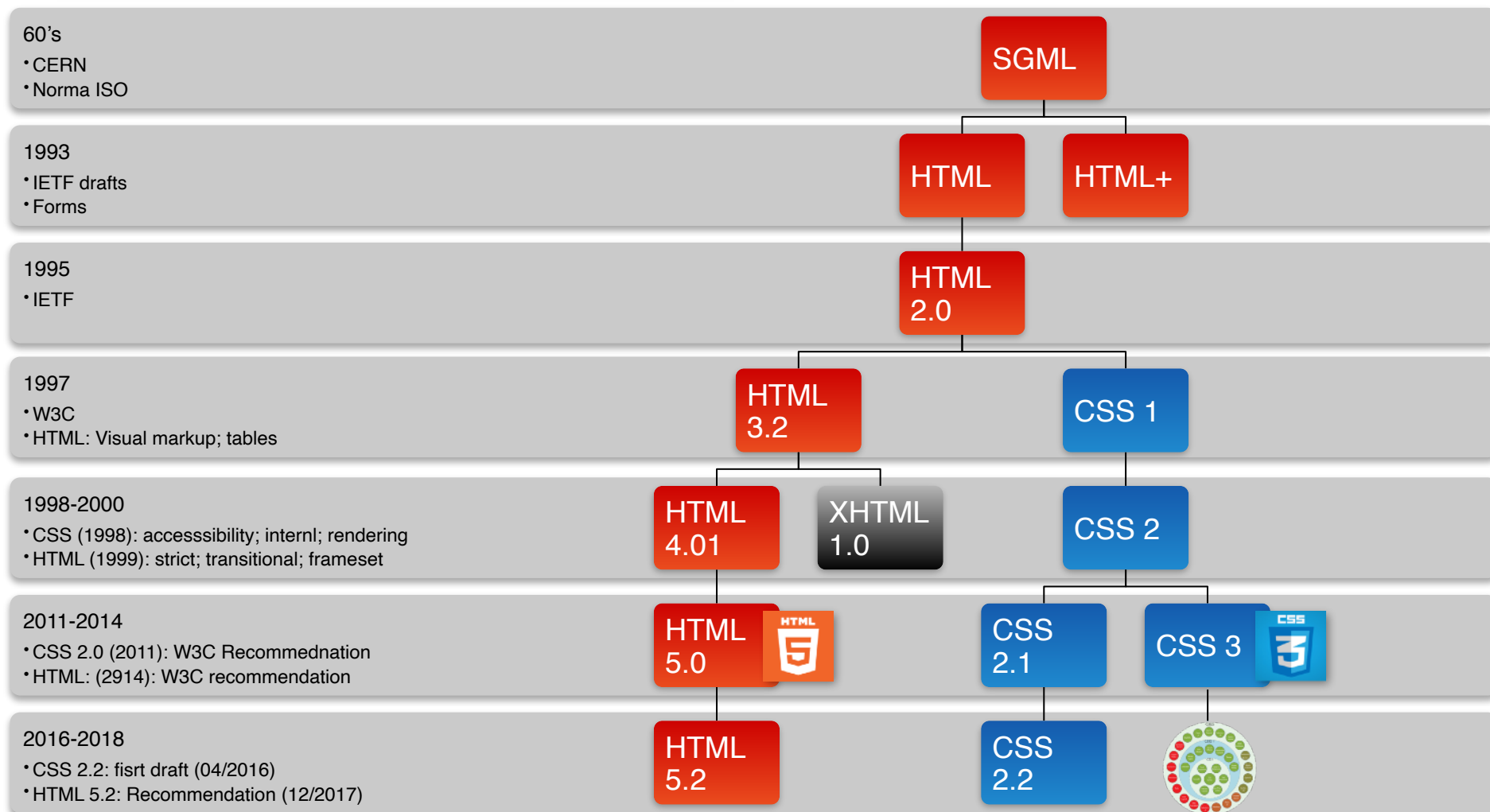


Módulo 10

PROG. WEB 1 - HTML + CSS

Um pouco de história...



HTML: Conteúdo vs. Forma

HTML (HyperText Markup Language)

- Propósito original: definir o **conteúdo** de um documento
- HTML com tags de formatação (HTML 3.2)
 - Uma má ideia!
 - Formatação feita página a página!!
- Controlo de diálogo via hiperligações e/ou alterações à página


CSS (Cascading Style Sheets)

- Suportadas a partir do HTML 4.0
- Definem **como** o HTML deve ser **apresentado**
- Permitem remover a informação de formatação do HTML

Exemplo: <http://www.di.uminho.pt/>

HTML

Programação declarativa

- Como é
... e não como é feito.
- Um conjunto de paradigmas:
 - Constraint programming
 - Functional programming
 - Logic programming
 - muitas *markup languages* (HTML!) 

Imperativo vs. Declarativo

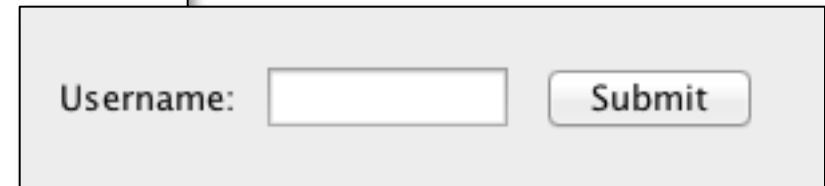
```
mainPanel = new javax.swing.JPanel();
jLabel1 = new javax.swing.JLabel();
jButton1 = new javax.swing.JButton();
jTextField1 = new javax.swing.JTextField();

jLabel1.setText("Username:");
jLabel1.setName("jLabel1");

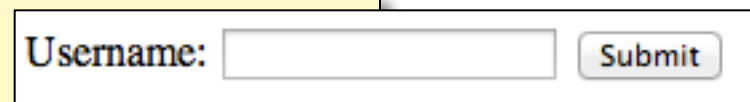
jButton1.setText("Submit");
jButton1.setName("jButton1");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        le_nome(evt);
    }
});

jTextField1.setText("");
jTextField1.setName("jTextField1");

setComponent(mainPanel);
```

A graphical user interface window with a light gray background. It contains a label "Username:" followed by a text input field and a "Submit" button.

```
<form action="le_nome.pl" method="get">
  Username:
  <input type="text" name="user" />
  <input type="submit" value="Submit" />
</form>
```

A graphical user interface window with a white background. It contains a label "Username:" followed by a text input field and a "Submit" button.

Estrutura tipo de um documento HTML 5

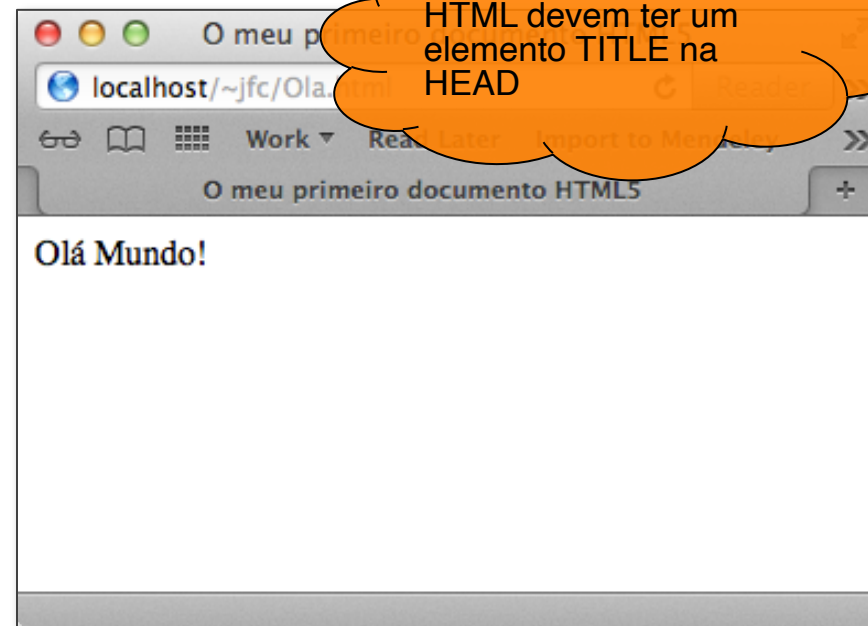
```
<!doctype html>
<html lang="pt">
  <head>
    <meta charset="utf-8" />
    <meta name="keywords" lang="en-us"
      content="example, hello" />
    <meta name="keywords" lang="pt" content="exemplo, olá" />
    <title>O meu primeiro documento HTML5</title>
  </head>
  <body>
    <p>Olá Mundo!</p>
  </body>
</html>
```

Doctype

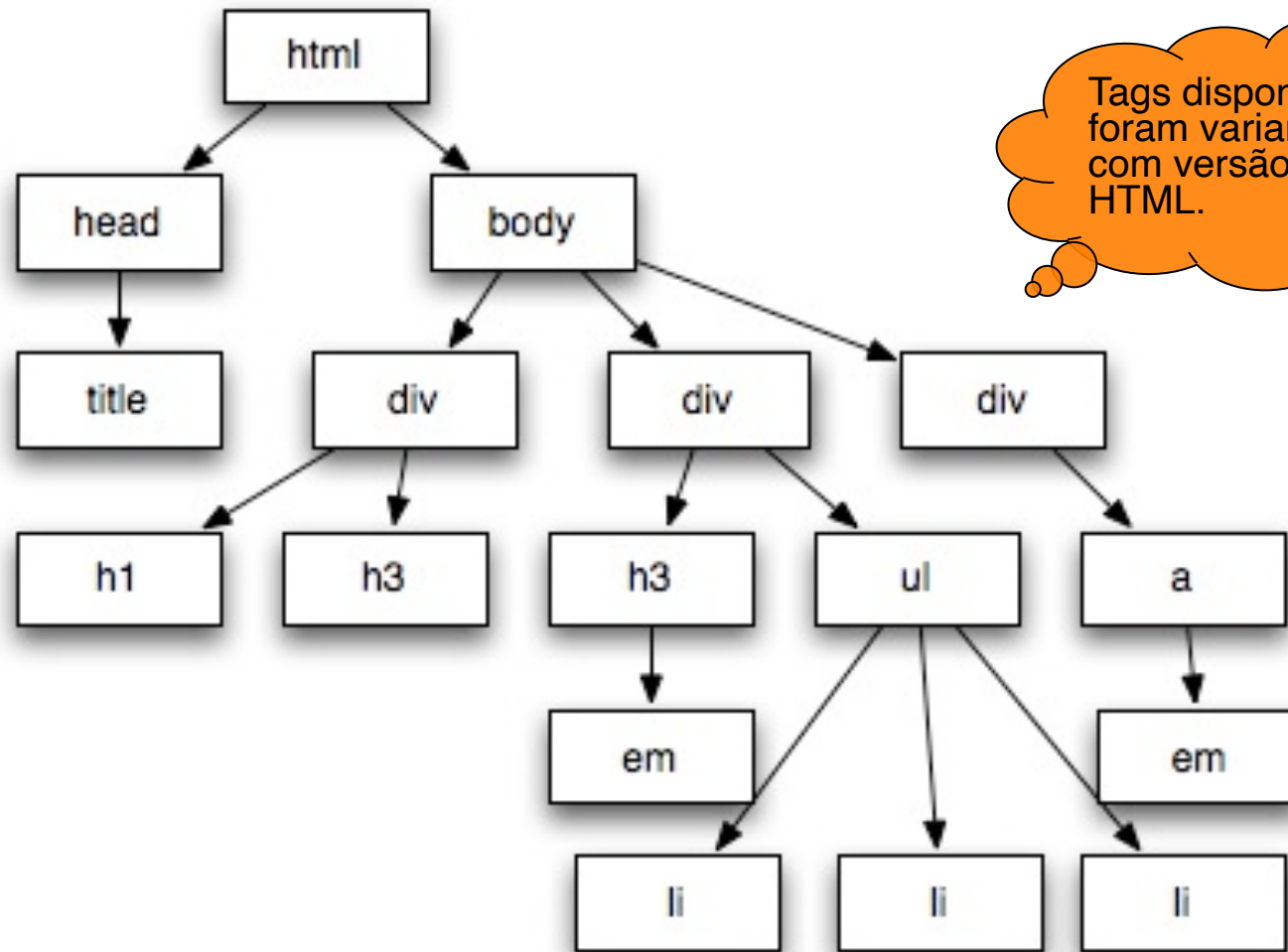
Com excepção do DOCTYPE, todo o documento está contido num elemento HTML

Informação sobre o documento (título, palavras chave, ...) que não é considerada conteúdo

Conteúdo do documento



Estrutura de um Documento HTML



Tags disponíveis foram variando com versão do HTML.

Elementos HTML 5.1

- Sintaxe:

- `<tag atrib=valor ... > conteúdo </tag>`
- `<tag atrib=valor ... />`

- Atributos comuns

- accesskey
- **class**
- contenteditable
- contextmenu
- dir
- draggable
- dropzone
- hidden
- **id**
- lang
- spellcheck
- style
- tabindex
- title
- translate

```
<p class="myClass" id="p1">  
    Olá Mundo  
</p>
```

HTML 5.1 Editor's draft (25/04/2016)

<http://w3c.github.io/html/dom.html#sec-global-attributes>

Elementos HTML 5.1 (além do `html`)

- Metadados
- Secções
- Agrupar conteúdo
- Anotação semântica de texto
- Edições (cf. track changes)
- Conteúdo embebido
- Links
- Tabelas
- Formulários
- Elementos interactivos
- Scripting

HTML 5.1 Editor's draft (25/04/2016)
<http://w3c.github.io/html/> (sec. 4)

HTML 5.1 – Metadados

- Informação sobre o documento:
 - **<head>**
 - **<title>** – título ou nome do documento
 - **<base>** – URL base para URL relativos
 - **<link>** – ligação a outros recursos
 - **<style>** – CSS embebido
 - **<meta>** – outro tipo de informações

<http://w3c.github.io/html/document-metadata.html>
<http://www.w3schools.com/tags/>

HTML 5.1 – Secções

- Permitem dividir o documento:
 - **<body>**
 - <article> — conteúdo auto-contido (notícia, post num blog)
 - <section> — uma secção (possivelmente com título – faz parte de qualquer coisa...)
 - <nav> — uma secção com links de navegação
 - <aside> — um aparte em relação ao conteúdo principal
 - <h1> a <h6> — títulos de decrescentes tamanhos
 - <header> — conteúdo introdutório
 - <footer> — nota de rodapé
 - <address> — informação de **contacto** sobre o documento (body) ou secção (article – HTML5)

<http://w3c.github.io/html/sections.html>
<http://www.w3schools.com/tags/>

HTML 5.1 – Agrupar conteúdo

- Permitem estruturar os documentos:
 - `<p>` — parágrafos
 - `<hr>` — quebra temática de parágrafo
 - `<pre>` — bloco de texto pré-formatado (mantém estrutura tipográfica)
 - `<blockquote>` — uma secção que é uma citação
 - `` e `` (``), `<dl>` (`<dt>` e `<dd>`) — diferentes listas
 - `<figure>` e `<figcaption>` — uma figura e a sua legenda
 - `<main>` — o conteúdo principal do documento (cf. nav)
 - `<div>` — define uma divisão/secção block-level genérica

<http://w3c.github.io/html/grouping-content.html>
<http://www.w3schools.com/tags/>

Markup semântico

- Utilização de tags para definir o tipo de conteúdo
 - Facilita manutenção
 - Facilita a indexação por motores de busca
 - Facilita formatação com CSS
 - Facilita adaptação a diferentes media
 - atributos `class` (e `id`) podem/devem ser utilizados para complementar as tags HTML5
- No entanto:
 - browsers mais antigos (IEs) não suportam as tags
 - (mas ver: https://www.w3schools.com/tags/ref_html_browsersupport.asp)
 - Grid systems das frameworks CSS impõe uma utilização das tags que é influenciada pela apresentação

Markup semântico – Exemplo

```
<!DOCTYPE html>
<html lang="en">
  <head> ...
</head>
<body>
  <header>GamesLibrary</header>
  <nav>
    <span id="nome">User name</span>
    <ul id="menu"> ...
  </ul>
</nav>
<aside>
  <form> ...
</form>
</aside>
<main>
  <table> ...
</table>
  <ul id="pages"> ...
</ul>
</main>
  <footer>Footer information &copy;</footer>
</body>
  <script src="script/script.js"></script>
</html>
```

HTML 5.1 – Anotações semânticas

- Atribuem semântica ao texto (sentido de algumas tags “alterado” no HTML5):
 - `<a>` — uma hiper-ligação
 - `` — ênfase
 - `` — importância
 - `<small>` — small print (as letras pequenas... - ≠ HTML4!!!)
 - `<s>`, `<i>`, `` — irrelevante, qualitativamente diferente, texto realçado
 - `<cite>` — o título de um trabalho – uma citação
 - `<dfn>` — identifica uma definição
 - `<abbr>` — uma abreviação ou acrónimo
 - `<data>` — anotar texto com valor machine readable (atributo `value`)
 - `` — define um elemento inline genérico
 - ... (são 30)

<http://w3c.github.io/html/textlevel-semantics.html>
<http://www.w3schools.com/tags/>

HTML 5.1 – Conteúdos embebidos

- Área em que a HTML5 tem muitas novidades
 - `<picture>`, `<source>`, `` — imagens
 - `<iframe>` — contexto aninhado (um documento dentro do documento)
 - `<embed>` — aplicação ou conteúdo externo (via plugin)
 - `<object>`, `<param>` — um recurso externo (imagem, contexto aninhado, ou recurso processado por plugin)
 - `<video>`, `<audio>`, `<source>` e `<track>` — integração de audio e vídeo sem plugins
 - `<map>` e `<area>` — image maps
 - `<math>` — embeber MathML
 - `<svg>` — embeber SVG (vector graphics)

<http://w3c.github.io/html/semantics-embedded-content.html>
<http://www.w3schools.com/tags/>

HTML 5.1 – mais elementos

- Anotar edições
 - `<ins>` — texto inserido
 - `` — texto apagado
- Elementos interactivos:
 - `<details>` e `<summary>` — conteúdo que pode ser mostrado/escondido pelo utilizador
 - `<menu>` e `<menuitem>` — menus
 - `<dialog>` — uma janela de diálogo (mal suportado)
- Ligações (Links):
 - atributo `href` em elementos `a` ou `area`

<http://w3c.github.io/html/index.html#contents>
<http://www.w3schools.com/tags/>

HTML 5.1 – mais elementos

- Tabelas:
 - `<table>` — tabelas
 - ... (muitos elementos para definir as tabelas)
- Formulários:
 - `<form>` — formulários
 - `<fieldset>` — agrupa elementos relacionados num formulário
 - `<progress>` — a progressão do trabalho desencadeado pelo formulário
 - `<meter>` — uma medida escalar ou fracção (espaço em disco, resultado de uma votação, ...)
 - ... (muitos mais elementos para definir formulários)

<http://w3c.github.io/html/index.html#contents>
<http://www.w3schools.com/tags/>

Formulários...

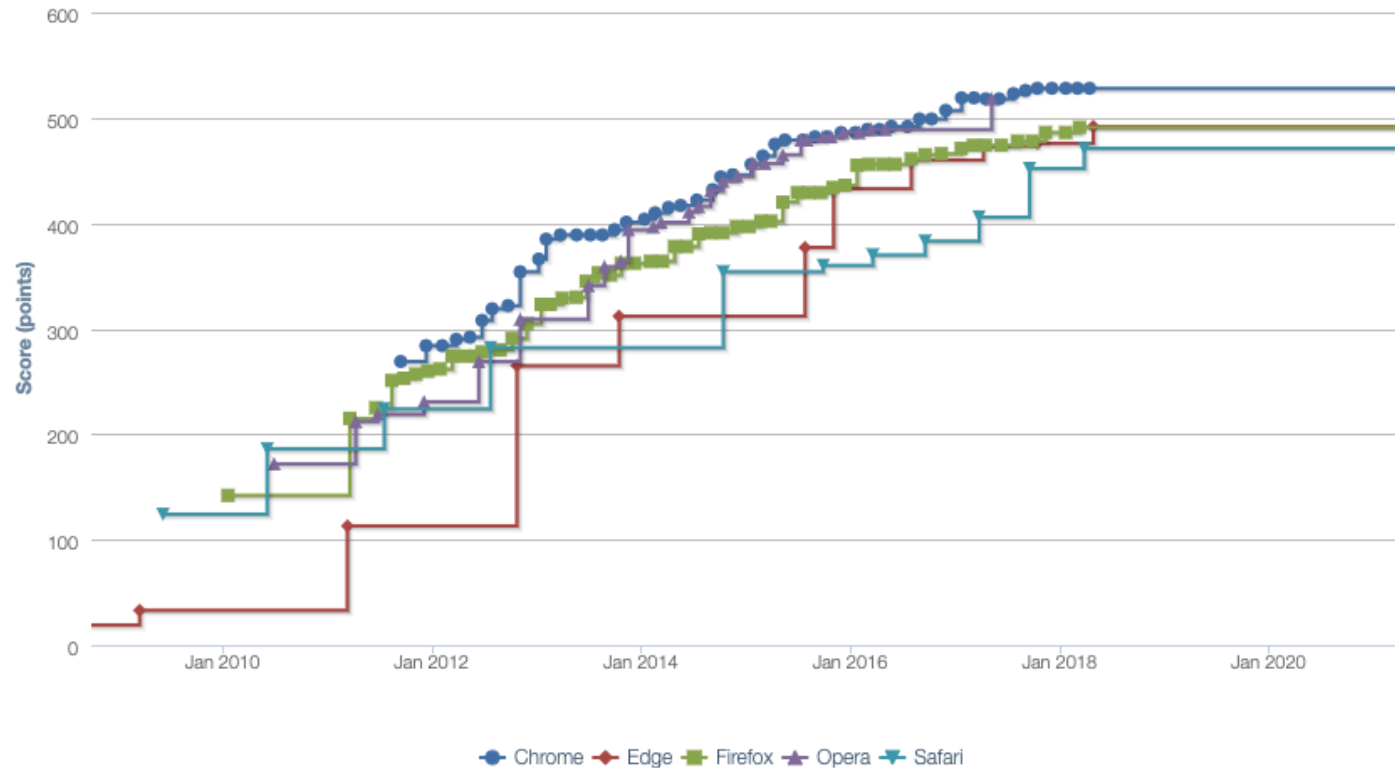
```
<form action="demo_action" autocomplete="on">  
  <label for="nameid" class="realce nome">Nome:</label>  
  <input type="text" id="nameid"  
    name="name"  
    placeholder="Indique o seu nome..."  
    required />  
  
  <label for="email">Email:</label>  
  
  <input type="email" id="email" name="email" />  
  
  <label for="password" class="realce">Password:</label>  
  
  <input type="password" id="password"  
    name="password" required />  
  
  <input type="submit" value="Login" />  
</form>
```



Nome: Email: Password:

Compatibilidade dos browsers

- <http://html5test.com>

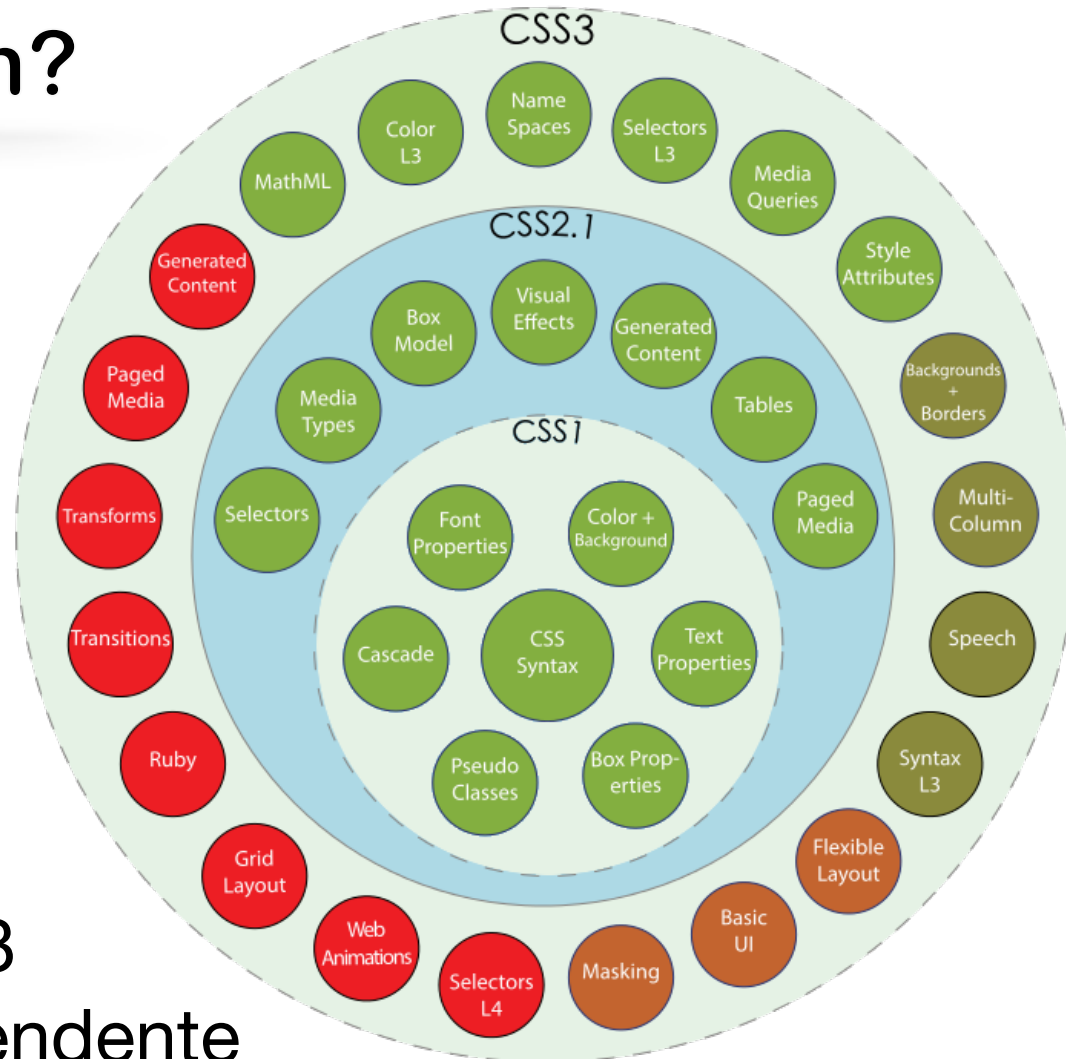


- 
– <http://validator.w3.org>

CSS

CSS Level 3 version?

- A CSS deixou de ter versões (níveis W3C)
 - CSS3 usado para distinguir das versões anteriores (CSS 2.x)
- Cada módulo da CSS3 evolui de forma independente



Cascading Style Sheets (CSS)

- Estilos (*Styles*) definem formatação dos elementos HTML
- Estilos podem ser guardados separadamente do HTML em *Style Sheets*:
 - Ficheiros .css
 - Permitem controlar estilo e *layout* de multiplas páginas
- Aplicação dos estilos dá prioridade a (*Cascading*):
 1. estilo *Inline* (*dentro* do elemento HTML)

```
<dt style="margin-bottom:20px;">...</dt>
```
 2. *style sheet* interna (definido na `<head>`)

```
<style type="text/css">  
    td {font-size: 0.8em;}  
</style>
```
 3. *style sheet* externa

```
<link href="mystyle.css" rel="stylesheet" type="text/css" />
```
 4. *estilo por omissão do browser*

Exemplo

- [Home page](#)
- [Musings](#)
- [My town](#)
- [Links](#)

My first styled page

Welcome to my styled page!

It lacks images, but at least it has style. And it has links, even if they don't go anywhere...

There should be more here, but I don't know what yet.

Made 5 April 2004 by myself.

```
<!DOCTYPE html>
<html>
  <head>
    <title>My first styled page</title>
    <!--link rel="stylesheet" href="
  </head>
  <body>
    <nav>
      <ul class="navbar">
        <li><a href="index.html">Home</a>
        <li><a href="musings.html">Musings</a>
        <li><a href="town.html">My town</a>
        <li><a href="links.html">Links</a>
      </ul>
    </nav>
    <main>
      <h1>My first styled page</h1>
      <p>Welcome to my styled page!</p>
      <p>It lacks images, but at least it has style. And it has links, even if they don't go
anywhere&hellip;</p>
      <p>There should be more here, but I don't know
what yet.</p>
    </main>
    <address>Made 5 April 2004 by myself.</address>
  </body>
</html>
```

```
body {
  font-family: Georgia, "Times New Roman", Times;
  color: purple;
  background-color: #d8da3d }
h1 {
  font-family: Helvetica, Geneva, Arial,
  SunSans-Regular, sans-serif }

nav {
  padding: 0;
  margin: 0;
  width: 10em }
nav > ul > a {
  text-decoration: none }
a:link {
  color: blue }
a:visited {
  color: purple }
address {
  margin-top: 1em;
  padding-top: 1em;
  border-top: thin dotted }

@media (min-width: 400px) {
  body { padding-left: 12em; }
  nav {
    position: absolute;
    top: 2em;
    left: 1em; }
  nav > ul {
    list-style-type: none; }
  nav > ul > li {
    background: white;
    margin: 0.5em;
    padding: 0.3em;
    border-right: 1px solid black; }
}
```

- [Home page](#)
- [Musings](#)
- [My town](#)
- [Links](#)

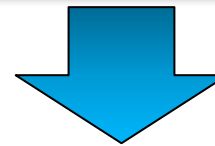
My first styled page

Welcome to my styled page!

It lacks images, but at least it has style. And it has links, even if they don't go anywhere...

There should be more here, but I don't know what yet.

Made 5 April 2004 by myself.



[Home page](#)

[Musings](#)

[My town](#)

[Links](#)

My first styled page

Welcome to my styled page!

It lacks images, but at least it has style. And it has links, even if they don't go anywhere...

There should be more here, but I don't know what yet.

Made 5 April 2004 by myself.

```
body {
  font-family: Georgia, "Times New Roman", Times, serif;
  color: purple;
  background-color: #d8da3d }
h1 {
  font-family: Georgia, "Times New Roman", Times, serif;
```

[Home page](#)
[Musings](#)
[My town](#)
[Links](#)

My first styled page

Welcome to my styled page!

It lacks images, but at least it has style. And it has links

There should be more here, but I don't know what yet.

Made 5 April 2004 by myself.

```
border-top: thin dotted }
@media (min-width: 400px) {
  body { padding-left: 12em;}
  nav {
    position: absolute;
    top: 2em;
    left: 1em;}
  nav > ul {
    list-style-type: none; }
  nav > ul > li {
    background: white;
    margin: 0.5em 0;
    padding: 0.3em;
    border-right: 1em solid black }
}
body { transition: padding-left 1s linear 0.1s; }

ul.navbar { transition: width 1s linear 0.1s; }
```

- Home page
- Musings
- My town
- Links

My first styled page

Welcome to my styled page!

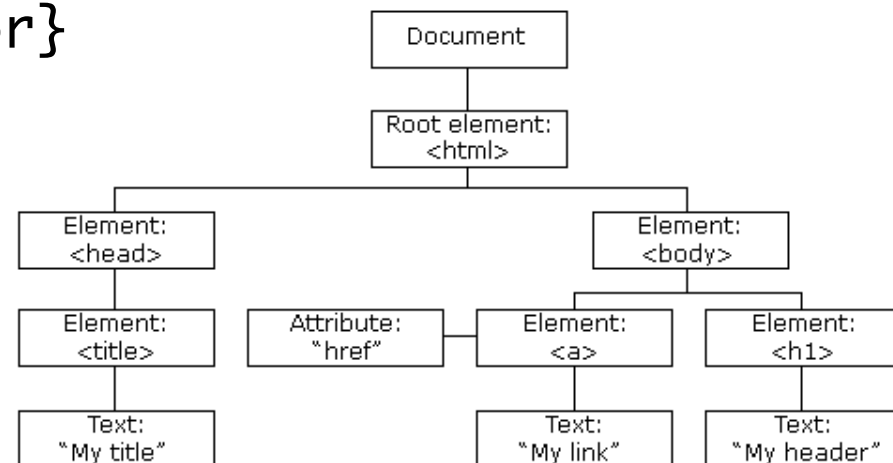
It lacks images, but at least it has style. And it has links, even if they don't go anywhere...

There should be more here, but I don't know what yet.

Made 5 April 2004 by myself.

Style Sheets – Sintaxe

- selector {propriedade: valor}
- Selectores
 - Universal
 - faz pattern matching com todos os elementos:
CSS: * {color: red;}
 - Selecção por tag
 - CSS: h1 {color: red;}
 - Selecção por classe
 - HTML: <p class="right">Alinhado à direita.</p>
 - CSS: .right {text-align: right;}
 - Selecção por identificador
 - HTML: <h1 id="chapter1">Título</h1>
 - CSS: #chapter1 {text-align: center;}



CSS – Combinar Selectores

- Múltiplos selectores
 - Regra aplica-se a cada um dos elementos identificados
 - CSS: h1, h2 {color: red}
- Selectores descendentes
 - Identifica um elemento dentro de outro
 - CSS: h1 em {colour: orange}
- Selectores filhos
 - Identifica um elemento que é filho de outro:
 - CSS: body > p {line-height: 1.3;}
- Irmãos adjacentes:
 - CSS: h1 + h2 {margin-top: -5mm;}
- Tags com uma dada classe
 - CSS: p.center { text-align:center; }

CSS – Selectores (pseudo-classes)

`selector:pseudo-class {property:value;}`

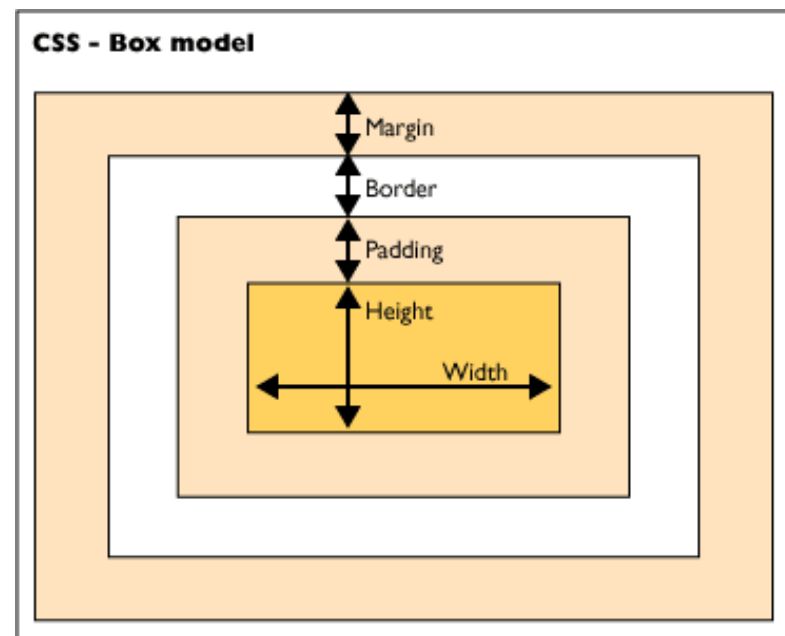
- `:first-child`
 - `div > p:first-child {text-indent: 0;}`
- `:link` e `:visited`
 - `a:link {color: red}`
 - `a:visited {color: brown}`
- `:hover`, `:active` e `:focus`
 - `a:hover {color: yellow;}`
 - `a:active {color: lime;}`
 - `a:focus {background: yellow;}`
 - `a:focus:hover {background: white;}`

CSS – Propriedades

- Estilo
 - Cores
 - Fontes
 - Espaçamento
 - Transformações
 - Transições e animações
 - ...
- Layout
 - Posicionar os elementos na página
 - Acabaram-se as tabelas!(!?)

CSS – Layout

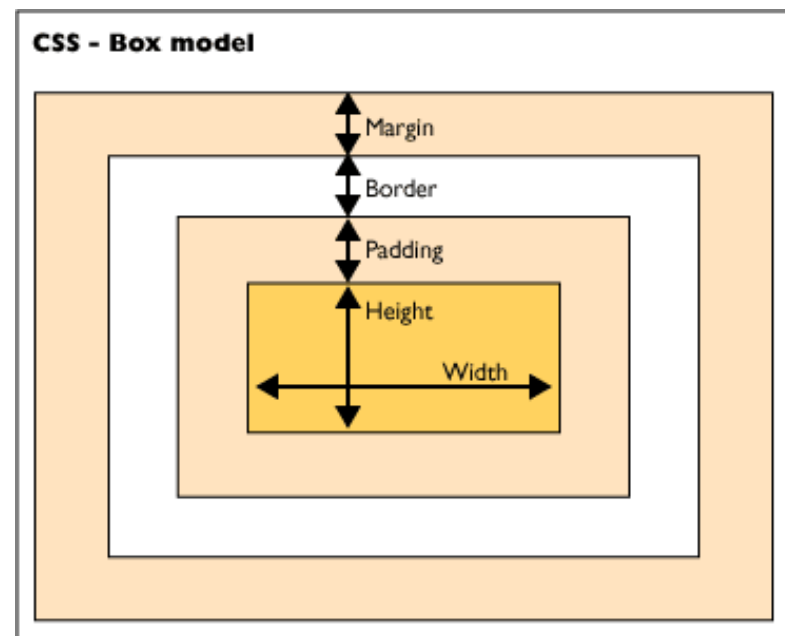
- Box model
 - Elementos HTML vistos como “caixas”
 - Propriedades CSS relevantes:
 - width, height – **do conteúdo!**
 - padding (padding: 12px 0px 12px 12px)
 - padding-X (padding-top: 12px)
 - border, border-X (border: 1px dotted black)
 - border-width, border-X-width (border-width: medium)
 - border-color, border-X-color (border-top-color: lime)
 - border-style, border-X-style (border-style: dashed)
 - margin, margin-X



X = {top, bottom, right, left}

CSS – Layout

- Box model
 - Largura efectiva dos elementos é superior à propriedade `width`!... ☹
 - $\text{largura} = \text{width} + \text{padding} + \text{border} + \text{margin}$
 - (mas... `box-sizing: border-box`)
 - `box-sizing: border-box`
 - `width` inclui `padding` + `border` (mas não `margin`)



X = {top, bottom, right, left}

CSS – Layout

- Por omissão elementos são posicionados sequencialmente
- Utilizam-se propriedades para alterar o posicionamento
 - display: none | inline | block | **flex**
 - position: static | relative | absolute | fixed | inherit
 - float: left | right | none
 - clear: left | right | both | none
 - visibility: visible | hidden | collapse
- **Flexbox layout module** (display: flex)
 - flex-direction: row | column
 - justify-content: flex-start | flex-end | center |
space-between | space-around
 - flex-wrap: nowrap | wrap | wrap-reverse

CCS – media queries

- `@media mediaType and (media features) { regras CSS }`
- `MediaType`
 - `(not | only) all | print | screen | speech`
- `Media features`
 - Operadores lógicos: `and`, `not`
 - `width | max-width | min-width` (o mesmo para `device-width`)
 - `height | max-height | min-height` (o mesmo para `device-height`)
 - `orientation | aspect-ratio | resolution`
 - `Color | color-index | monochrome`
 - ...
- Exemplos (utiliza-se `min-` e `max-` em vez de `>` e `<`)

```
@media screen and (max-width: 400) { ... }  
@media screen and (min-width: 400) and (max-width: 700) { ... }  
@media screen and (min-width: 700) { ... }
```

<http://www.w3.org/TR/css3-mediaqueries/>
https://www.w3schools.com/html/html_responsive.asp

Pre-processadores CSS

- Permitem utilizar linguagens mais expressivas para definir as regras CSS
 - e.g. evitar repetição de código
- Compilam para plain CSS
- Duas opções:
 - **Sass: Syntactically Awesome Style Sheets**
 - <http://sass-lang.com>
 - **Less:**
 - <http://lesscss.org>



Sass essencial

- Variáveis
 - `$border-def: 1px solid black;`
`td { border: $border-def; padding: $half-skip; }`
- Nesting (aninhamento de regras – incluindo @media!)
 - `main { ...`
`table { ... }`
`}`
 - em vez de: `main { ... }`
`main table { ... }`
- Mixins ('funções')
 - `@mixin flex-config($direction) { display: flex;`
`justify-content: space-between; flex-direction: $direction; }`
`nav { ...`
`@include flex-config(row);`
`...`
`}`
- @extend (herança)
 - `td { border: $border-def; padding: $half-skip; }`
`th { @extend td;`
`background-color: lightgray; }`

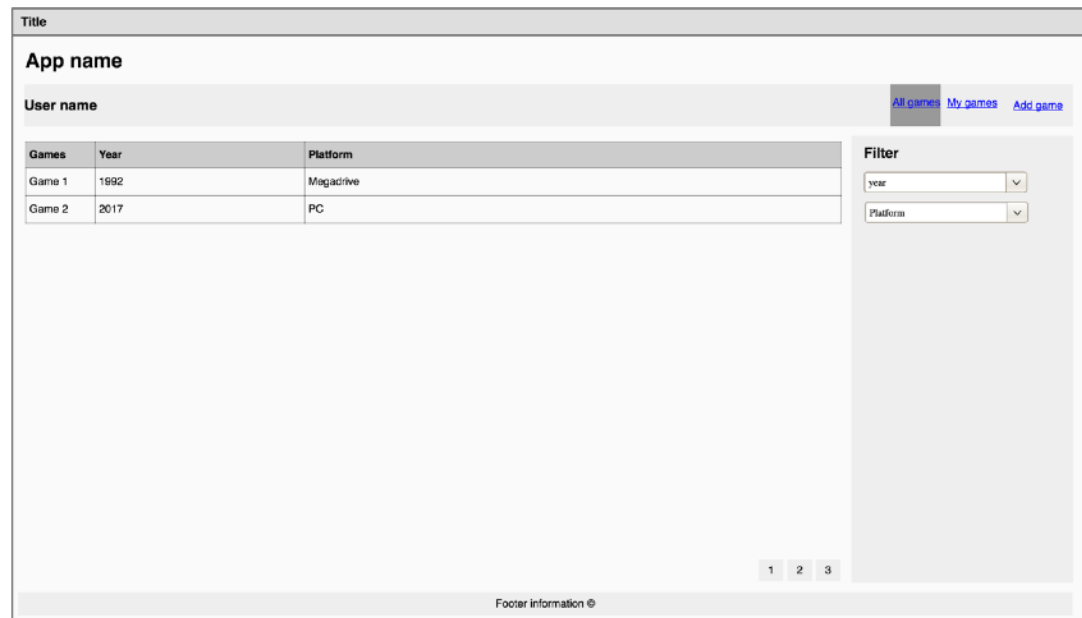
Exercício

- Tutorial 1

1. Construir página HTML com markup semântico
2. Definir layout em CSS
3. Definir estilo dos elementos em CSS
4. Acrescentar funcionalidades responsive
5. Melhorar a solução com Sass

- Tutorial 2

1. Repetir o exercício com Bootstrap



Exercício

- Tutorial 2
 - Construir a página com Bootstrap