

Explicar o que é o HTML.

HTML é uma linguagem de marcação (Hypertext Markup Language) criada por Tim Berners-Lee utilizada para a construção/desenvolvimento de elementos que constituem os websites. Ele publicou a primeira versão do HTML em 1991, consistindo em 18 tags. Desde então, cada versão do HTML vem com novas tags e atributos (modificadores de tags).

Qual a estrutura base dum documento HTML?

Um documento HTML válido precisa seguir obrigatoriamente a estrutura composta pelas tags `<html>`, `<head>` e `<body>` e a instrução `<!DOCTYPE>`.

<html> define o início de um documento html e indica ao navegador que todo o conteúdo posterior a este marcador deve ser tratado como uma série de marcadores html.

<head> define o cabeçalho do documento e possui informações sobre o documento que vai ser representado no navegador.

<body> define o conteúdo principal do documento e é a parte que é exibida no navegador. No marcador do corpo podem se definir atributos comuns a toda a página, como cor de fundo, margens e outras formatações.

Explicar o funcionamento do protocolo HTTP.

O HTTP é o protocolo utilizado para troca de documentos de hipertexto, ou seja, é a base para a comunicação de dados da world wide web.

Essa troca realiza-se da seguinte maneira:

O Web Cliente envia um pedido para o Server através do browser e recebe uma resposta do Server, após este comunicar com a Database e verificar que consegue emitir uma resposta (se não o conseguir fazer, temos um erro).

Explicar o significado de todos os campos da seguinte url:

<http://www.ua.pt/deti/PageCourse.aspx?id=383&p=4&a=9>

`<protocolo>://<servidor>[:<porta>]/<caminho>/<recurso>?dados&dados`

Porta é opcional, http (80), https (443)

http é o protocolo, *www.ua.pt* é o servidor, *deti* é o caminho, *PageCourse.aspx* é o recurso *?id=383&p=4&a=9* são os dados, em que *"?"*(primeiro) e *"&"*(seguintes) são os separadores.

Explicar o que é um marcador (tag) HTML, com exemplos. /Caracteriza um elemento HTML.

<marcador> texto </marcador>

Como o próprio nome diz, uma tag é um marcador que, na linguagem html, é responsável pela formatação do documento, indicando o tipo de elemento com que estamos a trabalhar (p.e.: a tag `<p>` diz-nos que estamos a trabalhar com um parágrafo).

Existem dois tipos de tags: as que precisam de abertura e fechamento, `</>`, e as que não precisam (p.e.: `
`)

Quais são os atributos a ter em conta quando se quer saber o tamanho total dum elemento HTML?

Conteúdo, padding, border, margin



Onde se deve colocar os **scripts num documento HTML e porquê?**

-Head -> só se tiveres um script pequeno que sabes que vai ser carregado rapidamente.

-Fim do body -> lugar ideal, pois assim a página é carregada, e caso haja erros é mostrada a parte visual na mesma.

Porque motivo se deve incluir o elemento (tag) <script> no final do <body> de uma página HTML?

Quando alguma tag é encontrada, o browser vai executá-la e com a tag <script> não é diferente, ou seja, quando o browser encontrar a tag <script> ele não vai executar mais nada enquanto não terminar de executar o que está dentro dela.

Colocar um <script> no final do body, permite que o conteúdo antes dele já apareça para o usuário sem ter de esperar a execução do script. Isso passa a impressão de um site mais rápido. A desvantagem é que - se o seu script modifica significativamente o conteúdo e/ou sua apresentação e funcionalidade - o usuário verá uma página "estranha" e "mal formatada" antes que o script a "corrija".

Onde se podem definir estilos e qual a sua ordem de precedência?

- Ficheiro css (Global) - colocadas num ficheiro externo que pode depois ser associado a um ou mais documentos html.

- Head (Document) - colocadas dentro de um marcador localizado no do documento

- Inline - colocadas na linha do marcador html

Inline-Documente-Global

Indique qual é a diferença entre classes e Id's

ID's → únicos

Os IDs são únicos, cada elemento pode ter apenas um ID, cada página pode ter apenas um elemento com esse ID já as classes não são únicas. Pode-se usar a mesma classe em vários elementos. Pode-se usar várias classes no mesmo elemento. Os estilos dos ID's começam com cardinal ("#"). Os estilos das classes começam com ponto (".");

O que significa 'responsividade' no contexto de desenvolvimento web?

Mudar o aspeto do conteúdo com base no dispositivo.

Explicar o que é o Responsive Web Design, utilizando o Twitter Bootstrap como exemplo.

O Responsive Web Design consiste numa abordagem de web site que tem como objetivo o desenvolvimento de um site que se adapte a qualquer dispositivo, seja ele de pequenas, médias ou grandes dimensões, através da redimensionação de elementos, para, assim, oferecer uma experiência de visualização e leitura fácil.

O Twitter Bootstrap é um exemplo de uma framework que oferece várias componentes de HTML, CSS e JavaScript de utilidade pública para o desenvolvimento de web projetos.

Enumera 3 vantagens do Bootstrap.

- Escreve-se menos código javascript
- Não é preciso definir Styles
- Responsividade embutida

Quais são as 3 linguagens que um web developer deve saber usar e para que serve cada uma?

HTML – conteúdo da página da web

CSS – O CSS é responsável pelo estilo como background, cores, layouts, espaçamentos e animações

JS – ações do conteúdo, programar o comportamento das páginas da web

Caracteriza o Javascript quanto à tipagem. Dá uma vantagem e uma desvantagem dessa tipagem.

Javascript é fracamente tipada

Vantagem: mais flexível

Desvantagem: type error

O que significa a sigla DOM e o que é que nos permite fazer?

Document Object Model

Permite utilizar/aceder/manipular os objetos de documento HTML através do JavaScript.

Explicar a estrutura e funcionamento de um formulário HTML.

Um formulário começa com a tag <form> e fecha com a tag </form>

Na tag <form> podemos ter ainda os atributos name, que dá um nome para o formulário para futura referência, method, que pode ser posto para ocultar os dados do usuário ou get para não o fazer, e action que é onde terá o endereço da entidade processadora.

A seguir podemos ter várias divisões no formulário, utilizando a tag <fieldset> </fieldset>, devidamente

legendadas com a tag <legend> </legend>.

Dentro dessas divisões opcionais podemos ter vários <input>. Cada input tem dentro de si um name, que dá um nome ao elemento que mais tarde servirá como referência; o type, que define o tipo de secção, podendo ser um radio, text, checkbox, password entre outras; um maxlength, que define o número máximo de caracteres aceites; um value, que pode ou não estar omissso ou vazio; um placeholder, que será um texto que irá pré aparecer na caixa do input.

Podemos também ter tags <textarea></textarea>. Dentro dessa tag também podemos ter os atributos name, placeholder e maxlength, e ainda podemos ter os atributos rows e cols que definem quantas linhas e colunas vão ser dispendidas para aquela área de texto.

Um formulário deve ainda ter bottons. Estes são inseridos através da tag <input> definindo o atributo type como o tipo de button desejado, p.e. type="submit" para o botão de enviar e type="resetbtn" para o botão de limpar formulário. Os botões podem ainda ter name e value, que no caso, esta última terá o texto que será exibido.

Depois de preenchido o formulário, ao clicar no botão submit, este deverá ser enviado para o link presente no action.

Var a = 5

Var b = "4"

Var c = "hello world"

Console.log(a+c) 5hello world int+string o + deixa de ter valor de soma e passa a ter valor de concatenação (juntar as variáveis)