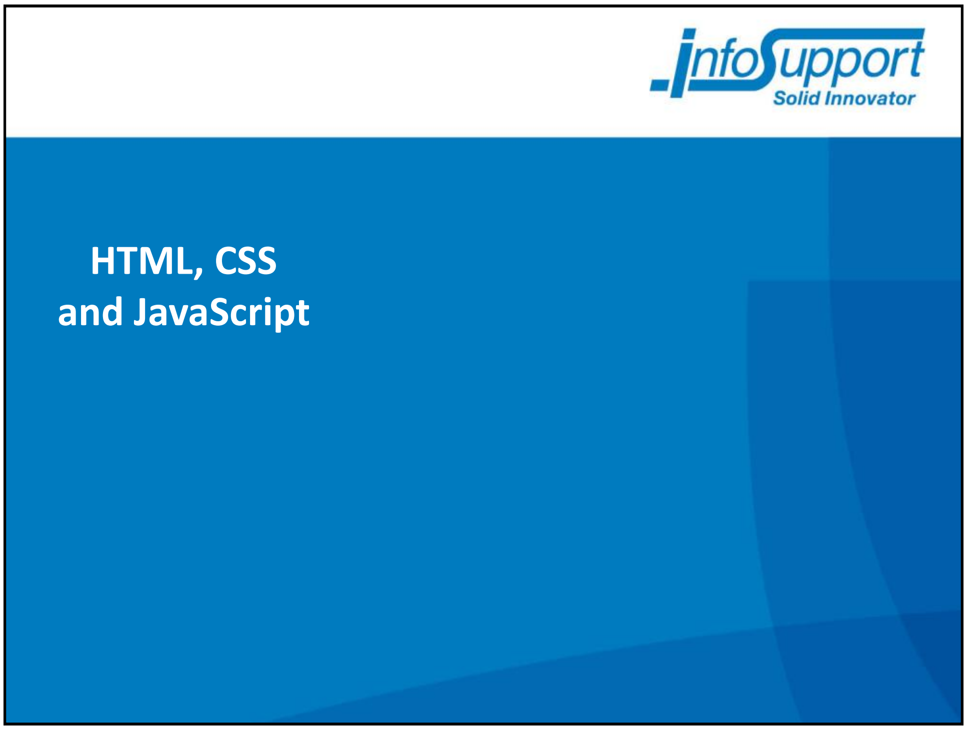
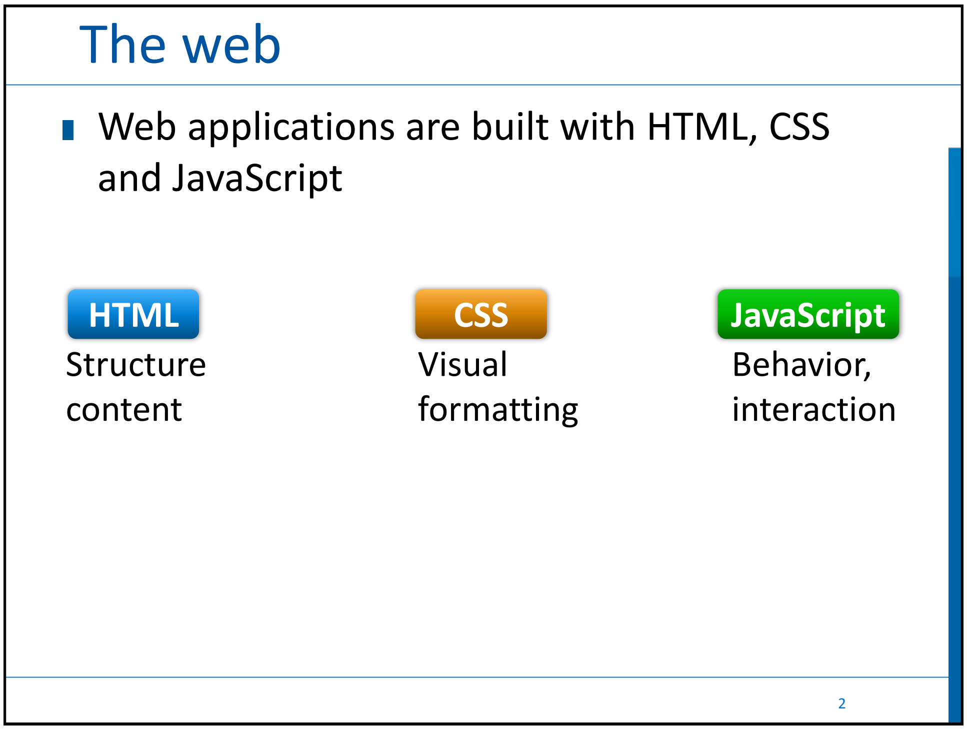


HTML, CSS and JavaScript

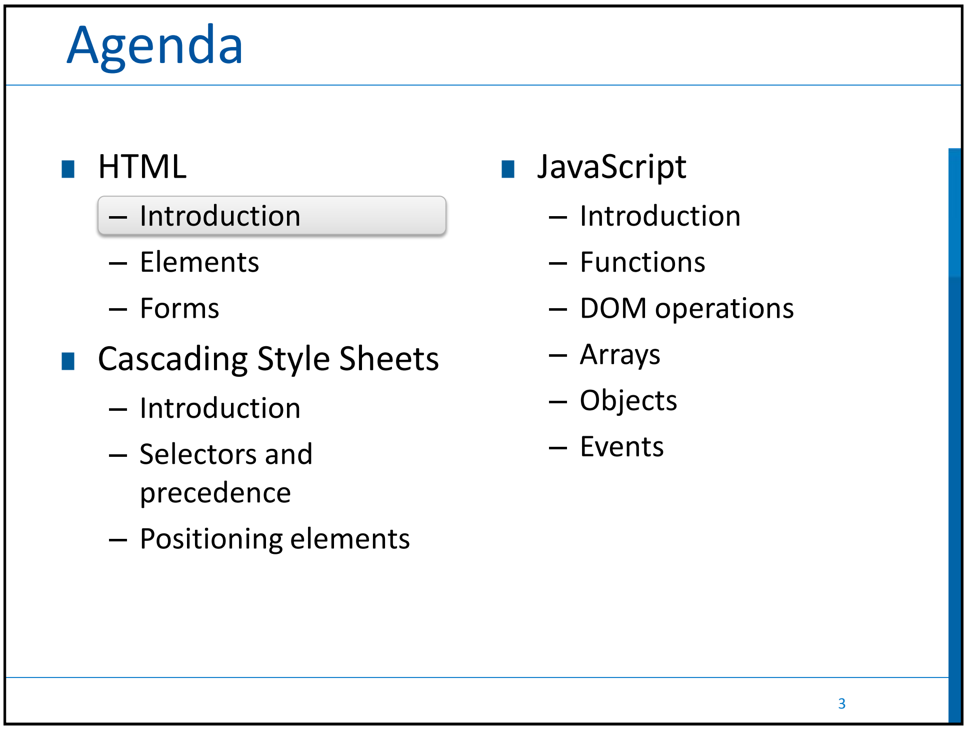


The web

- Web applications are built with HTML, CSS and JavaScript

HTML	CSS	JavaScript
Structure content	Visual formatting	Behavior, interaction

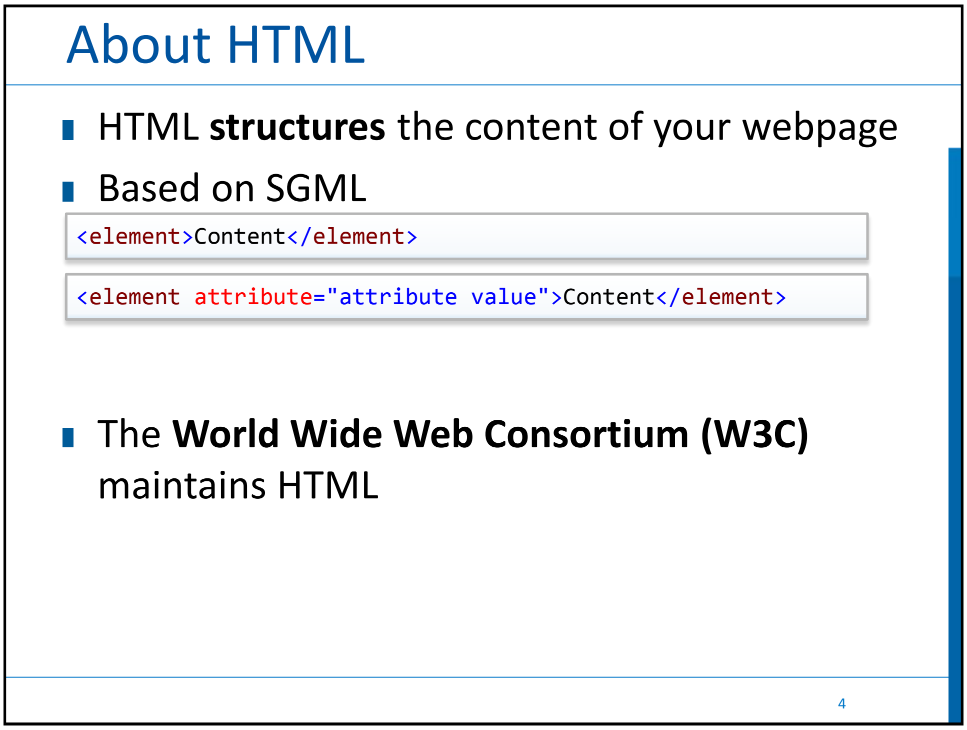
2



Agenda

- HTML
  - Introduction
  - Elements
  - Forms
- Cascading Style Sheets
  - Introduction
  - Selectors and precedence
  - Positioning elements
- JavaScript
  - Introduction
  - Functions
  - DOM operations
  - Arrays
  - Objects
  - Events

3



About HTML

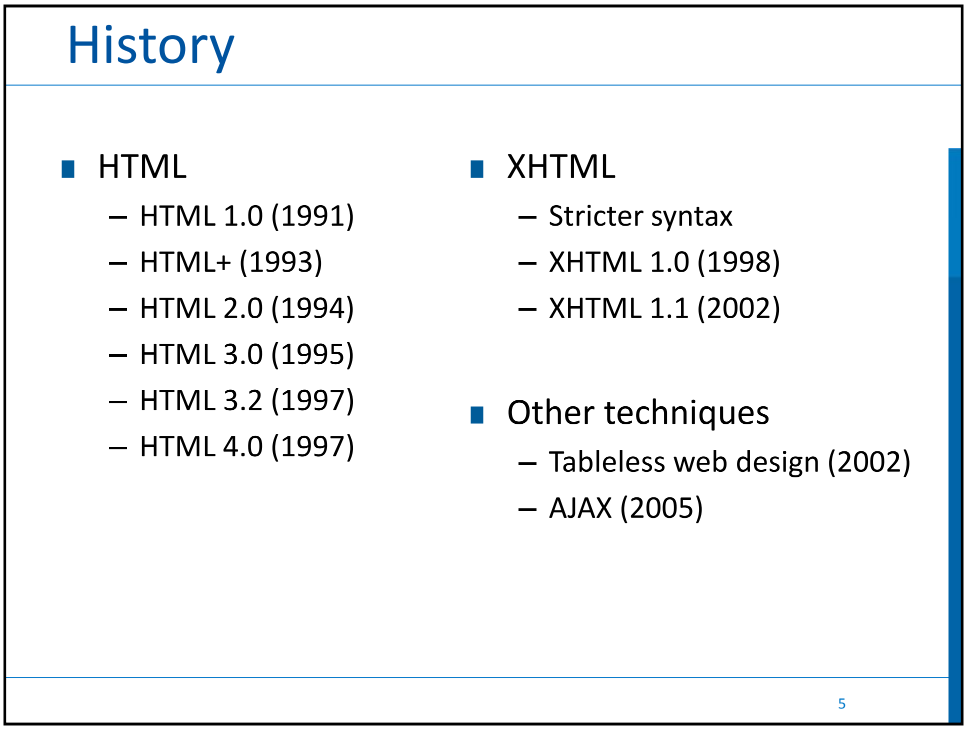
- HTML **structures** the content of your webpage
- Based on SGML

```
<element>Content</element>
```

```
<element attribute="attribute value">Content</element>
```

- The **World Wide Web Consortium (W3C)** maintains HTML

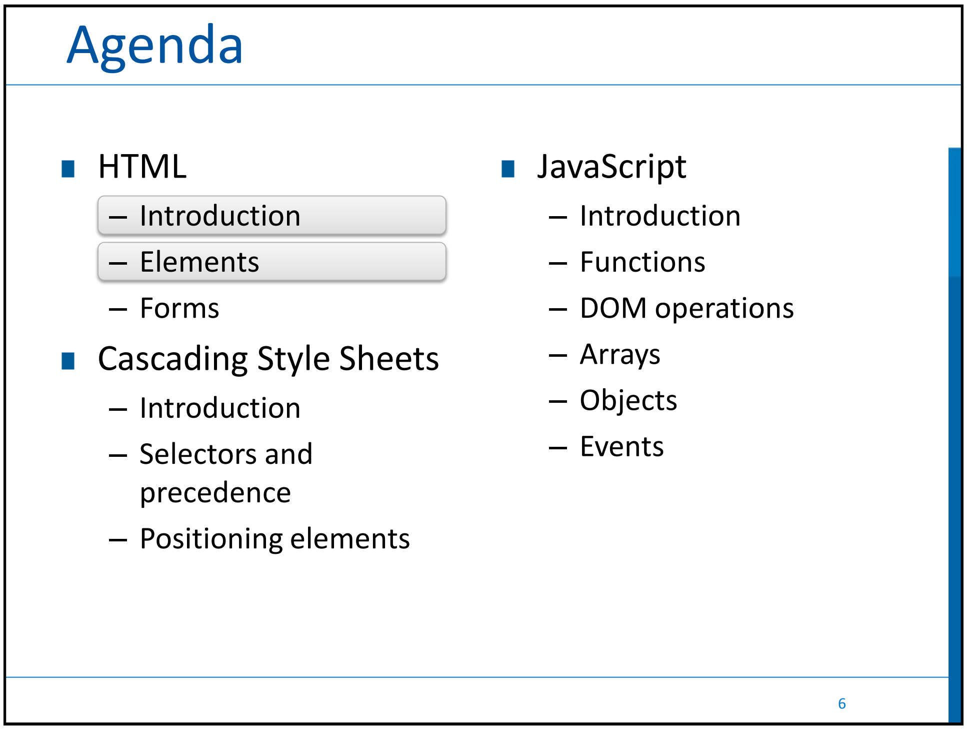
4



History

- HTML
  - HTML 1.0 (1991)
  - HTML+ (1993)
  - HTML 2.0 (1994)
  - HTML 3.0 (1995)
  - HTML 3.2 (1997)
  - HTML 4.0 (1997)
- XHTML
  - Stricter syntax
  - XHTML 1.0 (1998)
  - XHTML 1.1 (2002)
- Other techniques
  - Tableless web design (2002)
  - AJAX (2005)

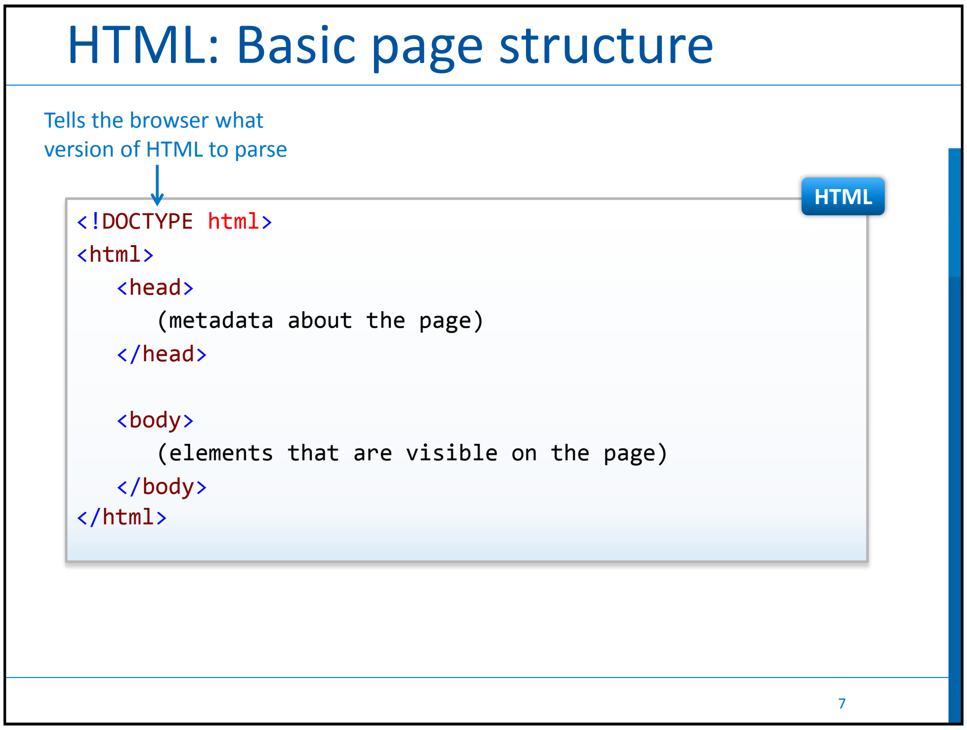
5



Agenda

- HTML
  - Introduction
  - Elements
  - Forms
- Cascading Style Sheets
  - Introduction
  - Selectors and precedence
  - Positioning elements
- JavaScript
  - Introduction
  - Functions
  - DOM operations
  - Arrays
  - Objects
  - Events

6



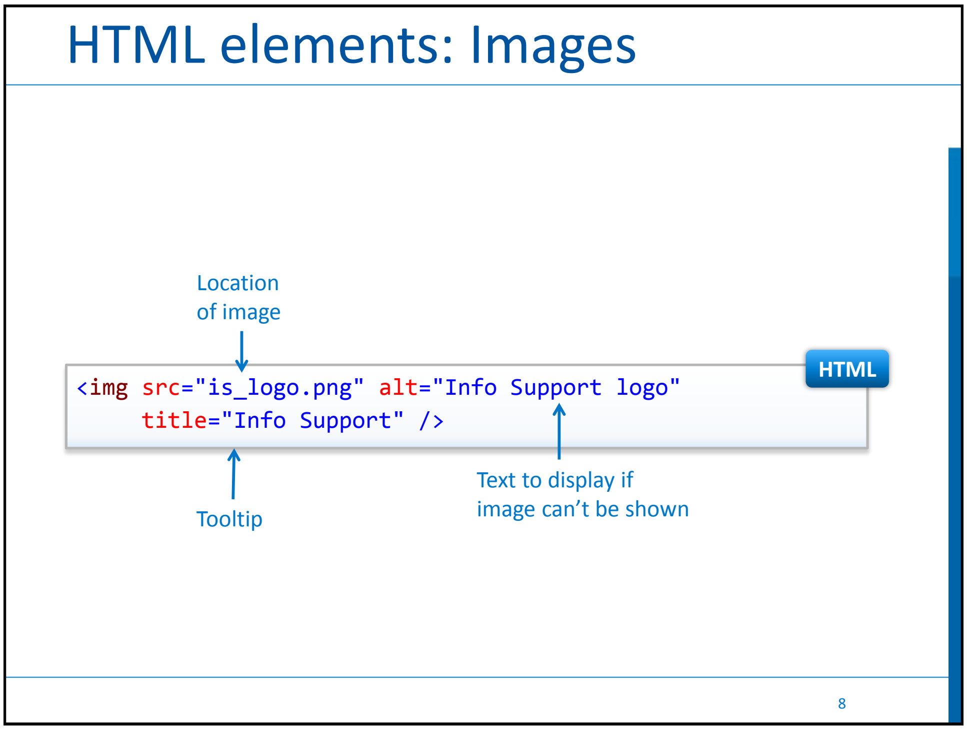
HTML: Basic page structure

Tells the browser what version of HTML to parse

HTML

```
<!DOCTYPE html>
<html>
   <head>
      (metadata about the page)
   </head>

   <body>
      (elements that are visible on the page)
   </body>
</html>
```

7



HTML elements: Images

Location of image

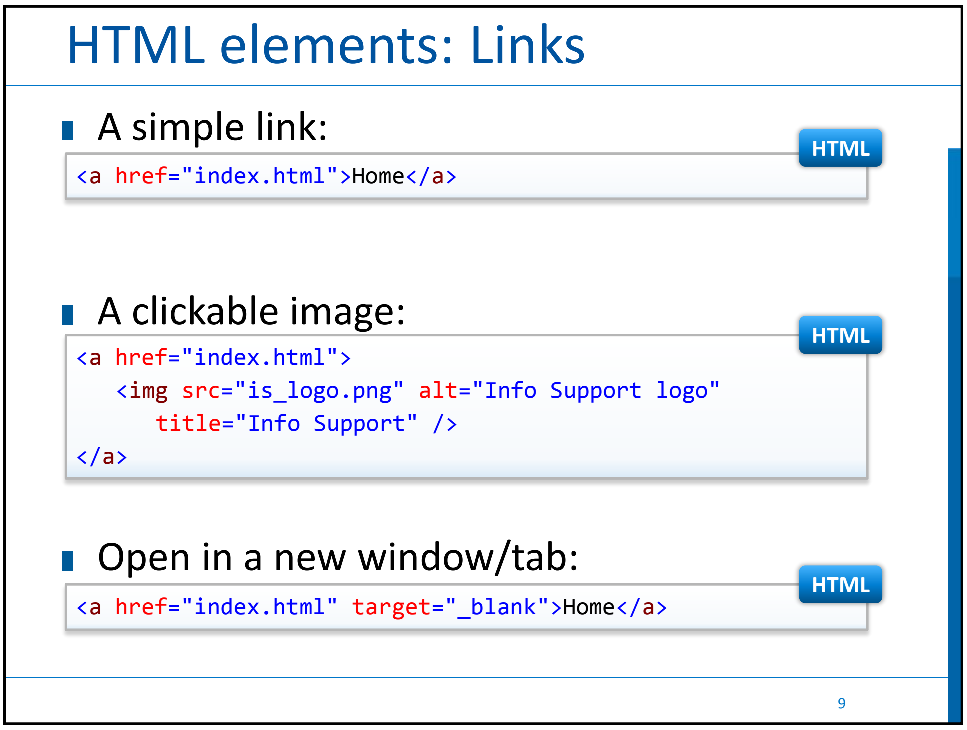
HTML

```
<img src="is_logo.png" alt="Info Support logo"
     title="Info Support" />
```

Tooltip

Text to display if image can't be shown

8



HTML elements: Links

- A simple link:

HTML
```
<a href="index.html">Home</a>
```

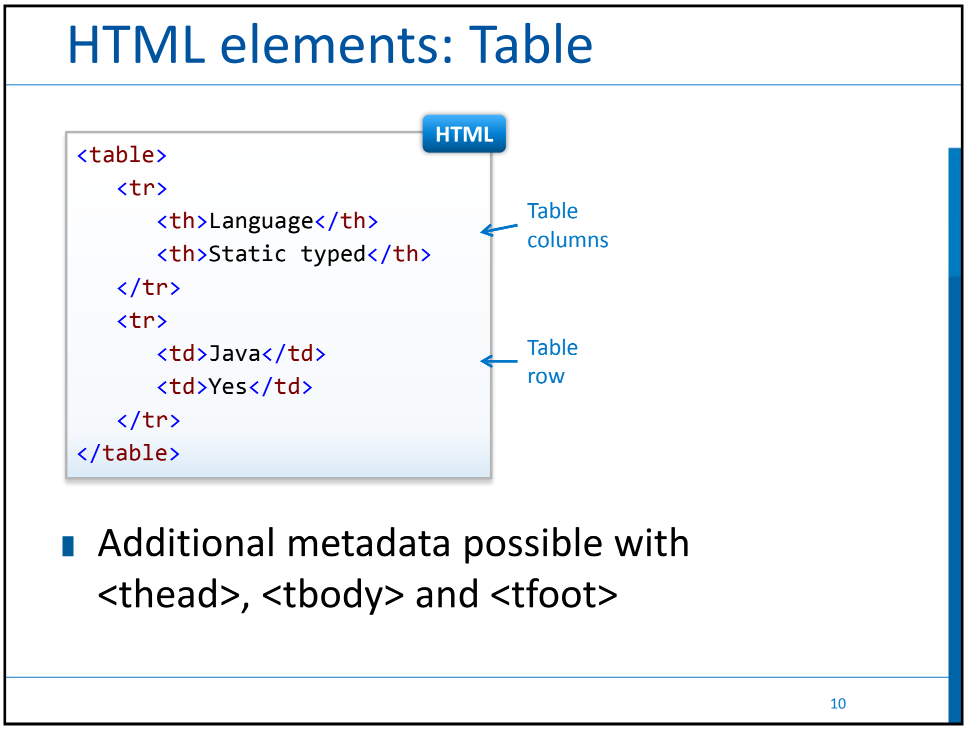
- A clickable image:

HTML
```
<a href="index.html">
   <img src="is_logo.png" alt="Info Support logo"
      title="Info Support" />
</a>
```

- Open in a new window/tab:

HTML
```
<a href="index.html" target="_blank">Home</a>
```

9



HTML elements: Table

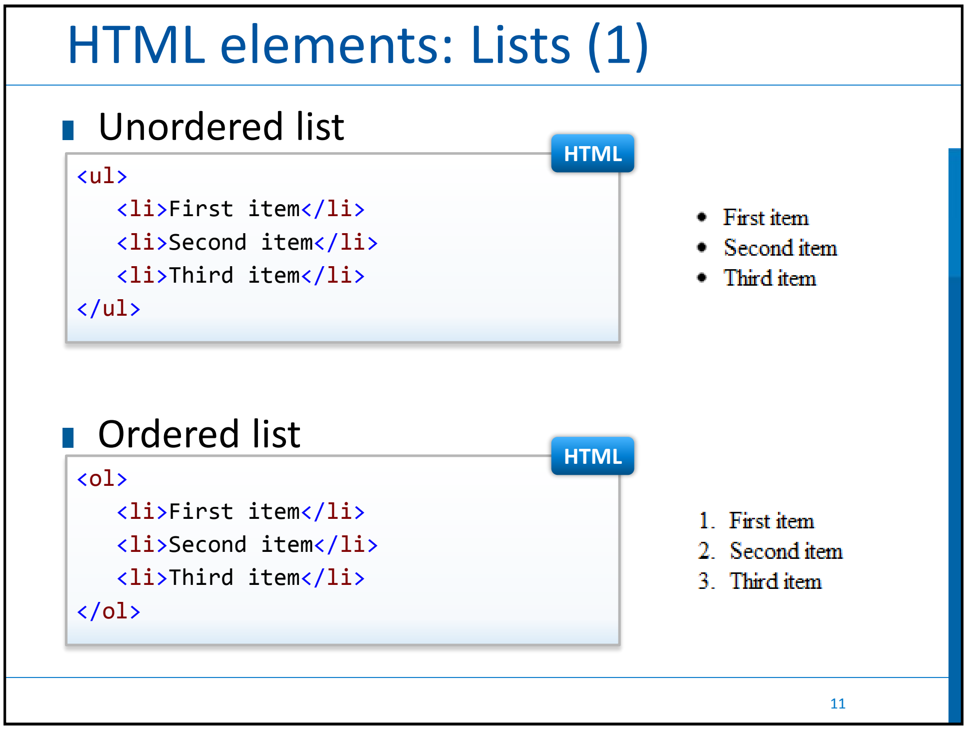
HTML
```
<table>
   <tr>
      <th>Language</th>
      <th>Static typed</th>
   </tr>
   <tr>
      <td>Java</td>
      <td>Yes</td>
   </tr>
</table>
```

Table columns

Table row

- Additional metadata possible with <thead>, <tbody> and <tfoot>

10



HTML elements: Lists (1)

- Unordered list

HTML
```
<ul>
   <li>First item</li>
   <li>Second item</li>
   <li>Third item</li>
</ul>
```

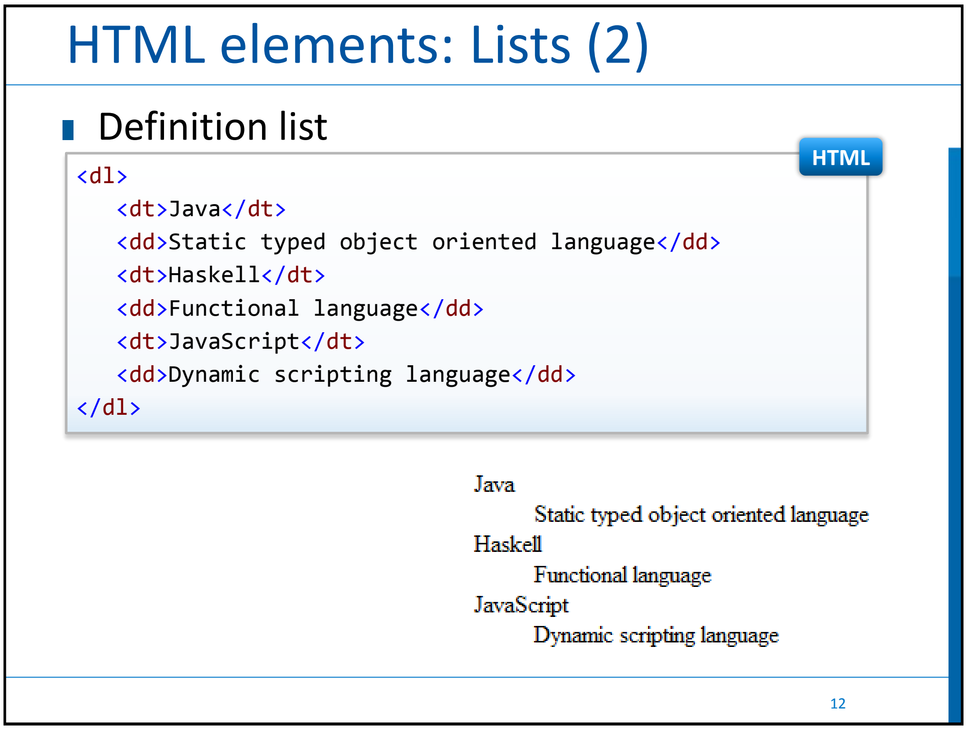
- First item
- Second item
- Third item

- Ordered list

HTML
```
<ol>
   <li>First item</li>
   <li>Second item</li>
   <li>Third item</li>
</ol>
```

1. First item
2. Second item
3. Third item

11



HTML elements: Lists (2)

- Definition list

HTML
```
<dl>
   <dt>Java</dt>
   <dd>Static typed object oriented language</dd>
   <dt>Haskell</dt>
   <dd>Functional language</dd>
   <dt>JavaScript</dt>
   <dd>Dynamic scripting language</dd>
</dl>
```

Java
    Static typed object oriented language
Haskell
    Functional language
JavaScript
    Dynamic scripting language

12

HTML elements: Frames

- Once used to represent a part of a page

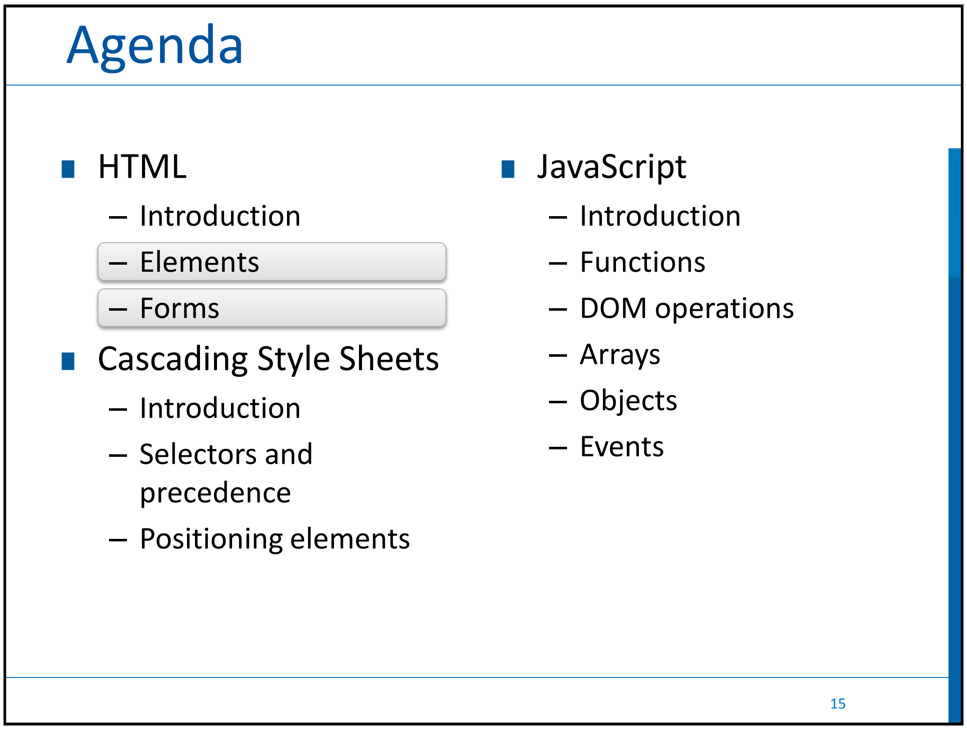
- Come with issues:
  - Broken bookmarks
  - Invisible navigation
  - Printing problems
  - Search engines reference incomplete documents

HTML elements: Div and Span

- **Meaningless** elements
- Highly useful for applying styles

- Difference between `div` and `span`
  - `<div>` is a block element
  - `<span>` is an inline element

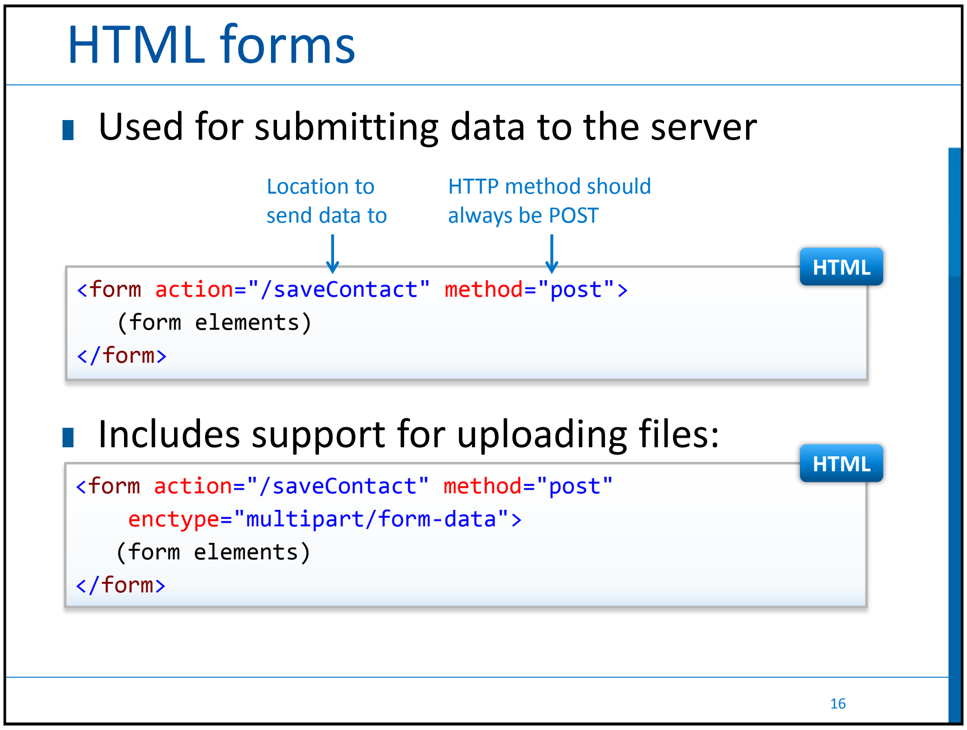
HTML
```html
<div>(more block elements)</div>
<span>(text or other inline elements)</span>
```



Agenda

- HTML
  - Introduction
  - Elements
  - Forms
- Cascading Style Sheets
  - Introduction
  - Selectors and precedence
  - Positioning elements
- JavaScript
  - Introduction
  - Functions
  - DOM operations
  - Arrays
  - Objects
  - Events

15



HTML forms

- Used for submitting data to the server

Location to send data to

HTTP method should always be POST

HTML

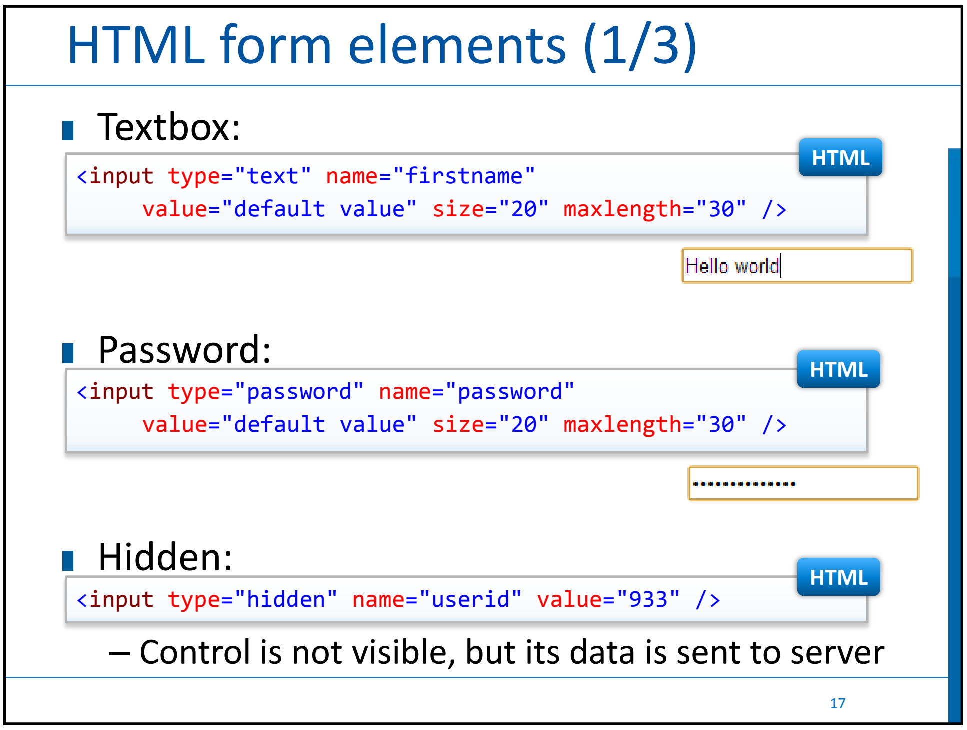
```html
<form action="/saveContact" method="post">
   (form elements)
</form>
```

- Includes support for uploading files:

HTML

```html
<form action="/saveContact" method="post"
    enctype="multipart/form-data">
   (form elements)
</form>
```

16



HTML form elements (1/3)

- Textbox:

HTML
```
<input type="text" name="firstname"
     value="default value" size="20" maxlength="30" />
```

Hello world

- Password:

HTML
```
<input type="password" name="password"
     value="default value" size="20" maxlength="30" />
```

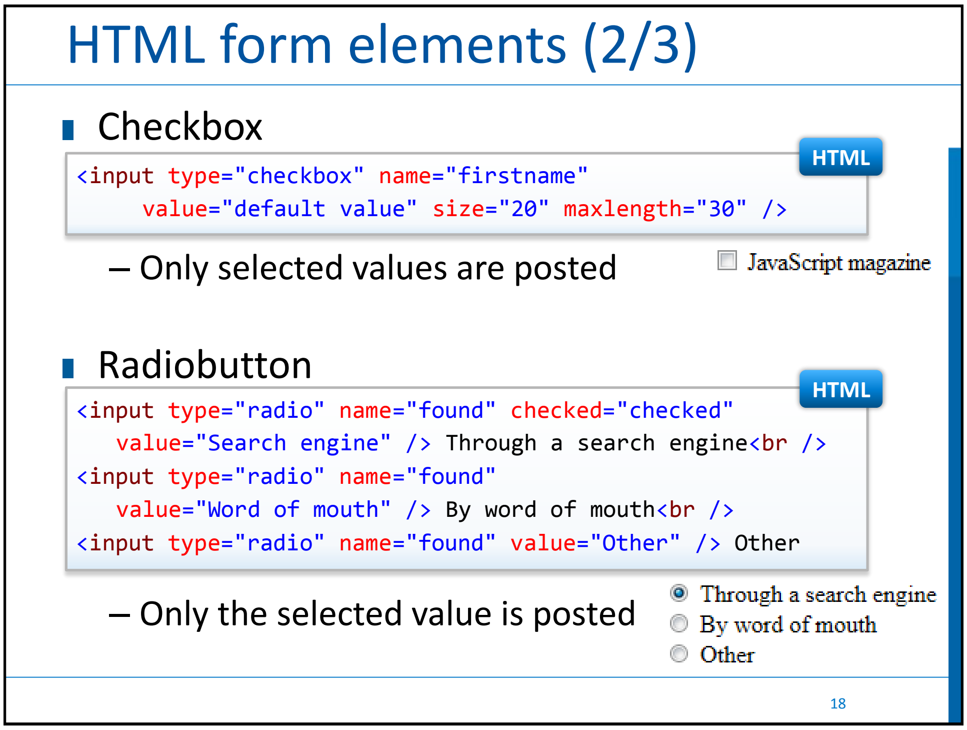
•••••••••••••

- Hidden:

HTML
```
<input type="hidden" name="userid" value="933" />
```

  – Control is not visible, but its data is sent to server

17



HTML form elements (2/3)

- Checkbox

HTML
```
<input type="checkbox" name="firstname"
     value="default value" size="20" maxlength="30" />
```

  – Only selected values are posted

☐ JavaScript magazine

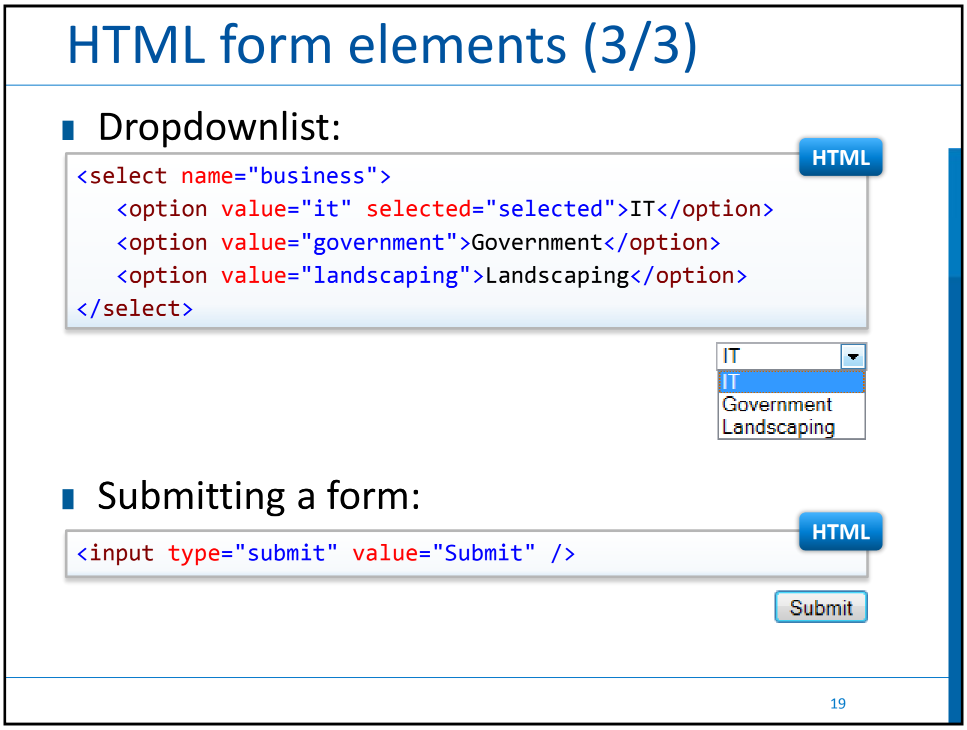
- Radiobutton

HTML
```
<input type="radio" name="found" checked="checked"
   value="Search engine" /> Through a search engine<br />
<input type="radio" name="found"
   value="Word of mouth" /> By word of mouth<br />
<input type="radio" name="found" value="Other" /> Other
```

  – Only the selected value is posted

● Through a search engine
○ By word of mouth
○ Other

18


IT
IT
Government
Landscaping
Submit

HTML form elements (3/3)

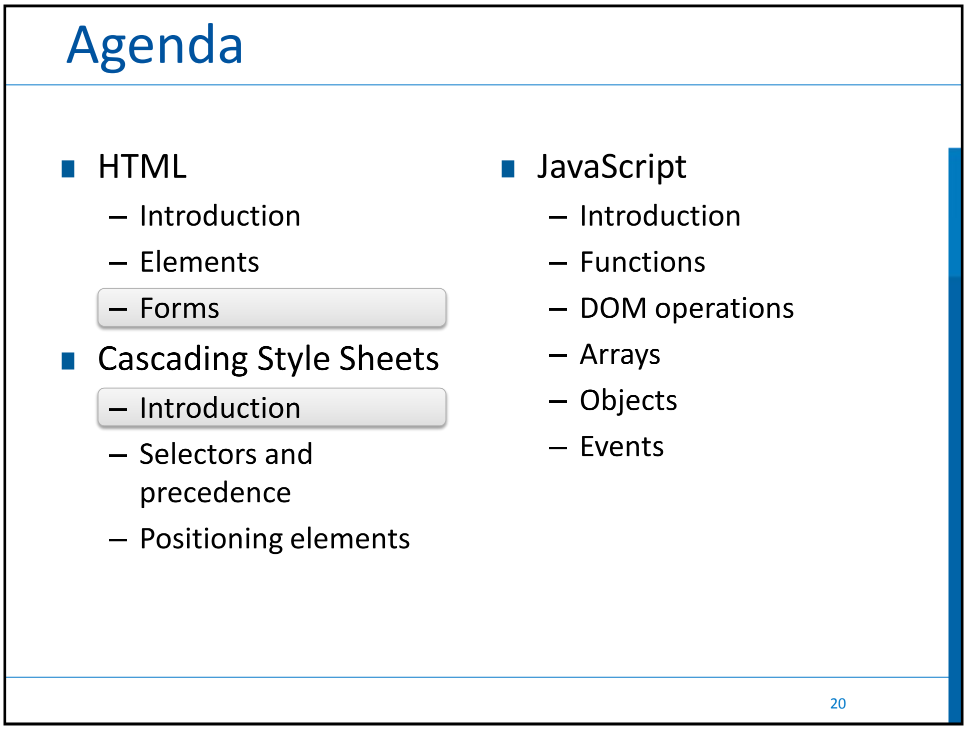
- Dropdownlist:

HTML
```
<select name="business">
   <option value="it" selected="selected">IT</option>
   <option value="government">Government</option>
   <option value="landscaping">Landscaping</option>
</select>
```

- Submitting a form:

HTML
```
<input type="submit" value="Submit" />
```

19



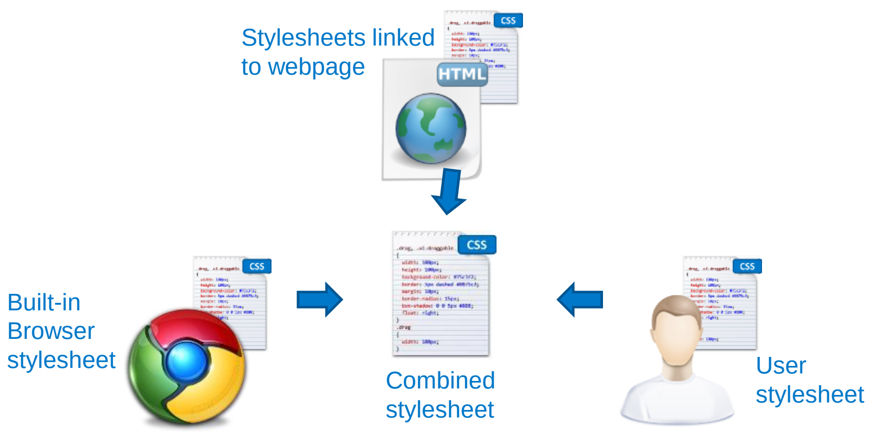
Agenda

- HTML
  - Introduction
  - Elements
  - Forms
- Cascading Style Sheets
  - Introduction
  - Selectors and precedence
  - Positioning elements
- JavaScript
  - Introduction
  - Functions
  - DOM operations
  - Arrays
  - Objects
  - Events

20

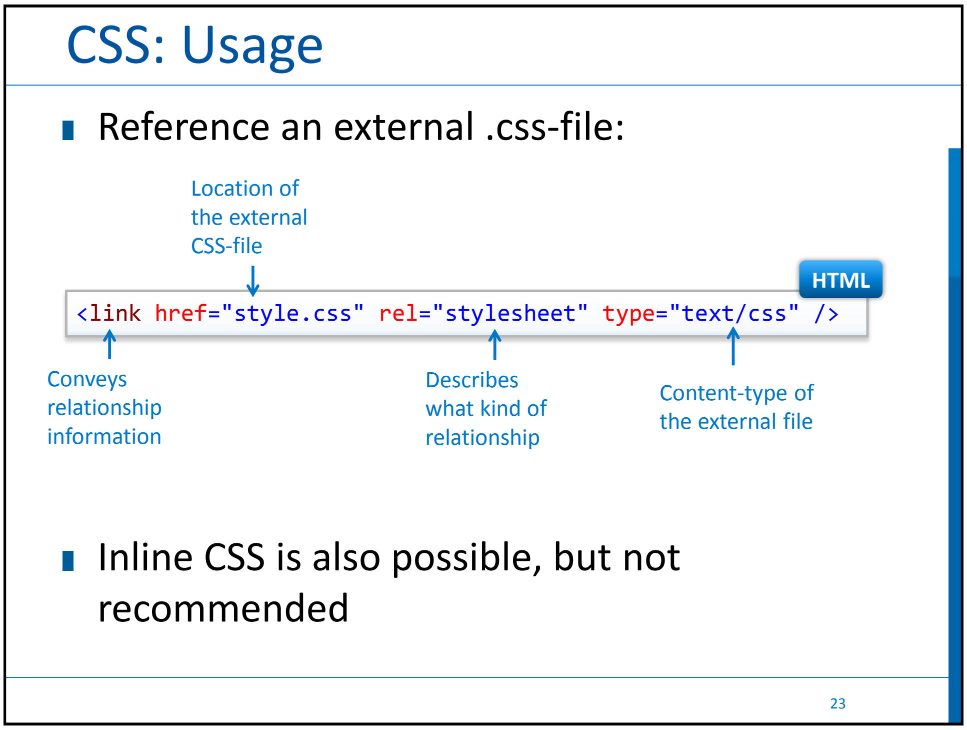
Cascading Style Sheets

- Used for styling your HTML elements



CSS: History

- CSS1 (1996)
  - Basic styling support
- CSS2 (1998)
  - Better positioning support
  - Targeting different media
- CSS2.1 (2011)
  - Contains Fixes for CSS2
  - The current standard
- CSS3
  - Divided into modules (several already approved)
  - Support for:
    - Transforming text
    - Animations
    - Shadows
    - Rounded corners
    - More



CSS: Usage

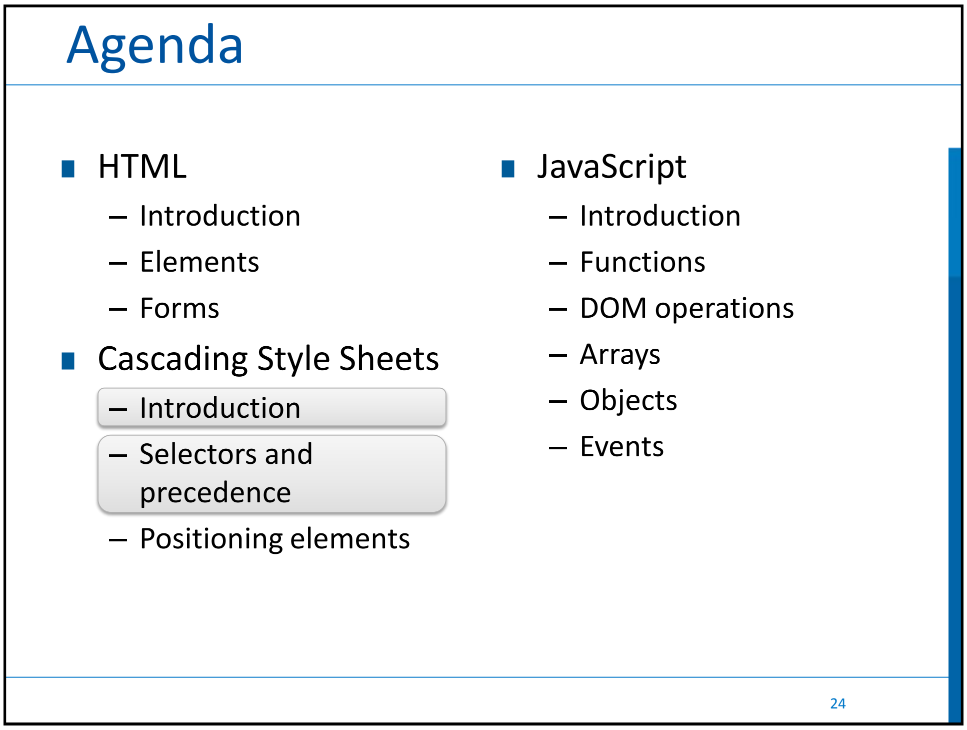
- Reference an external .css-file:

```html
<link href="style.css" rel="stylesheet" type="text/css" />
```

HTML
Location of the external CSS-file
Conveys relationship information
Describes what kind of relationship
Content-type of the external file

- Inline CSS is also possible, but not recommended

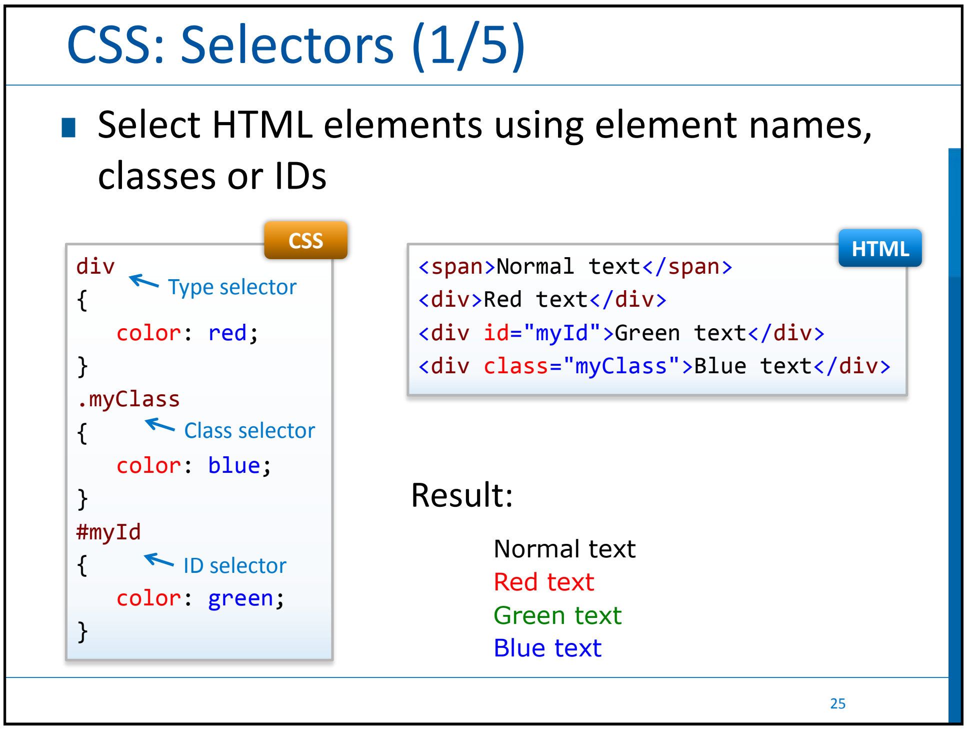
23



Agenda

- HTML
  - Introduction
  - Elements
  - Forms
- Cascading Style Sheets
  - Introduction
  - Selectors and precedence
  - Positioning elements
- JavaScript
  - Introduction
  - Functions
  - DOM operations
  - Arrays
  - Objects
  - Events

24



CSS: Selectors (1/5)

- Select HTML elements using element names, classes or IDs

CSS

```
div          ← Type selector
{
   color: red;
}
.myClass
{            ← Class selector
   color: blue;
}
#myId
{            ← ID selector
   color: green;
}
```

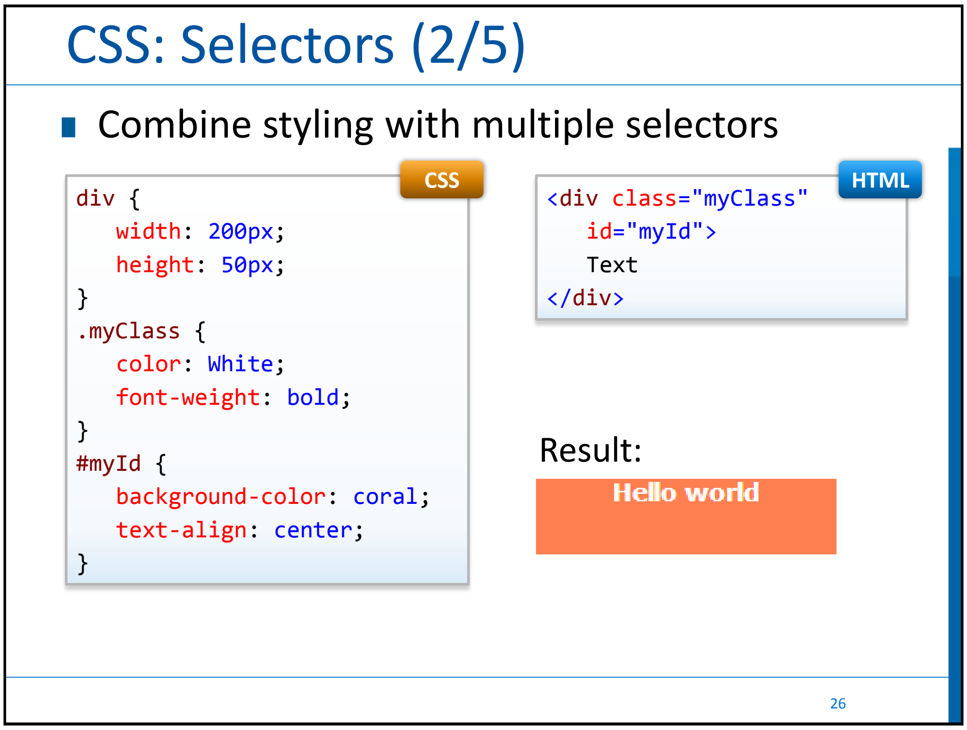
HTML

```
<span>Normal text</span>
<div>Red text</div>
<div id="myId">Green text</div>
<div class="myClass">Blue text</div>
```

Result:

Normal text
Red text
Green text
Blue text

25



CSS: Selectors (2/5)

- Combine styling with multiple selectors

CSS

```
div {
   width: 200px;
   height: 50px;
}
.myClass {
   color: White;
   font-weight: bold;
}
#myId {
   background-color: coral;
   text-align: center;
}
```

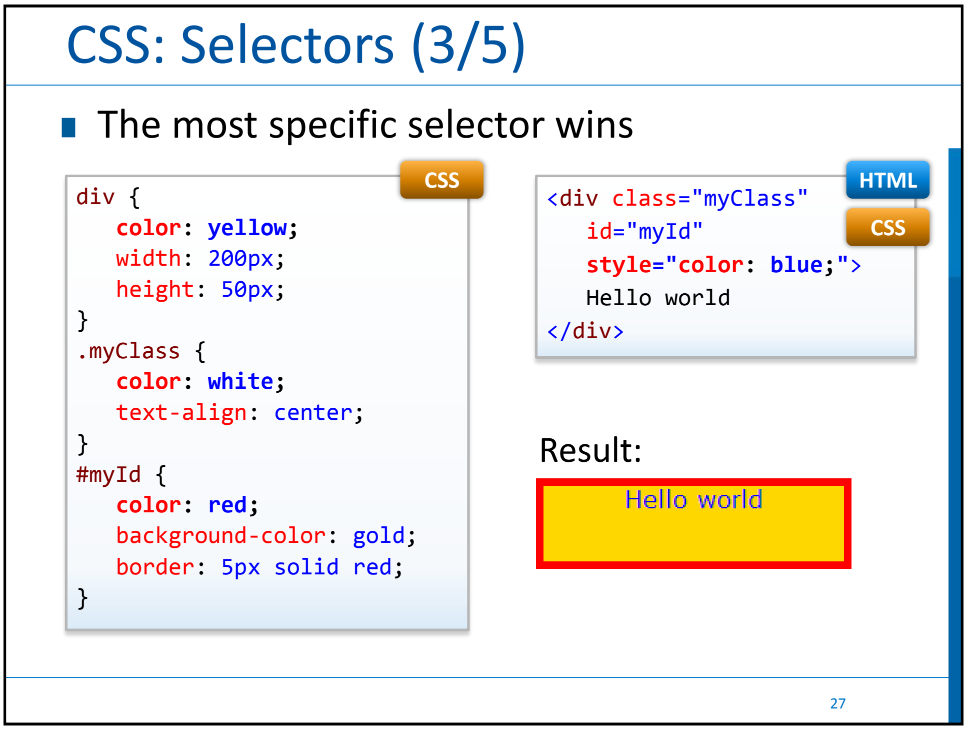
HTML

```
<div class="myClass"
   id="myId">
   Text
</div>
```

Result:

Hello world

26



CSS: Selectors (3/5)

- The most specific selector wins

CSS

```css
div {
    color: yellow;
    width: 200px;
    height: 50px;
}
.myClass {
    color: white;
    text-align: center;
}
#myId {
    color: red;
    background-color: gold;
    border: 5px solid red;
}
```

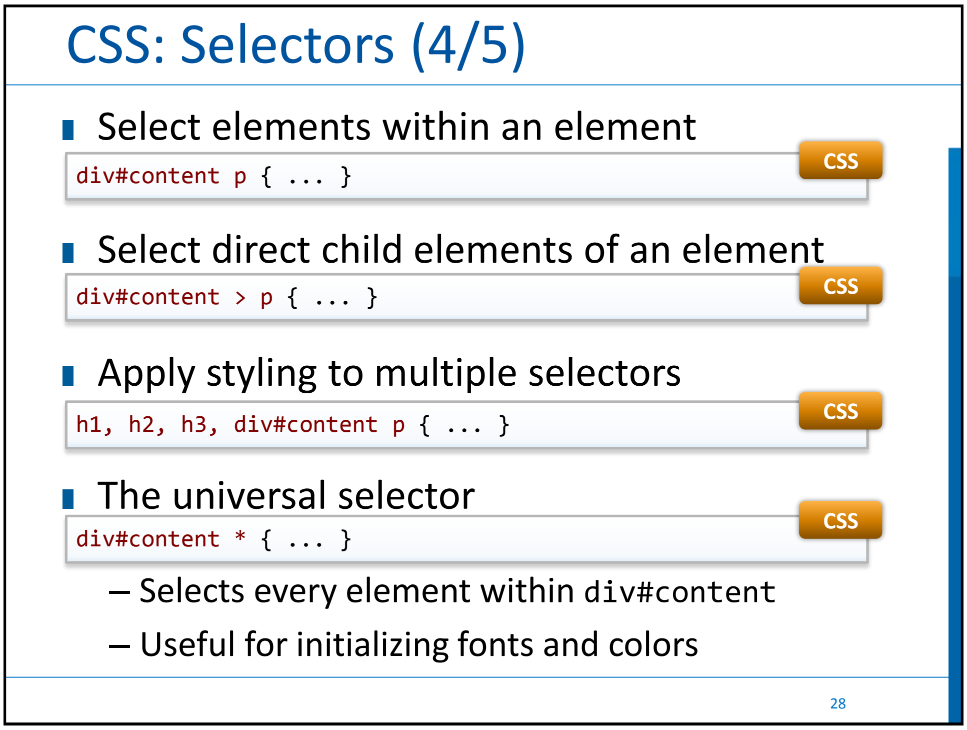
HTML CSS

```html
<div class="myClass"
    id="myId"
    style="color: blue;">
    Hello world
</div>
```

Result:

Hello world

27



CSS: Selectors (4/5)

- Select elements within an element

CSS

```css
div#content p { ... }
```

- Select direct child elements of an element

CSS

```css
div#content > p { ... }
```

- Apply styling to multiple selectors

CSS

```css
h1, h2, h3, div#content p { ... }
```

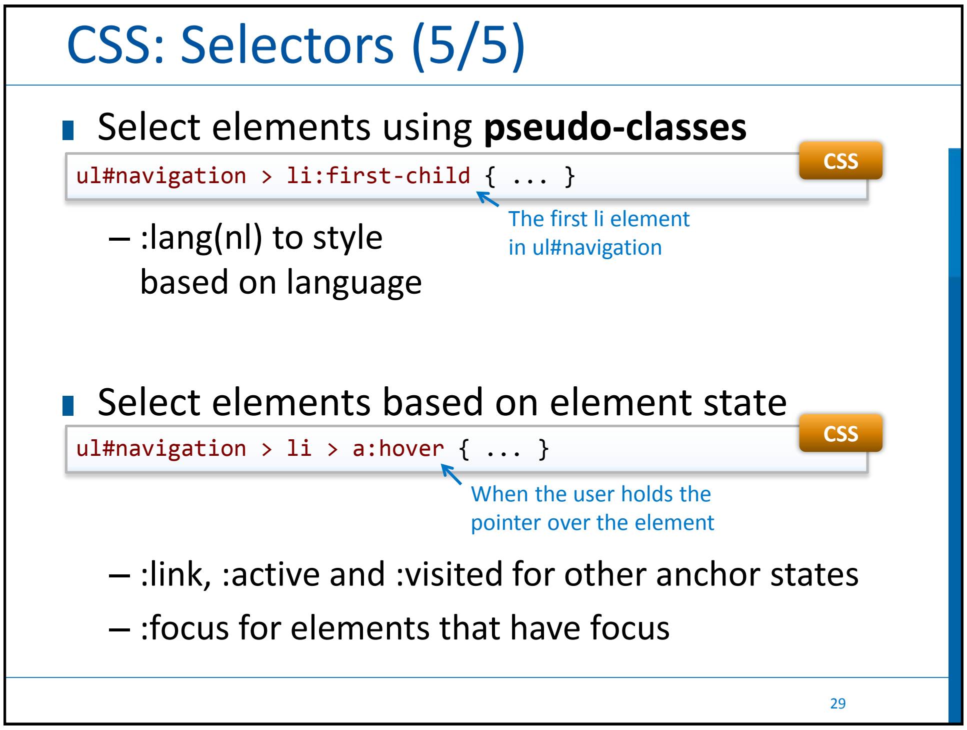
- The universal selector

CSS

```css
div#content * { ... }
```

  - Selects every element within `div#content`
  - Useful for initializing fonts and colors

28



CSS: Selectors (5/5)

- Select elements using **pseudo-classes**

```
ul#navigation > li:first-child { ... }
```

CSS

The first li element in ul#navigation

  - :lang(nl) to style based on language

- Select elements based on element state

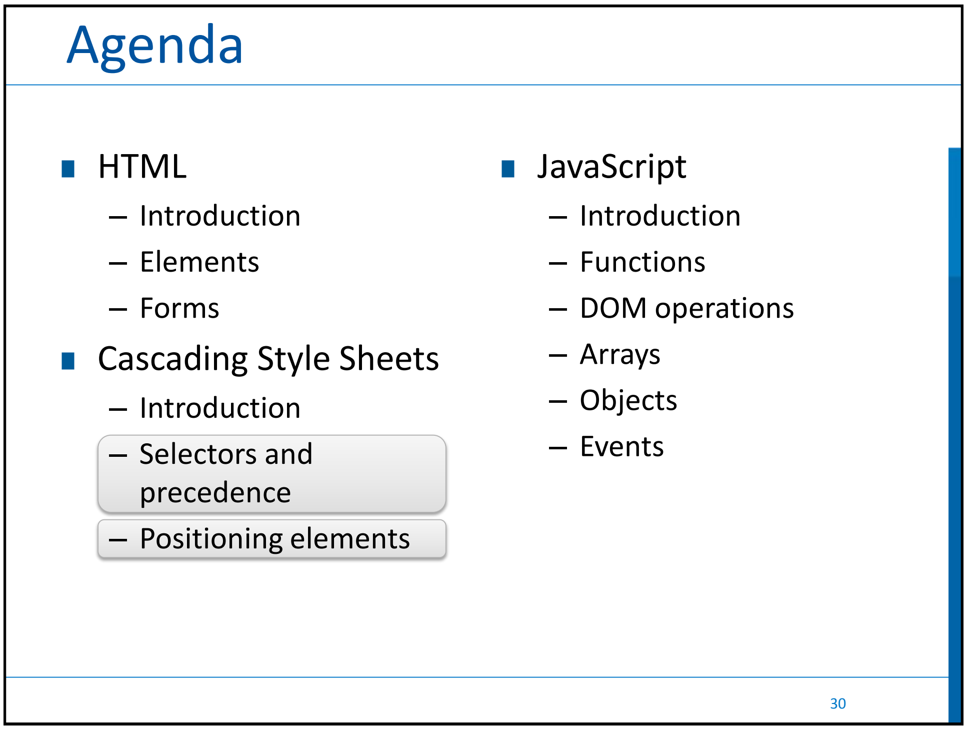
```
ul#navigation > li > a:hover { ... }
```

CSS

When the user holds the pointer over the element

  - :link, :active and :visited for other anchor states
  - :focus for elements that have focus

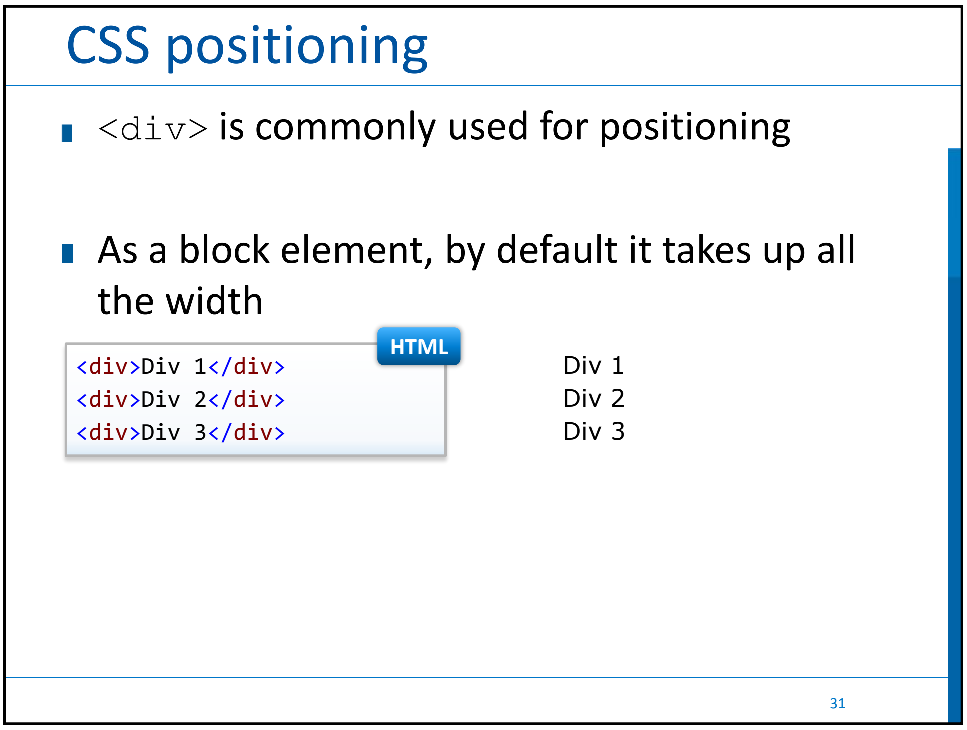
29



Agenda

- HTML
  - Introduction
  - Elements
  - Forms
- Cascading Style Sheets
  - Introduction
  - Selectors and precedence
  - Positioning elements
- JavaScript
  - Introduction
  - Functions
  - DOM operations
  - Arrays
  - Objects
  - Events

30



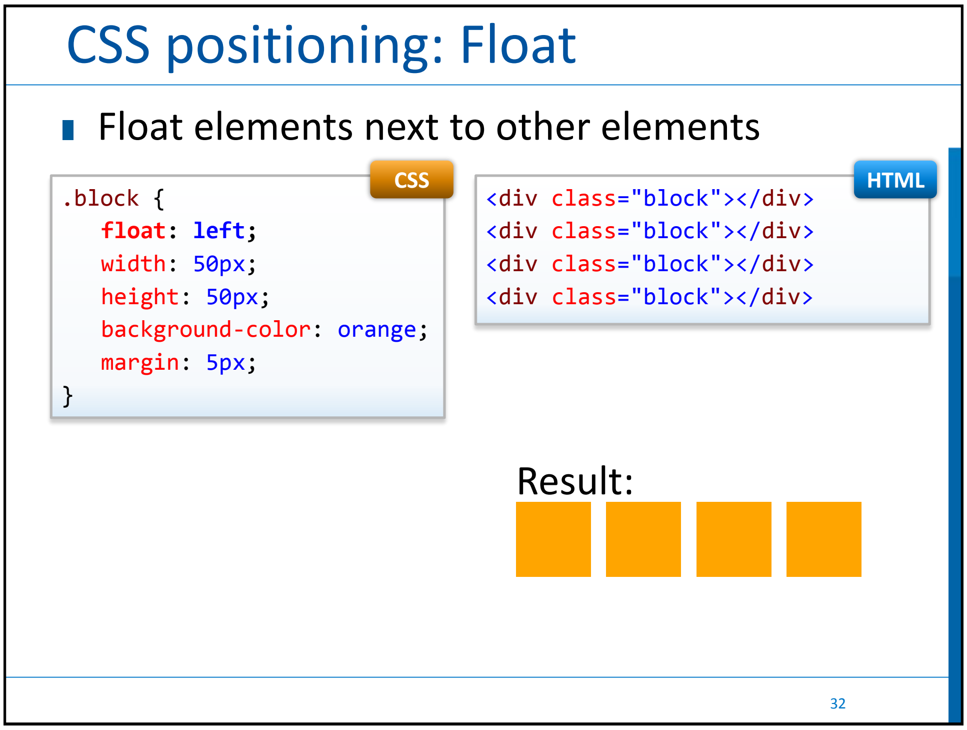
CSS positioning

- `<div>` is commonly used for positioning
- As a block element, by default it takes up all the width

HTML
```html
<div>Div 1</div>
<div>Div 2</div>
<div>Div 3</div>
```

Div 1
Div 2
Div 3

31



CSS positioning: Float

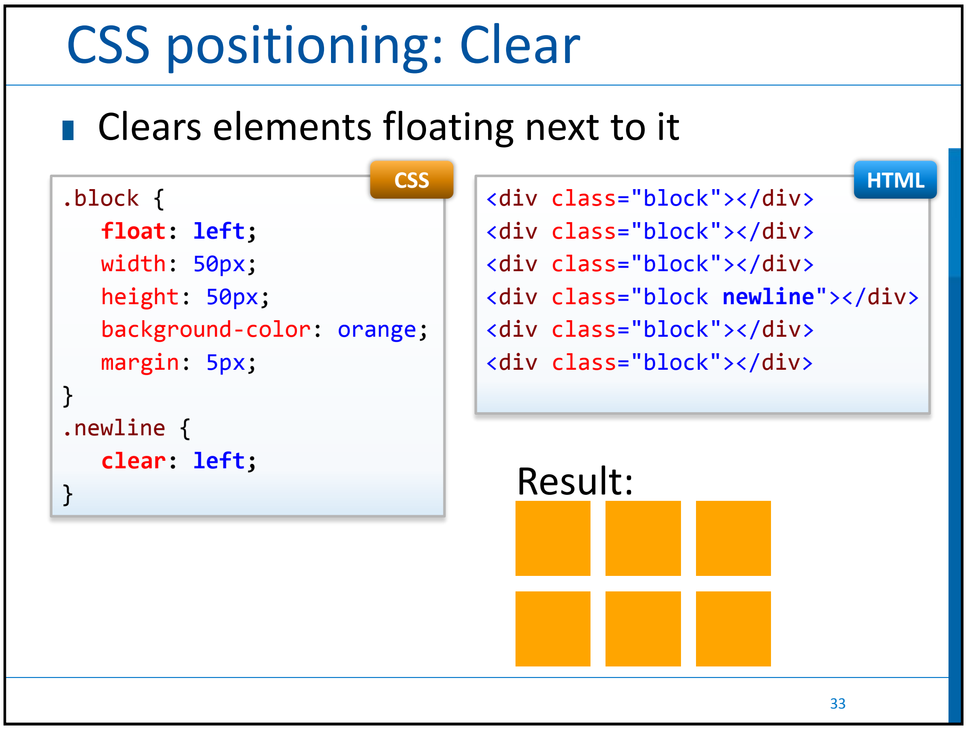
- Float elements next to other elements

CSS
```css
.block {
    float: left;
    width: 50px;
    height: 50px;
    background-color: orange;
    margin: 5px;
}
```

HTML
```html
<div class="block"></div>
<div class="block"></div>
<div class="block"></div>
<div class="block"></div>
```

Result:

32



CSS positioning: Clear

- Clears elements floating next to it

CSS

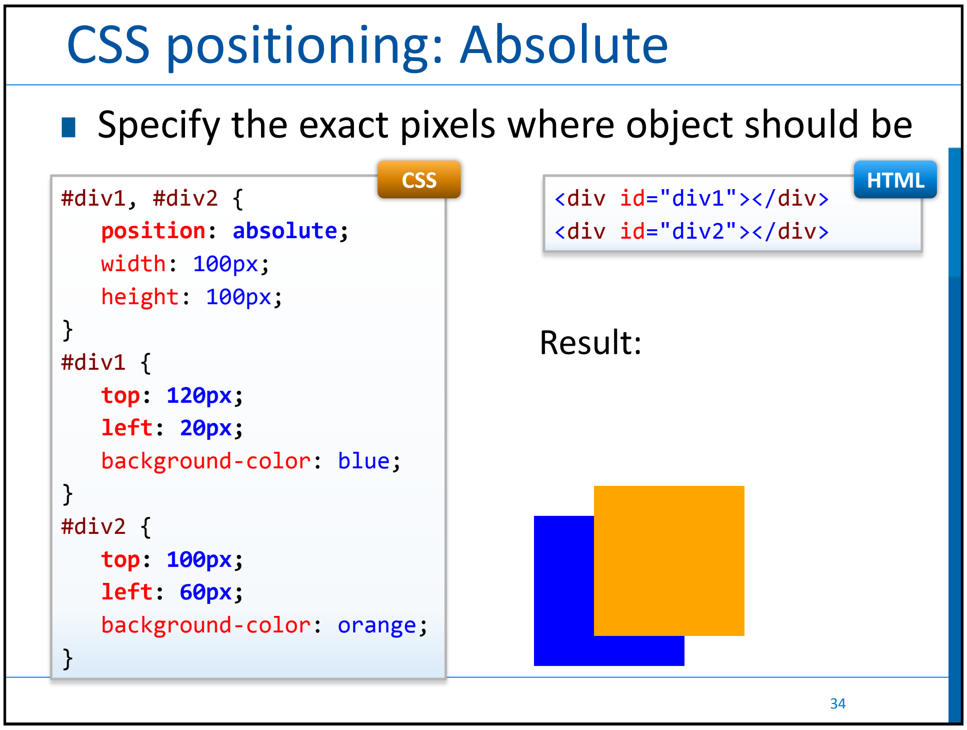
```css
.block {
   float: left;
   width: 50px;
   height: 50px;
   background-color: orange;
   margin: 5px;
}
.newline {
   clear: left;
}
```

HTML

```html
<div class="block"></div>
<div class="block"></div>
<div class="block"></div>
<div class="block newline"></div>
<div class="block"></div>
<div class="block"></div>
```

Result:

33



CSS positioning: Absolute

- Specify the exact pixels where object should be

CSS

```css
#div1, #div2 {
   position: absolute;
   width: 100px;
   height: 100px;
}
#div1 {
   top: 120px;
   left: 20px;
   background-color: blue;
}
#div2 {
   top: 100px;
   left: 60px;
   background-color: orange;
}
```

HTML

```html
<div id="div1"></div>
<div id="div2"></div>
```

Result:

34

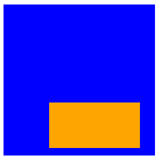
CSS positioning: Relative

- Position absolute within a parent element

CSS
```
#container {
    position: relative;
    background-color: blue;
    width: 100px;
    height: 100px;
}
#some-div {
    position: absolute;
    bottom: 5px;
    right: 10px;
    width: 60px;
    height: 30px;
    background-color: orange;
}
```

HTML
```
<div id="container">
   <div id="some-div"></div>
</div>
```

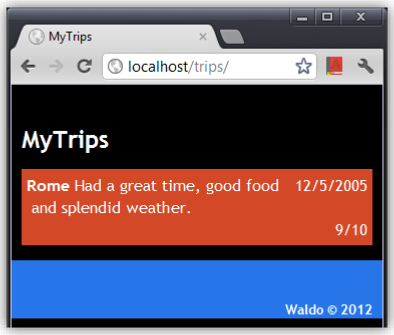
Result:



35

Lab: Setting up the Trips page

- Exercise 1: Basic structure
- Exercise 2: Styling

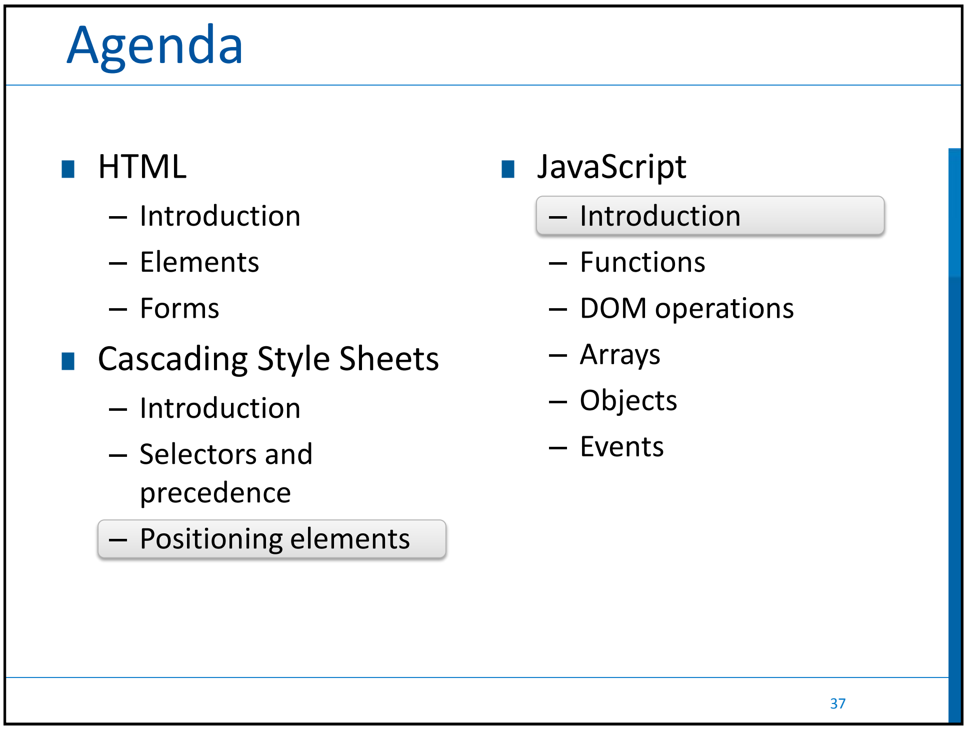


MyTrips

Rome Had a great time, good food and splendid weather. 12/5/2005 9/10

Waldo © 2012

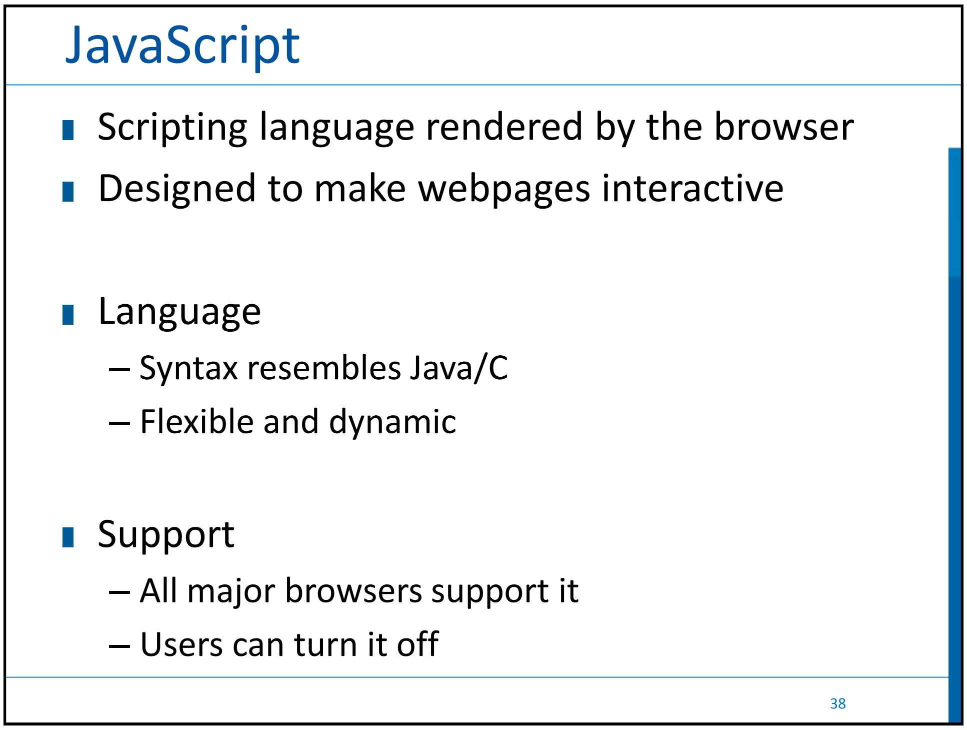
36



Agenda

- HTML
  - Introduction
  - Elements
  - Forms
- Cascading Style Sheets
  - Introduction
  - Selectors and precedence
  - Positioning elements
- JavaScript
  - Introduction
  - Functions
  - DOM operations
  - Arrays
  - Objects
  - Events

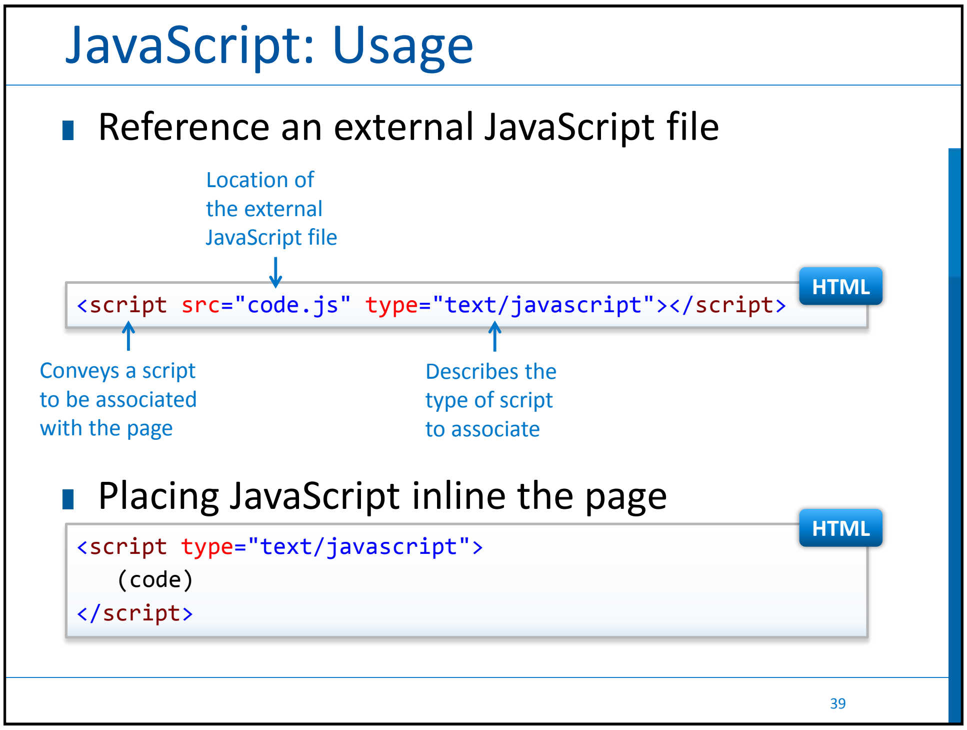
37



JavaScript

- Scripting language rendered by the browser
- Designed to make webpages interactive
- Language
  - Syntax resembles Java/C
  - Flexible and dynamic
- Support
  - All major browsers support it
  - Users can turn it off

38



JavaScript: Usage

- Reference an external JavaScript file

Location of the external JavaScript file

HTML

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

Conveys a script to be associated with the page

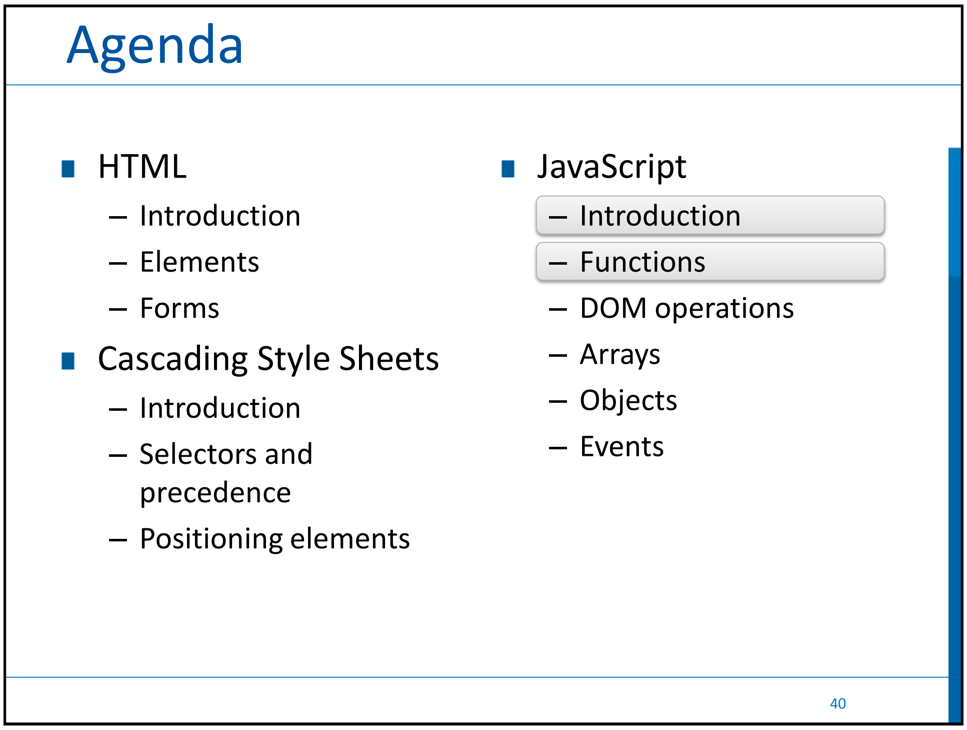
Describes the type of script to associate

- Placing JavaScript inline the page

HTML

```html
<script type="text/javascript">
   (code)
</script>
```

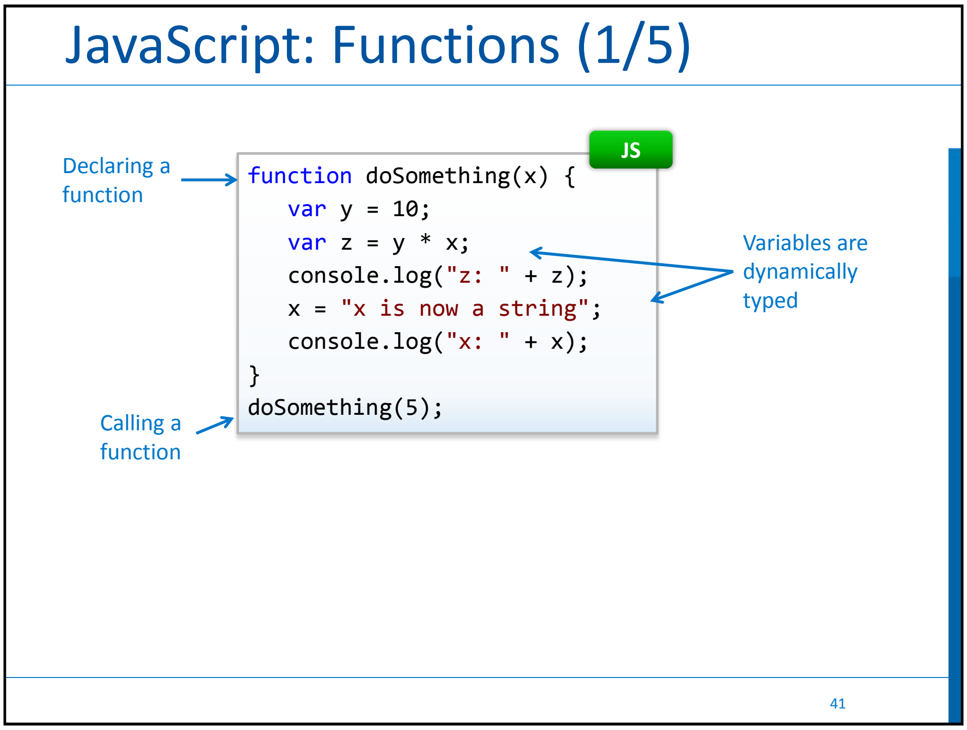
39



Agenda

- HTML
  - Introduction
  - Elements
  - Forms
- Cascading Style Sheets
  - Introduction
  - Selectors and precedence
  - Positioning elements
- JavaScript
  - Introduction
  - Functions
  - DOM operations
  - Arrays
  - Objects
  - Events

40



JavaScript: Functions (1/5)

Declaring a function

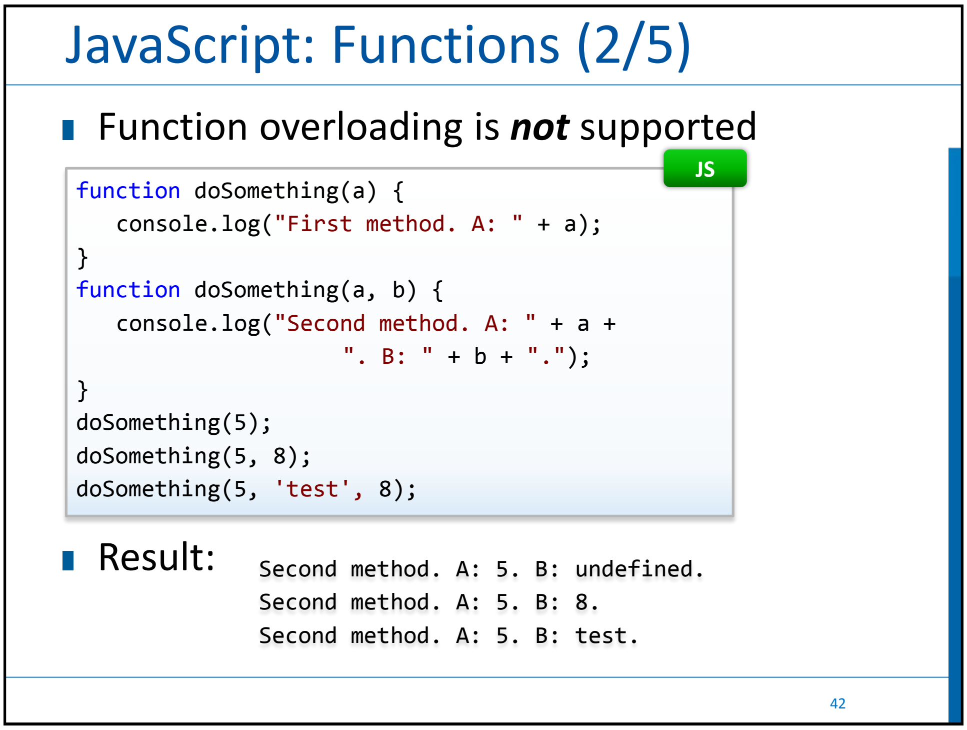
JS

```
function doSomething(x) {
   var y = 10;
   var z = y * x;
   console.log("z: " + z);
   x = "x is now a string";
   console.log("x: " + x);
}
doSomething(5);
```

Variables are dynamically typed

Calling a function

41



JavaScript: Functions (2/5)

- Function overloading is ***not*** supported

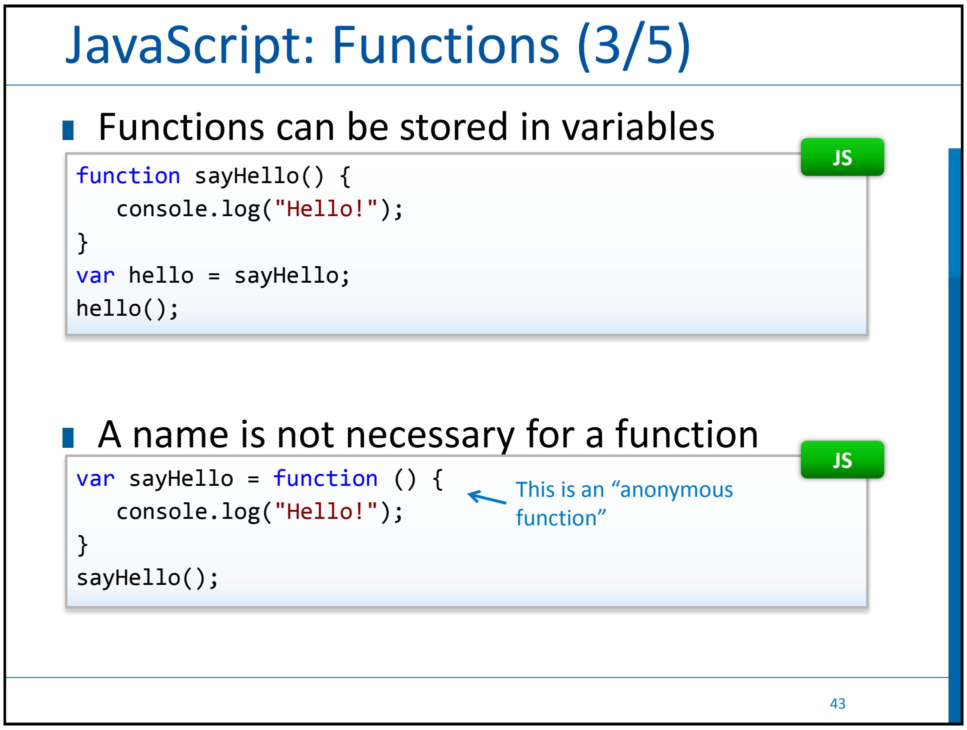
JS

```
function doSomething(a) {
   console.log("First method. A: " + a);
}
function doSomething(a, b) {
   console.log("Second method. A: " + a +
                 ". B: " + b + ".");
}
doSomething(5);
doSomething(5, 8);
doSomething(5, 'test', 8);
```

- Result:

```
Second method. A: 5. B: undefined.
Second method. A: 5. B: 8.
Second method. A: 5. B: test.
```

42



JavaScript: Functions (3/5)

- Functions can be stored in variables

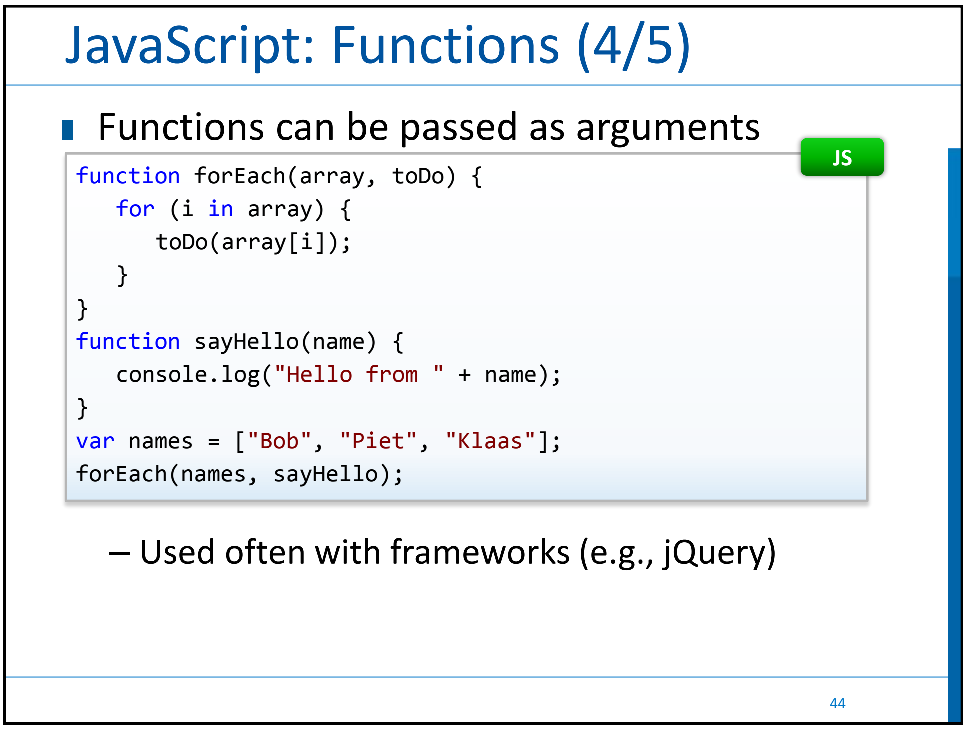
JS
```
function sayHello() {
   console.log("Hello!");
}
var hello = sayHello;
hello();
```

- A name is not necessary for a function

JS
```
var sayHello = function () {
   console.log("Hello!");
}
sayHello();
```

This is an “anonymous function”

43



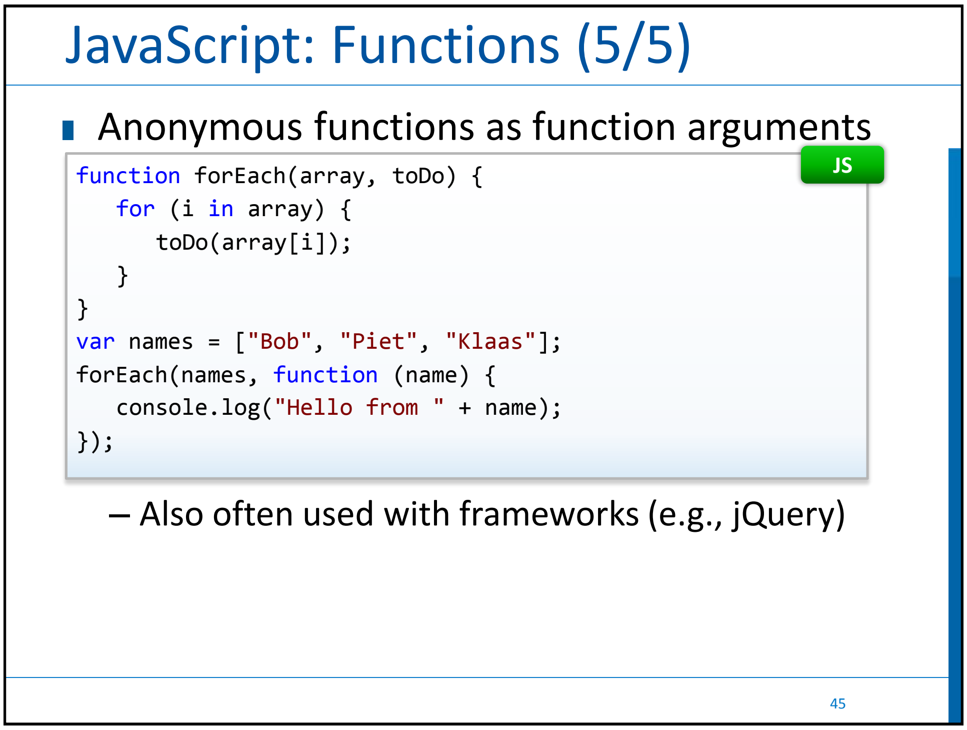
JavaScript: Functions (4/5)

- Functions can be passed as arguments

JS
```
function forEach(array, toDo) {
   for (i in array) {
      toDo(array[i]);
   }
}
function sayHello(name) {
   console.log("Hello from " + name);
}
var names = ["Bob", "Piet", "Klaas"];
forEach(names, sayHello);
```

   – Used often with frameworks (e.g., jQuery)

44



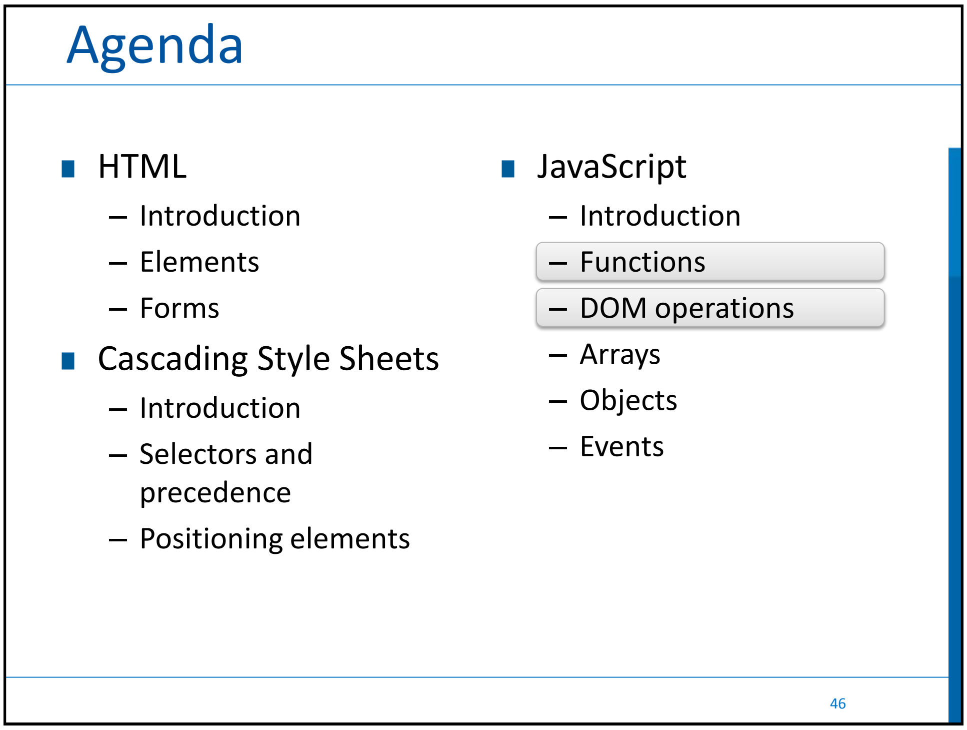
JavaScript: Functions (5/5)

- Anonymous functions as function arguments

JS
```
function forEach(array, toDo) {
   for (i in array) {
      toDo(array[i]);
   }
}
var names = ["Bob", "Piet", "Klaas"];
forEach(names, function (name) {
   console.log("Hello from " + name);
});
```

  – Also often used with frameworks (e.g., jQuery)

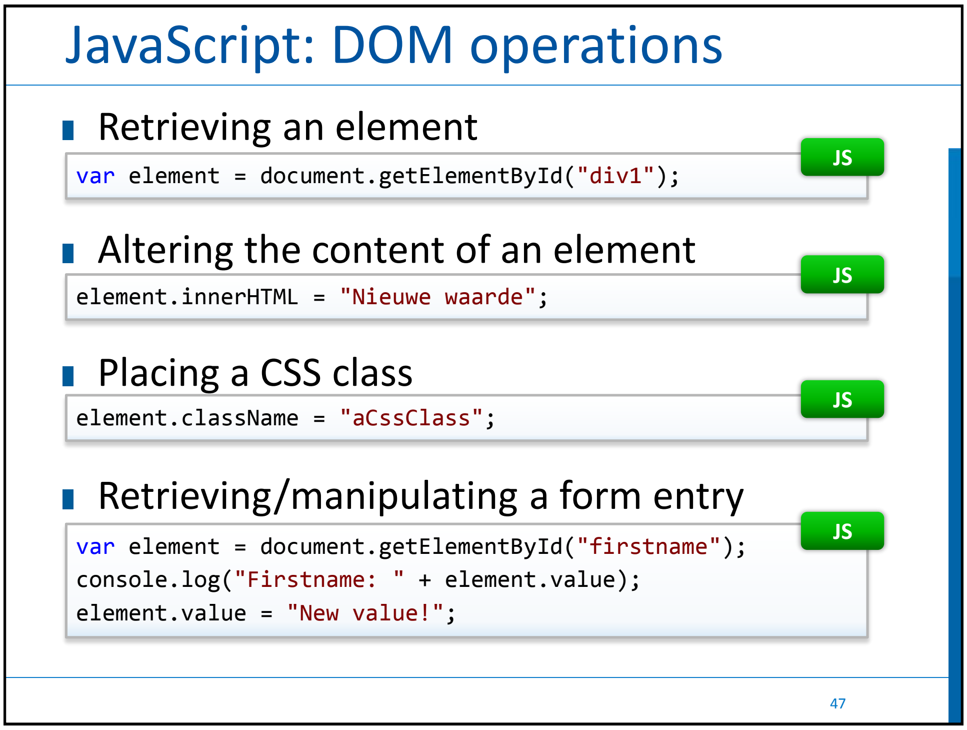
45



Agenda

- HTML
  – Introduction
  – Elements
  – Forms
- Cascading Style Sheets
  – Introduction
  – Selectors and precedence
  – Positioning elements
- JavaScript
  – Introduction
  – Functions
  – DOM operations
  – Arrays
  – Objects
  – Events

46



JavaScript: DOM operations

- Retrieving an element

```
var element = document.getElementById("div1");
```

- Altering the content of an element

```
element.innerHTML = "Nieuwe waarde";
```

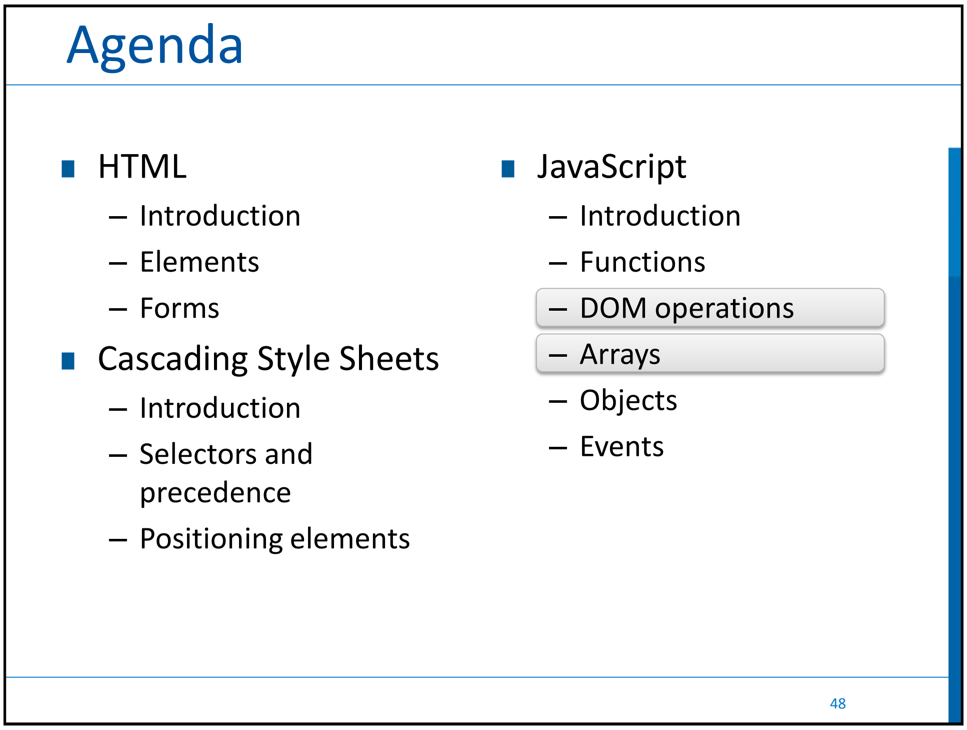
- Placing a CSS class

```
element.className = "aCssClass";
```

- Retrieving/manipulating a form entry

```
var element = document.getElementById("firstname");
console.log("Firstname: " + element.value);
element.value = "New value!";
```

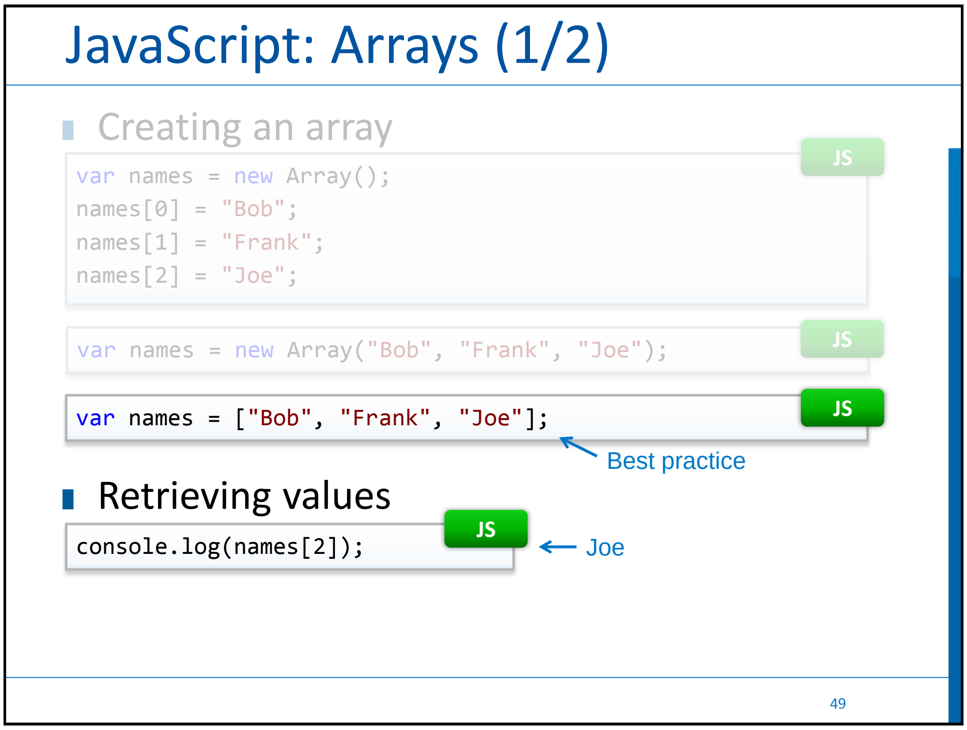
47



Agenda

- HTML
  - Introduction
  - Elements
  - Forms
- Cascading Style Sheets
  - Introduction
  - Selectors and precedence
  - Positioning elements
- JavaScript
  - Introduction
  - Functions
  - DOM operations
  - Arrays
  - Objects
  - Events

48



JavaScript: Arrays (1/2)

- Creating an array

```
var names = new Array();
names[0] = "Bob";
names[1] = "Frank";
names[2] = "Joe";
```

JS

```
var names = new Array("Bob", "Frank", "Joe");
```

JS

```
var names = ["Bob", "Frank", "Joe"];
```

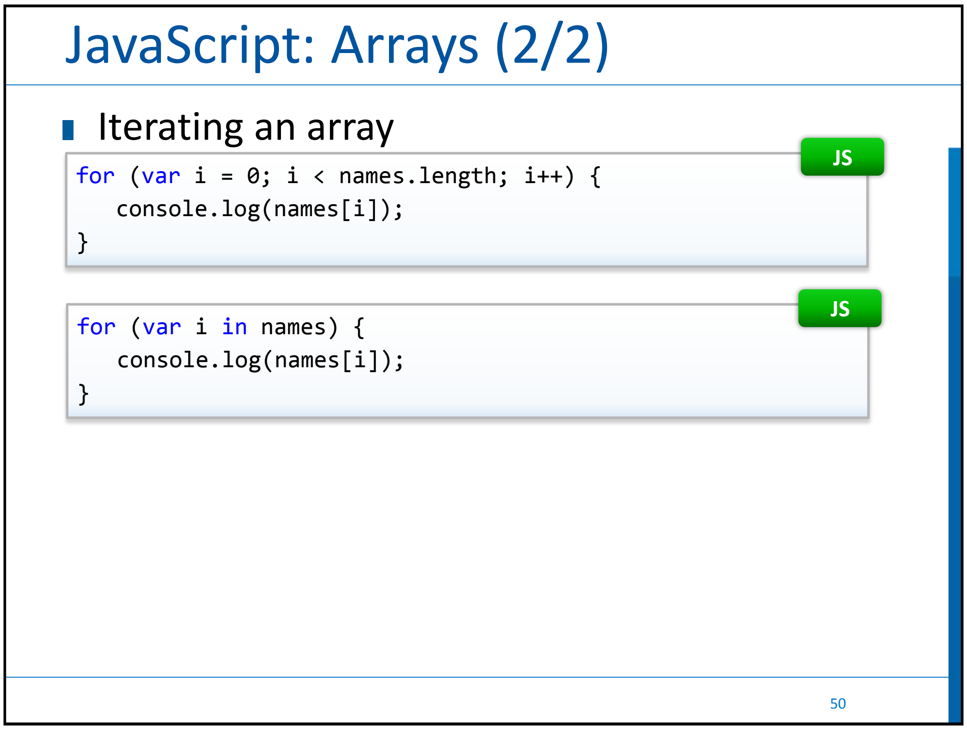
JS
Best practice

- Retrieving values

```
console.log(names[2]);
```

JS
Joe

49



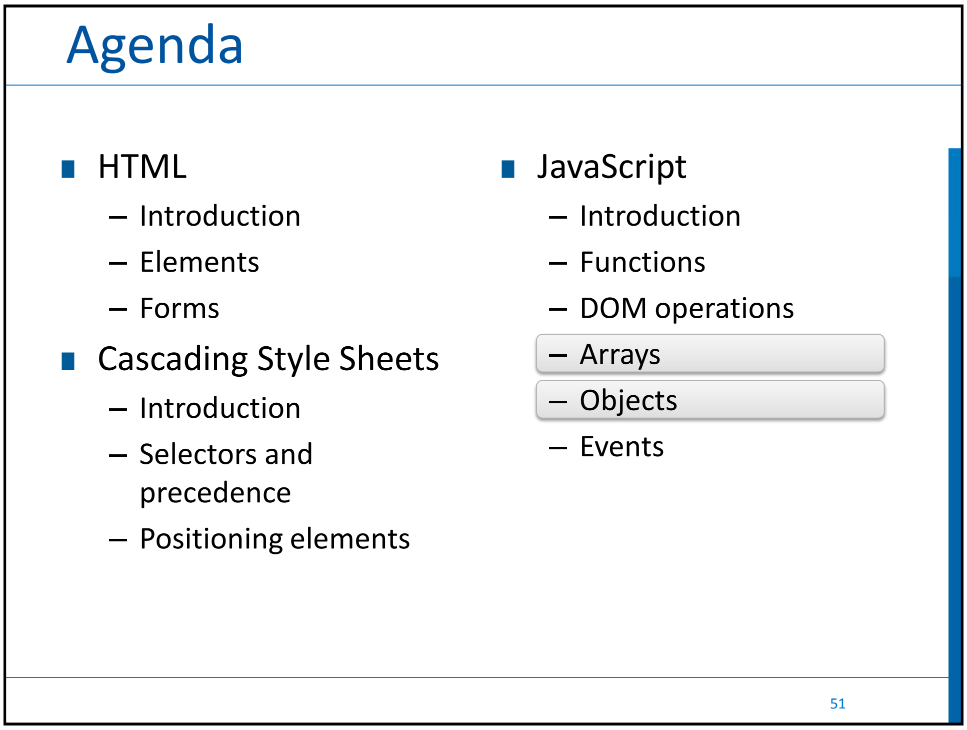
JavaScript: Arrays (2/2)

- Iterating an array

```
for (var i = 0; i < names.length; i++) {
    console.log(names[i]);
}
```

```
for (var i in names) {
    console.log(names[i]);
}
```

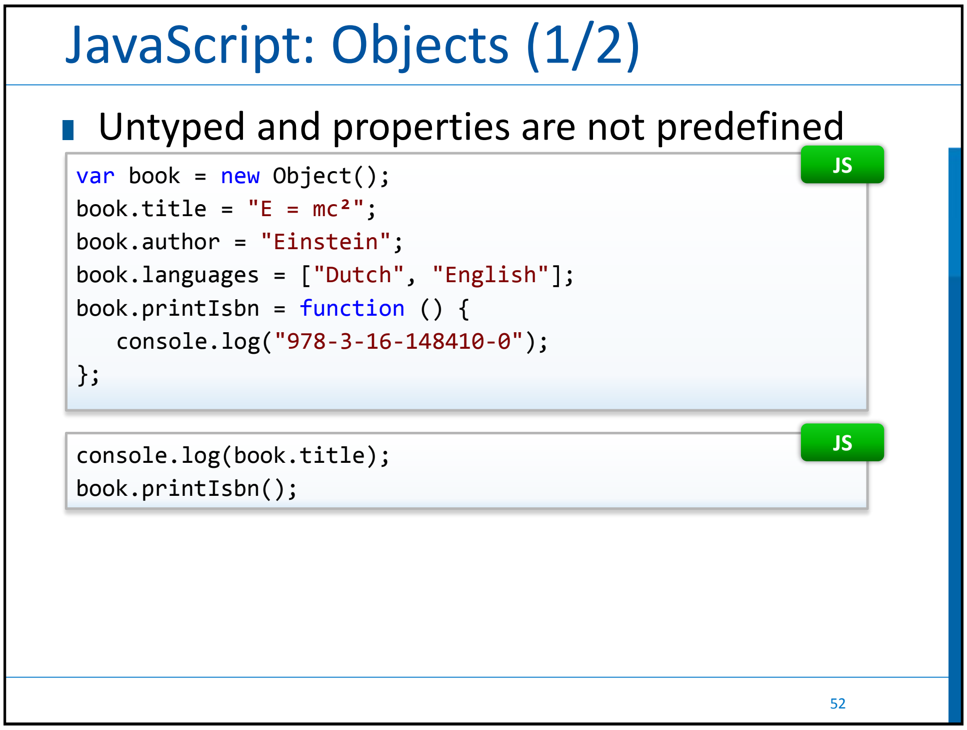
50



Agenda

- HTML
  - Introduction
  - Elements
  - Forms
- Cascading Style Sheets
  - Introduction
  - Selectors and precedence
  - Positioning elements
- JavaScript
  - Introduction
  - Functions
  - DOM operations
  - Arrays
  - Objects
  - Events

51



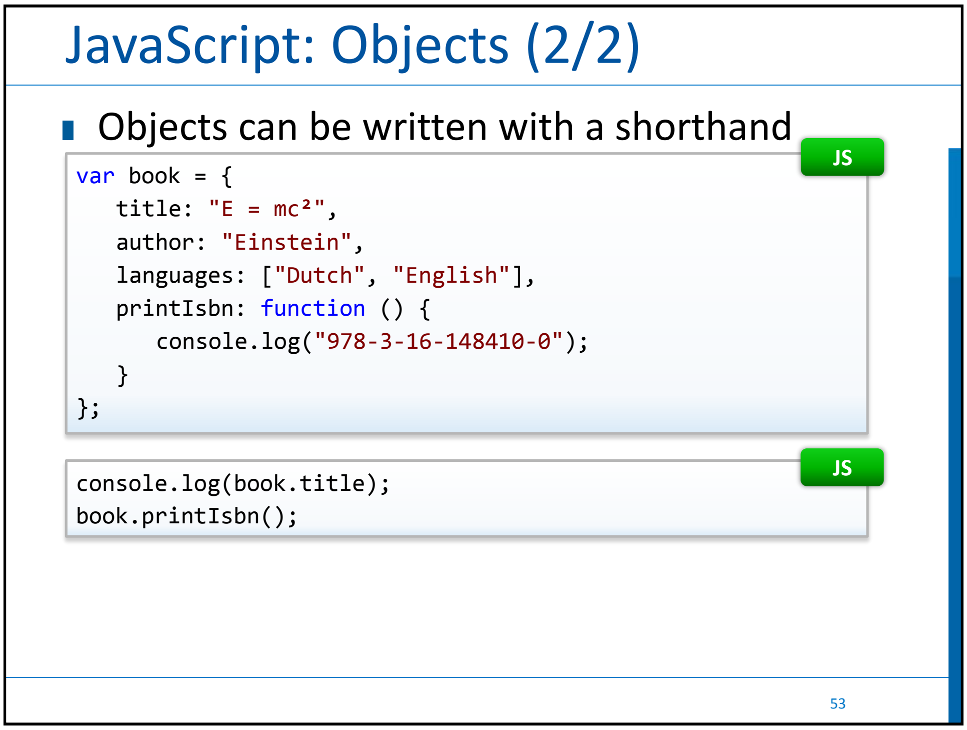
JavaScript: Objects (1/2)

- Untyped and properties are not predefined

JS
```
var book = new Object();
book.title = "E = mc²";
book.author = "Einstein";
book.languages = ["Dutch", "English"];
book.printIsbn = function () {
   console.log("978-3-16-148410-0");
};
```

JS
```
console.log(book.title);
book.printIsbn();
```

52



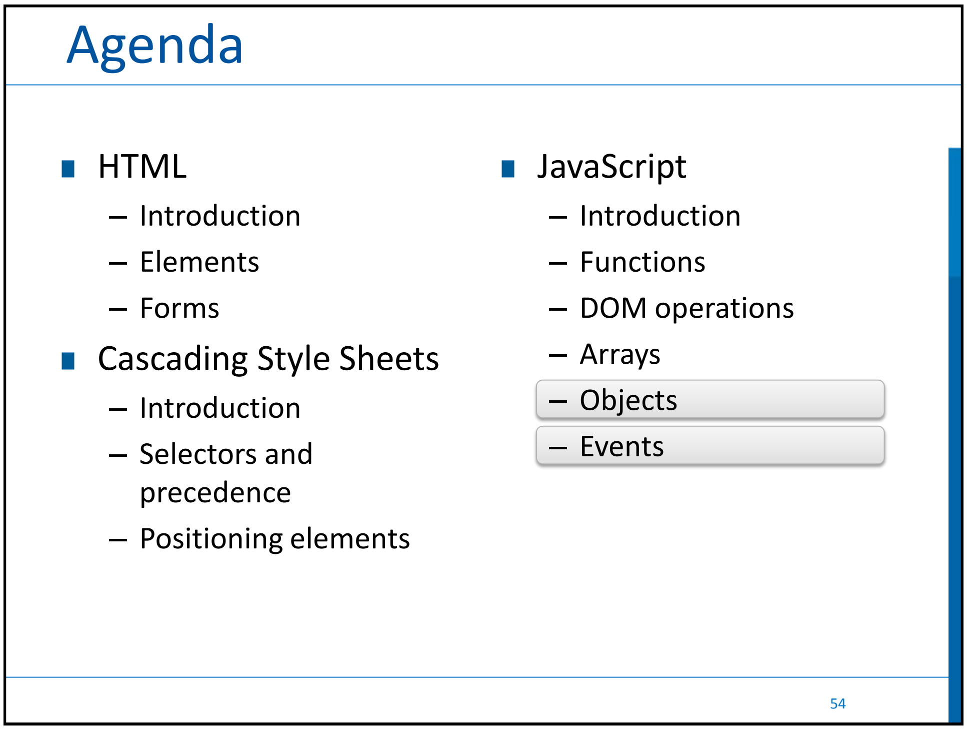
JavaScript: Objects (2/2)

- Objects can be written with a shorthand

JS
```
var book = {
   title: "E = mc²",
   author: "Einstein",
   languages: ["Dutch", "English"],
   printIsbn: function () {
      console.log("978-3-16-148410-0");
   }
};
```

JS
```
console.log(book.title);
book.printIsbn();
```

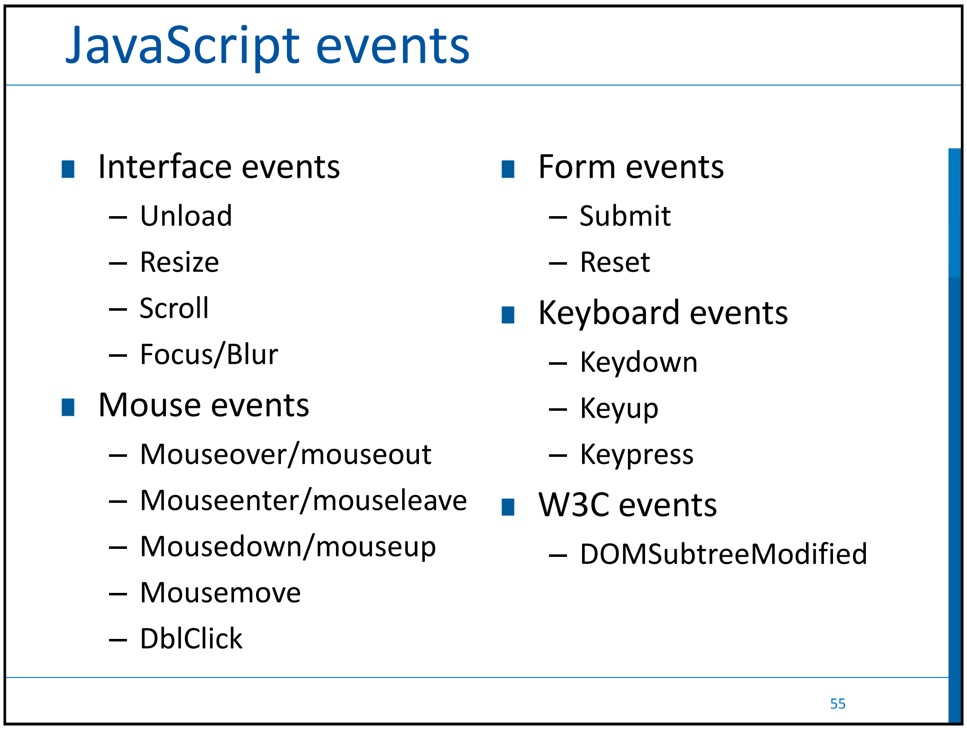
53



Agenda

- HTML
  - Introduction
  - Elements
  - Forms
- Cascading Style Sheets
  - Introduction
  - Selectors and precedence
  - Positioning elements
- JavaScript
  - Introduction
  - Functions
  - DOM operations
  - Arrays
  - Objects
  - Events

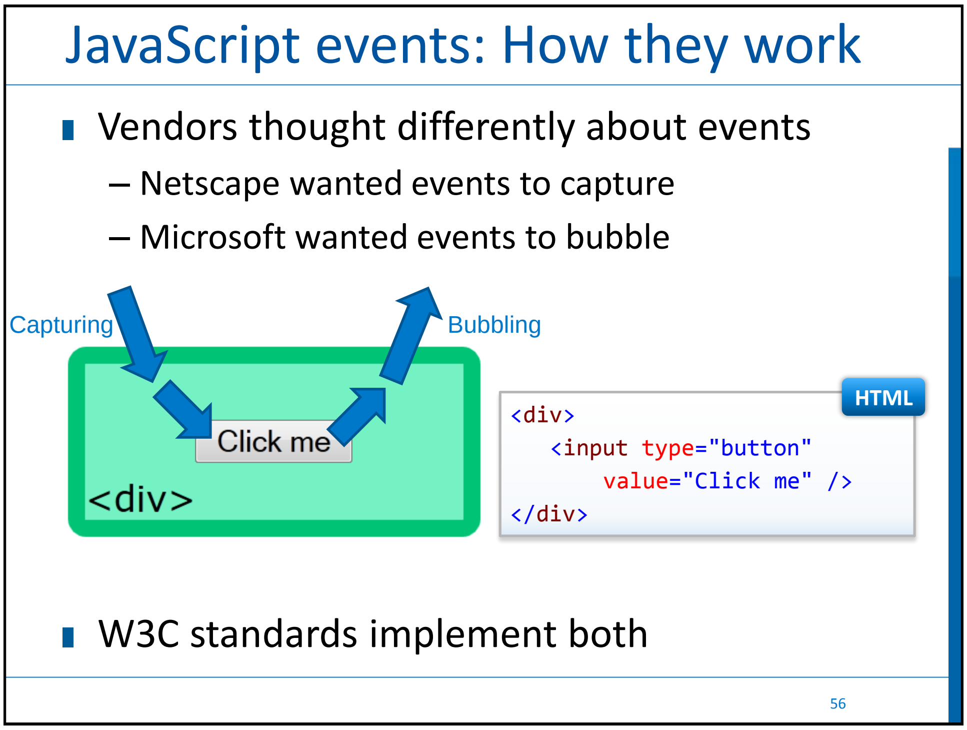
54



JavaScript events

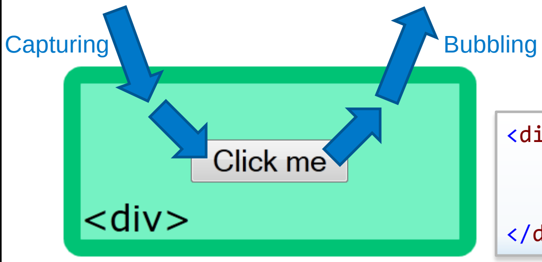
- Interface events
  - Unload
  - Resize
  - Scroll
  - Focus/Blur
- Mouse events
  - Mouseover/mouseout
  - Mouseenter/mouseleave
  - Mousedown/mouseup
  - Mousemove
  - DblClick
- Form events
  - Submit
  - Reset
- Keyboard events
  - Keydown
  - Keyup
  - Keypress
- W3C events
  - DOMSubtreeModified

55



JavaScript events: How they work

- Vendors thought differently about events
  - Netscape wanted events to capture
  - Microsoft wanted events to bubble

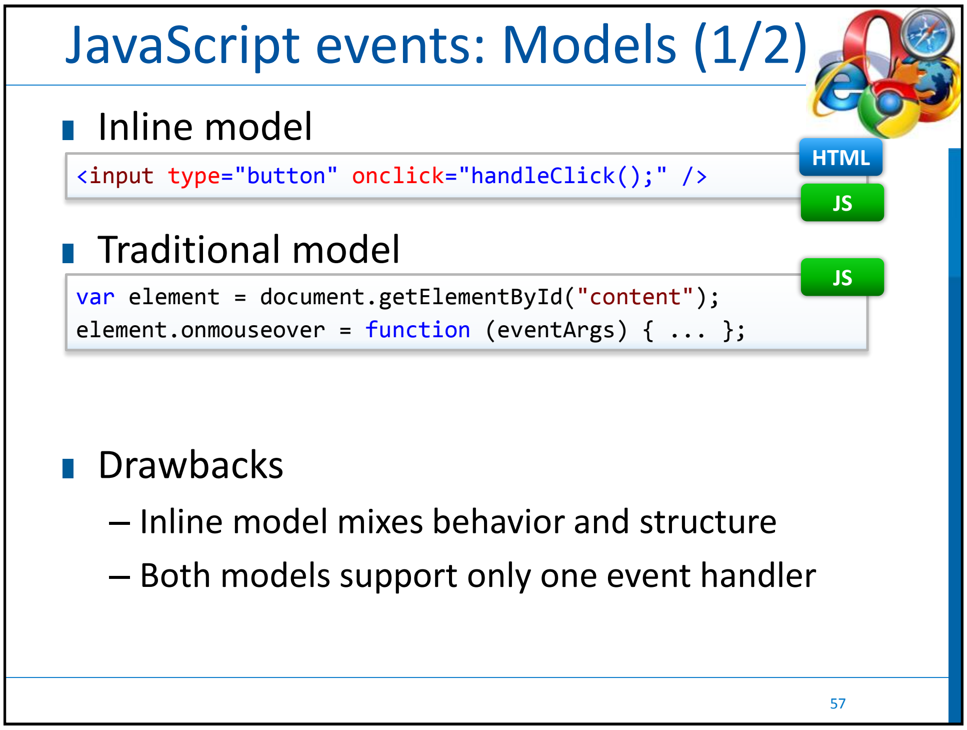


HTML

```
<div>
   <input type="button"
       value="Click me" />
</div>
```

- W3C standards implement both

56



JavaScript events: Models (1/2)

- Inline model

HTML JS

```
<input type="button" onclick="handleClick();" />
```

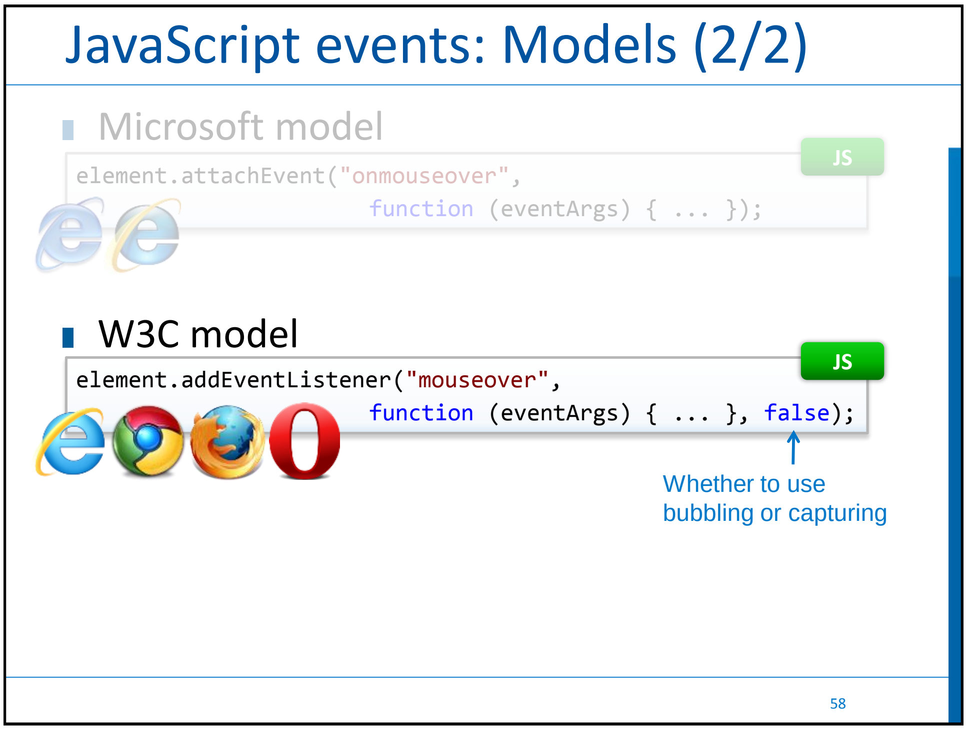
- Traditional model

JS

```
var element = document.getElementById("content");
element.onmouseover = function (eventArgs) { ... };
```

- Drawbacks
  - Inline model mixes behavior and structure
  - Both models support only one event handler

57



JavaScript events: Models (2/2)

- Microsoft model

JS

```
element.attachEvent("onmouseover",
                        function (eventArgs) { ... });
```

- W3C model

JS

```
element.addEventListener("mouseover",
                        function (eventArgs) { ... }, false);
```

Whether to use bubbling or capturing

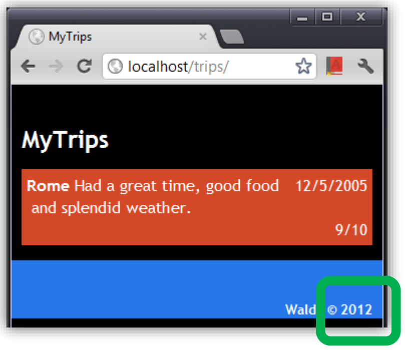
58

Questions

59

Lab: Setting up the Trips page

- Exercise 3: JavaScript



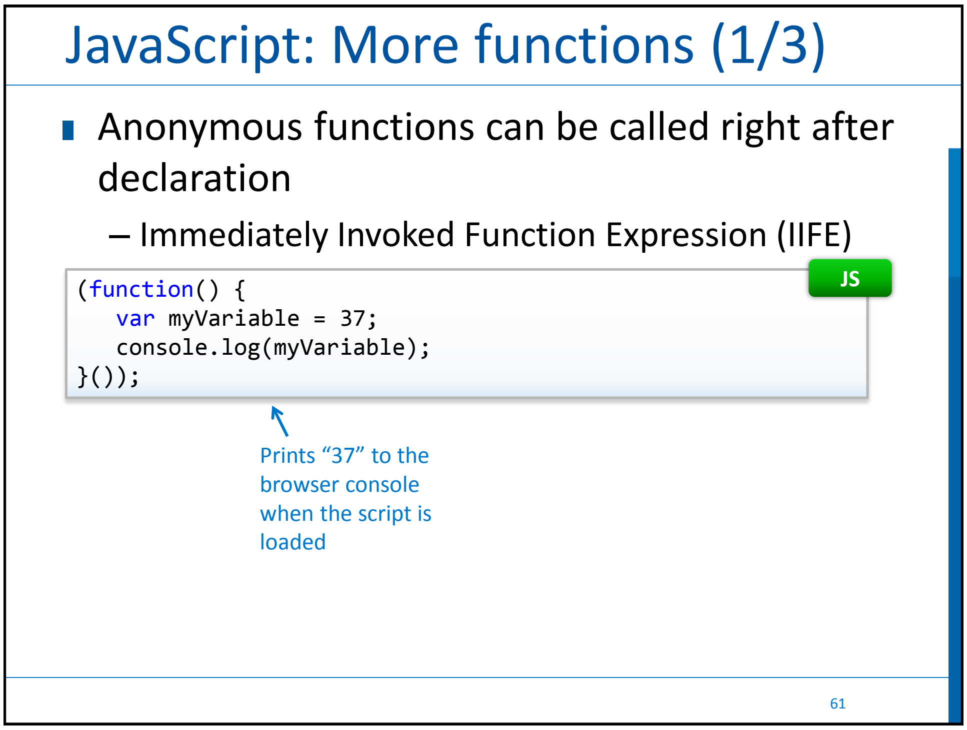
MyTrips

Rome Had a great time, good food and splendid weather. 12/5/2005

9/10

Wald © 2012

60



JavaScript: More functions (1/3)

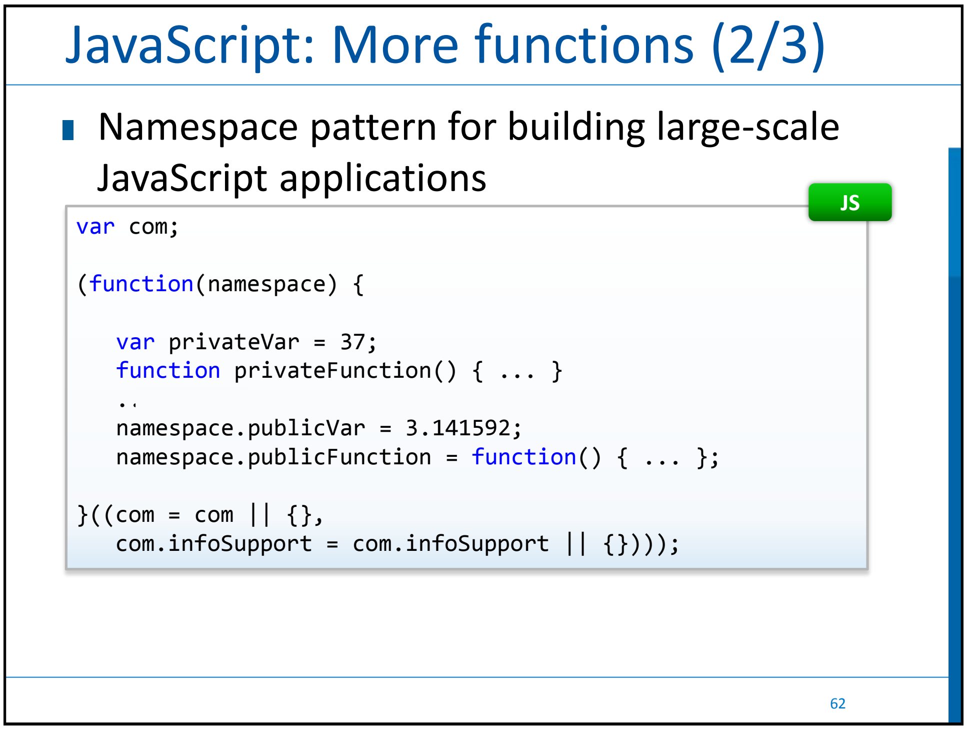
- Anonymous functions can be called right after declaration
  – Immediately Invoked Function Expression (IIFE)

JS

```
(function() {
   var myVariable = 37;
   console.log(myVariable);
}());
```

Prints “37” to the browser console when the script is loaded

61



JavaScript: More functions (2/3)

- Namespace pattern for building large-scale JavaScript applications

JS

```
var com;

(function(namespace) {

   var privateVar = 37;
   function privateFunction() { ... }
   ..
   namespace.publicVar = 3.141592;
   namespace.publicFunction = function() { ... };

}((com = com || {},
   com.infoSupport = com.infoSupport || {})));
```

62

JavaScript: More functions (3/3)

- Ensure undefined is really undefined
  - The undefined constant used to be mutable

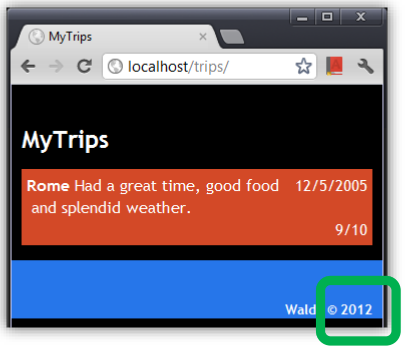
JS

```
var com;

(function(namespace, undefined) {

   var privateVar = 37;
   function privateFunction() { ... }

   namespace.publicVar = 3.141592;
   namespace.publicFunction = function() { ... };

}((com = com || {},
   com.infoSupport = com.infoSupport || {}));
```

63

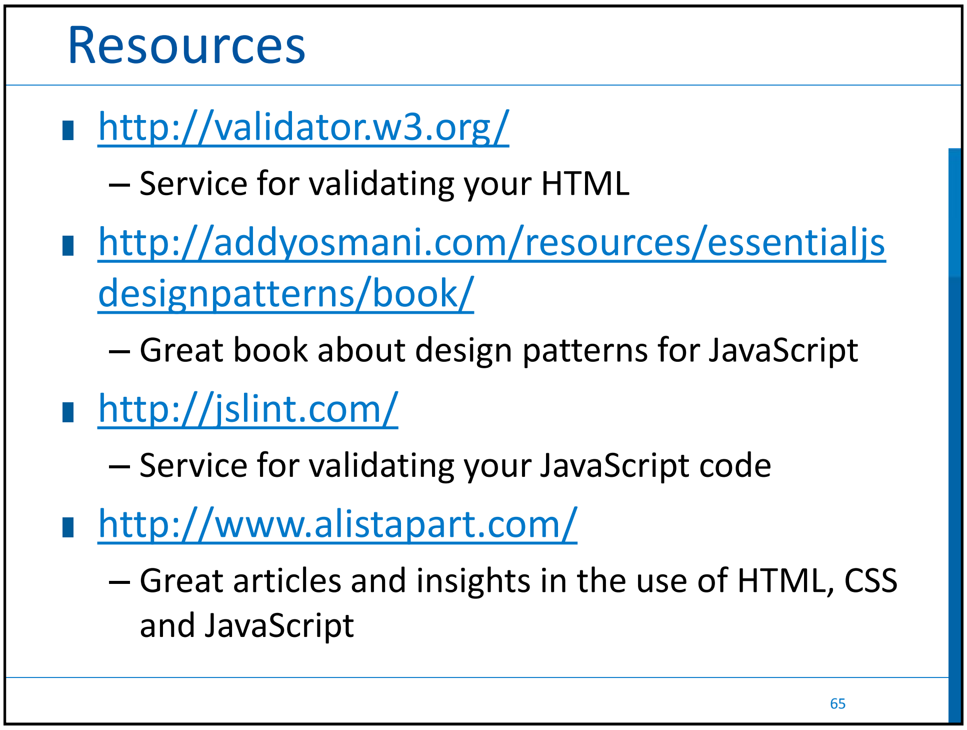
Lab: Setting up the Trips page

- Exercise 4: Namespacing your JavaScript



Rome Had a great time, good food and splendid weather. 12/5/2005
9/10

64



Resources

- http://validator.w3.org/
  - Service for validating your HTML
- http://addyosmani.com/resources/essentialjs designpatterns/book/
  - Great book about design patterns for JavaScript
- http://jslint.com/
  - Service for validating your JavaScript code
- http://www.alistapart.com/
  - Great articles and insights in the use of HTML, CSS and JavaScript

65