

<https://www.edureka.co/blog/interview-questions/javascript-interview-questions/>

Csak azokat a kérdéseket tartalmazza, amit a többi dokumentum nem

Mi a Callback?

A Callback egy egyszerű JavaScript-függvény, amelyet argumentumként átadnak valamilyen metódusnak. Ez egy olyan függvény, amelyet egy másik függvény végrehajtása után kell végrehajtani, ezért a „Callback” elnevezés. A JavaScript-ben a függvények objektumok. Emiatt a függvények a függvényeket argumentumként vehetik fel, és más függvények visszaadhatják őket.

Mi az a Closure? Adj egy példát.

A Closure akkor jön létre, amikor az aktuális hatókörön kívül definiált változó valamilyen belső hatókörön belül érhető el. Ez egy belső függvényből hozzáférést biztosít egy külső függvény hatóköréhez. A JavaScript-ben minden egyes függvény létrehozásakor Closure-ok jönnek létre. A Closure használatához egyszerűen definiáljon egy függvényt egy másik függvényen belül.

Mi a különbség az attribútumok és a property-k között?

Attribútumok - további részletek az elemről, például id, type, value stb.

Tulajdonság (property) – az attribútumhoz rendelt érték, például type = "text", value = 'Name' stb.

Milyen módszerekkel határozhat meg egy változót a JavaScript-ben?

A változó definiálásának három lehetséges módja a JavaScript-ben:

- var - A JavaScript változók egy var utasítással deklarálhatóak, és opcionálisan inicializálhatjuk a változó értékét. Példa: var a = 10; A változó deklarációkat a kód végrehajtása előtt dolgozzuk fel.
- Const - A const használata nem teszi lehetővé számukra az objektum módosítását. Ha egy függvényt const-ként deklarálunk, akkor bármilyen típusú objektumra meghívható.
- Let - Ez azt jelzi, hogy a változó újra deklarálható, például egy számláló egy ciklusban, vagy egy értékcsere egy algoritmusban. Azt is jelzi, hogy a változót csak abban a blokkban fogják használni, amelyben meghatározták.

Mi a különbség az undeclared és az undefined között?

Az undeclared változók azok, amelyek nem léteznek a programban, és nincsenek deklarálva. Ha a program megpróbálja beolvasni egy undeclared változó értékét, futásidejű hiba lép fel. Undefined változók azok, amelyeket a program deklarál, de nem kapott értéket. Ha a program megpróbálja beolvasni egy undefined változó értékét, akkor egy undefined értéket ad vissza.

Hogyan adhatók át a JavaScript primitív / objektumtípusok a függvényekben?

Az egyik különbség a kettő között az, hogy a primitív adattípusokat érték alapján, az objektumokat pedig referenciával adják át.

- Az érték típusú átadás alatt az eredeti MÁSOLATÁNAK létrehozását kell érteni.
- Hivatkozás átadása azt jelenti, hogy létrehozzunk egy ALIAS-t az eredetihez

Mi az export és az import?

Az import és az export segít a moduláris JavaScript kód megírásában. Az Import és Export használatával több fájlra oszthatjuk a kódunkat. Például

```
//----- lib.js -----</span>
export const sqrt = Math.sqrt;</span>
export function square(x) {</span>
  return x * x;</span>
}
export function diag(x, y) {
  return sqrt(square(x) + square(y));
}

//----- main.js -----</span>
{ square, diag } from 'lib';
console.log(square(5)); // 25
console.log(diag(4, 3)); // 5
```

Mi az oka annak, hogy egy JavaScript-forrásfájl teljes tartalmát egy függvénybe csomagolja?

Ez egyre gyakoribb gyakorlat, amelyet sok népszerű JavaScript könyvtár használ. Ez a technika bezárást (closure) eredményez a fájl teljes tartalma körül, ami talán a legfontosabb, hogy privát névteret hoz létre, és ezáltal segít elkerülni a különféle JavaScript-modulok és könyvtárak közötti esetleges névütközéseket.

Ennek a technikának egy másik jellemzője, hogy lehetővé teszi a globális változó könnyű álnévét. Ezt gyakran használják a jQuery beépülő modulokban.