

## 2026.02.18. JS test feladat

0. Ha eddig nem tettet töltsd le a [nodejs-t](#)
1. Futtasd le a projektmappában az `npm i` parancsot.
2. Implementáld a következő függvényeket a math.js file-ban (math.test.js-t ne írd felül):

Függvény	Leírás	Bemenet	Visszatérési érték	Hibakezelés / Megkötések	Példa
<b>add(a, b)</b>	Összeadás	két number	number vagy null	Nem szám bemenet → null	<code>add(2, 3) → 5</code>
<b>subtract(a, b)</b>	Kivonás	két number	number vagy null	Nem szám bemenet → null	<code>subtract(10, 4) → 6</code>
<b>multiply(a, b)</b>	Szorzás	két number	number vagy null	Nem szám bemenet → null	<code>multiply(3, 4) → 12</code>
<b>divide(a, b)</b>	Osztás	két number	number vagy null	Ha <code>b === 0</code> → null Nem szám bemenet → null	<code>divide(20, 5) → 4</code>
<b>sumArray(numbers)</b>	Tömb elemeinek összege	number tömb	number vagy null	Ha nem tömb → null Üres tömb → 0 Nem szám elemek esetén → null	<code>sumArray([1, 2, 3]) → 6</code>
<b>average(numbers)</b>	Tömb számainak átlaga	number tömb	number vagy null	Üres tömb → null Nem tömb → null	<code>average([10, 20]) → 15</code>
<b>max(numbers)</b>	Legnagyobb érték kiválasztása	number tömb	number vagy null	Üres / nem tömb → null	<code>max([3, 7, 2]) → 7</code>
<b>min(numbers)</b>	Legkisebb érték kiválasztása	number tömb	number vagy null	Üres / nem tömb → null	<code>min([3, 7, 2]) → 2</code>

Függvény	Leírás	Bemenet	Visszatérési érték	Hibakezelés / Megkötések	Példa
<b>isEven(number)</b>	Eldönti, hogy a szám páros-e	number	boolean vagy null	Nem egész szám → null, Nem szám → null	<code>isEven(4) → true</code>
<b>factorial(n)</b>	Faktoriális számítása ( $n!$ )	nemnegatív egész szám	number vagy null	Negatív szám → null Nem egész szám → null $0! = 1$ kötelező	<code>factorial(5) → 120</code>

3. Futtasd le az `npx jest` parancsot a teszteléshez