

Frequência 1

CSS

Estrutura



Seletores

- Usados para encontrar elementos HTML baseados nos seguintes critérios:
 - Por tipo de elemento:
 - `p {...}`
 - Por identificador:
 - `<p id="p1">...</p>`
 - `#p1 {...}`
 - Por classe:
 - `<p class="p1">...</p>`
 - `.p1 {...}`
 - Por atributo:
 - `[atributo*="valor"]`: `atributo` cujo valor contém `valor`;
 - `[atributo$="valor"]`: `atributo` cujo valor é sufixado por `valor`;
 - `[atributo^="valor"]`: `atributo` cujo valor é prefixado por `valor`;
 - `[atributo~="valor"]`: `atributo` cujo `valor` é uma lista de palavras separadas por espaços, uma das quais é **exatamente** `valor`;
 - `[atributo]="valor"`: `atributo` cujo valor pode ser exatamente `valor` ou começado com `valor` e exatamente seguido de um hífen;
 - Atributos específicos: `seletor[atributo]`
 - `...`
 - `a[target] {...}`
 - Valores de atributos:
 - `...`
 - `...`
 - `a[target="_blank"] {...}`
 - Baseado em combinadores:
 - Seletor descendente (`>`):
 - Combina dois seletores tal que os elementos correspondidos pelo segundo seletor são selecionados se tiverem um elemento parente ("ancestor") correspondente ao primeiro seletor;
 - Exemplo:

```
<div class="box"><p>Text in .box</p></div>
<p>Text not in .box</p>

<style>
.box p {
  color: red;
}
```

```
}  
</style>
```

Text in .box

Text not in .box

▪ Seletor de filho direto (>):

- Corresponde apenas aos elementos correspondentes ao segundo seletor que são filhos diretos dos elementos correspondentes ao primeiro seletor;
- Exemplo:

```
<ul>  
  <li>Unordered item</li>  
  <li>Unordered item  
    <ol>  
      <li>Item 1</li>  
      <li>Item 2</li>  
    </ol>  
  </li>  
</ul>  
  
<style>  
ul > li {  
  border-top: 5px solid red;  
}  
</style>
```

- Unordered item
- Unordered item
 - 1. Item 1
 - 2. Item 2

▪ Seletor irmão direto (+):

- Corresponde apenas aos elementos correspondentes ao segundo seletor que são o irmão direito do primeiro seletor;
- Exemplo:

```
<article>  
  <h1>A heading</h1>  
  <p>Veggies es bonus vobis, proinde vos postulo essum magis kohlrabi welsh onion daikon amaranth tatsoi tomatillo  
    melon azuki bean garlic.</p>  
  
  <p>Gumbo beet greens corn soko endive gumbo gourd. Parsley shallot courgette tatsoi pea sprouts fava bean collard  
    greens dandelion okra wakame tomato. Dandelion cucumber earthnut pea peanut soko zucchini.</p>  
</article>  
  
<style>  
h1 + p {  
  font-weight: bold;  
  background-color: #333;  
  color: #fff;  
  padding: .5em;  
}  
</style>
```

A heading

Veggies es bonus vobis, proinde vos postulo
essum magis kohlrabi welsh onion daikon
amaranth tatsoi tomatillo melon azuki bean
garlic.

Gumbo beet greens corn soko endive gumbo gourd.
Parsley shallot courgette tatsoi pea sprouts fava bean
collard greens dandelion okra wakame tomato.
Dandelion cucumber earthnut pea peanut soko
zucchini.

- Seletor irmão geral (~):

- Selecciona todos os irmãos de um elemento que não estão adjacentes ao mesmo;
- Exemplo:

```
<article>
  <h1>A heading</h1>
  <p>I am a paragraph.</p>
  <div>I am a div</div>
  <p>I am another paragraph.</p>
</article>

<style>
h1 ~ p {
  font-weight: bold;
  background-color: #333;
  color: #fff;
  padding: .5em;
}
</style>
```

A heading

I am a paragraph.

I am a div

I am another paragraph.

- Por pseudo-classes:

- Uma pseudo-classes é uma *keyword* adicionada a um seletor que especifica um estado especial do(s) elemento(s) selecionado(s);
- Exemplos: `:link`, `:visited`, `:hover` ;

- Por pseudo-elementos:

- Um pseudo-elemento é uma *keyword* adicionada a um seletor e permite estilizar uma parte específica do(s) elemento(s) selecionado(s);
- Exemplos: `::first-letter`, `::before`, `::after` ;

Integração

- **Local:** utilização do atributo `style` no elemento a estilizar;
- **Interna:** utilização do elemento `<style>` que é filho direto de `<head>` ;

- **Externa:** é criado um ficheiro à parte que é posteriormente adicionado ao `<head>` do documento HTML com o elemento `<link>` (é utilizado o atributo `href` para referenciado o ficheiro com as regras CSS, e.g. `<link rel="stylesheet" href="style.css">`);

Cascata

- É um algoritmo que define como combinar valores de propriedades de diferentes origens;
- Para “desempatar” qual a regra aplicada a um elemento, as seguintes regras são aplicadas em cascata:
 - **Cálculo de especificidade:** as mais específicas prevalecem; a especificidade é representada por 3 dígitos:
 1. Número de seletores por ID presentes;
 2. Número de seletores por classe, atributos e pseudo-classes;
 3. Número de seletores por elemento e pseudo-elemento.
 - Exemplo:
 - `#red.blue`: 110
 - `#green`: 100
 - `div.yellow#red`: 111
 - **Ordem de especificação:** se forem iguais, prevalece a última;
 - **Anotação `!important`:** utilizada para forçar uma regra específica.

Media Queries

- Utiliza-se a regra `@media` para incluir um bloco de propriedades CSS somente se uma determinada condição for verdadeira;

Sintaxe

```
@media not|only mediatype and (mediafeature and|or|not mediafeature)
{ ... };
```

- O resultado da *media query* será verdadeiro se o tipo de mídia especificado corresponder ao tipo de dispositivo no qual o documento é sendo exibido e todas as expressões na consulta de mídia forem verdadeiras;
- Quando uma consulta de mídia é verdadeira, as regras CSS correspondentes são aplicadas, seguindo as regras normais de cascata;
- Exemplo:
 - Esconder um elemento `<div>` com a classe `test` quando o ecrã for `>= 600px`:

```
@media screen and (max-width: 600px)
{
  div.test {
    display: none;
  }
}
```

JavaScript

Document Object Model

- Quando o browser carrega uma página HTML, cria um modelo de objetos associado do documento HTML (o Document Object Model, **DOM**);
- É através do DOM que podemos aceder e alterar, de forma dinâmica, o conteúdo da página;

Objetos

- `window`: objeto global que representa a "janela" do navegador e fornece métodos para a controlar;
- Browser Object Model (BOM):
 - `navigator`: fornece informações sobre o sistema operativo e navegador;
 - `location`: permite ler o URL atual e redirecionar o navegador para o novo URL;
- Document Object Model: modelo de objetos padrão e interface de programação para documentos HTML;
 - **Métodos de acesso:**

MÉTODO	DESCRIÇÃO
<code>getElementById</code>	Obtém uma referência para um elemento HTML a partir do seu atributo <code>id</code>
<code>getElementsByName</code>	Obtém uma referência para um conjunto de elementos HTML a partir do nome da sua tag
<code>getElementsByClassName</code>	Obtém uma referência para um conjunto de elementos HTML a partir do seu atributo <code>class</code>
<code>querySelector</code>	Obtém uma referência para um elemento HTML a partir de um seletor CSS específico
<code>querySelectorAll</code>	Obtém uma referência para um conjunto de elementos HTML a partir de um seletor CSS específico

- Métodos de adição/remoção de elementos:

MÉTODO	DESCRIÇÃO
<code>document.createElement</code>	Cria um elemento HTML
<code>document.createTextNode</code>	Cria um <i>node</i> de texto
<code><element>.removeChild</code>	Remove um elemento HTML
<code><element>.appendChild</code>	Adiciona um elemento HTML
<code><element>.replaceChild</code>	Substitui um elemento HTML

Eventos

- Quando um utilizador manipula uma página, são disparados eventos que podem ser tratados por meio de código JavaScript;
- Existem várias maneiras de tratar de eventos:
 - Atributos de elementos no HTML: adicionar o atributo `onClick` e especificar a função;
 - *Listeners* de eventos: `document.<elementSelector>.addEventListener(event, function, useCapture)`;
 - `event`: *string* com o tipo do evento (sem o prefixo *on*);
 - `function`: *callback* executada em resposta ao evento;
 - `useCapture`: indica como é feita a propagação do evento
 - `false` (**default**): o evento do elemento mais interno é tratado em primeiro lugar, seguidamente do externo (**bubbling**);
 - `true`: o evento do elementos mais externo é tratado em primeiro lugar, seguidamente do interno (**capturing**);

JQuery

- Funcionalidades:
 - Manipulação DOM/CSS;
 - Efeitos e animação;
 - AJAX.

Sintaxe

`$seletor.action()`

- `$`: acede ao objeto global do JQuery;
- `seletor`: seleciona um elemento HTML;
- `action()`: ação a ser realizada nos elementos HTML;

- Exemplo:
 - Acesso e modificação: `$('#<p>Teste</p>').appendTo('div');`
 - Eventos: `$("#p").click(() => $(this).hide());`
 - Adicionar CSS: `$("#p").addClass("backgroundColor");`

Node

- Permite usar JavaScript do lado do servidor;
- Permite aos programadores desenvolver aplicações em rede, em tempo real e de alta escalabilidade de forma simples e rápida;
- Inclui I/O não bloqueante;

Express

- Framework que disponibiliza um conjunto de funcionalidades para desenvolver aplicações *web* e *mobile*;
- Permite criar *middlewares* para responder a pedidos HTTP;

```
import express from 'express';
const app = express();

app.get('/', (req, res, next) => {
  // ...
  next(); // Calls the next middleware
})
```

- Se o *middleware* atual não terminar a resposta, este deve chamar `next()` de modo a chamar o próximo *middleware*.
- Define rotas que permitem executar diferentes ações baseado no método HTTP do pedido e do URL;

HTTP Requests

- Conjunto de informações enviados ao servidor pelo *browser*;
- Constituído por:
 - Linha do pedido: contém o método HTTP, o caminho do recurso desejado e a versão do protocolo utilizado;
 - Cabeçalhos (*headers*): dão informações adicionais sobre o cliente e/ou pedido;
 - Corpo do pedido (*body*): permite enviar dados com o método `POST` durante a comunicação;

HTTP Responses

- Representa o objeto que uma aplicação Express envia quando recebe um pedido HTTP;
- Se a resposta não for enviada, a solicitação do cliente será deixada em suspenso;

body-parser

- Permite-nos aceder à informação de um formulário no lado do servidor de uma forma mais legível;
- Os dados enviados são transmitidos no *request body* do pedido HTTP e podem ser acedidos a partir de `req.body` do *middleware*;

Notas:

- Se não for passado o argumento corresponde à rota para a chamada a `app.use()` (i.e. apenas temos a *callback*), a *callback* é executada sempre que é recebido um pedido.