

<https://www.guru99.com/javascript-interview-questions-answers.html>

Csak azokat a kérdéseket gyűjtöttem össze innen, amiket nem szerepeltek az előző kérdéssorban.

Mi a különbség a ViewState és a SessionState között?

A 'ViewState' a munkamenet (session) egyik oldalára jellemző.

A „SessionState” a felhasználóspecifikus adatokra vonatkozik, amelyek a webalkalmazás minden oldalán elérhetők.

Mi az === operátor?

Az === -t szigorú (strict) egyenlőségi operátornak hívják, amely igaz értéket ad vissza, amikor a két operandus konverzió nélkül azonos értékű.

Mit nevezünk variable type-nak a Javascriptben?

Variable typing számot rendelünk egy változóhoz, majd ugyanahoz a változóhoz rendelhető egy karakterlánc.

```
i = 10;  
i = "string";
```

Mi a delete operátor funkciója?

A delete kulcsszó a property, valamint annak értékének törlésére szolgál.

```
var student= {age:20, batch:"ABC"};  
delete student.age;
```

Mi a JavaScript-ben az undefined jelentése?

- A kódban használt változó nem létezik
- A változó nincs hozzárendelve egyetlen értékhez sem
- Property nem létezik

Milyen típusú Pop-up boxok érhetők el a JavaScript-ben?

- Alert
- Confirm
- Prompt

Mi a Void (0) használata?

- A Void (0) az oldal frissítésének megakadályozására szolgál, és a "null" paraméter átadásra kerül sor hívás közben.
- A Void (0) más metódus meghívására szolgál az oldal frissítése nélkül.

Említse meg, mi a hátránya az innerHTML JavaScript használatának?

- A tartalmat mindenhol kicseréli, tehát Nem használhatjuk a "inner HTML-hez való hozzáfűzést". Még akkor is, ha a + = karaktert használja, mint a "innerHTML = innerHTML + 'html'", a régi tartalmat mégis html váltja fel
- A teljes innerHTML L-tartalmat újra elemzik és elemekké építik, ezért sokkal lassabb a feldolgozás
- A innerHTML nem nyújt validálást, ezért nem valid HTML-t is be tudunk így illeszteni egy HTML-be

Milyen típusú hibák vannak a JavaScriptben?

- Load time errors: A weblap betöltésekor felmerülő hibák, mint a helytelen szintaxis hibák, Load time hibáknak nevezik, és dinamikusan generálják a hibákat.
- Run time errors: A HTML-nyelven belüli parancs nem megfelelő használatából eredő hibák.
- Logical Errors: Ezek azok a hibák, amelyek a különböző műveleteket végrehajtó függvények rossz logikájából adódnak.

Magyarázza meg az window.onload és onDocumentReady?

- Az onload funkció addig nem fut, amíg az oldalon található összes információ be nem töltődik. Ez jelentős késleltetést eredményez bármelyik kód végrehajtása előtt.
- onDocumentReady a DOM betöltése után tölti be a kódot. Ez lehetővé teszi a kód korai manipulálását.

Mi a különbség a .call () és a .apply () között?

A .call () és .apply () függvények nagyon hasonlóak a használatukban, egy kis különbség kivételével. A .call () akkor használható, ha a függvény argumentumainak száma ismert a programozó számára, mivel ezeket argumentumként kell megemlíteni a call utasításban. Másrészt a .apply () akkor használható, ha ez a szám nem ismert. A .apply () függvény elvárja, hogy az argumentum tömb legyen.

Az `.call()` és az `.apply()` közötti alapvető különbség abban rejlik, hogy az argumentumok átadódnak a függvénynek. Használatukat a megadott példa szemlélteti.

```
var someObject = {
  myProperty : 'Foo',

  myMethod : function(prefix, postfix) {

    alert(prefix + this.myProperty + postfix);
  }
};
someObject.myMethod('<', '>'); // alerts '<Foo>'
var someOtherObject = {

  myProperty : 'Bar'

};
someObject.myMethod.call(someOtherObject, '<', '>'); // alerts '<Bar>'
someObject.myMethod.apply(someOtherObject, ['<', '>']); // alerts '<Bar>'
```

Mi az event bubbling?

A JavaScript lehetővé teszi a DOM elemek egymásba ágyazását. Ilyen esetben, ha a gyermek kezelőjére kattintanak, a szülő kezelője is úgy fog működni, mintha rá is kattintanának.

Magyarázza el a halasztott szkriptek (deferred script) szerepét a JavaScript-ben?

Alapértelmezés szerint a HTML-kód elemzése az oldal betöltésekor szünetel, amíg a parancsfájl nem áll le. Ez azt jelenti, hogy ha a szerver lassú vagy a parancsfájl különösen bonyolult, sok időbe telik, akkor a weboldal késéssel jelenik meg. A Halasztott szkriptek (deferred script) használata közben a parancsfájlok késleltetik a parancsfájl végrehajtását, amíg a HTML elemző (parser) fut. Ez lerövidíti a weboldalak betöltési idejét, és gyorsabban jelennek meg.

Melyek a JavaScript funkcionális komponensei?

First-class functions: A JavaScript függvényeit első osztályú objektumként használják. Ez általában azt jelenti, hogy ezeket a függvényeket argumentumként továbbadhatjuk más függvényeknek, visszaadhatjuk más függvények értékeként, változókhoz rendelhetjük vagy tárolhatjuk az adatstruktúrákban is.

Beágyazott függvények: A többi függvényen belül definiált függvényeket beágyazott függvényeknek nevezzük. Úgy hívják őket, hogy minden alkalommal a fő függvényt használják.

Magyarázza az unshift () metódust?

Ez a módszer a tömb kezdetén működik, ellentétben a `push()`-val. Hozzáadja a kívánt számú elemet egy tömb elejére. Például –

```
var name = [ "john" ];
name.unshift( "charlie" );
name.unshift( "joseph", "Jane" );
console.log(name);
```

Megoldás:

```
["joseph ", "Jane ", "charlie ", "john "]
```

Vázolja az `unescape()` és az `escape()` függvényeket?

Az `escape()` függvény felelős egy karakterlánc kódolásáért annak érdekében, hogy az információt az egyik számítógépről a másikra továbbítsa a hálózaton.

```
<script>
    document.write(escape("Hello? How are you!"));
</script>
```

Kimenet: Hello% 3F% 20Hogy% 20are% 20you% 21

Az `unescape()` függvény nagyon fontos, mivel dekódolja a kódolt karakterláncot.

A következő módon működik. Például:

```
<script>
    document.write(unescape("Hello%3F%20How%20are%20you%21"));
</script>
```

Output: Hello? How are you!

Mi a `decodeURI()` és az `encodeURIComponent()`?

Az `encodeURIComponent()` az URL-t hex kódolássá alakítja. A `decodeURI()` pedig a kódolt URL normálissá alakítására szolgál.

```
<script>
    var uri="my test.asp?name=ståle&car=saab";

    document.write(encodeURIComponent(uri)+ "<br>");

    document.write(decodeURI(uri));
</script>
```

Output -

my%20test.asp?name=st%C3%A5le&car=saab

my test.asp?name=ståle&car=saab