Lisbon School of Economics & Management
Universidade de Lisboa

LISBOA
UNIVERSIDADE DE LISBOA

WEB CLIENT

HTML, CSS and JavaScript fundamentals

Carlos J. Costa / João Tiago Aparício



HTML

- HyperText Markup Language (HTML)
- The standard language used to access and display web pages
- Code is interpreted by a browser, which shows the result to users
- A document is composed of tags and text

EXAMPLE

```
<h1>Hello, web!</h1>
<p>Tags wrap <b>text</b>.</p>
```



How to write HTML

Start every page with the document type, then declare the character encoding and the language.

HTML

```html
<!DOCTYPE html>
<!-- Character encoding: two equivalent ways -->
<meta http-equiv="content-type" content="text/html; charset=UTF-8" />
<meta charset="UTF-8">
<!-- Language -->
<html lang="pt-PT">
```

Improving your HTML

Elements

Structure content with headings, paragraphs and lists.

```
<h1>Title</h1>
<p>Paragraph</p>
<ul><li>Item</li></ul>
```

Links

Connect pages and resources with anchors.

```
<a href="page.html">
  Read more
</a>
```

Images

Embed pictures and describe them with alt text.

```
<img src="a.png"
  alt="Logo">
```

…and more: tables, lists, headings, forms.

Forms

A form collects user input and sends it to a server-side script.

HTML

```
<form method="post" action="test.php">
  <input type="text" name="city" value="Lisbon" size="15" maxlength="15">
  <input type="submit" value="enviar">
</form>
```

`method`

How the data is sent to the server (post or get).

`action`

The script that receives and processes the data.

`maxlength`

The maximum number of characters accepted.

HTML attributes

`id`	Assigns a name to an element. The name must be unique in a document.
`class`	Assigns a class name or a set of class names to an element.
`style`	Specifies style information for the current element.

EXAMPLE

```
<p id="meuparagrafo">paragrafo.</p>
```

CSS

- Cascading Style Sheets
- Customise the appearance of HTML
- Defined as a set of rules
- Rules can be placed:
  - Inline, in an element
  - In the head of the page
  - In a separate file
- A separate file can be shared by several HTML pages



CSS: Defining a rule

SYNTAX

```
element {
  attribute1: value1;
  attribute2: value2;
}
```

EXAMPLE

```
h1 {
  font-size: 36px;
}
```

A rule **selects an element** and sets one or more declarations, each written as **`attribute: value;`**

CSS: Inline style

Set the style directly on an element with the style attribute.

HTML

```
<p style="color:blue; margin-left:20px;">this paragraph.</p>
```

RESULT

this paragraph.

CSS: In the head

Declare the rules once, inside a style element in the head of the page.

HTML

```
<head>
  <style>
    body { background-color: yellow; }
    p { color: blue; }
  </style>
</head>
```

How it works

body: yellow background

p: blue text

Applies to the whole page

CSS: In a separate file

Link an external stylesheet from the head of each page.

HTML

```
<head>
  <link rel="stylesheet" type="text/css" href="formatos.css">
</head>
```

One stylesheet, many pages. The same file can be linked from several HTML pages, so a style change is made in one place.

JavaScript



A language that is…

- Lightweight
- Interpreted
- Object-oriented, with first-class functions
- Prototype-based
- Dynamic
- Multiparadigm: object-oriented, imperative and functional styles

Where it runs

Best known as the scripting language for web pages

ALSO OUTSIDE THE BROWSER

- node.js
- Apache CouchDB

JavaScript: Calling a function

1. Create a web page
2. Add a form with a button
3. The onclick event calls teste()

HTML

```
<form method="post">
  <input type="button" onclick="teste()">
</form>
```

JavaScript: Showing an alert

```
<!DOCTYPE html>
<html>
<head>
<script>
function teste(){
  alert("olá");
}
</script>
</head>
<form method="post">
  <input type="button" value="Ola" onclick="teste();">
</form>
<img style="position:absolute; top:50px; left:100px;" src="boneco.jpg" />
</html>
```

Clicking the button runs teste(), which shows an alert box.

JavaScript: Reading a style

```
<!DOCTYPE html>
<html>
<head>
<script>
function teste(){
  var a=document.getElementById("b1").style.top;
  alert(a);
}
</script>
</head>
<form method="post">
  <input type="button" value="Ola" onclick="teste();">
</form>
<img id="b1" style="position:absolute; top:50px; left:100px;" src="boneco.jpg" />
</html>
```

getElementById finds the image by its id; style.top returns its position as text, e.g. "50px".

JavaScript: Useful snippets

```
document.getElementById("demo").innerHTML = Date();
```

Writes the current date into the element with id “demo”.

```
var a=parseInt(a);
```

Converts a text value, such as “50px”, into a number.

```
"Hello, " + name
```

The + operator concatenates strings.

JavaScript: Moving an element

```
<!DOCTYPE html>
<html>
<head>
<script>
function teste(){
  var a=document.getElementById("b1").style.top;
  var a=parseInt(a)+10;
  //alert(a);
  document.getElementById("b1").style.top=a+"px";
}
</script>
</head>
<form method="post">
  <input type="button" value="Ola" onclick="teste();">
</form>
<img id="b1" style="position:absolute; top:50px; left:100px;" src="boneco.jpg" />
</html>
```

Each click reads top, adds 10 and writes it back with "px", moving the image down.

Bibliography

- **W3C: HTML 4.01 specification**
  http://www.w3.org/TR/html401/struct/global.html
- **MDN Web Docs: JavaScript**
  https://developer.mozilla.org/en-US/docs/Web/JavaScript
- **W3Schools: web development tutorials**
  http://www.w3schools.com