**Interview Questions 15m**

1. **Could you name some built-in methods in JavaScript?**
   1. **charAt():** Returns the character at the specified index.
   2. **charCodeAt():** Returns a number indicating the Unicode value of the character at the given index.
   3. **concat():** Combines the text of two strings and returns a new string.
   4. **indexOf():** Returns the index within the calling String object of the first occurrence of the specified value, or -1 if not found.
   5. **lastIndexOf():** Returns the index within the calling String object of the last occurrence of the specified value, or -1 if not found.
   6. **localeCompare():** Returns a number indicating whether a reference string comes before or after or is the same as the given string in sort order.
   7. **length():** Returns the length of the string.

**2. What are the escape characters in JavaScript?**

JavaScript uses the \ (backslash) as an escape character for:

**\'** single quote.

**\"** double quote.

**\\** backslash.

**\n** new line.

**\r** carriage return.

**\t** tab.

**\b** backspace.

**\f** form feed.

1. **Who developed JavaScript ?**

The first **ever** JavaScript was created by Brendan Eich at Netscape, and has since been updated to conform to ECMA-262 Edition 5 and later versions.

1. **How to use a prompt box in JS?**

A prompt box is often used if you want the user to input a value before entering a page.

When a prompt box pops up, the user will have to click either "OK" or "Cancel" to proceed after entering an input value.

If the user clicks "OK" the box returns the input value. If the user clicks "Cancel" the box returns null.

1. **What is JavaScript ‘Strict Mode’?**

JavaScript's strict mode, introduced in ECMAScript 5, is a way to opt in to a restricted variant of JavaScript, thereby implicitly opting-out of "sloppy mode". Strict mode isn't just a subset: it intentionally has different semantics from normal code. Browsers not supporting strict mode will run strict mode code with different behavior from browsers that do, so don't rely on strict mode without feature-testing for support for the relevant aspects of strict mode. Strict mode code and non-strict mode code can coexist, so scripts can opt into strict mode incrementally.

Strict mode makes several changes to normal JavaScript semantics:

Eliminates some JavaScript silent errors by changing them to throw errors.

Fixes mistakes that make it difficult for JavaScript engines to perform optimizations: strict mode code can sometimes be made to run faster than identical code that's not strict mode.

Prohibits some syntax likely to be defined in future versions of ECMAScript.

*JavaScript'in ECMAScript 5'te tanıtılan katı modu, JavaScript'in kısıtlı bir türevini seçmenin ve dolayısıyla örtük olarak "özensiz modu" devre dışı bırakmanın bir yoludur. Katı mod yalnızca bir alt küme değildir: kasıtlı olarak normal koddan farklı anlamlara sahiptir. Katı modu desteklemeyen tarayıcılar, bunu yapan tarayıcılardan farklı davranışlarla katı mod kodu çalıştıracaktır, bu nedenle katı modun ilgili yönleri için destek için özellik testi yapmadan katı moda güvenmeyin. Katı mod kodu ve katı olmayan mod kodu bir arada bulunabilir, bu nedenle komut dosyaları kademeli olarak katı modu seçebilir.*

*Katı mod, normal JavaScript semantiğinde birkaç değişiklik yapar:*

*Bazı JavaScript sessiz hatalarını, hataları atmak üzere değiştirerek ortadan kaldırır.*

*JavaScript motorlarının optimizasyon gerçekleştirmesini zorlaştıran hataları düzeltir: katı mod kodu bazen katı mod olmayan özdeş koddan daha hızlı çalışacak şekilde yapılabilir.*

*ECMAScript'in gelecekteki sürümlerinde tanımlanması muhtemel bazı sözdizimlerini yasaklar.*

1. **What is the difference between ‘var’ and ‘let’ keyword?**

ECMAScript 6 introduced the let statement. it's described as a local variable.

The var variables belong to the global scope when you define them outside a function.

1. **What is JavaScript Hoisting?**

JavaScript Hoisting refers to the process whereby the interpreter appears to move the declaration of functions, variables or classes to the top of their scope, prior to execution of the code. ... Variable and class declarations are also hoisted, so they too can be referenced before they are declared. JavaScript only hoists declarations, not initializations.

1. **What is mean by NULL in JavaScript?**

The value null represents the intentional absence of any object value. It is one of JavaScript's primitive values and is treated as falsy for boolean operations.

It simply represents the **absence** of **any object value (**If you are from Java backgound then it could simply mean an object pointing to nothing**).**