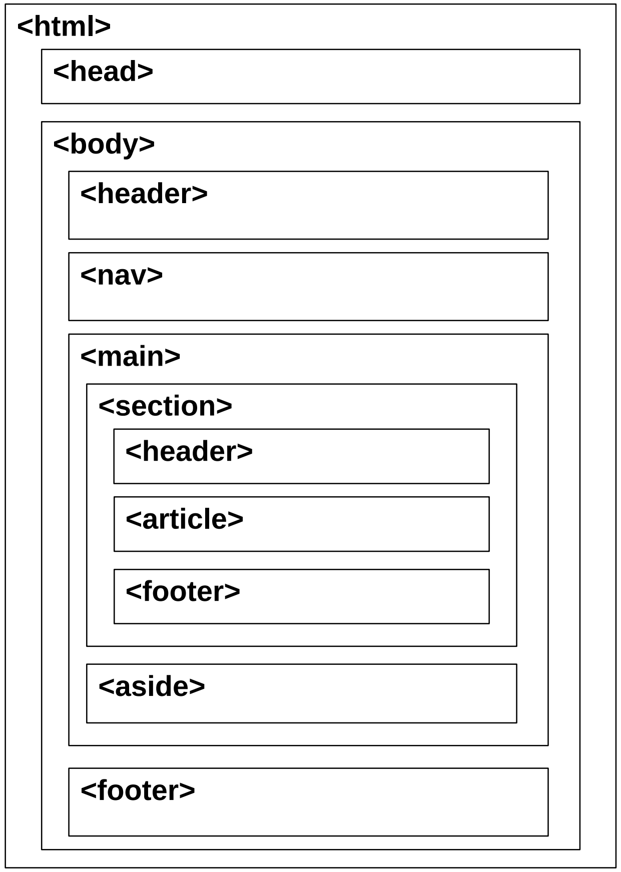
```
<nav role="navigation" aria-label="main navigation">
    <ul>
        <li><a href="/">Home</a></li>
        <li><a href="/products">Products</a></li>
    </ul>
</nav>
```

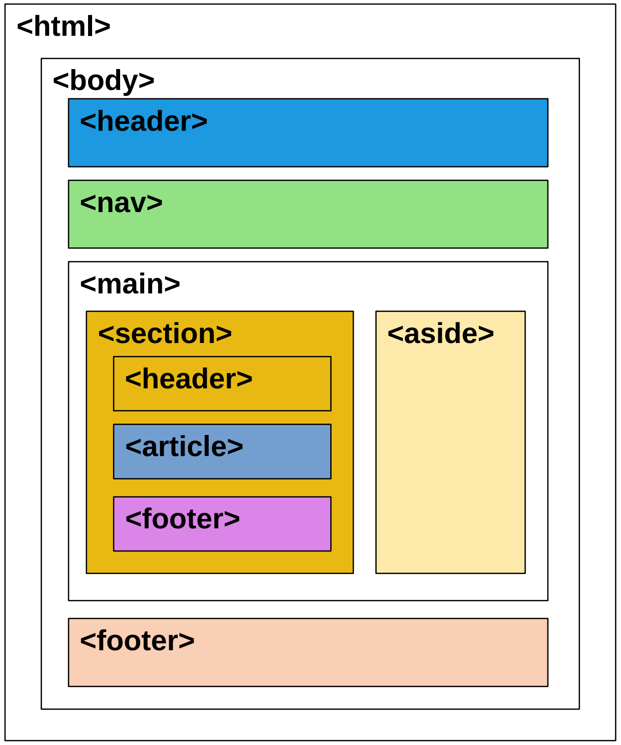

Hello world!
www.example.org/
Lorem ipsum...
Home - Products

CSS

- V1 released in 1996 by Håkon Wium Lie and Bert Bos;
- Specifications maintained by World Wide Web Consortium (W3C) since 1998
- CSS2.1 is standard, CSS3 is still working draft, but 4 modules announced as recommendations (media queries, namespaces, selectors level 3, color)
- Uses .css for filename extension, MIME: text/css

The CSS concept



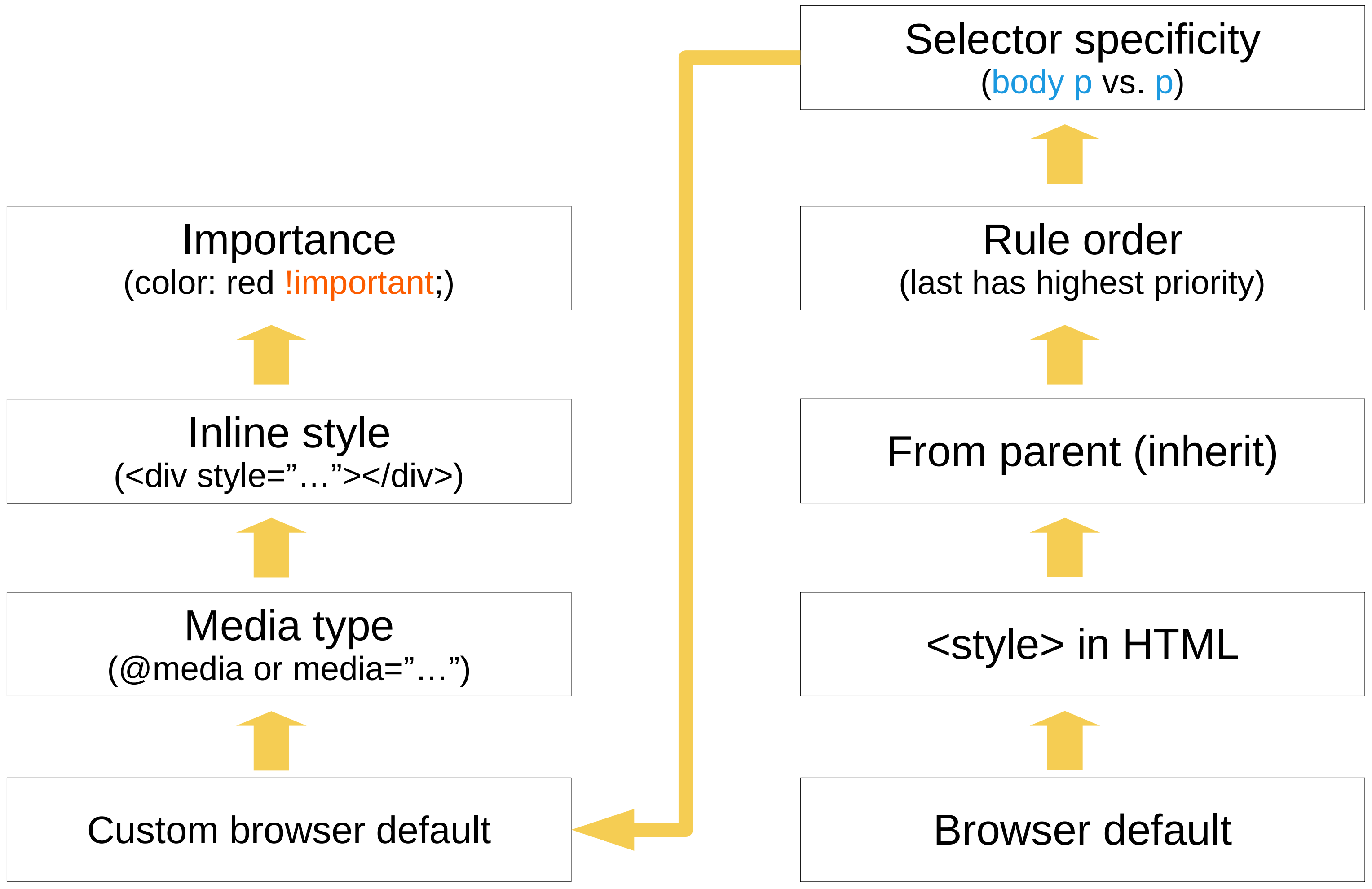


The CSS syntax

```
// Rule

selector {

    // Declarations block

    property: value;

}


selector children {

    property: value;

}
```

```
// Rule-set

selector1,

selector2 {

    property: value;

}


selector:pseudo-class {

    property: value;

}
```

CSS priority scheme



CSS selectors

- Selectors are unable to ascend
- Selectors can be:
  - elements of specific type (p, div, h1, section...)
  - elements specified by “id” or “class” attributes (#unique-el, .style-group)
  - elements depending on how they are placed relative to others in the document tree (section > article, .wrapper .module strong)
- Browser reads selector from right to left

Reading of selectors

```
.offer  h1
.offer-title
```

```
<div class=“ad“>
    <h1 class=“ad-title“></h1>
</div>
<article class=“offer“>
    <h1 class=“offer-title“></h1>
</article>
<article class=“offer“>
    <h1 class=“offer-title“></h1>
</article>
```

Reading of selectors

```
.offer h1
.offer-title
```

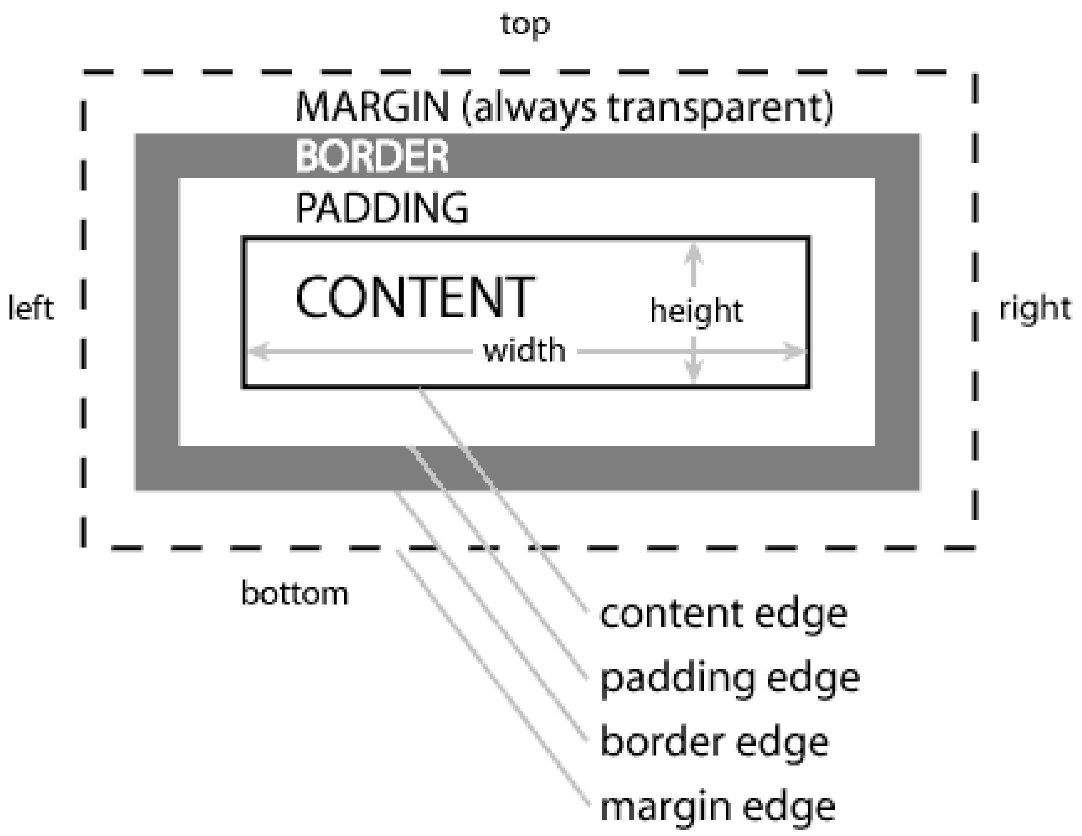
```
<div class="ad">
    <h1 class="ad-title"></h1>
</div>
<article class="offer">
    <h1 class="offer-title"></h1>
</article>
<article class="offer">
    <h1 class="offer-title"></h1>
</article>
```

Reading of selectors

```
.offer  h1
.offer-title
```

```
<div class=“ad“>
    <h1 class=“ad-title“></h1>
</div>
<article class=“offer“>
    <h1 class=“offer-title“></h1>
</article>
<article class=“offer“>
    <h1 class=“offer-title“></h1>
</article>
```

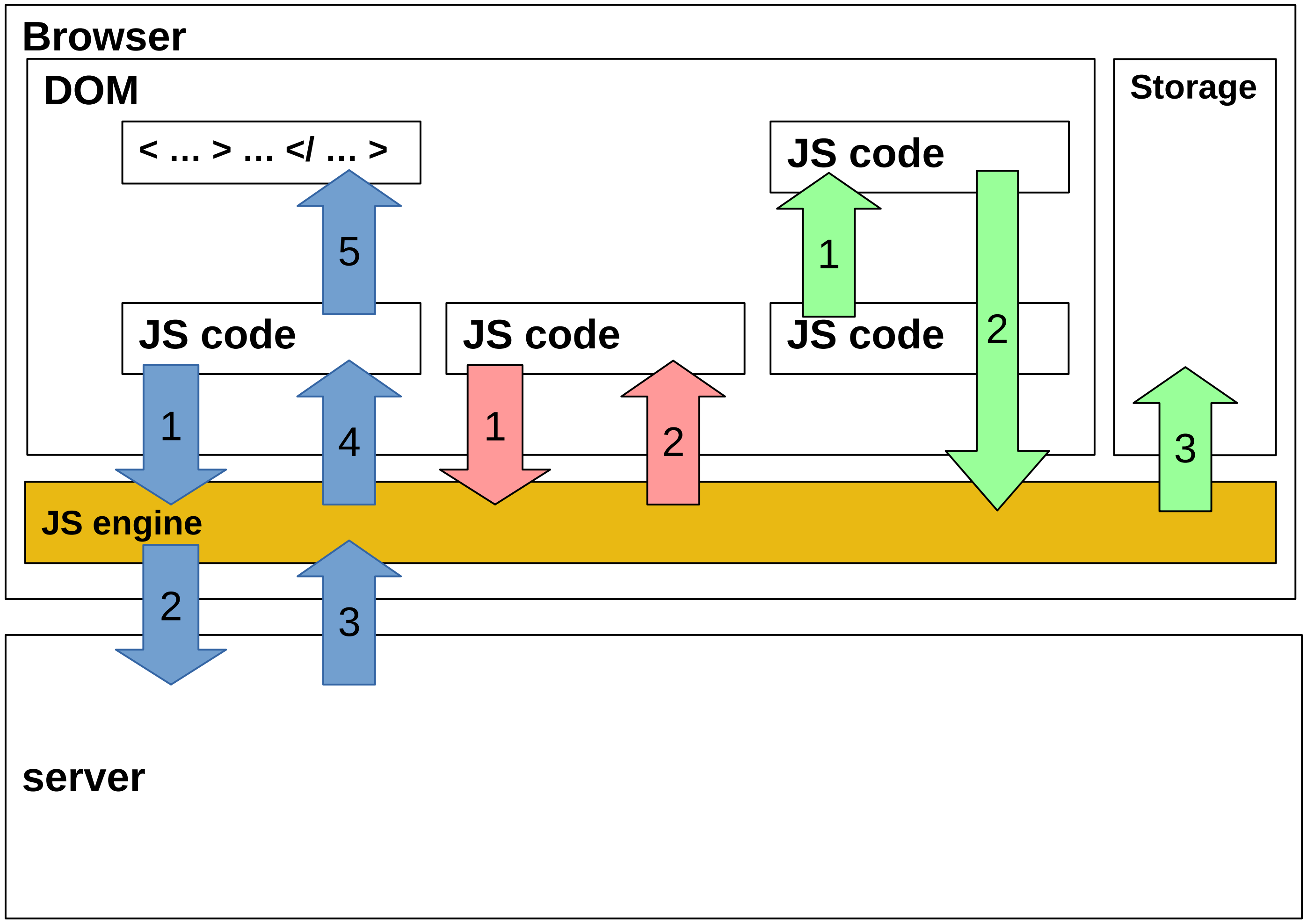
CSS Box Model



JavaScript

- Designed by Brendan Eich, shipped in Netscape Navigator in 1995
- Multi-paradigm: scripting, object oriented (prototype-based), imperative, functional
- Standardized as ECMAScript since 1996. Current standard is ECMAScript 5.1
- Officially managed by Mozilla Foundation
- Uses .js for filename extension, MIME: application/javascript (text/javascript is obsolete by RFC 4329)

The JavaScript Concept



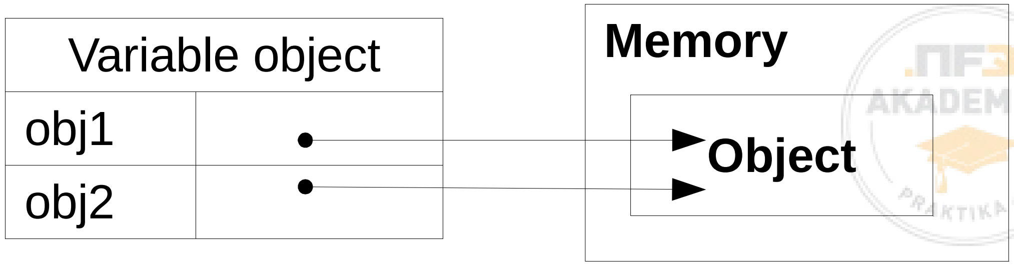
Types

- JS uses 2 kinds of types:
  - Primitive – simple data types
  - Reference – objects which are references to locations in memory
- 5 primitive types:
  - Boolean – true/false or new Boolean()
  - Number – integer or floating-point numeric value or new Number()
  - String – characters delimited by either single or double quotes or new String()
  - Null – null value
  - Undefined – undefined value (not initialized variable has undefined value, an ex. var a;)

Reference types

- Reference values are instances of reference types and are synonymous with objects (unordered lists of properties consisting of a name and a value)
- Reference types do not store the object directly into variable. Variable holds a pointer of object
- When assigned a variable with object pointer to another, each variable get a copy of pointer and both reference the same object in memory

```
var obj1 = new Object();

var obj2 = obj1;
```



Instantiating Built-in Types

- Built-in types:
  - Array – an ordered list of numerically indexed values; new Array() or []
  - Date – a date and time; new Date()
  - Error – a runtime error; new Error()
  - Function – a function; new Function() or function()
  - Object – a generic object; new Object() or {}
  - RegExp – a regular expresion; new RegExp() or //
  - Math - provides properties and methods for manipulating mathematical data. Does not store data

The Logical and Comparison Operators

- Data comparison operators - compare operands and return Boolean values:
  - == - the left and right operands are equal
  - === - the left and right operands are equal and the data types are equal
  - != - the left operand is not equal to the right
  - > - the left operand is greater than the right
  - >= - the left operand is greater than, or equal to, the right
  - < - the left operand is less than the right
  - <= - the left operand is less than, or equal to, the right
- Logical operators - test for more than one condition:
  - && - And
  - || - Or
  - ! - Not

Data Manipulation Operators

- JavaScript has a number of operators you can use to manipulate the data:
  - + - adds numbers or concatenates strings; 1 + 2 = 3 or 'a' + 'b' = 'ab'
  - - - subtracts the second number from first; 2 – 1 = 1
  - * - multiples two numbers; 2 * 2 = 4
  - / - divides the first number by the second; 4 / 2 = 2
  - % - finds the modulus; 3 % 2 = 1
  - -- - decreases the number by 1; var x = 2; x--; (x === 1)
  - ++ - increases the number by 1; var x = 2; x++; (x === 3)

Twitter Bootstrap

- Developed by Mark Otto and Jacob Thornton, released in 2011
- Current The Bootstrap 3 release was announced in 2013
- The Bootstrap 3 is mobile first and using flat design
- In addition to the regular HTML elements, Bootstrap contains other commonly used interface elements (button groups, drop-downs, tabs, navigation, etc.) implemented as CSS classes
- Bootstrap comes with several JS components (Modal, Dropdown, Scrollspy, etc.) in form of jQuery plugins

jQuery

- Was originally released in January 2006
- Designed to simplify the client-side scripting of HTML
- Eliminates cross-browser incompatibilities
- The framework is extensible, new events, elements and methods can be easily added and then reused as a plugin
- Reserves $ as global variable

Readings

- Mark Myers, A Smarter Way to Learn HTML & CSS
- Nicholas C. Zakas, The Principles of Object-Oriented JavaScript
- Russ Ferguson, Christian Heilmann, Beginning JavaScript with DOM Scripting and Ajax: Second Edition
- http://smashingmagazine.com
- http://sitepoint.com
- http://css-tricks.com
- http://html5doctor.com

Resources

- http://caniuse.com
- http://validator.w3.org
- https://jigsaw.w3.org/css-validator/
- https://regex101.com/#javascript
- https://developer.mozilla.org/en-US/docs/Web
- http://getbootstrap.com
- http://api.jquery.com
- http://docs.emmet.io/cheat-sheet/

Home Work

Create a responsive ready CV (at least for mobile and desktop viewports) with Twitter's Bootstrap. CV should have Table of Contents. Its links should navigate into sections (personal information, working history, education etc.).

Home work should be pushed into Github's public repository. The link of repository should be send via email martynas.barzda@nfq.lt.