O que é DOM?

R: É uma interface padronizada que permite que navegadores e scripts possam manipular o conteúdo de uma página web sem a necessidade de realizar atualizações e upgrades. É ele, ainda, que põe ordem na navegação web para que o conteúdo fique acessível a todos.

Quais a diferença do html5 pro html4?

R: Ao contrário das versões antigas do **HTML**, que permitiam criar principalmente sites estáticos que precisavam ser apimentados com CSS e JavaScript, o HTML5 é muito mais dinâmico e inclui elementos multimídia. Ele suporta nativamente vídeo e áudio, e você pode até mesmo fazer jogos ou animações com ele.

Quantas tags de title existem?

R:

Qual a diferença de div para um spa?

R: A tag <**div**> é usada para demarcar um bloco maior de código, contendo várias tags HTML ou mesmo uma única tag, mas de tamanho grande. Já a tag <**span**> é para trechos mais pontuais, menores, dentro de um parágrafo ou uma linha.

Cite algumas técnicas de SEO

R: Capture palavras-chave de relevância.
Crie conteúdos relevantes e completos. ...

- Realize a otimização on-page. ...
- Realize a otimização off-page. ...
- Saiba utilizar textos âncoras.

Cite técnicas pra deixar um site mais performático.

R: otimize as imagens

Escolha uma hospedagem de confiança

Fazer um teste de performance

Regular os rastreadores web

Abandone recursos desnecessários

Revise o código (manter o código limpo)

Cite algumas técnicas de acessibilidade.

R: Cartilha de **acessibilidade** web. ...

- HTML Semântico. ...
- Leitor de Tela. ...
- Leitor de Libras. ...
- Alto Contraste. ...
- Ferramenta de Lupa e Zoom.

Como você estrutura sua aplicação?

R: Eu agrupo o conteúdo relacionado por tópicos

- Destacar as páginas mais importantes
- Manter o conteúdo simples e organizado em uma hierarquia lógica

___

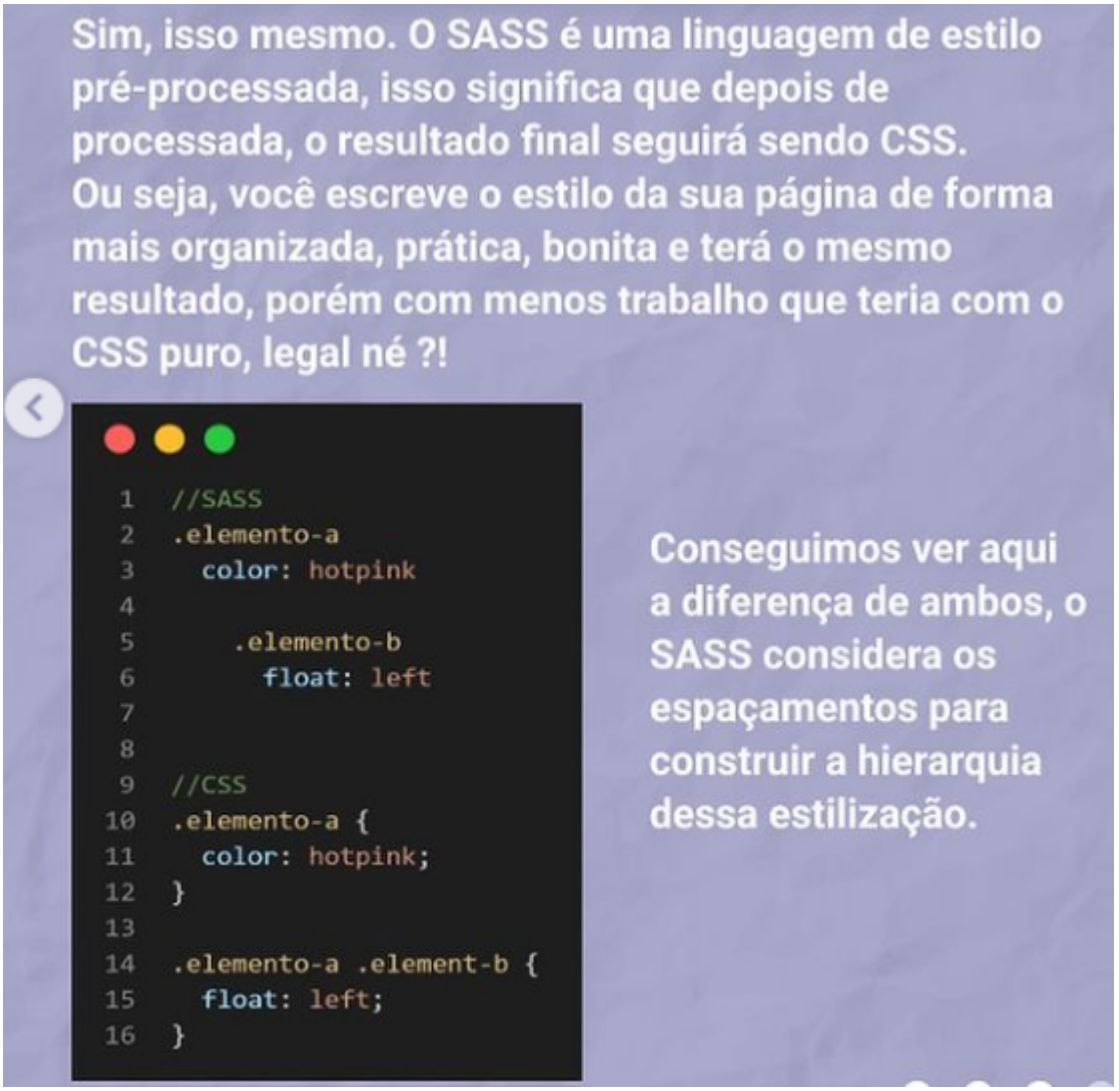
O que é SASS?

R: Ele é um pré-processador do CSS de folhas de estilo para auxiliar na produtividade de códigos, principalmente no que diz respeito a repetição de uma mesma ação, diversas vezes.

SASS é uma linguagem de extensão do CSS, a sigla significa “Syntactically Awesome Style Sheets”, que quando traduzido quer dizer “folhas de estilo com uma sintaxe incrível”.

Qual a vantagem de utilizá-la?

- Torna o processo mais simples e eficiênte;
- É possível adicionar recursos especiais como variáveis, mixins (trecho de código reaproveitável), funções e operações;
- É possivel quebrar o arquivos em várias pequenas partes e assim reutilizar a folha de estilo de forma mais prática.
- É possível adicionar até mesmo operadores matemáticos nesse tipo de arquivo.



Qual a diferença de position: relative e absolute?

O Position Absolute: é um tanto diferente do Relative. Enquanto o elemento com Position Relative utiliza seu próprio canto para referenciar sua posição, o elemento com Position Absolute se utiliza do ponto superior esquerdo de outros elementos. Estes elementos são os parentes dele do elemento com position absolute. Mais especificamente o pai.

O position fixed: irá fixar a posição do elemento na coordenada que você definir. A medida que a página é rolada, o elemento continua fixo na posição que você definiu e o conteúdo da página rola normalmente. Geralmente é usado para fixar elementos como cabeçalhos ou sidebars.

O position relative: posiciona o elemento em relação a si mesmo. Ou seja, o ponto zero será o canto superior esquerdo, e ele começará a contar a partir dali. Todos os positions precisam de um ponto para iniciar o cálculo da coordenada para assim posicionar o elemento na tela. Ao contrário do que muitos acham, esse ponto não é

o ponto central do elemento, o ponto base é o canto superior esquerdo do elemento. A partir deste canto, o browser irá calcular a coordenada que você definiu e irá posicionar o elemento no viewport.

O Position Static: é o valor padrão para todos os elementos de uma página - todos são inicializados como 'static'. Não há complicação aqui, simplesmente significa que o elemento vai seguir o fluxo da página normalmente. A única boa razão para atribuir explicitamente o valor 'position: static' de um elemento é forçar a remoção de um posicionamento previamente definido.

Qual a diferença entre classe e id?

O **atributo Id** especifica uma identificação única para o elemento HTML. Por questões de boas práticas, não deve ser reutilizado e nem conter espaços em seu nome, pois o navegador irá identificar o espaço como parte dele, já que os elementos não podem ter mais de um Id.

O **atributo global class** especifica uma ou mais classes para o elemento HTML. Esse atributo pode ser reutilizado, ajudando a pessoa desenvolvedora a não repetir códigos, além de permitir o uso de diferentes classes simultaneamente.

Pq o CSS Modules é tão vantajoso?

Os **css**-**modules** são arquivos **css** em que os classNames e animações são definidos localmente, isso significa que os estilos ali criados, só serão declarados dentro daquele escopo, e não globalmente, evitando conflitos entre estilos.

Quais as boas práticas que você utiliza na hora de escrever CSS?

R: Eu faço uma divisão lógica do meu CSS

Quando podemos usar o before ou after?

R: **::before** cria o pseudo-elemento antes do elemento selecionado
::after cria o pseudo-elemento após o elemento selecionado

Qual a diferença de DOM pra o Virtual DOM?

DOM é a representação dos compomentes na página. Você manipula o **DOM** a fim de manipular estes componentes (criar, recriar, alterar seu estado). **Virtual DOM** é um framework para manipulação do **DOM**.

O que é hook?

Resumidamente, **Hooks** é uma nova proposta que irá nos permitir utilizar estado, ciclo de vida, entre outras funcionalidades, sem a necessidade de escrevermos componentes com classe. A proposta já foi aceita e está disponível na versão 16.8 do **React**.

Como funciona o useEffect?

O **useEffect**() recebe como primeiro parâmetro uma função que será executada assim que o componente renderizar. Então é um ótimo lugar para fazer requisições. Dessa maneira como escrevemos, a função passada ao **useEffect**() será executada sempre que o componente for atualizado.

Quando devemos usar o useCallback?

O useCallback é utilizado para manter a referência da função passada, o useMemo deve ser utilizado para **cálculos caros**, e não para guardar a referência de um objeto, por exemplo.

Qual a diferença de um componente funcional e componente utilizando classe?

Hoje a **diferença** basicamente é em questões de estilos, um segue o principio de orientação à objetos enquanto outro segue para uma abordagem **funcional**

Como funciona uma promise?
Uma **Promise** representa um proxy para um valor que não é necessariamente conhecido quando a promessa é criada. Isso permite a associação de métodos de tratamento para eventos da ação assíncrona num caso eventual de sucesso ou de falha.

Qual a diferença de promise pra async await?

Ao usarmos o then, as **Promises** são executadas em paralelo, enquanto o **async**/**await** trata as **Promises** de forma sequencial, como se estivesse realmente executando um código síncrono – que imprime os resultados um após o outro, como vimos no artigo sobre **Promises** – mas, de maneira assíncrona.

O que é uma closure?

Uma closure é a combinação de uma função agrupada (incluída) com referências ao seu estado circundante (o ambiente léxico). Em outras palavras, um closure lhe dá acesso ao escopo de uma função externa a partir de uma função interna. Em JavaScript, os closures são criados toda vez que uma função é criada, no momento da criação da função.

Qual a diferença entre um for e um map?

- map(): O método **map()** invoca a função `callback` passada por argumento para cada elemento do Array e devolve um novo Array como resultado.
- foreach(): O método `forEach()` executa uma dada função em cada elemento de um array.

Como o método reduce funciona?

O método **reduce**() executa a função de callback uma vez para cada elemento presente no array, excluindo furos (valores indefinidos), recebendo quatro argumentos: acumulador - valor inicial (ou o valor do callback anterior), valorAtual - o valor do elemento atual.

Qual a feature que você mais gosta do es7?

Como o webpack funciona?

De uma maneira geral, o **webpack** pode ser visto como uma ferramenta responsável pela construção, onde todos os seus ativos, o que inclui JavaScript, imagens, fontes e CSS, são direcionados para um gráfico de dependência.

Me explica a pirâmide de testes. Testes unitários, integração e end to end e quais ferramentas podemos utilizar para cada uma dessas fases.

Testes de integração visam testar se todos os módulos **do** sistema funcionam bem juntos, os **testes integrados** (ponto a ponto, **End-to-End** ou **de** sistema) são os **testes** que visam analisar o sistema como um todo. Enqunato que o **teste unitário**, é responsável por testar a menor unidade em um sistema

Pq code-splitting é tão importante?

Ele é importante para dividir o código de sua aplicação pode te ajudar a carregar somente o necessário ao usuário, o que pode melhorar dramaticamente o desempenho de sua aplicação.

O que é uma função pura?

É uma **função** que dada a mesma entrada, sempre retornará a mesma saída e não tem efeitos colaterais. Sendo assim, é mais fácil de testar e fazer a manutenção. Como no exemplo acima, a **função** sempre retorna o mesmo valor baseado na entrada sem manipular nenhuma variável fora.

O que é Babel?

É a principal ferramenta para pré-processar JavaScript. Ele é um transpilador JavaScript em código aberto e multiplataforma.

Como o promise race funciona?

Race faz exatamente o que o nome diz, ele recebe um array de **Promises**, inicia todas elas, a que retornar Primeiro vai ser o retorno do método por completo. Ele é um caso especial do **Promise**. all onde, ao invés de esperar todas as **Promises** serem resolvidas, simplesmente retorna o primeiro resultado que obtiver.

Por que você gosta de typescript? Quais são as vantagens?

O potencial de detecção de erros durante o desenvolvimento de projetos e a possibilidade de incluir a inteligência (IntelliSense) da IDE (ambiente de desenvolvimento integrado) que está sendo usada. Isso reflete num ambiente muito mais ágil e seguro enquanto o código está sendo digitado pelo usuário.
Além de ajudar o ambiente de desenvolvimento, o TypeScript possibilita o uso de diferentes funcionalidades do superset que não estão inclusas na forma nativa e, também, a migração de linguagem gradativamente, refletindo no aumento da produtividade das equipes.

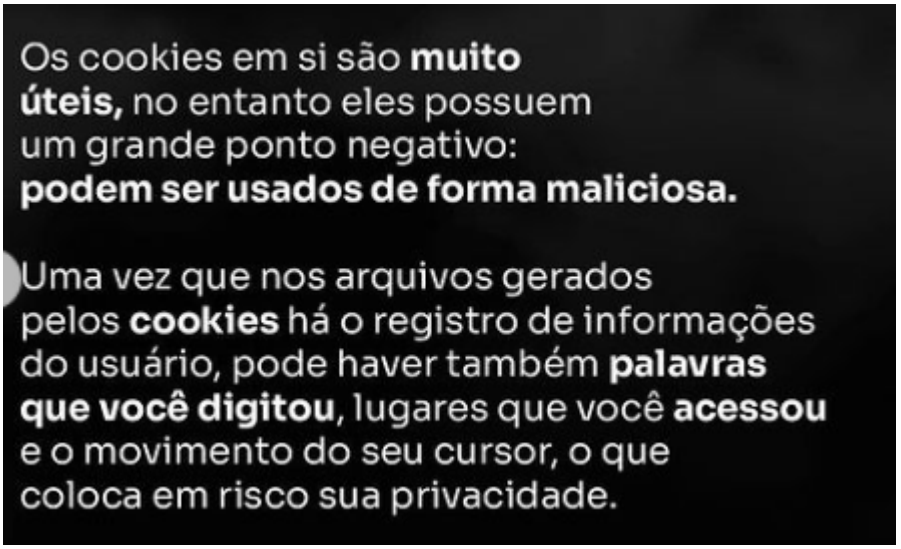
Como você organiza as suas tarefas do dia a dia?

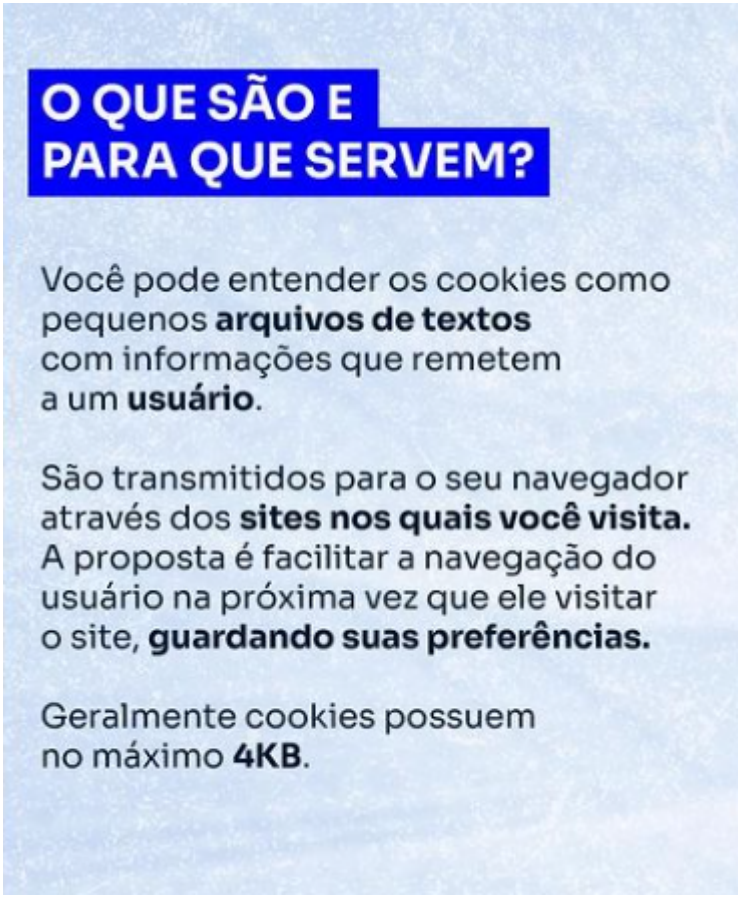
O que você anda estudando ultimamente?

Qual foi o maior desafio técnico que você já teve em sua carreira?

O que é Crud?

O que são cookies?





—----------


Lista de propriedades
CSS QUE TODO FRONT-END
Deve conhecer

text

color: definir a cor

font: escolhendo a fonte de um texto

font-family: especifica a fonte de um elemento

font-size: definir o tamanho do texto

font-weight: define a espessura do caractere

letter-spacing: definir espaço entre caracteres

line-height: define espaço entre linhas

text-align: alinhamento de um texto

text-decoration: linhas decorativas no texto

text-indent: recuo da primeira linha.

text-transform: capitalização do texto

vertical-align: alinhamento vertical de um elemento

Background

background: definir todo o estilo de fundo

background-attachment: define se uma imagem de fundo deve rolar com o resto da página ou ficar fixa

background-color: definir cor de fundo

background-image: definir imagem de fundo

background-position: definir a posição inicial de uma imagem de fundo

background-repeat: repita a imagem de fundo na horizontal e na vertical

Position

position: definir como um elemento é posicionado

top: posiciona o elemento no topo da tela.

bottom: posiciona o elemento no rodapé da tela

left: posiciona o elemento a esquerda da tela

right: posiciona o elemento a direita da tela

z-index: ordem de pilha de um elemento

Display

display: exibir o comportamento de um elemento

float: elemento flutuante

clear: controla o fluxo próximo aos elementos flutuantes

overflow: transborda uma caixa de elementos

visibility: define a visibilidade do elemento

Box

border: definir todo o estilo de borda

border-color: definir uma cor para a borda

border-style: definir um estilo para a borda

border-width: definir a largura da borda

height: definir a altura do elemento

margin: definir as margens para o elemento

padding: espaço entre o conteúdo e a fronteira

width: definir a altura do elemento

box-sizing: especifica o comportamento de largura e altura

List

list-style: define o marcador de um elemento de item de lista

list-style-image: definir uma imagem como o marcador de item de lista

list-style-position: posição dos marcadores de item de lista

list-style-type: tipos de marcador de item de lista


VOCÊ
SABE A DIFERENÇA
ENTRE C, C++ E C#?
arrasta pro lado

Você sabia?

O nome C++ é uma **brincadeira com o operador** de incremento da linguagem C.

i++

C# é algo como **(C++)++**, com os sinais de adição **empilhados.**

♯♯

Diferenças fundamentais:

- **A linguagem C é procedural, as outras são orientadas a objetos.**
- **C# faz o gerenciamento de memória de forma automática, C e C++ não.**