

Best Tech Tips

Best Tech Tips (/pt) / Dicas De Programação (/pt/parts/36-programming-tips)

/ 50 perguntas e respostas mais frequentes da entrevista sobre JavaScript (/pt/chapters/6184-50-frequently-asked-javascript-interview-questions-and-answe)

50 perguntas e respostas mais frequentes da entrevista sobre JavaScript

(/pt/chapters/6184-50-frequently-asked-javascript-interview-questions-and-answe)

Categoria Dicas De Programação (/pt/parts/36-programming-tips) | 📅 August 03, 2021 00:54

JavaScript provou ser uma linguagem de script versátil e escalonável ao longo do tempo. É uma das linguagens de script mais populares na indústria de desenvolvimento web. Ele oferece mais confiabilidade; é fácil de executar e executar. Isso abre oportunidades especiais para desenvolvedores. Esta é a razão pela qual milhões de desenvolvedores (*quase 94 por cento de todos os sites são feitos de JavaScript*) tendem a usar essa linguagem.

Um desenvolvedor iniciante com conhecimento básico de JavaScript pode ganhar US \$ 70-80.000 por ano. JavaScript pode ser realmente uma bênção para sua carreira, e habilidades de trabalho de longa data nessa linguagem podem torná-lo um dos funcionários mais bem pagos do ano. Portanto, não é à toa que você deve procurar empregos que ofereçam uma posição como desenvolvedor de JavaScript. Você pode ser um novato ou profissional. Para se envolver, é importante estar pronto para as perguntas da entrevista sobre JavaScript.

Perguntas e respostas mais frequentes da entrevista JS

As perguntas da entrevista sobre JavaScript vêm em três seções. Uma parte é determinada para os desenvolvedores iniciantes, depois vem o nível intermediário e a última é para os profissionais. Aqui, começaremos primeiro com as perguntas da entrevista de codificação de JavaScript de nível básico.

Perguntas básicas da entrevista JS

Esses tipos de perguntas de entrevista básicas sobre JavaScript podem ser respondidas por quase todos os níveis de candidatos. Para as pessoas que começaram com JavaScript, essas perguntas parecerão fáceis de responder até mesmo para elas.

1. O que é JavaScript?



JavaScript é uma das linguagens de script mais populares e fáceis de aprender. É uma linguagem de programação muito leve, compatível com os elementos do lado do cliente e do lado do servidor. É uma linguagem de programação orientada a objetos.

É uma linguagem de programação interpretada (um programa que pode implementar instruções diretamente) e pode ser facilmente embutida em HTML. Juntos, eles podem construir elementos dinamicamente interativos para páginas da web estáticas. Ele oferece melhor usabilidade e aprimora a experiência das pessoas com as páginas da web a um nível totalmente novo.

2. Você pode explicar a diferença entre Java e JavaScript?



Java é uma linguagem de programação muito comum. Ajuda a desenvolver um programa completo. Java é desenvolvido de forma a seguir apenas alguns números de dependências de execução. Java pode desenvolver aplicativos independentes que podem ser executados em navegadores e também em outros dispositivos eletrônicos.

Por outro lado, JavaScript é uma linguagem de programação composta. Ele pode criar elementos dinâmicos e interativos para páginas da web estáticas. JavaScript pode ser facilmente iniciado com HTML. Ao contrário do Java, os aplicativos JavaScript são executados apenas em navegadores.

Observe, Java e JavaScript dois são funcionalmente diferentes e não são dependentes um do outro.

3. O que é uma linguagem de script?

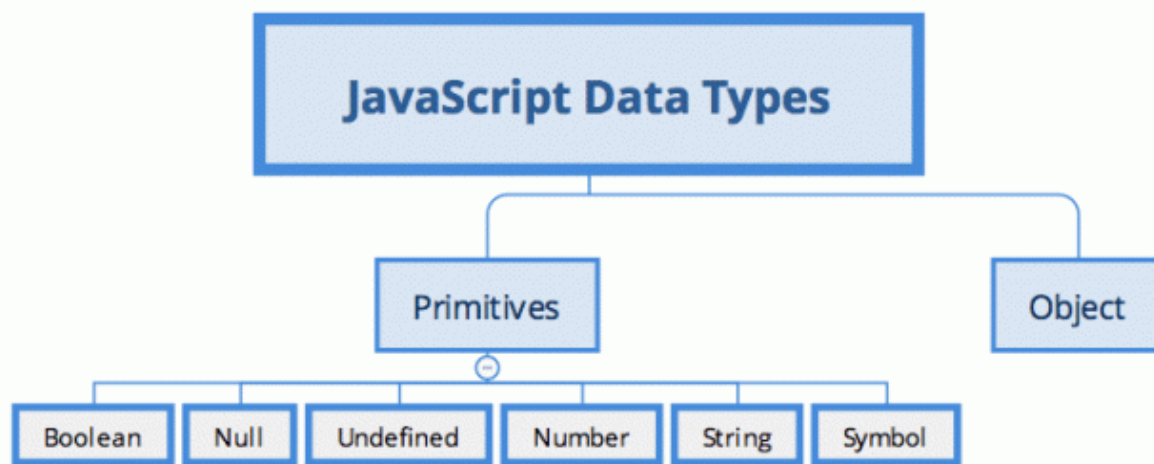
Uma linguagem de script também é uma linguagem de programação. A principal diferença entre uma linguagem de programação e uma linguagem de script é que as linguagens de programação precisam de um compilador para traduzir a linguagem de alto nível, enquanto as linguagens de script precisam de um intérprete como um tradutor.

Uma linguagem de script é dedicada a circunstâncias específicas de tempo de execução. Trabalha com a execução de diversos problemas e tarefas. JavaScript é uma das linguagens de script mais conhecidas. Algumas outras linguagens de script são ASP, PHP, Python, Perl, etc.

4. Você acha que o JavaScript tem uma interface mais rica?

sim. JavaScript é uma linguagem de interface mais rica. Ele fornece alguns recursos premium, como arrastar e soltar elementos, controles deslizantes também raramente disponíveis em outras linguagens de script. JavaScript tenta o seu melhor para fornecer aos usuários uma experiência de interface melhor e mais rica.

5. Quais são os tipos de dados suportados por JavaScript?



JavaScript oferece suporte a vários tipos de dados, e são eles:

- Número
- booleano
- Corda
- Objeto
- Símbolo
- Nulo
- Indefinido

6. Você pode nos contar alguns recursos do JavaScript?



JavaScript & its Features

Sim eu posso. JavaScript é usado em um grande número de sites mundiais. Uma das principais razões para trabalhar por trás disso é sua simplicidade e viabilidade. Aqui estão alguns recursos úteis do JavaScript:

- É uma linguagem de script leve e sensível a maiúsculas e minúsculas.
- JavaScript torna mais fácil organizar os elementos do lado do cliente. Ele pode permitir que os clientes usuários executem tarefas simples no navegador. E para fazer isso, não é necessário esperar a resposta do servidor.
- Oferece melhor controle aos usuários. Este idioma oferece alguns recursos adicionais para os navegadores e, portanto, reduz a carga do servidor.
- JavaScript é uma linguagem independente de plataforma. E, portanto, ele pode ser executado em sistemas baseados em Windows, Macintosh e Netscape.
- Ele pode criar facilmente conteúdo HTML.
- JavaScript ajuda a verificar a entrada dos usuários. E se houver algum erro na entrada fornecida, esta linguagem de script irá detectá-los antes de enviar a informação para o servidor web.
- JavaScript pode ser integrado com Java. No entanto, não é Java.
- É uma linguagem de programação interpretada.
- Ele está focado na criação de aplicativos baseados em navegador.

7. Por que devo escolher JavaScript em vez de outras linguagens?

Aqui está porque você deve escolher JavaScript:

Autentica as informações do usuário: JavaScript pode validar a entrada do cliente, mesmo se seus usuários-alvo cometerem algum erro ao inserir quaisquer dados. JavaScript pode detectá-lo. Além disso, o JS fornecerá uma resposta rápida ao usuário se ele se esquecer de inserir ou se enganar com qualquer informação.

Não totalmente dependente do servidor: JavaScript permite que você faça cálculos básicos no navegador. E para isso, não precisa pedir permissão ao servidor. Conseqüentemente, economizando seu tempo.

Gerencia data e hora: JavaScript oferece alguns recursos premium. Recursos que não estão disponíveis em nenhuma outra linguagem de programação. Por exemplo, JavaScript pode gerenciar datas e horas, o que dificilmente é acessível por qualquer outra linguagem. Ele também oferece alguns recursos exclusivos, como arrastar e soltar ou controles deslizantes.

Ciente dos navegadores do usuário: JavaScript pode aprender facilmente sobre o navegador do usuário e seu sistema operacional. Isso ajuda em situações em que o servidor precisa de informações sobre o navegador do usuário para processamento posterior.

8. Entre JS e ASP, o que funciona rápido?

JavaScript funciona mais rápido.

ASP, abreviado como página de servidor ativa, é uma linguagem de script do lado do servidor, enquanto o JavaScript é uma linguagem de script do lado do cliente. JavaScript não precisa de ajuda do lado do servidor para implementar tarefas simples. No entanto, o ASP sim. Conseqüentemente, tornando mais fácil para o JavaScript funcionar mais rápido do que o ASP.

9. Quem desenvolveu o JavaScript?

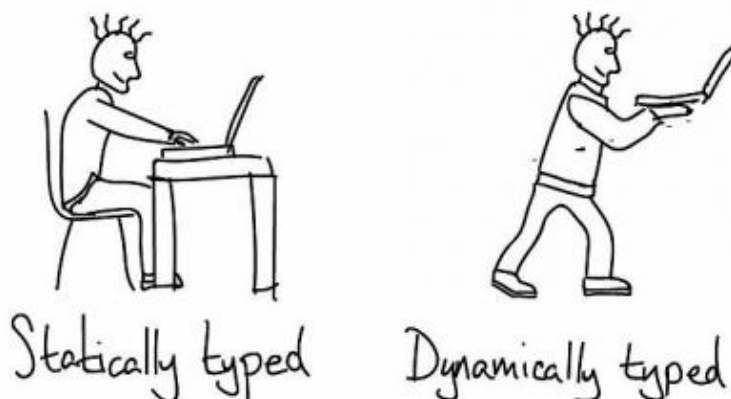


History Of JavaScript



JavaScript foi desenvolvido por Brendan Eich em 1995. Ele introduziu o JavaScript enquanto era membro do Netscape Communications. Brenden foi motivado por Java, Self e Scheme e decidiu criar JavaScript.

10. O que é um idioma digitado?



A linguagem digitada, em programação, é um sistema em que os valores são conectados apenas aos valores. Aqui, os valores não estão associados a variáveis. A linguagem digitada vem em dois tipos:

Linguagem digitada dinamicamente: Aqui, uma variável pode assumir vários tipos. Quando uma variável pode ter mais do que números, como char, é uma linguagem digitada dinamicamente.

Linguagem digitada estaticamente: Quando uma variável pode conter apenas um tipo, ela é chamada de linguagem com tipagem estática. Por exemplo, as variáveis em Java podem ter apenas um tipo de cada vez.

11. Você pode nos contar algumas limitações do JavaScript?



Como você está se candidatando a um emprego com base em um tópico especial, também precisa saber sobre seus limites. Isso o ajudará a tomar as medidas adequadas em caso de necessidade. Esses tipos de perguntas de entrevista de JavaScript não são totalmente incomuns em um quadro de entrevista. Aqui está a resposta,

Tudo tem seus prós e contras. Provavelmente, o JavaScript também tem algumas limitações. Eles estão:

- Embora o JavaScript ofereça um bom serviço do lado do cliente, o JS do lado do cliente não aprova os usuários a escrever ou ler arquivos. Esta política foi estabelecida devido a algumas questões de segurança.
- JS às vezes pode ser traduzido de maneiras diferentes por vários navegadores. Embora o JS do lado do servidor ofereça o mesmo resultado o tempo todo, a linguagem de script do lado do cliente pode variar um pouco de tempos em tempos.
- JavaScript não é aplicável para aplicativos de rede porque não oferece suporte para tais aplicativos.
- JavaScript não vem com nenhum componente essencial de multiprocessador.

12. Você pode nos contar sobre os erros no JavaScript?



Sim eu posso. Existem três tipos de erros em JavaScript:

- **Erros lógicos:** Erros lógicos acontecem em JavaScript quando há uma lógica pobre colocada no script. Este tipo de lógica é o mais difícil de rastrear, pois não é detectado durante a execução em tempo de execução.
- **Erros de tempo de execução:** Os erros de tempo de execução ocorrem em JavaScript quando há um deslocamento de comando na linguagem HTML. Acontece durante o período de execução de um programa.
- **Erros de sintaxe:** Um dos erros vitais é o erro de tempo de carregamento durante o carregamento de uma página da web. Isso acontece devido a qualquer sintaxe incorreta colocada no programa. Esse tipo de erro também é conhecido como erros de análise.

13. Por que o JavaScript é uma linguagem que diferencia maiúsculas de minúsculas?

JavaScript é uma linguagem que diferencia maiúsculas de minúsculas. Ao usar JavaScript, cada palavra-chave, array, nome das funções e outros identificadores devem seguir a capitalização adequada dos alfabetos.

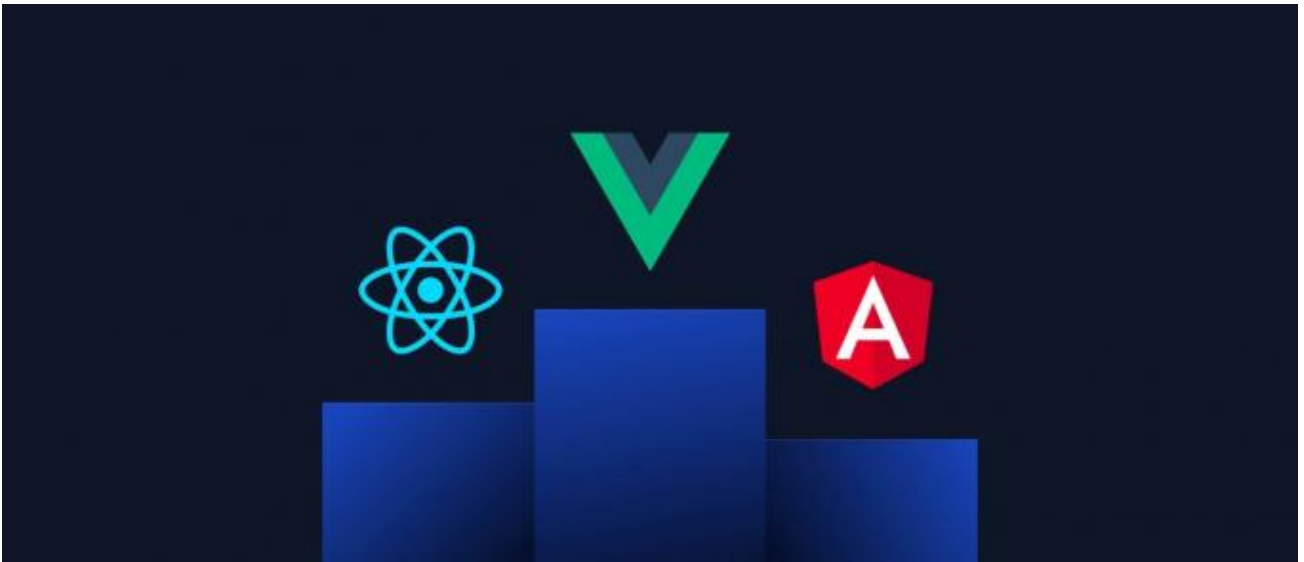
Por exemplo, se você escrever a palavra-chave 'quebra' em JavaScript como BREAK, será exibida uma mensagem de erro. Tem que ser escrito como uma "pausa".

14. Mencionar as convenções sobre como nomear uma variável em JavaScript?

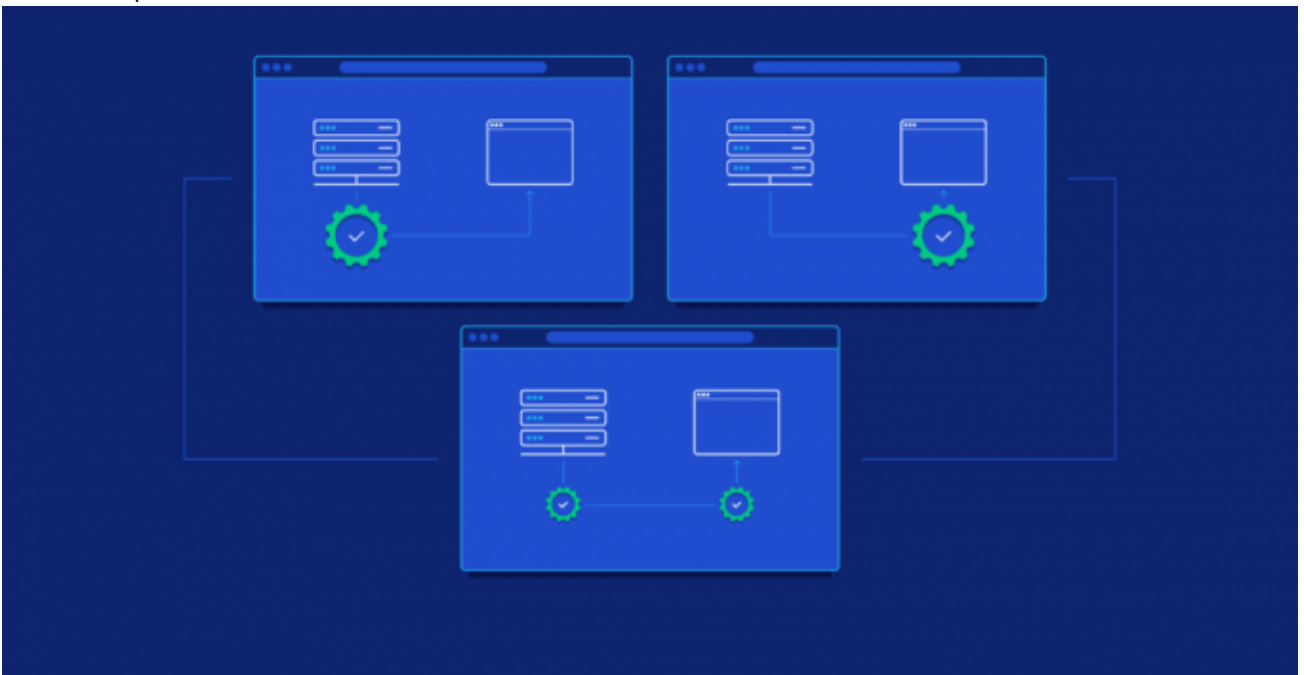
Sim eu posso. Existem certas regras e regulamentos ao nomear uma variável em JavaScript. Eles estão:

- Os nomes das variáveis diferenciam maiúsculas de minúsculas em JavaScript. Por exemplo, X e x são duas variáveis diferentes em JavaScript.
- Não comece um nome de variável com números de 0 a 9. Os nomes das variáveis devem começar com uma letra. No entanto, você também pode usar um sublinhado para nomear uma variável ou usar números após as letras ou o sublinhado no nome. Por exemplo, 345xyz é um nome inválido para uma variável JavaScript, enquanto _345xyz ou xyz345 é um nome de variável válido em JavaScript.
- Não use palavras-chave reservadas de JavaScript como um nome para uma variável. Por exemplo, você não pode usar while ou break para nomear uma variável.

15. O que são frameworks JavaScript?



Os frameworks JavaScript são escritos em JavaScript. Uma estrutura em cada aspecto define todo o design do sistema. Portanto, um Framework JavaScript e uma biblioteca JavaScript não são iguais. Existem muitos frameworks disponíveis em JavaScript. Então



às vezes, escolher o caminho certo em uma lista enorme pode ser muito desafiador. Aqui está uma lista das estruturas de JavaScript mais populares,

- Angular
- Vue. Js
- Reagir
- Meteoro

16. Você pode nos falar sobre os cookies JavaScript?

JavaScript

Cookies

Os cookies em JavaScript são um dado. Nos cookies, os dados são reservados em arquivos de texto pequenos e compactos. Quando uma página da web é enviada ao seu navegador, a conexão é encerrada. E com o término da conexão, todos os dados referentes ao usuário serão apagados pelo servidor. Os cookies facilitam o controle das informações do usuário (como nome, ID, e-mail). Para que, da próxima vez que um usuário fizer uma visita, ele não precise inserir essas informações novamente.

Obviamente, antes de armazenar as informações do usuário, é necessária a permissão do usuário.

17. O que é NULL em JavaScript?

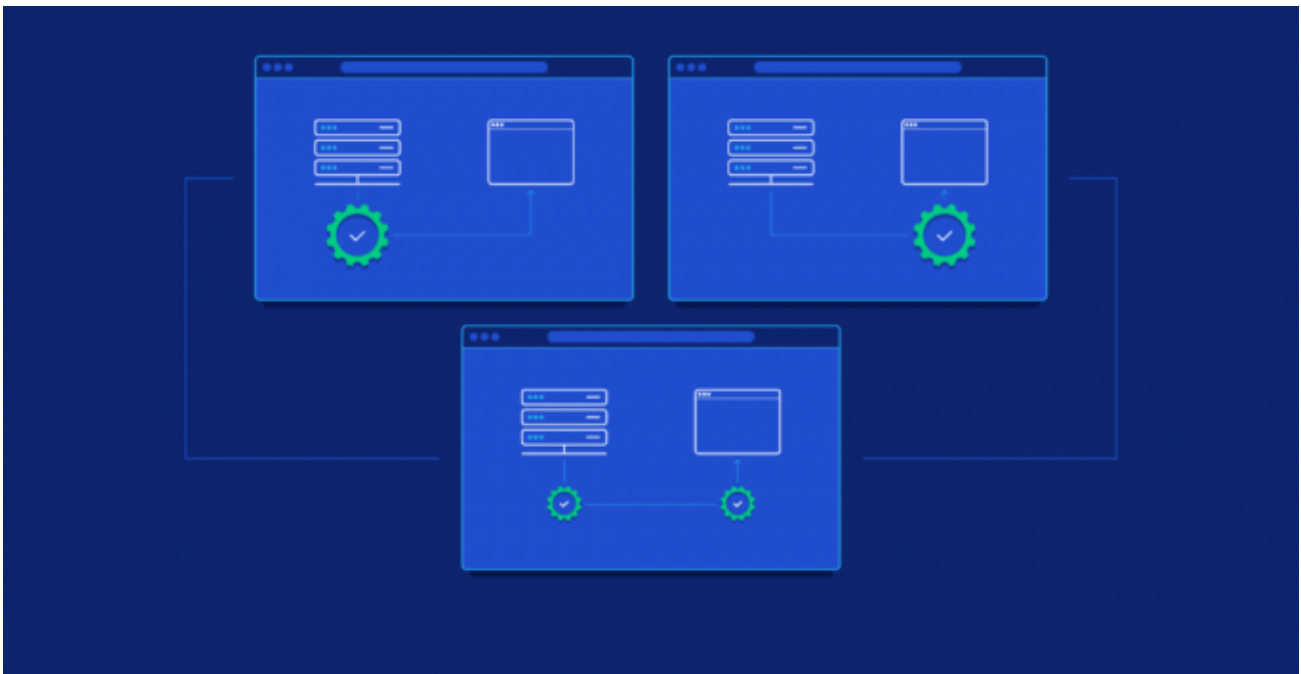
Nulo é uma palavra-chave do JavaScript. Ele não representa nenhum valor ou valor sem existência. Em qualquer condição em que um desenvolvedor precise usar uma variável sem um valor, palavras-chave nulas são usadas em JavaScript.

Palavras-chave nulas não representam nenhum caractere ou string, nenhum número ou qualquer valor condicional ou qualquer array. Os valores nulos, ao contrário de quaisquer outros objetos, não são indiretamente configurados para uma variável.

Com isso, termina nosso nível básico de Perguntas para Entrevistas em JavaScript. Agora, vamos para a próxima etapa.

Como concluímos as perguntas básicas da entrevista do JavaScript, é hora de ir para o próximo nível. Haverá uma seção dedicada em cada quadro de entrevista onde o candidato terá que enfrentar algumas questões de nível intermediário sobre o seguinte tópico. Aqui está nossa coleção de algumas possíveis perguntas de entrevista de JavaScript de nível intermediário.

18. Como você pode detectar as informações do sistema do cliente com JavaScript?



Usando JavaScript, pode-se detectar facilmente os detalhes do sistema (como sistema operacional, informações do navegador) de outras partes. Para fazer isso, basta usar `navigator.appVersion` ou `navigator.userAgent`.

Aqui, `navigator.appVersion` é somente leitura. Ao usar esta propriedade, ela retornará uma string revelando informações sobre a versão do navegador do cliente.

`navigator.userAgent` também é uma propriedade somente leitura do JavaScript. Usando esta propriedade, ele retornará informações contidas sobre o nome do navegador, versão e plataforma do navegador do cliente.

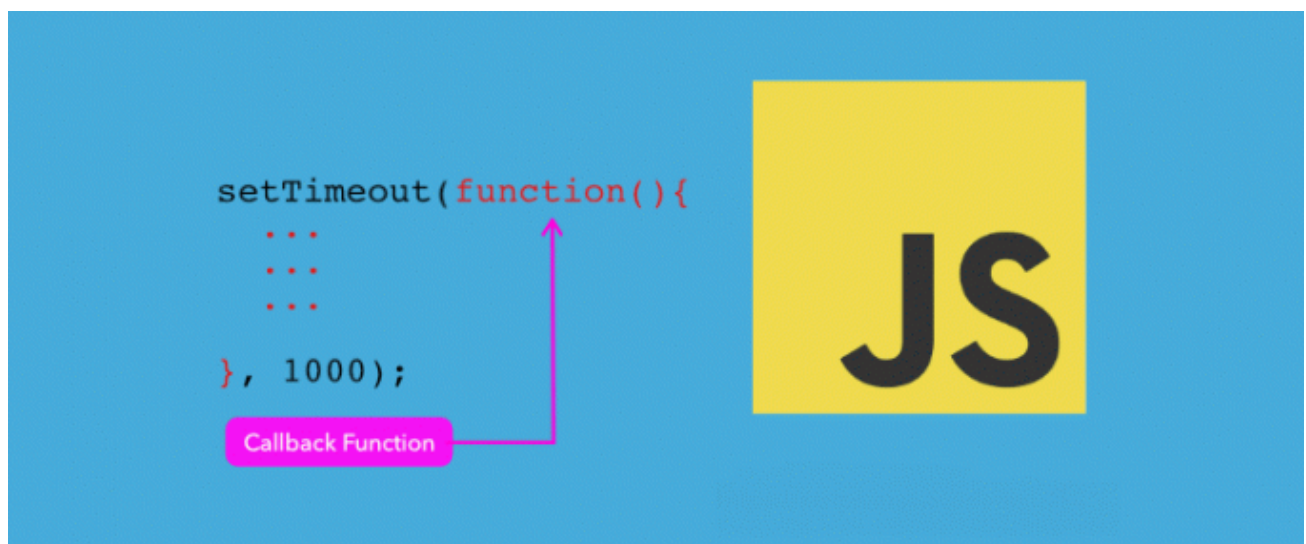
19. Você pode explicar os escopos das variáveis em JavaScript?

Sim eu posso. Os escopos de variáveis em JavaScript são descritos como a região onde as variáveis são declaradas. Variáveis em JavaScript têm dois escopos. Eles estão:

Variáveis globais: Quando uma variável é visível fora de uma função e basicamente em qualquer lugar no código escrito, ela é conhecida como variável global. Uma variável global pode ser obtida de qualquer parte de seu código JavaScript.

Variáveis locais: O escopo da variável é local quando uma variável é definida dentro de uma função e pode ser obtida a partir dessa função e é apenas uma função aninhada; é conhecido como variáveis locais.

20. O que é “retorno de chamada” em JavaScript?

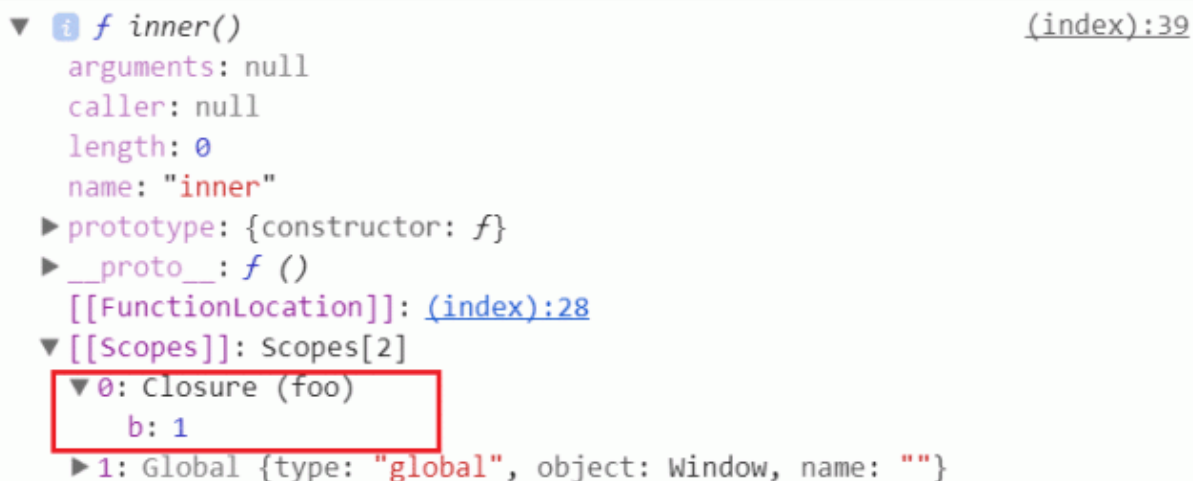


Um retorno de chamada é uma função JavaScript simples. É executado após a execução de uma função. As funções de retorno de chamada possuem a capacidade de serem passadas como um parâmetro ou um argumento para uma função

de outra função JavaScript. Depois de passar a função de retorno de chamada, ela será executada dentro da função para a qual foi passada.

Observe que as funções de retorno de chamada podem ser usadas de forma síncrona e assíncrona. Eles possibilitam que uma função receba outras funções como parâmetros ou argumentos.

21. O que é “fechamento” em JavaScript?



O fechamento em JavaScript é uma composição de uma função encerrada em seu ambiente circundante. Fechamentos são desenvolvidos toda vez que um programador introduz uma função.

Em JavaScript, você pode simplesmente usar um encerramento definindo uma variável fora do escopo contínuo. Esta variável deve ser acessada por outro escopo interno. De forma simples, os fechamentos permitem que você entre no escopo de uma função externa a partir de uma função interna.

22. Quais são os atributos e propriedades do JavaScript?

Os atributos em JavaScript são um somatório de objetos. Esses objetos pertencem à classe Attr. Os atributos fornecem mais informações sobre um elemento em JavaScript. Como id, valor, tipo e outros.

Por outro lado, a propriedade em JavaScript representa a característica de um objeto. As propriedades também narram atributos relacionados a diferentes estruturas de dados. Em JavaScript, existem dois tipos de propriedades,

- **Propriedades da instância:** Esses tipos de propriedades em JavaScript carregam os dados que são definidos para um determinado objeto.
- **Propriedades estáticas:** As propriedades estáticas em JavaScript transportam os dados que são divididos em todas as instâncias do objeto.

23. O que é o operador TypeOf em JavaScript?

TypeOf é uma palavra-chave de JavaScript, usada para obter o tipo de dados de um operando. Este operando, entretanto, pode ser uma estrutura literal ou de dados como uma variável. No caso de uma variável, Typeof funciona para retornar o tipo de uma variável quando ela está sendo chamada.

Além de derivar o tipo de operandos, o operador TypeOf é muito útil ao autenticar parâmetros de diferentes funções. Ele também monitora se as variáveis são declaradas corretamente. No entanto, o operador typeof retorna sua saída como uma string.

24. Como você pode ler um cookie com JavaScript?

How to Create, Read, Delete Cookies



Com JavaScript, ler um cookie é muito fácil. Você simplesmente precisa usar a string `document.cookie`. E o valor de `document.cookie` retornará o cookie. Esta string contém um registro do nome e valor (o valor da string do cookie) de um cookie que é isolado por ponto-e-vírgula. Com esta string `document.cookie`, você pode obter acesso ao cookie de qualquer lugar em sua programação.

No entanto, para vários cookies, `document.cookie` retornará todos os valores e nomes em uma string.

25. Um cookie pode ser excluído usando JavaScript?

Sim, podemos excluir um cookie usando JavaScript. Primeiro, você precisa declarar o caminho do cookie. Isso garantirá que você está excluindo o cookie correto. Acima de tudo, se você não definir o caminho do cookie, alguns navegadores podem não permitir que você remova um cookie.

Para excluir um cookie e esperar que ele não retorne mais dados do cliente, defina o valor do cookie como vazio. E simplesmente defina uma data de validade para uma data passada.

26. Como você pode definir uma variável em JavaScript?

Você pode declarar variáveis em JavaScript de três maneiras. Eles estão:

Var: Você pode usar uma instrução `Var` em JavaScript para declarar uma variável. Você também pode usar a instrução `var` para inicializar o valor dessa variável específica. Se houver uma instrução `var` em seu código, ela será executada antes de executar toda a programação.

Const: Qualquer função pode ser declarada como `Const` em JavaScript. E quando é declarada, a função pode ser chamada em qualquer objeto. No entanto, as funções `const` não permitem a modificação de objetos que foram chamados.

Deixar: `Let` é um sinal em JavaScript. Indica quando uma variável deve ser reatribuída. Também é usado para sinalizar o uso de uma variável no bloco apenas onde ela foi declarada.

27. Mencione uma diferença entre armazenamento local e armazenamento de sessão.

Local Storage!

VS

Session Storage!

Armazenamento local: O armazenamento local ajuda a diminuir o tráfego entre o cliente e o servidor. O armazenamento local não contém a possibilidade de ser destruído até que seja excluído manualmente por meio de qualquer configuração. Portanto, os dados salvos no armazenamento local não têm período de eliminação natural.

Armazenamento de sessão: O armazenamento de sessão, em função, é quase o mesmo que o armazenamento local. A única diferença entre eles é que os dados armazenados no armazenamento da sessão serão excluídos após a expiração de cada sessão de página.

28. Mencione a diferença entre nulo e indefinido.

JS JavaScript

Undefined Vs Null

Em JavaScript, indefinido se refere à condição quando uma variável foi anunciada, mas ainda não foi atribuída com valor.

Pelo contrário, nulo é um valor já atribuído. Não tem valor algum. Uma variável pode ser atribuída com um nulo, que não conterá nenhum valor.

Observe que undefined é um tipo, enquanto null é um objeto em JavaScript.

29. Mencione algumas diferenças indefinidas e também não declaradas.

Indefinido e não declarado são dois tipos em JavaScript. Uma variável não será declarada quando não tiver sido declarada. Portanto, ele não existe em seu programa. No entanto, se você ainda colocar valor para uma variável não declarada, seus compiladores mostrarão um erro de tempo de execução.

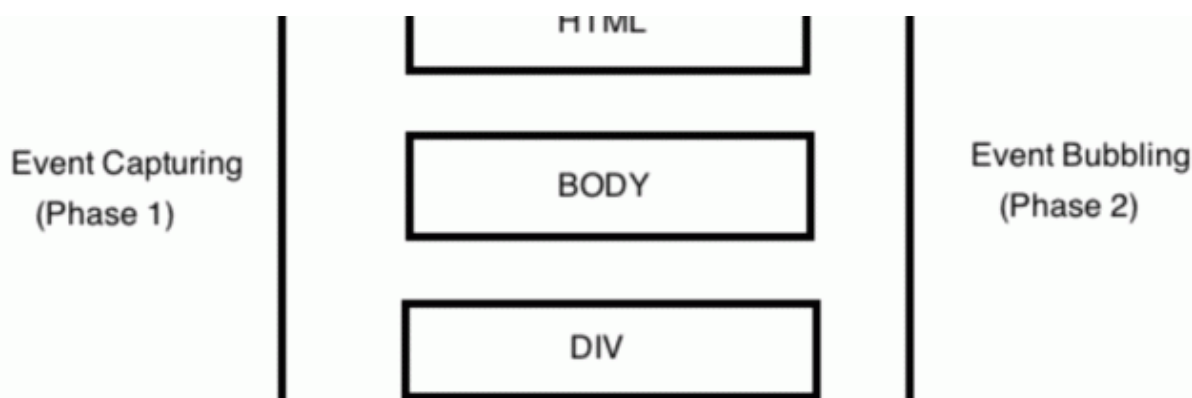
Por outro lado, variáveis indefinidas são quando uma variável é declarada, mas ainda não foi definida com nenhum valor. Seu programa retornará um valor indefinido quando o compilador executar uma variável indefinida.

30. Mencione a diferença entre janela e documento em JS.

O objeto janela em JavaScript representa a janela de um navegador e é compatível com todos os navegadores. A janela JavaScript contém todos os elementos globais (como variáveis, histórico, funções) do JavaScript. O documento também é outro objeto do Window.

Um documento é considerado uma propriedade da janela. Objeto de documento refere-se aos documentos HTML que são apresentados em uma janela.

31. Descreva o borbulhamento de eventos em JavaScript.



Borbulhamento de eventos em JavaScript é um termo muito comum para quem usa JavaScript regularmente. É um processo de extensão de evento para HTML DOM API. Este ato ocorre quando um elemento de um evento está aninhado dentro de outro segundo elemento e ambos os elementos listaram um identificador para o mesmo evento.

Com esse processo de borbulhamento de evento, o evento é primeiro gerenciado pelo elemento interno e, em seguida, é enviado ao elemento externo. O borbulhamento de eventos leva um evento ao seu destino específico. Se não houver um manipulador definido para esse elemento, o evento aparecerá em seu elemento pai.

32. Você conhece o NaN em JavaScript?

NaN in JavaScript



NaN em JavaScript significa não um número. É uma propriedade do JavaScript. Ele representa que o valor de uma variável não é um número legal. Conseqüentemente, ele mostra uma situação de falha para uma função que deve enviar de volta um número válido.

A propriedade NaN aparece quando uma string está sendo traduzida em um número que não é totalmente possível.

A seguir, seguiremos nossa seleção de algumas perguntas de entrevista sobre JavaScript de nível profissional.

Perguntas da entrevista de JavaScript da categoria avançada

Conseqüentemente, concluímos o nível básico e intermediário das perguntas da entrevista em JavaScript e ficamos com a última seção de nosso artigo - perguntas da entrevista de nível avançado em JavaScript. Esta seção é geralmente dedicada às pessoas que têm experiência neste campo há algum tempo.

33. Defina o modo estrito em JavaScript.

```
> (function() {  
    'use strict'  
    variable = 'hey'  
})();
```

```
✖ ▶ Uncaught ReferenceError: variable is not defined  
    at <anonymous>:3:12  
    at <anonymous>:4:3
```

```
> (() => {  
    'use strict'  
    myname = 'Flavio'  
})();
```

```
✖ ▶ Uncaught ReferenceError: myname is not defined  
    at <anonymous>:3:10  
    at <anonymous>:4:3
```

O modo estrito em JavaScript permite um melhor gerenciamento de erros em seu código JavaScript. Quando o modo estrito é ativado, atos específicos não podem ser executados. Por exemplo, você não pode usar variáveis não

declaradas enquanto este modo estiver ativado. Outros atos que serão evitados incluem: definir valores para propriedades somente leitura, adicionar propriedades a elementos inextensíveis.

Observe para habilitar o modo estrito em JavaScript e você precisa adicionar "usar estrito" antes de introduzir um arquivo ou programa.

34. Qual é a diferença entre "==" e "===" em JavaScript?

Em JavaScript, '==' indica igualdade entre dois valores, enquanto '===' é seguido por um teste de igualdade. E ele retornará um "falso" se algum dos valores for diferente. No entanto, também fornecerá uma declaração falsa se a natureza das variáveis também for diferente.

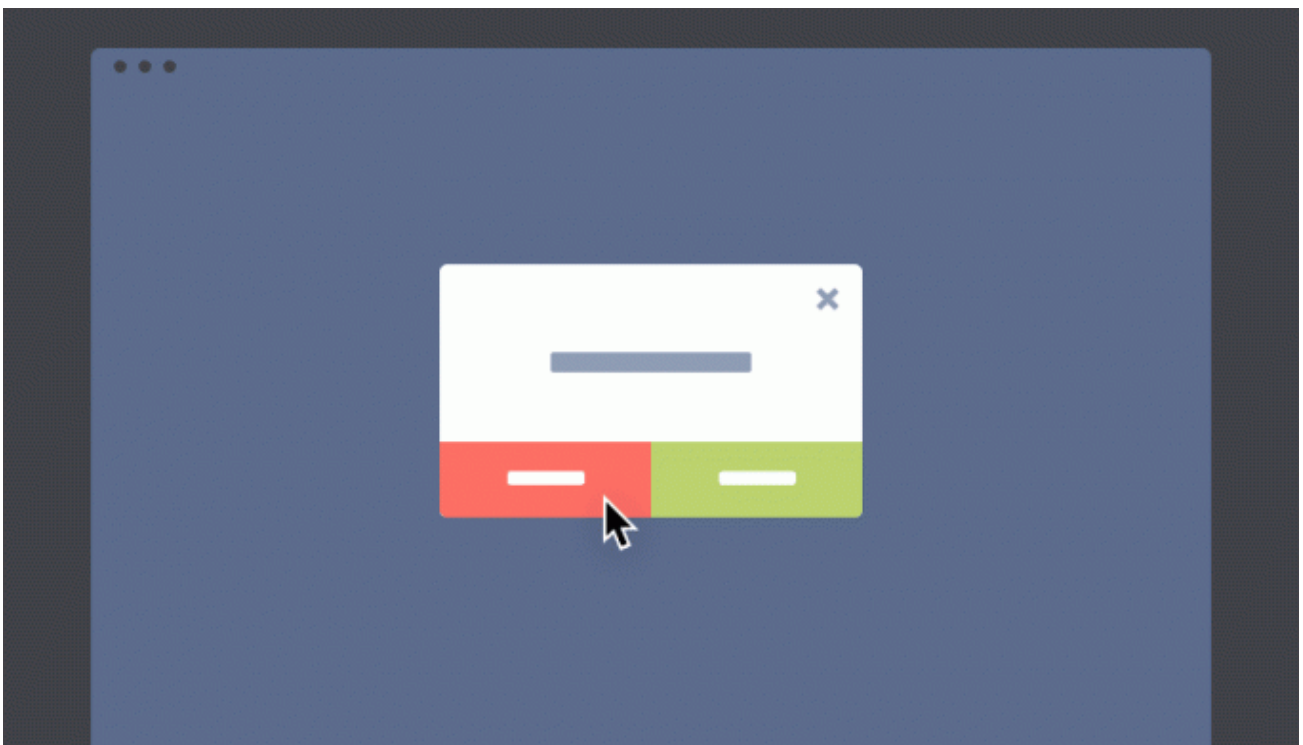
35. O que são innerHTML e innerText em JavaScript?

innerHTML é uma propriedade do JavaScript. É uma parte do DOM (Document Object Model). Esta propriedade do JavaScript ajuda a influenciar a maneira como um site deve ser exibido. innerHTML oferece a leitura e remoção de tudo o que está dentro do objeto DOM.

Por outro lado, a propriedade innerText do JavaScript é usada para atribuir ou retornar os elementos de texto de um nó específico e todos os seus sucessores. Enquanto você estiver usando a propriedade innerText, cada nó filho será removido.

Observe, innerHTML gerencia a tag HTML, enquanto innerText não.

36. Descreva as caixas pop-up em JavaScript.



As caixas pop-up em JavaScript são usadas principalmente para mostrar mensagens ou notificações aos visualizadores. Existem três tipos de caixas pop-up em JavaScript. Eles estão:

- **Caixa de Alerta:** Caixas de alerta em JavaScript são usadas para mostrar mensagens de aviso ao usuário.
- **Caixa de confirmação:** As caixas de confirmação são usadas para solicitar permissão ou autorização do usuário.
- **Caixa de aviso:** A caixa de prompt é usada para solicitar a entrada do usuário para ser usada posteriormente.

Observe que, em todos os casos, o usuário deve selecionar ok ou cancelar para prosseguir com a ação.

37. Descreva a caixa de prompt do JavaScript.

A caixa de prompt em JavaScript é usada para obter qualquer entrada dos usuários antes de prosseguir com uma página da web. Uma caixa de prompt mostrará uma caixa de diálogo (geralmente uma caixa de texto) onde o visitante deve inserir as informações necessárias antes de entrar na página. Com o pop-up de uma caixa de prompt, o usuário será solicitado a selecionar ok ou cancelar para se aproximar mais do site.

38. Qual será a saída de `8 + 1 + "5"` em JavaScript?

8 e 1 são inteiros. Eles terão uma adição adequada. No entanto, 5 é uma string. Portanto, ele permanecerá o mesmo após a execução. E a saída será de 95.

Observe que, aqui, a diferença é feita apenas por `""`. Apresenta 5 como uma string em vez de um número.

39. Mencione a diferença entre `ligar` e `aplicar`.

Javascript `call()` and `apply()`

Chamar e aplicar JavaScript; ambos podem chamar funções. A diferença elementar entre o método `call()` e o método `apply()` é que o método `call` aceita argumentos e o valor de uma função independentemente. Enquanto o método `apply` concede argumentos e valores de função como uma matriz.

Em palavras simples, é uma lista de argumentos para o método `call()`, enquanto é um array exclusivo de argumentos para o método `apply()`.

40. Como você cria um cookie com JavaScript?

Criar um cookie com JavaScript é muito simples e fácil. Tudo que você precisa fazer é definir um valor de string para a propriedade `document.cookie`. Aqui está como deve ser feito,

Sintaxe:

```
document.cookie = "nome de usuário = jane doe";
```

Você também pode definir uma data de expiração para que o cookie expire quando o navegador for desligado por padrão.

Sintaxe:

```
document.cookie = "nome de usuário = jane doe, expira = dia, data e hora em UTC";
```

41. Descreva as diferentes maneiras de como o código JS pode se envolver com HTML.

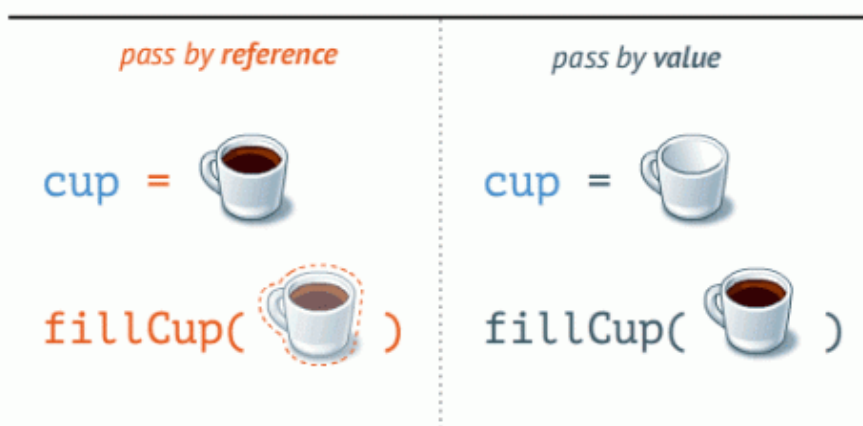
Existem três maneiras diferentes em que o código JavaScript pode ser associado a um arquivo HTML. Eles estão:

- **Na linha**
- **interno**
- **Externo**

Na linha: Uma função embutida em JavaScript é uma função anônima atribuída a uma variável. As funções embutidas podem ser facilmente reutilizadas. É um tipo especial de função e frequentemente considerado quase o mesmo que uma função anônima; ao contrário de outras linguagens de programação, o conceito convencional de uma função embutida não é compatível com JS.

Interno e externo: Essas duas maneiras são aplicáveis quando o JavaScript é usado para uma função. Nesse caso, quando você pode integrar o script JS na página atual, é chamado de caminho interno. Por outro lado, quando você coloca o script em um arquivo separado, ele é conhecido como caminho externo.

42. Como você passa dados primitivos JS e objetos em função?

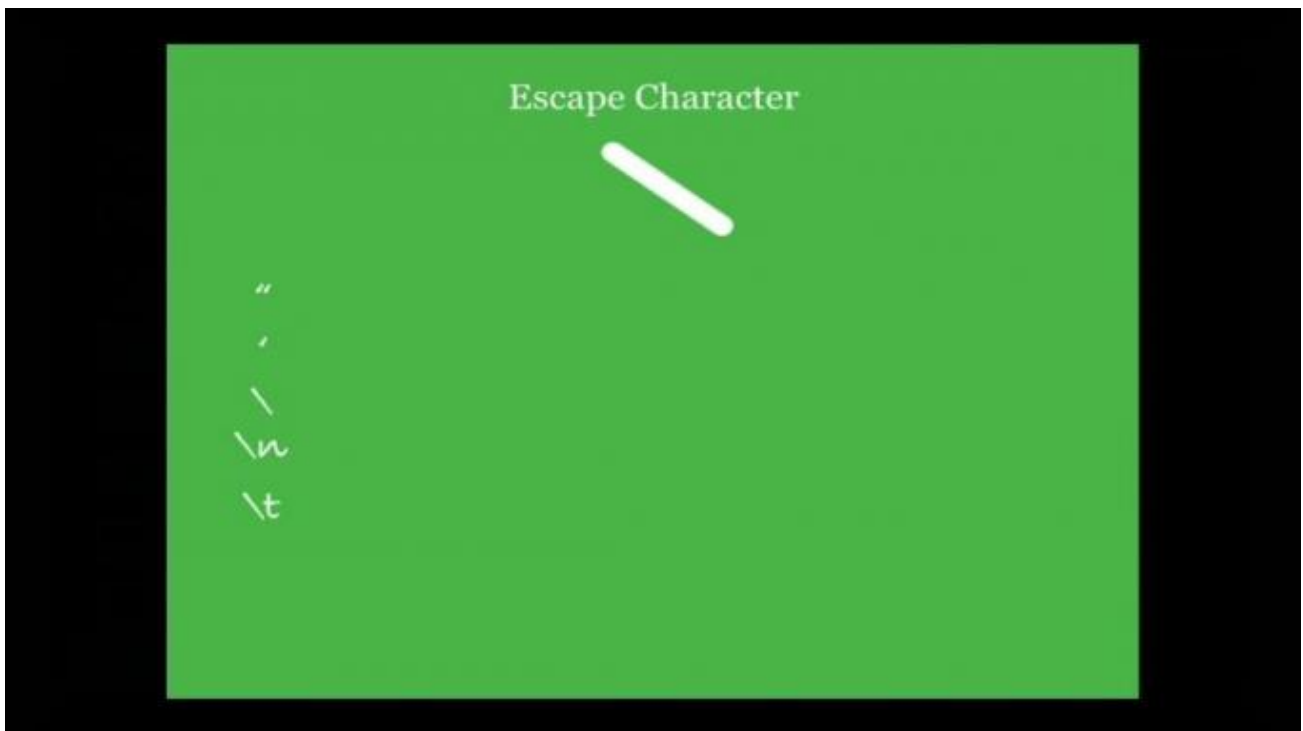


Em JavaScript, os tipos de dados primitivos progridem por valor, enquanto os objetos são passados por referência. Aqui está uma descrição:

Por valor: Quando você passa dados primitivos por este processo, uma cópia do original é feita. Isso significa que quando você está transferindo dados primitivos em uma função, o original não é alterado de forma alguma. A cópia dos dados passa por todos os tipos de modificações.

Por referência: Os tipos de dados primitivos não são transmitidos neste processo. Os objetos são passados por referência. Neste processo, um alias para o original é feito. Aqui, o processo obtém uma referência do objeto. O valor de referência aqui é o endereço desse objeto na memória.

43. Descreva os caracteres de escape em JavaScript.



Os caracteres de escape em JavaScript permitem que os programadores escrevam caracteres especiais sem ter que quebrar o aplicativo. Os caracteres de escape também são conhecidos como folga e também são usados com outros caracteres específicos, como aspas simples e duplas, e comercial, apóstrofes e outros.

No entanto, você deve colocar a folga antes do personagem para fazê-lo aparecer. Por exemplo,

a document.write "ela é uma garota" legal ""

44. Por que o conteúdo completo de um arquivo-fonte JavaScript envolvido em um único livro de funções?

Muitas bibliotecas JavaScript bem conhecidas, mais ou menos, seguem essa prática para desenvolver uma conclusão em torno do conteúdo completo do arquivo de origem JS. Ele permite que os desenvolvedores criem um namespace pessoal, o que ajuda a evitar qualquer tipo de conflito entre os nomes de diferentes bibliotecas e módulos JavaScript.

Esse processo também ajuda a obter um apelido para variáveis globais com muita facilidade.

45. Mencione como um elemento HTML pode ser acessado em um código JavaScript.

```

<head>
  <title>
    :::Welcome to javascript:::
  </title>
</head>

<body>
  <script type = "text/javascript">
    function verifyText ()
    {
      var text = document.getElementById("username").value;
      if(text == "")
      {
        document.getElementById("userlabel").innerHTML
      }
    }
  </script>

  <table>
    <tr>
      <td>Name:</td>
      <td><input type = "text" id = "username" onblur = "verifyText();">
    </tr>
  </table>

```

As seguintes formas são mencionadas abaixo:

getElementById ('id_name'): Esta função irá buscar um elemento de JS por seu nome de ID.

getElementsByClass ('class_name'): Este método obterá todos os elementos com os nomes de classe atribuídos.

getElementsByTagName ('tag_name'): Este método irá buscar todos os elementos com o nome de tag atribuído.

querySelector (): Esta função, após pegar o seletor de estilo CSS, traz de volta seu primeiro elemento escolhido.

46. O que são importações e exportações em JavaScript?

Quando os módulos JavaScript são usados para exportar funções, objetos e outros valores primitivos, a instrução export é usada para que possam ser usados por outros aplicativos com a instrução import. Por exemplo,

```

// — lib.js —
export const sqrt = Math.sqrt;
exportar função quadrada (x) {
  return x * x;
}
função de exportação diag (x, y) {
  retornar sqrt (quadrado (x) + quadrado (y));
}

// — main.js —
{quadrado, diag} de 'lib';
console.log (quadrado (5)); // 25
console.log (diag (4, 3)); // 5

```

47. Qual é a saída do código a seguir?

```

var Saída = (função (x)
{
  Excluir X;
  return X;
})
(0);
console.log (saída);

```

O resultado será zero. O operador Delete em JavaScript é usado para encerrar propriedades de um objeto e não influencia as variáveis locais de forma alguma. E neste programa, x é uma variável local. Portanto, a exclusão do operador não afetará x, e o resultado será zero.

48. Qual é o resultado do seguinte código:

```
var X = {foo: 1};
var Saída = (função ()
{
delete X.foo;
return X.foo;
}
)();
console.log (saída);
```

O resultado será indefinido. Aqui, sendo x um operador local, contém a propriedade foo. X também é uma função autoexecutável. E o operador delete é usado para excluir qualquer propriedade de um objeto. Portanto, a exclusão removerá a propriedade foo do objeto x. Em seguida, a saída tentará ter uma referência em vez da propriedade foo levando ao resultado indefinido.

49. Determine a saída do código a seguir.

```
var Employee =
{
empresa: 'abc'
}
var Emp1 = Object.create (funcionário);
excluir Emp1.company Console.log (emp1.company);
```

A saída aqui será abc. O operador delete em JavaScript exclui a propriedade de um objeto. Aqui, porém, emp1 é um objeto, mas tem companhia como propriedade de protótipo. Conseqüentemente, o operador delete exclui a propriedade company imediatamente do objeto Employee.

50. Por favor, descubra a saída do código abaixo.

```
// nfe (expressão de função nomeada)
var Foo = Barra de funções ()
{
return 7;
};
barra de tipo ();
```

Em JavaScript, uma função pode carregar uma única variável de referência como sua identidade de função. Portanto, o resultado mostrará um erro de referência.

Palavras Finais

Aqui termina nossa coleção de perguntas frequentes sobre a entrevista sobre JavaScript. Com isso, espera-se cobrir a maioria das possíveis perguntas da entrevista de JavaScript. No entanto, se perdemos algo ou se você tiver alguma dúvida em relação a todo o processo da entrevista, avise-nos. Além disso, compartilhe este artigo com seus amigos que estão tentando acessar o quadro de perguntas da entrevista JS.

instagram stories viewer (<https://dumpor.com>)

Search

Categorias

- Comandos A Z (/pt/parts/10-a-z-commands)
- Android (/pt/parts/11-android)
- Ecossistema Apple (/pt/parts/12-apple-ecosystem)
- Apple Tv (/pt/parts/13-apple-tv)
- So Chrome (/pt/parts/14-chrome-os)
- Computação Em Nuvem (/pt/parts/15-cloud-computing)
- Dicas De Informática (/pt/parts/16-computer-tips)
- Sites Legais (/pt/parts/17-cool-websites)
- Criptomoeda (/pt/parts/18-cryptocurrency)
- Ciência De Dados (/pt/parts/19-data-science)
- Downloads De Software Grátis (/pt/parts/20-free-software-downloads)
- Coisas Divertidas (/pt/parts/21-fun-stuff)
- Aparelhos (/pt/parts/22-gadgets)
- Jogos (/pt/parts/23-gaming)
- Software / Dicas Do Google (/pt/parts/24-google-software-tips)
- Dicas Do Internet Explorer (/pt/parts/25-internet-explorer-tips)
- Iot Internet Das Coisas (/pt/parts/26-iot--internet-of-things)
- Ipad (/pt/parts/27-ipad)
- Iphone (/pt/parts/28-iphone)
- Linux (/pt/parts/29-linux)
- Postagens Em Destaque (/pt/parts/2-featured-posts)
- Mac (/pt/parts/30-mac)
- Mac Os X (/pt/parts/31-mac-os-x)
- Ml & Ai (/pt/parts/32-ml--ai)
- Sra. Dicas De Escritório (/pt/parts/33-ms-office-tips)
- Ferramentas Online (/pt/parts/34-online-tools)
- Software Livre (/pt/parts/35-open-source-software)
- Dicas De Programação (/pt/parts/36-programming-tips)
- Computador De Placa única (/pt/parts/37-single-board-computer)
- Windows Xp (/pt/parts/38-windows-xp)

Última postagem do blog

- AUG** 100 perguntas e respostas mais frequentes da entrevista sobre Java
3 (/pt/chapters/6146-100-frequently-asked-java-interview-questions-and-answers)
- AUG** Os 20 melhores livros de Matlab para desenvolvedores iniciantes e especialistas
3 (/pt/chapters/6031-the-20-best-matlab-books-for-beginner-and-expert-developers)
- AUG** Os 20 melhores livros de Java para a aprendizagem da programação básica de Java
2 (/pt/chapters/6015-the-20-best-java-books-for-learning-core-java-programming)

Categorias

- › Comandos A Z (/pt/parts/10-a-z-commands)
- › Android (/pt/parts/11-android)
- › Ecossistema Apple (/pt/parts/12-apple-ecosystem)
- › Apple Tv (/pt/parts/13-apple-tv)

- › [So Chrome \(/pt/parts/14-chrome-os\)](/pt/parts/14-chrome-os)
- › [Computação Em Nuvem \(/pt/parts/15-cloud-computing\)](/pt/parts/15-cloud-computing)
- › [Dicas De Informática \(/pt/parts/16-computer-tips\)](/pt/parts/16-computer-tips)
- › [Sites Legais \(/pt/parts/17-cool-websites\)](/pt/parts/17-cool-websites)
- › [Criptomoeda \(/pt/parts/18-cryptocurrency\)](/pt/parts/18-cryptocurrency)
- › [Ciência De Dados \(/pt/parts/19-data-science\)](/pt/parts/19-data-science)

Mais recentes

- › [Como usar a navegação Caret \(/pt/chapters/11109-how-to-use-caret-browsing\)](/pt/chapters/11109-how-to-use-caret-browsing)
- › [Os Chromebooks têm Bluetooth? \(/pt/chapters/11110-do-chromebooks-have-bluetooth\)](/pt/chapters/11110-do-chromebooks-have-bluetooth)
- › [Como fazer a interface da tela de sete segmentos com o Raspberry Pi 4 \(/pt/chapters/11064-how-to-interface-seven-segment-display-with-raspberry-pi-4\)](/pt/chapters/11064-how-to-interface-seven-segment-display-with-raspberry-pi-4)

© 2022 Best Tech Tips (<https://ciksiti.com/>) All Rights Reserved.

