

# JavaScript gyakorlás

## Bevezetés

Az alábbi feladatokat egyszerűen elvégezhetjük online szerkesztőben, pl.: [StackBlitz](#) (Static típusú projektet válasszunk). A megoldások egy külön dokumentumban találhatók.

## Változók definiálása

### Const vs let kulcsszavak

Milyen kulcsszavakat (`const` vagy `let`) írnál a kérdőjelek helyére? Miért?

```
const f = (scores) => {  
  ??? min = 30;  
  ??? sum = 0;  
  
  scores.forEach((score) => (sum += score));  
  
  if (sum < min) {  
    console.log('Failed');  
  }  
  
  return sum;  
};  
  
f([5, 3, 2]);  
f([7, 12, 20]);
```

## Érték típusok

### Sztringek - template string szintakszis

Mit ír ki az alábbi kódrészlet?

```
const firstName = 'John';  
const age = 25;  
console.log(  
  `Hello, my name is ${firstName} and I'm ${age} ${  
    age === 1 ? 'year' : 'years'  
  } old`  
);
```

## Sztringek - műveletek

Írjunk egy függvényt, amely két sztringet vár paraméterül, a függvény fűzze össze a két sztringet és térjen vissza az eredménnyel. Hívjuk meg ezt a függvényt, majd írjuk ki a visszatérési értéket a console-ra, írjuk még ki az eredmény sztring hosszát, valamint az 5. karaktert. Mit kapunk, ha az összefűzött sztring nincs 5 karakter hosszú?

## Függvények - arrow function szintakszis

Írjuk át egyszerűbb szintaktikai formára az alábbi formára, milyen karaktereket hagyhatunk el?

```
const f = (a) => {  
  return a + 4;  
}
```

## Tömbök - műveletek

Hozzunk létre egy tömböt, amelyben olyan objektumok találhatók, amelyek kulcsai: "firstName", "lastName", "age". Létrehozáskor egy elemet tegyünk a tömbbe, majd a "push" metódus segítségével még két elemet adjunk a tömbhöz. Írjuk ki a tömb hosszát, és a legelső elemet a console-ra. A "splice" metódus segítségével töröljük a 2. elemet a tömbből.

## Primitív vs referencia típusok

Osztályozzuk az alábbi érték típusokat aszerint, hogy primitív vagy referencia típusnak számítanak-e:

- string
- Object
- Function
- Array
- bool
- Date
- number

## Ciklusok és elágazások

### Tömbökön iterálás

Keressünk három féle módot, amivel egy tömbön végigiterálhatunk.

### If-else

Írjuk át az alábbi kifejezést if-else helyett ?: operátorra:

```
const f = (y) => {  
  let res;  
  if (y >= 0) {  
    res = 2 * y;  
  } else {  
    res = -2 * y;  
  }  
  
  return res;  
};  
  
console.log(f(3));  
console.log(f(-5));
```

## Truthy vagy falsy

Mit ír ki?

```
const truthyOrFalsy = (n) => {  
  if (n) {  
    console.log('truthy');  
  } else {  
    console.log('falsy');  
  }  
}  
  
truthyOrFalsy('');  
truthyOrFalsy([]);  
truthyOrFalsy(0);
```

## Logikai operátorok

! operátor - mit ír ki, miért?

```
console.log(!5);
```

