

6.3.1 DOM E innerHTML

00:00:00 - 00:00:30

Agora que a gente já viu o que é variável, o que é array, o que é objeto que é laço de repetição usando for, hoje, na nossa aula, vamos fazer uma breve revisão sobre o que é DOM, algo que vocês viram bem por alto no nosso curso 1. E também como fazer inserção de conteúdos de tags HTML dentro do nosso HTML usando JavaScript. E para começar eu trouxe essa árvore genealógica. Para quem não sabe, a árvore genealógica é aquela árvore que fala quem são nossos ascendentes.

00:00:30 - 00:01:00

Descendentes, fala da questão de ancestralidade também, quem são nossos parentes diretos e indiretos. E eu trouxe essa árvore porque como a gente fala muito sobre DOM, para quem não sabe o DOM é um modelo de documento do objeto, é tudo aquilo que engloba a estrutura de documentos e objetos que a gente tem no nosso navegador a partir do momento que a gente acessa essa página. Quando a gente fala sobre esse assunto podemos dizer que o DOM é uma árvore genealógica.

00:01:00 - 00:01:29

Só que em vez de ser uma árvore genealógica familiar, como a gente está acostumado a gente fala que é uma árvore genealógica um pouco meio que ao contrário assim, mas abordando a estrutura de como a nossa página HTML é renderizada e de como o documento também é exibido e informado para o nosso navegador. O que que eu estou querendo dizer com isso? Estamos vendo aqui, passando na tela, saindo do lúdico e indo para uma questão mais técnica, a nossa árvore genealógica do nosso DOM.

00:01:29 - 00:02:06

Nós temos o "document", que tá acima de tudo o nosso documento. E esse documento armazena todos os nossos arquivos de texto que é o nosso HTML, certo!? E dentro desse HTML temos "head", lembra do esqueleto? É a nossa cabeça. E tem o "body" O corpo desse documento. Então, no "head" armazenamos o "title" e o "title" armazena o "text". O "body" armazena o corpo desse texto e temos ali as tags: o parágrafo, "ul", "div", "link"... e por aí vai.

00:02:06 - 00:02:36

E assim como tem no "title" e no "head" eles também armazenam textos. O parágrafo armazena texto, "ul" ainda tem alguns filhos aqui que são os "li" e que também armazenam texto. Então, se a gente for pensar numa árvore genealógica o "document" ali é o topo, é o mais antigo, vamos dizer assim, se formos falar em questão de família. E depois temos o HTML, e do HTML tem o "head" e o "body" e cada um tem seus respectivos objetos filhos que são os seus atributos que a gente utiliza.

00:02:36 - 00:03:07

Por que eu falei que é um pouquinho ao contrário o DOM? Porque acima do "document" existe a janela, que a gente chama de "window", na visão de código. Que nada mais é que a janela do nosso navegador que engloba tudo isso, engloba inclusive o "document" Um teste que vocês podem muito fazer é digitar... sabe o console sabe o console do Chrome ou de qualquer outro navegador? É só digitar "window" naquela tag de console.

00:03:07 - 00:03:32

Digite "window", e aí vocês vão ver o que vai ser retornado. Eu vou fazer um exemplo bem prático para vocês, para vocês verem como é a estrutura do nosso DOM, do nosso "Document Object Mode". Se a gente for aqui e depois em inspecionar, por exemplo... a gente vem aqui... e vamos em console, se a gente der "document", o que ele vai retornar para gente?

00:03:32 - 00:04:05

Olha aqui, o "document" vai retornar uma informação que é, como eu falei, o nosso arquivo HTML. Lembra da nossa estrutura, olha aqui. É exatamente essa árvore que é retornada. Então, temos aqui o DOCTYPE html, que é o nosso documento de texto, e temos o "head" e o "body". Tudo certinho. Se a gente for digitar um "window", aqui no nosso navegador, olha o que ele retorna. Ele retorna tudo que engloba o que está dentro dessa janela do navegador.

00:04:05 - 00:04:40

Dentro desse "window" vai ter milhares de coisas que não vai entrar no assunto, mas para fim de curiosidade, para saber o que retorna o "document", o que retorno um "window". O "window" retorna muitas coisas aqui, tudo que a gente trabalha em relação ao navegador, à janela do navegador. Eventos, cookies, requisições, tudo... tudo que está aqui. E aí, temos o nosso "window" bem lá embaixo. Aqui ó... olha o nosso "window" aqui. A gente tem o "document", a gente tem tudo. Tudo que está englobado na janela "window".

00:04:40 - 00:05:12

Quando a gente digita "window" aqui a gente consegue retornar. Então, o "document" retorna exatamente todo o documento que tá inserido aqui dentro dessa janela, que é como eu falei, o que está presente aqui nessa arquitetura. Por que eu estou falando isso? Porque quando a gente fala, estuda e entende sobre manipulação de elementos, precisamos entender o que estamos pegando ali, que tipo de informação estamos querendo pegar para poder manipular alguma coisa, trazer alguma informação.

00:05:12 - 00:05:43

E lá no curso anterior, vocês aprenderam sobre o "get element by Id". Não sei se vocês lembram, mas o que a gente usa antes? O "document". A gente usa o "document" antes para indicar que dentro daquele documento a gente quer pegar o ID de alguma coisa. Dentro daquele documento, que você já sabe que o documento é o arquivo HTML, a gente quer pegar alguma informação e essa informação significa o ID.

00:05:43 - 00:06:09

Então, qualquer coisa em que estamos fazendo manipulação, uma forma de você pegar aquela informação para depois você manipular ou fazer outras coisas.

É muito importante entendermos isso. Pode até parecer que eu estou sendo repetitiva, mas é para, justamente, vocês entenderem a importância e também entenderem mesmo como é a estrutura. Porque não adianta a gente só fazer código e não entendeu por trás dos panos.

00:06:09 - 00:07:05

Por que nós usamos um "document"? Por que a gente usa um "window"? Por que a gente usa um clique para fazer um evento, como vocês viram com o "advent listener"? Por que a gente usa um "for"? É para você, justamente, entender o "por trás" dos panos da web.

Inclusive, até recomendo para quem quiser depois pesquisar, entender mais a fundo. É bem interessante a gente entender como é que funciona, de forma mais aprofundada, essa coisa da arquitetura web. Porque eu acho que só vai enriquecer mais o conhecimento.

00:06:42 - 00:06:48

E muita das coisas que vocês estão vendo neste curso, e podem ver em outros lugares, vai começar a fazer muito sentido para vocês. Mas como eu falei, lá no início da aula, eu comentei sobre falarmos de manipulação de elementos E a gente falou do "get element by id", que a gente consegue pegar o ID de uma informação, mas como é que a gente consegue colocar uma informação lá no HTML?

00:07:05 - 00:07:35

E como é que a gente faz isso? A gente usa o "innerHTML", que significa a gente inserir um trecho de HTML dentro do nosso documento.

Olha o nosso "document" aqui de novo. `document.getElementById`, que é para acessar o ID O que eu estou fazendo aqui nesse código?

Eu estou informando ao documento que eu quero pegar aquele ID, como você já sabem, de uma determinada página dentro daquele arquivo HTML e dentro desse ID eu quero inserir o HTML.

00:07:35 - 00:08:09

Então, toda vez que vocês pensarem em "innerHTML" lembre-se sempre que é inserir algum trecho de código HTML. E aí, a gente usa o igual para fazer a atribuição e a gente coloca entre aspas as etiquetas, nossas tags.

Pode ser qualquer tag HTML, desde que seja HTML, até porque a ideia é justamente fazer inserção de código HTML. Lembre-se que quando estiver trabalhando com manipulação de elemento você tem que começar sempre com "document".

00:08:09 - 00:08:45

Acessar o ID dessa informação e depois utilizar o "innerHTML" para fazer a inserção dessas tags, desse trecho de código no qual você quer colocar no seu arquivo HTML. Eu sei que é muita informação, eu sei que esse módulo de JavaScript, como eu estou falando praticamente em toda a aula, é muita coisa, eu vou mostrar isso no exemplo prático para vocês porque eu acho que é de muito importância vocês entenderem o que estamos fazendo aqui. E vamos fazer um exemplo bem prático, novamente com o nosso projeto final para vocês entenderem mais a fundo do que funciona o "innerHTML".