

AQUI ESTÃO AS 20 PERGUNTAS SOBRE JAVASCRIPT QUE SERÃO FEITAS EM SUA PRÓXIMA ENTREVISTA

Tecnologia 117 Views



Uma entrevista é uma parte importante do processo de contratação. Ele fornece uma visão sobre as habilidades, conhecimentos e capacidade de um indivíduo para realizar tarefas desafiadoras. Essa também é uma das maneiras mais confiáveis de um empregador filtrar candidatos inadequados para um anúncio de emprego.

Como desenvolvedor de JavaScript, você precisa se preparar antes de aparecer em um exame ou entrevista. Isso aumenta suas chances de ser contratado por uma empresa conceituada.

Agora você deve estar se perguntando por onde começar quando se trata de preparar e conhecer as perguntas que estão sendo feitas. Para ajudá-lo, reuni uma lista de perguntas frequentes sobre entrevistas JavaScript. Cada uma dessas perguntas tem uma resposta curta que você pode verificar a seguir.

No final, você estará um pouco mais preparado e confiante para responder a quaisquer perguntas que seu entrevistador possa fazer. Então vamos começar.

O que é JavaScript?

JavaScript é uma linguagem de script criada pela Netscape em 1995. Ele foi originalmente usado em um navegador da Web popular chamado 'Netscape Navigator'. 'No entanto, hoje em dia podemos usá-lo para o desenvolvimento de aplicativos do lado do cliente e do lado do servidor.

Bem ... Acho que ninguém vai perguntar sobre isso, mas não devemos esquecer de onde vem o JavaScript.

Você pode listar algumas vantagens do JavaScript?

- JavaScript pode funcionar offline no navegador da web.
- Suporta vários paradigmas de programação. Por exemplo, podemos usar conceitos de programação orientada a objetos, funcionais e imperativos.
- Possui a maior coleção de bibliotecas e estruturas de código aberto.
- JavaScript pode criar jogos online e offline, software de desktop, sites e aplicativos móveis.
- Você não precisa aprender linguagens de programação separadas para criar o front-end e o back-end de um site. JavaScript é compatível com todos os navegadores da web comuns e pode ser executado no servidor com Node.js.
- É uma linguagem interpretada. Dito isso, não precisamos construir ou compilar o código antes de usá-lo. As instruções JavaScript são executadas diretamente.

Devemos usar JavaScript interno ou externo?

O JavaScript interno é mais adequado quando só precisamos usá-lo em uma única página da web. Sempre use um arquivo JavaScript externo para sites com várias páginas da web.

Por que plug-ins do WordPress como o Autoptimize agregam o código JavaScript em um arquivo?

Ao combinar o código-fonte JavaScript em um único arquivo, o número de solicitações ao servidor ao gerar uma página da web é reduzido. Isso, por sua vez, acelera o carregamento de um site.

Suponha que incluamos dez arquivos JavaScript em uma página da web. Quando abrimos esta página da web, nosso navegador da web envia dez solicitações de HTTP ao servidor para obter esses arquivos. Por outro lado, se combinarmos o código de todos esses arquivos em um, tudo o que temos a fazer é fazer uma solicitação ao servidor.

Qual é o levantamento do JavaScript?

Lifting é um conceito em JavaScript que nos permite usar variáveis e funções antes mesmo de serem declaradas.

Quando executamos o código JavaScript, ele primeiro extrai automaticamente todas as declarações de variáveis e funções do código e as move para o início de seu escopo. Em seguida, o código é executado.

A principal vantagem do levantamento é que nosso código funciona corretamente e não exibe erros como “variável indefinida” ou “função indefinida”.

Para obter mais informações sobre levantamento, consulte meu artigo, [Compreendendo variáveis, escopo e levantamento em JavaScript](#).

Você pode prever a saída do código a seguir?

Código:

```
username = "Juan Cruz Martinez";  
var username; // variable declaration  
var username; // redeclared the variable  
console.log(username);
```

Resultado:

Juan Cruz Martinez

Explicação:

- Inicialmente, a declaração da variável na linha 2 e linha 3 é considerada uma instrução.
- Agora o conceito de levantamento é aplicado. Ou seja, JavaScript move a declaração da variável para cima. Em seguida, o código é executado.
- Além disso, lembre-se de que o valor armazenado em uma variável não será perdido, mesmo se redeclarmos a variável.

O que é JavaScript ‘Modo estrito’?

O comportamento padrão do JavaScript é muito tolerante no caso de cometermos um pequeno erro. Isso significa que nenhuma mensagem de erro será exibida. Mas às vezes, no desenvolvimento, precisamos ver todos os tipos de erros e avisos para depurar o código.

É aqui que entra o uso do “Modo estrito” em JavaScript. Basicamente, é uma variante restrita em que o JavaScript exibe todos os erros e avisos, mesmo que sejam erros silenciosos.

Podemos ativar o “Modo estrito” usando o

```
"use strict";
```

Diretiva no início do nosso script.

Quais são algumas alternativas ao Svelte?

Svelte é uma estrutura de desenvolvimento de front-end para a linguagem de programação JavaScript.

Algumas de suas alternativas populares são:

O que são funções de chamada automática?

As funções de chamada automática são ligeiramente diferentes das funções normais, pois são executadas imediatamente onde são declaradas.

Normalmente declaramos uma função primeiro e apenas a chamamos mais tarde. No entanto, o JavaScript executa automaticamente o código das funções de chamada automática no tempo de execução.

Deve-se observar que essas funções não têm nome. Novamente, não podemos nos lembrar desses tipos de funções. Eles também são conhecidos como “funções anônimas”.

Aqui está um exemplo de funções de chamada automática:

```
(function () {  
    var a = 12;  
    var b = 3;  
    console.log(a * b);  
})();
```

Qual é a diferença entre ‘var’, ‘let’ e ‘const’?

var	let	const
It is available right from the beginning when the JavaScript was introduced.	It is a new way to declare variables in JavaScript, starting from ES6.	const is used to store a value that will not be changed throughout the execution of the script. It is also introduced recently in ES6.
It has a global/function scope.	It has block scope.	It also has block scope.
Can be updated or re-declared in its scope.	We can't re-declare them.	const represents a constant value, so it can't be updated or re-declared.

Para obter mais informações sobre isso, consulte meu artigo “Noções básicas sobre variáveis, escopo e levantamento em JavaScript”.

Qual é a diferença entre ‘==’ e ‘===’?

Ambos são usados em JavaScript para fazer uma comparação entre dois valores.

== operator	=== operator
It is a Type Converting Operator .	It is a Strict Equality Operator .
It is only used to compare two values.	It compares two values as well as their types.

Há alguma diferença entre palavras-chave “nulas” e “indefinidas”?

Ambas as palavras-chave representam um valor vazio. No entanto, existem duas diferenças fundamentais entre nulo e indefinido.

null	undefined
We declare a variable and assign it a value of <code>null</code> .	We declare a variable without assigning any value.
type of (null) object	type of (undefined) object

Você pode dizer a diferença entre ‘declaração de função’ e ‘expressão de função’?

Explicação basicamente funcional’ nada mais é do que o processo normal definir uma função com a palavra-chave

```
function
```

, seu nome único, Parâmetros e o corpo da função. Enquanto, quando atribuímos uma função Explicação para uma variável então se torna ‘expressão de função’. É interessante notar que as expressões de função não são puxadas para cima. Isso significa que se você tentar invocá-los antes de definir, eles mostrarão um erro.

Explicação da função:

```
function full_name(first_name, last_name)
{
    return first_name + " " + last_name;
}
```

Expressão da função:

```
var addition = function add(value_1, value_2)
{
    return value_1 + value_2;
}
```

O que é um fechamento?

Um 'encerramento' em JavaScript é uma função dentro de outra função. A função interna tem acesso às suas próprias variáveis, às variáveis definidas na função externa e às variáveis globais.

Exemplo de fechamento:

```
var a = 2;

function outer()
{
    var b = 4;

    function inner()
    {
        var c = 6;

        console.log(a + b + c);
    }

    inner();
}

outer();
```

Você pode prever a saída dessas duas funções? E eles vão devolver o problema imediatamente ou não?

```
function user1()
{
    return {
        name: "Juan"
    };
}

function user2()
{
    return
    {
        name: "Juan"
    };
}

console.log(user1());
console.log(user2());
```

Saída da função user1 ()

```
{name: "Juan"}
```

Saída da função user2 ()

```
undefined
```

O que é NaN?

Em JavaScript, NaN significa “Not a Number”. Este é um valor especial que ocorre quando não podemos realizar uma operação.

Por exemplo, e se tentarmos dividir uma string por um número (por exemplo, “Hello World” / 5)?

Você pode explicar o loop for-in?

O loop for-in é especialmente projetado para percorrer todas as propriedades do objeto. Em cada iteração, ele seleciona uma propriedade do objeto e executa as operações necessárias sobre ela.

Vamos tentar entender com um exemplo:

```
var user = {  
  "name": "Juan",  
  "country": "Germany",  
  "website": "livecodestream.dev"  
};  
  
for(var key in user)  
{  
  if(user.hasOwnProperty(key))  
  {  
    console.log(key + " -> " + user[key]);  
  }  
}
```

Resultado:-

```
name -> Juan  
country -> Germany  
website -> livecodestream.dev
```

O que é “borbulhamento de evento” e “captura”?

No JavaScript DOM, os elementos HTML são aninhados uns nos outros para formar uma hierarquia.

Agora, se pais e filhos registraram um identificador para um evento específico, qual é a ordem em que o evento é propagado?

Ele pode ser determinado de duas maneiras, conhecidas como formação e detecção de bolhas de eventos.

Quando os eventos borbulham, a criança primeiro captura o evento e depois o passa para os pais.

Enquanto na captura de evento, o pai captura o evento primeiro e depois o passa para os filhos.

Qual é a diferença entre JavaScript e ECMA Script?

JavaScript é uma linguagem de script, enquanto ECMA Script é uma coleção de diretrizes e regras para padronizar o JavaScript em vários navegadores da web.

Como você cria um cookie com JavaScript?

Em JavaScript, um cookie pode ser criado usando um objeto `document.cookie`. Basta atribuir a ele um valor de string que é apenas uma série de pares de valores-chave separados por ponto e vírgula.

Exemplo de cookie de criação de JavaScript:

Conclusão

Os empregadores geralmente tentam confundir os candidatos fazendo perguntas complicadas. Portanto, se você não estiver bem preparado, provavelmente perderá a oportunidade.

Então, hoje tentei responder a algumas perguntas comuns sobre entrevistas de JavaScript. Você pode até mesmo usá-lo como referência antes de ir para uma entrevista.

Esta Itens foi publicado originalmente em Transmissão de código ao vivo por Juan Cruz Martinez (Twitter: @bajcmartinez), Fundador e editor do Live Code Stream, empresário, desenvolvedor, autor, palestrante e criador de coisas.

Transmissão de código ao vivo também está disponível como um boletim informativo semanal gratuito. Inscreva-se para receber atualizações sobre tudo relacionado a programação, IA e ciência da computação em geral.

Publicado em 14 de outubro de 2020 – 06:30 UTC

Source link
