Aula 1

Http - Hypertext Transfer Protocol: protocolo utilizado para transferir documentos de hipertexto e os seus recursos de máquinas remotas.

Caracteristicas: protocolo da camada de aplicação
modelo de funcionamento baseado em pedido-resposta
cabeçalho das msg é texto puro (não binário)
não guarda o estado entre conexões distintas, cada conexão é vaz para o servidor

HTML Hyper Text Markup Language. Linguagem de marcação utilizada para produzir página web

Caracteristicas: marcadores / tags sempre entre < >, etiquetas/marcadores são responsaveis pela formatação de linguagem.
<marcador> texto </marcador>

um elemento é formado por um nome de etiqueta, atributos, valores e filhos. Os atributos modificam os valores padrão dos elementos e os valores caracterizam essa mudança
> elemento simples (não possui filhos): <hr/>
> elemento c/ atributos: Universidade de Aveiro
> elemento c/ filhos e atributos A <a href ... > Universidade de Aveiro é a minha Universidade

Os elementos básicos de um documento html:
<html> define o inicio de um doc. html, indica ao navegador que todo o conteúdo posterior ao marcador deve ser tratado como uma série de marcadores html
<head> define o cabeçalho - possui inf. sobre o documento que vai ser representado
<body> define o conteúdo principal, parte que é exibida no navegador, neste marcador podem-se definir atributos comuns a toda a página (cor, fundo, margem)

Elementos do cabeçalho: • <title> título de pag., exibido na barra de título do browser

• <style type = "text/css"> formatação das etiquetas em css

• <script type = "text/javascript"> programação de certas funções da pag. escritas em javascript

• </link> ligações da pag. com outros arquivos

• <meta> propriedades da pag.

Etiquetas do corpo: • <h1>, <h2>, ... <h6> cabeçalhos e títulos

• ... novo parágrafo

•
 impõe uma quebra de linha num texto

• <a> ... cria hiperligação para um outro local

• <table> ... </table> cria uma tabela

linhas criadas com <tr> ... </tr>

células criadas com <td> ... </td>

cabeçalhos das colunas <th> ... </th>

• ... , ... determinam uma divisão na pag. a qual pode possuir variadas formatações

• <img/> insere imagem

• <textarea> ... </textarea> delimita uma caixa de texto (com + de 1 linha)

- ... negrito
- ... itálico
- <u> ... </u> sublinhado
- <s> ... </s> riscado

} formatação de ktre

Parametros de hiperligação href destino

Aula 1 ver (35 - 38)

Aula 2

HTML (continuação):

Listas: ordenadas <ol>... : elementos da lista
(slide 5) tipo de marcador da lista. - Atributo type
(valores possíveis: 1, A, a, I, i)

não ordenadas <ul>... : elementos
(slide 6) Atributo type
(valores possíveis (disc, circle, square)

dicionários / listas de definições <dl>...</dl>
(slide 7) marcador de identificação do termo <dt>
marcador para a definição do termo <dd>

Caracteres Especiais: Há um conjunto de caracteres que, ou ñ possuem representação direta ou ñ pertencem a todos os alfabetos, por isso precisam de uma forma especial de representação

© → © → fim do caracter + no slide 11
início do caretes → corpo do caracter

Formulários: <form>...</form>, utilizados para recolha de informação por parte dos utilizadores.

Atributos: Name - nome do formulário
Action - endereço de identidade que vai processar a informação*
Method - forma de envio de dados para a entidade processadora. (GET, POST, PUT, DELETE

* entidade. da UA no slide 13

Campos de um formulário:

* input → dos principais responsáveis pela recolha de informação
  > sintaxe: <input type="???" />
  > atributos type dependendo do valor assumido por este campo o marcador altera-se (button, checkbox, file, hidden, image, password, radio, range, reset, submit, text)

(Slide 15)

* Texto - Linha Simples <input type="text" /> : permite a inserção de uma linha de texto.

> atributos: type - define o tipo de input

name - define o nome do atributo (obrigatório)

value - (pode estar vazio ou omisso), quando preenchido contém o valor a apresentar pela linha de texto

placeholder - texto que aparece na caixa quando está vazia, serve para ajuda do utilizador

(Slide 16)

* Texto - password <input type="password" /> : permite a inserção de uma linha de texto sem que o seu conteúdo possa ser lido na interface

> atributos type - define o tipo de input

name - define o nome do atributo (obrigatório)

value - (normalmente vazio) - não faz sentido introduzir um texto que se pretende secreto

placeholder - (= a conter)

* Texto - Hidden <input type="hidden" /> : permite a inserção de texto sem que o seu conteúdo seja mostrado na interface

> atributos - type

- name

- value contem o valor a enviar para entidade processadora

* Texto multilinha <textarea> ... </textarea> : permite a inserção de um texto em várias linhas

> atributos - name

- rows número de linhas

- cols número de colunas

- placeholder

* Botão de submit <input type="submit" /> : é o que permite o envio dos dados do formulário para entidade processadora

> atributos - type

- name define o nome do atributo (opcional)

- value contém o texto do botão, tb enviado para entidade processadora, caso o botão tenha nome

* Botão reset <input type="reset"/> : permite reverter o estado atual de um formulário para o seu estado inicial

> atributos type
- name - (opcional)
- value - texto a mostrar no botão

* Secções do formulário <fieldset> ... </fieldset> : permite criar secções dentro de um formulário

> Marcadores filhos: - cabeçalho da seleção : <legend>...</legend>
- todos os campos do formulário

> Atributos: name - define o nome do fieldset

* Checkboxes ☐ : permite a escolha de 0 ou mais opções de uma lista

> Atributos - type define o tipo de input
- name define o nome de atributo, obrigatório
- value contém o valor a enviar para a entidade processadora

* Radio Boxes ☐ : permite a escolha de 0 ou 1 opção de uma lista

> Atributos - type
- name
- value

* Botão com imagem <input type="image"/> : o botão com uma imagem comporta-se como um botão de submit mas quando a informação é enviada à entidade processadora, são enviadas as coordenadas do ponto em que a imagem foi selecionada:

> Atributos - type
- name define o atributo
- value nome - opcional

* Botão genérico <input type="button"/> : o botão genérico não possui um comportamento associado por omissão, depende do que for configurado pelo utilizador

> Atributos - type define o tipo do input, para outras opções é reset
- value contém o texto a mostrar no botão

* Botão genérico <input type="file" /> : permite o envio de ficheiros para o servidor

> Atributos : - type (= anteriores)

- name contém o nome do ficheiro a enviar para a entidade processadora

Img. Para os ficheiros serem recolhidos no servidor incluir marcador <form> o atributo enctype com o valor multipart/form-data

* Listas de valores - seleção simples <select>...</select> : importantes quando se pretende que o utilizador selecione valores dentro de uma gama pré definida.

> As opções da lista são delimitadas por marcadores <option>... </option>

> Atributos : - valor valor a enviar à entidade processadora

- selected (opcional). Indica a opção pré-selecionada, toma sempre o valor selected

* Listas de valores - seleção multipla <select multiple="multiple"> ... </select> : lista em que o utilizador pode escolher + do que um elemento.

> Atributos: - multiple indica que é possível escolher + que um elemento na lista, caso esteja presente, toma sempre o valor "multiple"

Usabilidade e acessibilidade:

Labels → utilizados para associar um texto explicativo a um marcador de um formulário, a associação entre ambos faz-se através do ID. Ou seja, para além do name, os marcadores passam também a necessitar de um atributo ID que pode, ou não ser = ao name.

→ se o campo associado for um input do tipo "text" / <textarea> o campo respetivo fica selecionado

se for um input do tipo radio a opção fica imediatamente selecionada

se for um input do tipo "checkbox" a troca de estado (selecionado/ desselecionado)

Os Labels são obrigatórios para que um formulário seja considerado "acessível" nos testes respetivos.

Tabindex → controla a ordem por que os campos e hiperligações são apresentados sempre que carregamos na tecla [Tab].

São um elemento de auxílio e de usabilidade, pois a utilização de rato como dispositivo apontador nem sempre é uma opção.

Aula 2

CSS → Cascading Style Sheets (folhas de estilos)

- Permite fazer uma separação entre a estrutura do documento HTML e a sua representação.

↳ o HTML define que um elemento é um cabeçalho de nível 1 (<h1></h1>) ou um paragrafo ( )

↳ a CSS controla as fontes, cores, margens, linhas, alturas, larguras, imagens de fundo, o seu posicionamento entre outras.

Formas de definição e hierarquia: As instruções CSS podem ser definidas de 3 maneiras distintas:

Global colocadas num ficheiro externo que pode depois ser associado a 1 ou + documentos html

Document colocadas dentro de um marcador <style></style> localizado no <head>

In-line colocadas nas linhas do ~~documento~~ marcador html.

A instrução que prevalece é a que está + proxima do elemento.

Cores: A propriedade color define a cor de um elemento.

Documento / Global — p { color: #F0FFFF ; }

In-line: ...

- Os 1os dois símbolos no código HTML representam a intensidade da cor vermelho 00 é o mínimo FF o + intenso
- O 3º e o 4º a intensidade de cor verde
- O 5º e o 6º representam a intensidade de azul

A cor também pode ser representada na forma decimal através de função rgb (rr, gg, bb), outra forma similar rgba (rr, gg, bb, tt), em que tt é a transparência e pode variar entre 0,0 (transparente) e 1,0 (opaco), as cores são separadas por uma vírgula.

As cores também podem ser definidas pelo seu nome.

Fundos:

- background-color
- background-image - url("url da imagem")
  Ex: url("http://grangetexturas....");
- background-repeat
  - > background-repeat: repeat-x repete-se na horizontal
  - > " " -y " - vertical
  - > background-repeat: repeat repete-se tanto na horizontal como na vertical
  - > background-repeat: no-repeat não se repete
- background-attachment
  - > background-attachment: scroll img move-se qd é arrastada
  - > " " : fixed img fica fixa qd é arrastada
- background-position
  - > background-position: 2cm 2cm img a 2cm da esquerda e 2cm para baixo
  - > " " : 50% 25% img. centrada na horizontal e a um quarto (25%) para baixo na pag.
  - > background-position: top right img. no canto superior direito

(15) É possível representar o background combinando diversas partes do mesmo.

Fontes de texto: • font-family - é uma propriedade usada para definir a lista das fontes a utilizar num marcador e qual a sua prioridade para apresentação.

NOTA Se a 1ª fonte não estiver definida/instalada, deverá ser usada a 2ª e assim por diante até ser encontrada uma fonte instalada.

- font-style - normal | italic | oblique | initial | inherit
- font-variant normal | small-caps | initial | inherit
- font-weight normal | bold | bolder | lighter | initial | inherit
  (200 = light, 400 = normal, 700 = bold)
- font-size
  - > medium | xx-small | x-small | small | large | x-large | xx-large smaller | larger | initial | inherit
  - > Valor numérico (10px, 8pt, 1.2cm, ...)
  - > % percentagem relativa ao elemento anterior (elemento pai) (80%, 75%)

Aula 3

Elementos HTML de agrupamento:

 ... → Utilizado para agrupar elementos "inline"

Ex: A minha mãe tem olhos de cor azeitona .

 ... → define uma divisão ou uma secção. (7)

Normalmente ocupam linhas inteiras e os browsers colocam uma quebra de linha antes e outra depois do elemento. Contudo isso pode ser mudado através de CSS.

CSS (continuação)

Seletores de Classe → Num doc. html, os seletores de classe permitem configurar apenas um conjunto de elementos pertencentes a essa classe e não todos os elementos de um determinado tipo.

Sintaxe: element { style properties }

Ex span {
background-color: Olive; (todos os elementos do tipo
color: #ffffff; (span) ficaram afetados por o
} estilo)

 A minha mae tem olhos de cor azeitona .

Sintaxe: element.classname { style properties } (10 e 11)

Ex: span.inBlue { background-color: blue; }

 Neste span o texto é azul.

 Neste span o texto fica na cor normal.

→ quando se usam classes, apenas os elementos pertencentes às mesmas são alterados / afetados.

(12)

Seletores de ID → O "ID de um elemento" deve ser único dentro de um documento, assim, este estilo destina-se a ser aplicado apenas num elemento em particular

NOTA: Classes nas definições de estilo começam com um .
ID, nome de um identificador nas definições de estilo começa com um #

Seletores CSS: (ver aplicação)

- Seletor adjacente → ou "próximo seletor irmão" irá abranger apenas o elemento especificado que segue imediatamente o antigo elemento especificado (14)

- Seletor irmão → o combinador ~ separa 2 seletores e aplica-se ao segundo elemento somente se ele é precedido pelo primeiro, e se ambos compartilharem um pai comum (15)

- Seletor filha → o combinador > separa 2 seletores e corresponde aos elementos que coincidem com o segundo seletor que são filhos diretos de elementos que coincidem com o primeiro (16)

- Seletor descendente (filho, neto, bisneto..) → o combinador " " (espaço) combina 2 seletores de modo a que o seletor combinado corresponde apenas aos elementos correspondentes ao segundo 2º seletor para o qual há um elemento ancestral combinando o 1º seletor.

Semelhantes aos seletores filhos mas não exigem relação entre elementos de pai - filho (17)

Pseudo Classes → é uma palavra-chave precedida de dois pontos (:) que é adicionada no final dos seletores para especificar que se deseja modelar os elementos selecionados somente quando estes estiverem num determinado estado. (ver melhor 19-20)

Pseudo Elementos → é uma palavra-chave precedida de dois dois pontos (::) que se aplica somente a uma parte do elemento, ex 1º letra, 1ª linha de um parágrafo, antes/depois de um elemento

:: after	:: first-letter	:: selection
:: before	:: first-line	:: backdrop

Elementos: (23)

Content: conteúdo de um elemento ou imagem

Padding: espaçamento entre o conteúdo e o bordadura

Border: bordadura à volta do elemento

Margin: distancia entre a border e o limite exterior do elemento

(exemplos 24 ao 29)

Dimensões dos elementos: controladas através de largura (width) e de altura (height) (30)

Posicionamento dos elementos: (31)

- absoluto - os pontos de referência são a página
- relativo - os pontos referência são o elemento pai dentro do qual o elemento está.

Barras de navegação: (exemplos do 33 ao 35)

Icons utilização de fontes como imagens.
(font awesome) (38)

Aula 4:

Responsive web design (RWD) é uma abordagem de web design que visa a elaboração de sites que forneçam:

> uma experiência de visualização e leitura fácil

> navegação optimizada com um mínimo de redimensionamento

> visão panorâmica e rolagem através de uma gama muito ampla de dispositivos (telemóveis, tablets, computador...)

Twitter Bootstrap: Contém um conjunto de marcadores HTML e modelos de design em CSS para tipografia, formulários, botões (....) entre outros componentes de interface.

Possui ainda um conjunto de extensões opcionais que obrigam à utilização de programação em JavaScript.

Funcionalidades Bootstrap:

Grelhas de informação: (20) grelha

Efeito na representação da informação: Os "containers" fornecem um meio para centrar o conteúdo do seu site. Use a classe .container para obter uma largura fixa ou a classe .container-fluid para obter uma largura total. (22)

> As linhas delimitadas pela classe "row" são grupos horizontais de colunas que garantem que as suas colunas estejam alinhadas corretamente.

> As classes de colunas: col-sm-, col-md-, col-lg-; indicam o número de colunas que pretende usar da 12 possíveis por linha.

> As larguras das linhas são definidas em %, são sempre fluidas e dimensionadas em relação a um elemento pai - podemos ter grelhas dentro de grelhas

> Os níveis da grelha são baseados em larguras mínimas, o que significa que se aplicam a esse nível e a todos os acima (Ex: especificamos só col-sm-4, aplica-se a todos os dispositivos peq. médios e grandes)

Jumbotron → elemento que pode ocupar toda a largura do browser (25,26)

No caso das imagens, o efeito Jumbotron é fortemente afetado pela diferença entre a relação largura x altura da imagem e a largura do dispositivo de visualização. (27 - 30)

Funcionalidades Bootstrap:

- **Carousel** o carroussel é um componente de apresentação de slides que comuta entre os diversos slides. Os carrosséis aninhados (carrosséis dentro de carrosséis) não são suportados. (37 - 36)

- Tipos e tamanhos de letras (37)
- Tables (38)
- Formulários (39 - 40)
- Botões (41 - 42)
- Tabs (43)
- Modal windows (44)
- Labels (45)
- Babges (46)
- Alert boxes (47)
- Panal (48)

Aula 5

Linguagens de programação:

→ O que são? Uma "linguagem de programação" é uma linguagem que possui uma sintaxe (formato) e uma semântica (significado), utilizada para expressar uma sequência de ações computacionais que formam um programa.

→ Conceitos básicos: compilador → verifica a sintaxe do código escrito para garantir que está de acordo com a semântica adequada. Se este estiver correto, gera um código executável a partir do código fonte (código executável não possui o código fonte ⇒ linguagens compiladas melhores para distribuir quando o programador não quer que o seu código seja público) (4)

interpretador → executa diretamente o código fonte escrito pelo programador; lê trechos do código fonte em tempo de execução, converte para um formato que o computador consegue ler e realiza a sua execução. (4)

Linguagem compiladas ou de script? Compilada precisa de um passo de compilação antes de executar; script não.

Linguagens script costumam ter performance inferior a linguagens compiladas pois exigem + passos para dispanilizar ao programa em tempo de ~~edição~~ execução. Porém, são muito + produtivas, pois eliminam a necessidade de compilar o código fonte de cada vez que uma alteração é feita.

Linguagens "tipadas" (typed): Operações aplicadas em estruturas de dados bem definidas e cada operação define os tipos de dados que deve receber. Já nas linguagens fracamente tipadas, operações aplicadas para qualquer estrutura de dados, porém podem falhar em tempo de execução caso a extrutura não suporte a operação. (6)

JavaScript: programa o comportamento das páginas web

Inclusão de javascript numa página html: O processo de inclusão é semelhante à da inclusão de CSS (<style></style>) nos nate caso usa-se <script></script>, normalmente no cabeçalho ou no final do documento.

Note: jquery 3.2.1 necessário

bootstrap 3.3.7

Versatibilidade vs Segurança:

- Elevada versatibilidade → pode ser executada em qualquer browser de qualquer sistema operativo, permite desenvolver aplicações que podem ser distribuídas de forma muito mais eficaz.

- Segurança reduzida → o código javascript é sempre enviado ao cliente na sua forma textual, podendo rápidamente ser copiado.

Para dificultar a leitura do código, protegendo a sua autoria, poupando ainda no espaço ocupado (não prejudica o carregamento, nem a apresentação) o código é muitas vezes "minimizado" (tradução livre de minified)

Ex: bootstrap.min.css, bootstrap.min.js

Linguagem Javascript: • baseada em instruções, terminando sempre com (;)
• case-sensitive (Sim ≠ sim ≠ SIM)

Sintaxe de linguagem Javascript:

- Declaração e atribuição de variáveis

> declaração → através da utilização da palavra reservada **var** seguida pelo nome-de-variavel

> atribuição → faz-se de modo convencional, <name-de-variavel> <operação> <valor>

- Funções: evitam replicação desnecessária; declaração através da palavra reservada function; elementos constituidos por um nome, uma lista de argumentos e um corpo

```
function  nome-função (arg1, arg2, arg3) {
    (corpo)
}
```

- Declaração de variáveis → o conteúdo das vareaveis pode ser apresentado na consola do navegador (console.log(x);) F12 para abrir consola

- Operações podem ser aplicados operadores aritméticos (+) (-) (x) (/)
Significado da opesação varia consuante o tipo de variavel, Ex: + com numeros calcula a soma, no caso de sequencias de caracteres, concatena-os.

- Condições (if ... else) → execução condicional

```
if (comparação) {
  /* instruções no caso positivo */
} else {
  /* instruções no caso negativo
```

= = igual
< menor
> maior
>= maior ou igual
!= diferente

Nota: = = = Compara o valor e o tipo (28)

- Condições (swich ... case) → Quando há + que uma condição para testar
  - para cada comparação há uma instrução case
  - Cada instrução case deve ser separada de uma instrução break
  - a instrução default será executada caso nenhuma das instruções anteriores tenha sido valida, não precisa de break (29)

- Ciclos (30)
  - while
    ```
    while (condição) {
      /* instruções */
    }
    ```
    → realizado 0 ou + vezes, ciclo só se realiza se a condição se verificar
  - do-while
    ```
    do {
      /* inst */
    } while (condição);
    ```
    → executado pelo menos 1 vez pois a comparação realiza-se no fim do ciclo
  - for
    ```
    for (inicio; comparação; incremento) {
      /* inst */
    }
    ```
    → executado um número fixo de vezes, desde o início até à comparação com o incremento

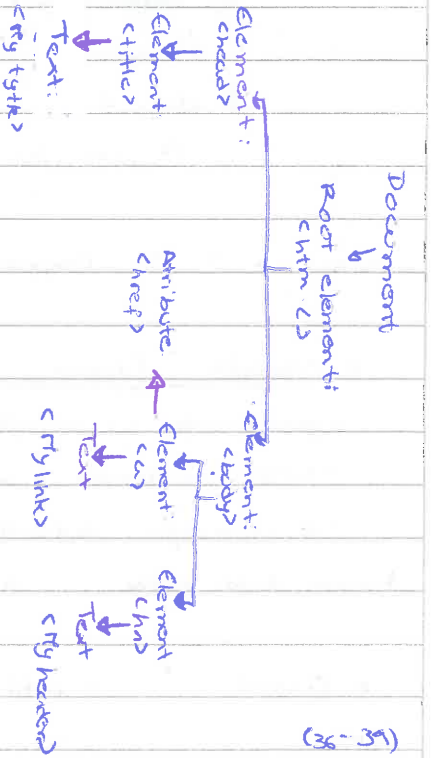
Document Object Model (DOM) → permite utilizar / aceder / manipular os objetos do documento html através do javascript.

Como aceder aos objetos do DOM: No html DOM tudo são nós:

documento = nó do tipo document
elementos html = nós do tipo element
atributos html = nós do tipo attribute
texto nos elementos html = nós do tipo text
comentários = nós do tipo coment

document = nó raiz do documento html e o "proprietário" de todos os outros nós (elements, text, atributes, comment.

<!doctype html> (35)
<html lang="en">
<head>
<title> My title </title>
</head>
<body>
<h1> My header </h1>
 My link
</body>
</html>



Iteração com o DOM: (36-39)

- Noção de objeto → podemos considerar que cada nó do documento html é ele próprio um objeto

Cada objeto é constituido por um conjunto de propriedades, métodos e eventos

ex: <a> é um objeto, também

Para aceder programaticamente a este objeto cujo id="URL-UA" faremos:

var x = document.getElementById("URL-UA");

- Elementos inexistentes → um elemento com ID inexistente, o valor devolvido será null.

NOTA: alert(msg): → caixa alerta com a msg especificada e um botão OK, garantem que a inf. chega ao utilizador

parseFloat(x.value) - converte o valor x do tipo string em um valor real do tipo float

Sintaxe da linguagem JavaScript:

- Eventos (40-48)

event.target devolve o objeto que despoletou um evento, util quando o código é comum a varios elementos em que apenas varia o nome

Glossário: (49-57)

Aula 6

Sintaxe da linguagem JavaScript:

• Temporizadores → permite controlar o tempo de execução das funções, util em animações, aumentam a interatividade de páginas. (4)

Os temporizadores podem ser relógios programáveis com uma resolução (diferença entre estados) de 1 milisegundo.

Torna possível: ativar funções de forma periódica, controlar o intervalo entre execuções. Em animações controla a duração de animação.

(6)

> setInterval ("função", intervalo): Define a função a ser invocada, de modo repetitivo a cada intervalo de tempo

- tempo → milisegundos
- função devolve um objeto, para ser possível cancelar o temporizador

> clearInterval (temporizador): Apaga o temporizador passado no argumento

> setTimeout ("função", atraso): Define a função a ser invocada depois do atraso especificado, em milisegundos. Função executada apenas 1 vez.

Exemplos (7 - 11)

Validação de formulários Para fazer a validação dos campos do formulário do lado do cliente, deve ser utilizado um <script></script> javascript de modo a garantir que os dados estão de acordo com o desejado antes de serem enviados para o servidor.

NOTA: Alguns problemas em passar de um formulário html para javascript:

1 o javascript refere-se aos elementos pelo ID ñ pelo name

2 como converter <input> em números ?

Sdução: 1 substituir o name pelo ID

2 utilizar name e id ao mesmo tempo - aconselhável para usar labels

Reunidas todas as condições necessárias (NOTA) para a validação do formulário antes de seu envio para o servidor.

1 Garantir que "firstName"/"personalData - firstName" tem no mín 3 letras

2 Em cada erro, mostra mensagem explicativa de modo a avisar o utilizador

→ Propriedades úteis de uma string: length - indica o tamanho da string

→ Métodos úteis para manipulação de strigs: trim - remove os espaços nos extremos da string

→ A propriedade classList de um elemento retorna uma lista com os nomes das classes desse elemento:

Métodos úteis para manipulação desta lista

- add("className") / contains("className") / remove("className")