**3. JavaScript, ECMAScript - Tesztkérdések**

**1. Melyik adattípus NEM létezik JavaScriptben?**

* boolean
* **double** ✅
* undefined

**2. Hogyan lehet egy változót deklarálni JavaScriptben?**

* variable x = 5;
* let x == 5;
* **let x = 5;** ✅

**3. Mi a különbség a let és const kulcsszavak között?**

* **A const nem módosítható, a let igen** ✅
* A let csak globálisan érvényes
* A const törölhető, a let nem

**4. Mi lesz a következő kód kimenete?**

console.log(typeof null);

* null
* **object** ✅
* undefined

**5. Hogyan lehet egy függvény alapértelmezett paraméterértéket adni?**

function sayHello(name = "Guest") {

console.log("Hello " + name);

}

* sayHello(name : "Guest")
* **name = "Guest" a paraméterlistában** ✅
* sayHello.default(name, "Guest")

**6. Mi az a closure JavaScriptben?**

* Egy speciális adatstruktúra
* **Egy belső függvény, amely hozzáfér a külső függvény változóihoz** ✅
* Egy módja a memóriakezelésnek

**7. Mi történik, ha egy változót var-ral deklarálunk egy függvényen belül?**

* **A változó lokális lesz a függvényen belül** ✅
* Globális változó jön létre
* Hibát eredményez

**8. Mi a különbség az == és a === operátorok között?**

* **A === típus-érzékeny, az == nem** ✅
* Mindkettő pontosan ugyanazt csinálja
* Az == gyorsabb

**9. Melyik NEM érvényes JavaScript függvénydeklaráció?**

* function hello() {}
* const hello = function() {}
* **def hello():** ✅

**10. Melyik metódussal lehet egy tömb utolsó elemét eltávolítani?**

* **pop()** ✅
* shift()
* splice(0, 1)

**11. Hogyan lehet egy objektum kulcsait kilistázni?**

* **Object.keys(obj)** ✅
* obj.getKeys()
* obj.mapKeys()

**12. Hogyan lehet egy aszinkron műveletet kezelni JavaScriptben?**

* **Promise vagy async/await segítségével** ✅
* Csak callback-ekkel
* Nem lehet aszinkron műveleteket végezni

**13. Hogyan lehet egy JSON objektumot átalakítani JavaScript objektummá?**

* **JSON.parse(jsonString)** ✅
* JSON.stringify(jsonString)
* JSON.objectify(jsonString)

**14. Melyik metódus NEM létezik egy JavaScript tömbön?**

* map()
* reduce()
* **assign()** ✅

**15. Hogyan lehet egy DOM elemet kiválasztani az ID alapján?**

* document.querySelectorAll("#myId")
* document.getElementsByClassName("myId")
* **document.getElementById("myId")** ✅

**16. Melyik eseménykezelő fut le egy űrlap elküldésekor?**

* **submit** ✅
* change
* keydown

**17. Mi történik, ha egy nem létező objektum tulajdonságára hivatkozunk?**

* **undefined** ✅
* null
* NaN

**18. Hogyan lehet egy tömb összes elemét egy stringgé alakítani?**

* **array.join(", ")** ✅
* array.concat(", ")
* array.toString(", ")

**19. Melyik ES6 funkció teszi lehetővé az objektumok egyszerű másolását?**

* Object.map()
* **Object.assign({}, obj)** ✅
* Object.copy(obj)

**20. Hogyan lehet egy függvényt exportálni egy másik modulba?**

* **export function myFunction() {}** ✅
* module.export = function myFunction()
* export {myFunction} as default

**3. JavaScript, ECMAScript még 20 kérdés**

**3.1. Melyik nem alapértelmezett adattípus JavaScript-ben?**

* **Symbol** ✅
* String
* Boolean

**3.2. Hogyan hozunk létre egy változót JavaScript-ben?**

* let myVar = 10;
* **var myVar = 10;** ✅
* let var myVar = 10;

**3.3. Melyik nem egy érvényes JavaScript függvény szintaxis?**

* function myFunc() {}
* **function = myFunc() {}** ✅
* const myFunc = () => {}

**3.4. Mi a következő kódban a kimenet: console.log(3 + '3');?**

* 6
* **33** ✅
* 9

**3.5. Mi történik, ha egy változót nem deklarálunk let, var vagy const kulcsszóval?**

* A változó nem jön létre
* **Globális változó jön létre** ✅
* A böngésző hibát dob

**3.6. Mi az undefined a JavaScript-ben?**

* Egy változó értéke, amely még nem lett hozzárendelve
* **Egy változó, amely nem létezik** ✅
* A függvény visszatérési értéke, ha nincs return

**3.7. Mi a különbség az == és === operátorok között?**

* **== összehasonlítja az értékeket, === pedig az értékeket és a típusokat** ✅
* Nincs különbség
* == mindig típuskonverziót végez

**3.8. Mi az eseménykezelő JavaScript-ben?**

* **Függvény, amely reagál a felhasználói interakciókra** ✅
* A DOM manipulációs eszköz
* Az alkalmazás indítása

**3.9. Hogyan hozunk létre egy objektumot JavaScript-ben?**

* let myObject = {};
* **let myObject = new Object();** ✅
* let myObject = Object.create();

**3.10. Mi az a this kulcsszó JavaScript-ben?**

* **A függvényhez tartozó objektumra hivatkozik** ✅
* Az aktuális függvényt jelöli
* Az objektum értékére hivatkozik

**3.11. Hogyan írhatunk egy aszinkron függvényt JavaScript-ben?**

* async function myFunc() {}
* **async function myFunc() {}** ✅
* await function myFunc() {}

**3.12. Mi a Promise JavaScript-ben?**

* **Egy objektum, amely egy jövőbeli értéket képvisel** ✅
* Egy aszinkron függvény
* Egy blokk, amely várakozik a kód futtatására

**3.13. Mi a különbség a setTimeout és a setInterval között?**

* **setTimeout egyszer futtatja le a kódot, míg a setInterval folyamatosan ismétli** ✅
* A setInterval csak egyszer hajtódik végre
* A setTimeout nem támogat aszinkron kódot

**3.14. Mi a try-catch blokkok szerepe JavaScript-ben?**

* **Hibák kezelése és elkapása** ✅
* Kódrészletek végrehajtása
* Változók deklarálása

**3.15. Mi az a closure JavaScript-ben?**

* Egy globális változó, amely elérhető minden függvényből
* **Egy függvény, amely hozzáfér a saját lokális változóihoz és a külső függvények változóihoz** ✅
* Egy funkció, amely csak egyszer fut le

**3.16. Mi az event delegation technika JavaScript-ben?**

* A DOM elemre való közvetlen eseménykezelés
* **A szülőelemhez rendelt eseménykezelő, amely a gyermekelemek eseményeit is kezeli** ✅
* A böngésző beállítása, amely lehetővé teszi az események kezelését

**3.17. Melyik nem érvényes JavaScript szintaxis?**

* let myVar = 20;
* **let 1myVar = 20;** ✅
* let my\_var = 20;

**3.18. Hogyan kérhetünk le adatokat egy API-ból JavaScript-ben?**

* **fetch()** ✅
* getData()
* axios()

**3.19. Mi az ECMAScript 6 (ES6) egyik legfontosabb új funkciója?**

* **Arrow function** ✅
* function kulcsszó eltávolítása
* for ciklus eltávolítása

**3.20. Mi az a destructuring JavaScript-ben?**

* **Objektumok és tömbök elemeinek egyszerű kinyerése változókba** ✅
* A változók összeadásának módja
* Egy szintaxis, amely megakadályozza az objektumok módosítását