Agenda

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

2

About HTML

- HTML defines the structure of your webpage
- Based on SGML
- Maintained by the World Wide Web Consortium (W3C)

3

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)

4

Agenda

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

5

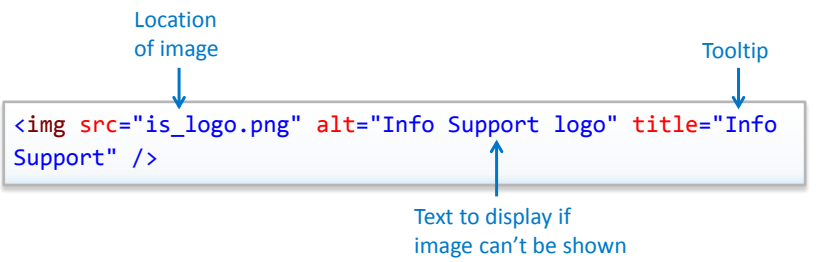
HTML: Basic page structure

Tells the browser what version of HTML to parse

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

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

6

HTML elements: Images



7

HTML elements: Links

- A simple link:

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

- A clickable image:

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

- Open in a new window/tab:

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

8

HTML elements: Table

```
<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>

9

HTML elements: Lists (1)

- Unordered list

- First item
- Second item
- Third item

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

- Ordered list

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

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

10

HTML elements: Lists (2)

- Definition list

```
<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

11

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

12

HTML elements: Div and Span

- Generic mechanisms for adding structure and applying styling
  - `<div>` is a block element
  - `<span>` is an inline element
- No meaning, little styling

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

13

Agenda

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

14

HTML forms

- Used for submitting data to the server

Location to send data to

HTTP method should always be POST

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

- Includes support for uploading files:

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

15

HTML form elements (1/3)

- Textbox:

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

- Password:

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

- Hidden:

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

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

16

HTML form elements (2/3)

- Checkbox:

JavaScript magazine

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

  – Only selected values are posted

- Radiobutton:

Through a search engine
By word of mouth
Other

```
<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 of every group is posted

17

HTML form elements (3/3)

- Dropdownlist:

IT
IT
Government
Landscaping

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

- Submitting a form:

Submit

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

18

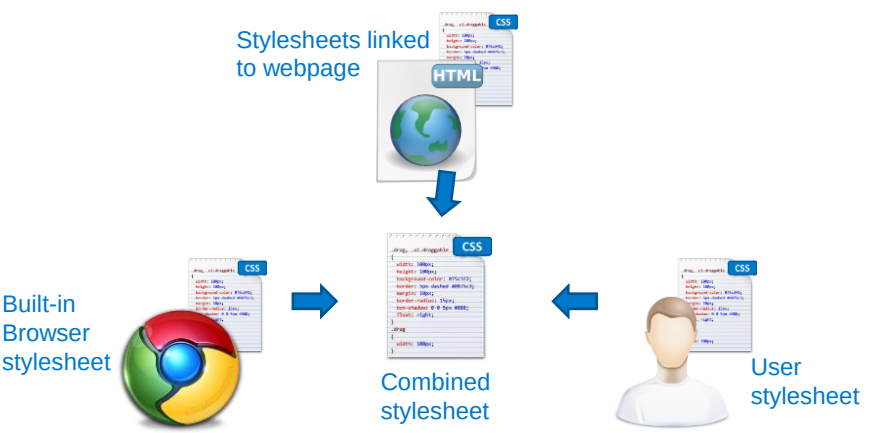
Agenda

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

19

Cascading Style Sheets

- Used for styling your HTML elements



20

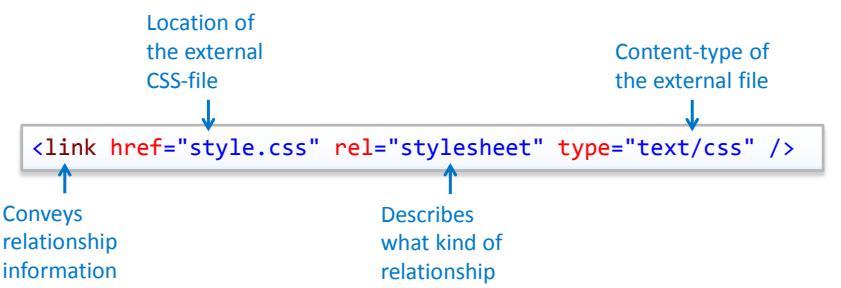
CSS: History

- 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

21

CSS: Usage

- Reference an external .css-file:



- Inline CSS is also possible, but not recommended

22

Agenda

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

23

CSS: Selectors

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

24

CSS: Combining styling

- Combine styling with multiple selectors

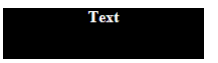
CSS:

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

HTML:

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

Result:



25

CSS: Selector precedence

- The most specific selector wins

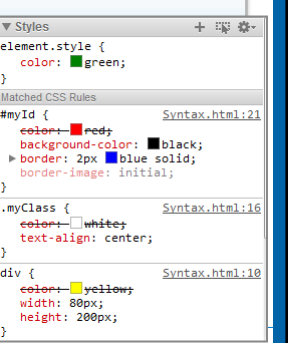
CSS:

```
div {
     color: yellow;
     width: 200px;
     height: 80px;
}
.myClass {
     color: white;
     text-align: center;
}
#myId {
     color: red;
     background-color: black;
     border: 2px blue solid;
}
```

HTML:

```
<div id="myId"
     class="myClass"
     style="color: green;">
     Text
</div>
```

Result:



26

CSS: More selectors

- Select elements within an element

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

- Select direct child elements of an element

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

- Apply styling to multiple selectors

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

- The universal selector

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

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

27

CSS: Pseudo-classes

- Selection beyond simple selectors

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

The first li element in ul#navigation

  - :last-child (CSS3)
  - :nth-child(even) (CSS3)
  - :enabled & :disabled (CSS3)

- Selection based on data outside the DOM

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

When the user holds the pointer over the element

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

28

Agenda

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

29

CSS positioning

- DIV is commonly used for positioning
- By default, are placed below each other:

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

Div 1
Div 2
Div 3

30

CSS positioning: Float

- Used to 'float' next to other elements

CSS:

```
#div1 {
     float: left;
     width: 50px;
     background-color: red;
}
#div2 {
     float: right;
     width: 50px;
     background-color: blue;
}
#div3 {
     width: 100%;
     background-color: green;
}
```

HTML:

```
<div id="div1">Div 1</div>
<div id="div2">Div 2</div>
<div id="div3">Div 3</div>
```

Result:

31

CSS positioning: Absolute

- Specify the exact pixels where object should be

CSS:

```
#div1 {
     position: absolute;
     top: 100px;
     left: 20px;
}
#div2 {
     position: absolute;
     top: 100px;
     left: 60px;
}
#div3 {
     position: absolute;
     top: 80px;
     left: 30px;
}
```

HTML:

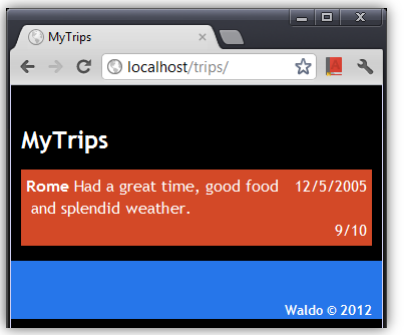
```
<div id="div1">Div 1</div>
<div id="div2">Div 2</div>
<div id="div3">Div 3</div>
```

Result:

32

Lab: Setting up the Trips page

- Exercise 1: Basic structure
- Exercise 2: Styling



33

Agenda

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

34

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

35

JavaScript: Usage

- Reference an external JavaScript file

Location of the external JavaScript file

```
<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

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

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: Functions (1/5)

Declaring a function

```
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

38

JavaScript: Functions (2/5)

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

```
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.
```

39

JavaScript: Functions (3/5)

- Functions can be stored in variables

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

- A name is not necessary for a function

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

← This is an “anonymous function”

40

JavaScript: Functions (4/5)

- Functions can be passed as arguments

```
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)

41

JavaScript: Functions (5/5)

- Anonymous functions as function arguments

```
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)

42

Agenda

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

43

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!";
```

44

Agenda

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

45

JavaScript: Arrays (1/2)

- Creating an array

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

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

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

Best practice

- Retrieving values

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

← Joe

46

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]);
}
```

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: Objects (1/2)

- Untyped and properties are not predefined

```
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");
};
```

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

49

JavaScript: Objects (2/2)

- Objects can be written with a shorthand

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

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

50

Agenda

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

51

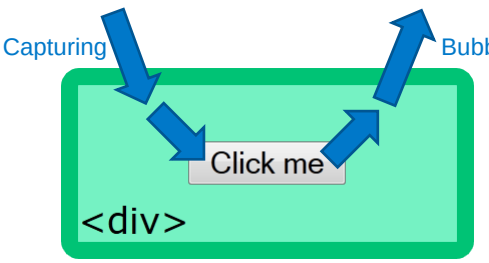
JavaScript events

- 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

52

JavaScript events: How they work

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




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

- W3C standards implement both

53

JavaScript events: Models (1/2)

- Inline model

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

- Traditional model

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

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

54

JavaScript events: Models (2/2)

- Microsoft model

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

- W3C model

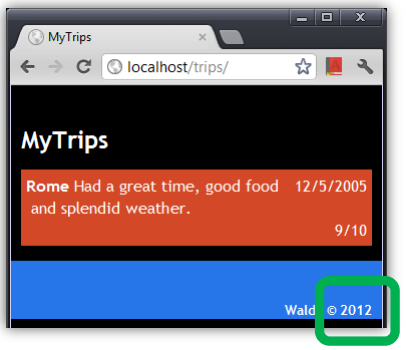
```
element.addEventListener("mouseover",
                    function (eventArgs) { ... }, false);
```

Whether to use bubbling or capturing

Questions

Lab: Setting up the Trips page

- Exercise 3: JavaScript



57

JavaScript: More functions (1/3)

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

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

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

58

JavaScript: More functions (2/3)

- Namespace pattern for building large-scale JavaScript applications

```
var com;

(function(namespace) {

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

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

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

59

JavaScript: More functions (3/3)

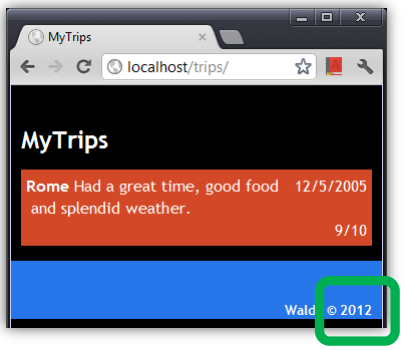
- Ensure undefined is really undefined
  - The undefined constant used to be mutable

```
var com;

(function(namespace, undefined) {

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

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

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

60

Lab: Setting up the Trips page

- Exercise 4: Namespacing your JavaScript



61

Resources

- http://validator.w3.org/
  - Service for validating your HTML
- http://addyosmani.com/resources/essentialjsdesignpatterns/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

62