

# WEBFEJLESZTÉS HAGYOMÁNYOS ESZKÖZÖKKEL

4. forduló

TRAINING360

A kategória támogatója: Training360 Kft.

## Ismertető a feladathoz

**A 4. forduló után elérhetőek lesznek a helyezések %-os formában: azaz kiderül, hogy a kategóriában a versenyzők TOP 20% - 40% - 60% -ához tartozol-e!**

Szeretnénk rá felhívni figyelmedet, hogy a játék nem Forma-1-es verseny! Ha a gyorsaságod miatt kilököd a rendesen haladó versenyzőket, kizárást vonhat maga után!

Felhasznált idő: 20:00/20:00

Elért pontszám: 8.57/28

## 1. feladat 2/2 pont

Egy olyan oldalt kell készítened, ahol scrollozáskor pontosan egy teljes képernyőnyi tartalom gördül. Akár lefelé, akár felfelé, ahogy itt is láthatod:



Mit írál az 1. és 2. számmal jelölt helyekre?

```

1  <head>
2    <meta charset="UTF-8">
3    <meta name="viewport" content="width=device-width, initial-scale=1.0">
4    <meta http-equiv="X-UA-Compatible" content="ie=edge">
5    <title>Scroll</title>
6    <style>
7      body {
8        margin: 0;
9      }
10
11     .warapper {
12       overflow-y: scroll;
13       height: 100vh;
14       1: y mandatory;
15     }
16
17     .page {
18       height: 100vh;
19       2: start;
20     }
21   </style>
22 </head>
23
24 <body>
25   <div class="warapper">
26     <div class="page">1</div>
27     <div class="page">2</div>
28     <div class="page">3</div>
29     <div class="page">4</div>
30   </div>
31 </body>

```

## Válasz

- ☐ scroll-snap-align, scroll-snap-type
- ☒ scroll-snap-type, scroll-snap-align  
Ez a válasz helyes, és meg is jelölted.
- ☐ scroll-snap-start, scroll-snap-stop
- ☐ scroll-snap-type, scroll-snap-stop

## Magyarázat

Először is a container-en be kell állítani a *scroll-snap-type*-ot, ami megadja, hogy a tartalom rögzítése melyik tengely mentén és milyen módon történjen.

A *scroll-snap-align* pedig megadja, hogy az elemnek melyik részének kell a containerhez igazodnia.

[Scroll Snap Type](#)

[Scroll Snap Align](#)

## 2. feladat 2.57/18 pont

Rengeteg módszer van arra, hogy mind horizontálisan, mind vertikálisan középre igazítsd a tartalmat.

Válaszd ki, hogy az alábbiak közül melyik kód esetében jelenik meg a szöveg mindkét tengely mentén pontosan középen (desktopon, legfrissebb Chrome), mint a mintában! A piros keret csak a szemléltetés miatt van berajzolva!

**A szöveg megjelenése a lényeg, nem az azt tartalmazó elem pozíciója vagy mérete.**

**Az %-os, vh-s mértékegységek miatt esetlegesen előforduló 1-2 pixellel nem kell foglalkozni!**

Centered content... or not

**Válaszok**



```
1 <head>
2   <meta charset="UTF-8">
3   <meta name="viewport" content="width=device-width, initial-scale=1.0">
4   <meta http-equiv="X-UA-Compatible" content="ie=edge">
5   <title> Centered content... or not</title>
6   <style>
7     body {
8       height: 100vh;
9       margin: 0;
10    }
11
12    table {
13      width: 100%;
14      height: 100%;
15    }
16
17    td {
18      text-align: center;
19      vertical-align: center;
20    }
21  </style>
22 </head>
23
24 <body>
25   <table>
26     <tr>
27       <td>
28         Centered content... or not
29       </td>
30     </tr>
31   </table>
32 </body>
```

Ez a válasz helyes, és meg is jelölted.



```
1  <head>
2    <meta charset="UTF-8">
3    <meta name="viewport" content="width=device-width, initial-scale=1.0">
4    <meta http-equiv="X-UA-Compatible" content="ie=edge">
5    <title>Centered content... or not</title>
6    <style>
7      body {
8        height: 100vh;
9        margin: 0;
10     }
11
12     .table {
13       display: table;
14       width: 100%;
15       height: 100%;
16     }
17
18     .table-cell {
19       display: table-cell;
20       text-align: center;
21       vertical-align: middle;
22     }
23   </style>
24 </head>
25
26 <body>
27   <div class="table">
28     <div class="table-cell">
29       Centered content... or not
30     </div>
31   </div>
32 </body>
```

Ez a válasz helyes, és meg is jelölted.



```
1 <head>
2   <meta charset="UTF-8">
3   <meta name="viewport" content="width=device-width, initial-scale=1.0">
4   <meta http-equiv="X-UA-Compatible" content="ie=edge">
5   <title>Centered content... or not</title>
6   <style>
7     body {
8       height: 100vh;
9       margin: 0;
10    }
11
12    .table {
13      display: table;
14      width: 100%;
15      height: 100%;
16    }
17
18    .table-row {
19      display: table-row;
20    }
21
22    .table-cell {
23      display: table-cell;
24      text-align: center;
25      vertical-align: middle;
26    }
27  </style>
28 </head>
29
30 <body>
31   <div class="table">
32     <div class="table-row">
33       <div class="table-cell">
34         Centered content... or not
35       </div>
36     </div>
37   </div>
38 </body>
```

Ez a válasz helyes, de nem jelölted meg.



```
1 <head>
2   <meta charset="UTF-8">
3   <meta name="viewport" content="width=device-width, initial-scale=1.0">
4   <meta http-equiv="X-UA-Compatible" content="ie=edge">
5   <title>Centered content... or not</title>
6   <style>
7     body {
8       position: relative;
9       height: 100vh;
10      margin: 0;
11    }
12
13    .center-center {
14      position: absolute;
15      top: 25%;
16      left: 25%;
17      height: 50%;
18      width: 50%;
19      margin:auto;
20    }
21  </style>
22 </head>
23
24
25 <body>
26   <div class="center-center">
27     Centered content... or not
28   </div>
29 </body>
```

Ez a válasz helytelen, de megjelölted.



```
1 <head>
2   <meta charset="UTF-8">
3   <meta name="viewport" content="width=device-width, initial-scale=1.0">
4   <meta http-equiv="X-UA-Compatible" content="ie=edge">
5   <title>Centered content... or not</title>
6   <style>
7     body {
8       position: relative;
9       height: 100vh;
10      margin: 0;
11    }
12
13    .center-center {
14      position: absolute;
15      top: 25%;
16      left: 50%;
17      width: 50%;
18      height: 50%;
19      margin-left: -25%;
20    }
21  </style>
22 </head>
23
24 <body>
25   <div class="center-center">
26     Centered content... or not
27   </div>
28 </body>
```

Ez a válasz helytelen, de megjelölted.





```
1 <head>
2   <meta charset="UTF-8">
3   <meta name="viewport" content="width=device-width, initial-scale=1.0">
4   <meta http-equiv="X-UA-Compatible" content="ie=edge">
5   <title>Centered content... or not</title>
6   <style>
7     body {
8       position: relative;
9       height: 100vh;
10      margin: 0;
11    }
12
13    .center-center {
14      position: absolute;
15      top: 50%;
16      left: 50%;
17      transform: translate(-50%, -50%);
18    }
19  </style>
20 </head>
21
22 <body>
23   <div class="center-center">
24     Centered content... or not
25   </div>
26 </body>
```

Ez a válasz helyes, és meg is jelölted.



flex



```
1 <head>
2   <meta charset="UTF-8">
3   <meta name="viewport" content="width=device-width, initial-scale=1.0">
4   <meta http-equiv="X-UA-Compatible" content="ie=edge">
5   <title>Centered content... or not</title>
6   <style>
7     body {
8       display: flex;
9       align-items: center;
10      justify-content: center;
11      margin: 0;
12      height: 100vh;
13    }
14  </style>
15 </head>
16
17 <body>
18   <div>
19     Centered content... or not
20   </div>
21 </body>
```

Ez a válasz helyes, de nem jelölted meg.



```
1 <head>
2   <meta charset="UTF-8">
3   <meta name="viewport" content="width=device-width, initial-scale=1.0">
4   <meta http-equiv="X-UA-Compatible" content="ie=edge">
5   <title>Centered content... or not</title>
6   <style>
7     body {
8       display: grid;
9       align-items: center;
10      justify-content: center;
11      height: 100vh;
12      margin: 0;
13    }
14  </style>
15 </head>
16
17 <body>
18   <div>
19     Centered content... or not
20   </div>
21 </body>
```

Ez a válasz helyes, de nem jelölted meg.



```
1 <head>
2   <meta charset="UTF-8">
3   <meta name="viewport" content="width=device-width, initial-scale=1.0">
4   <meta http-equiv="X-UA-Compatible" content="ie=edge">
5   <title>Centered content... or not</title>
6   <style>
7     body {
8       display: grid;
9       place-items: center;
10      height: 100vh;
11      margin: 0;
12    }
13  </style>
14 </head>
15
16 <body>
17   <div>
18     Centered content... or not
19   </div>
20 </body>
```

Ez a válasz helyes, de nem jelölted meg.

## Magyarázat

Mindössze kettő megoldás volt rossz.

A D.



```
1 <head>
2   <meta charset="UTF-8">
3   <meta name="viewport" content="width=device-width, initial-scale=1.0">
4   <meta http-equiv="X-UA-Compatible" content="ie=edge">
5   <title>Centered content... or not</title>
6   <style>
7     body {
8       position: relative;
9       height: 100vh;
10      margin: 0;
11    }
12
13    .center-center {
14      position: absolute;
15      top: 25%;
16      left: 25%;
17      height: 50%;
18      width: 50%;
19      margin: auto;
20    }
21  </style>
22 </head>
23
24
25 <body>
26   <div class="center-center">
27     Centered content... or not
28   </div>
29 </body>
```

és az E.

```
1 <head>
2   <meta charset="UTF-8">
3   <meta name="viewport" content="width=device-width, initial-scale=1.0">
4   <meta http-equiv="X-UA-Compatible" content="ie=edge">
5   <title>Centered content... or not</title>
6   <style>
7     body {
8       position: relative;
9       height: 100vh;
10      margin: 0;
11    }
12
13    .center-center {
14      position: absolute;
15      top: 25%;
16      left: 50%;
17      width: 50%;
18      height: 50%;
19      margin-left: -25%;
20    }
21  </style>
22 </head>
23
24 <body>
25   <div class="center-center">
26     Centered content... or not
27   </div>
28 </body>
```

Ezekkel is csupán annyi a probléma, hogy nem maga a szöveg kerül középre vertikálisan és horizontálisan, hanem a szöveget tartalmazó div. A div-nek a szélessége és a magassága adott, a tartalom pedig a bal felső sarokban fog kezdődni.

### 3. feladat 4/8 pont

Az alábbiakban 4 kódrészletre látsz 2 vagy több megoldást.

Melyikiek esetében igaz, hogy a különböző verziók valóban ekvivalensek egymással?

#### Válaszok



```
1 // 1. verzió
2 String(42)
3 // 2. verzió
4 42.0.toString()
5 // 3. verzió - a dupla pont nem elírás
6 42..toString()
7 // 4. verzió
8 42.0.toFixed()
```

Ez a válasz helyes, és meg is jelölted.



```
1 // 1. verzió
2 const job = 'hacker';
3 const hasDog = false;
4 const user = { name: 'Warai Otoko' }
5 if(job === 'hacker') user.it = true;
6 if(hasDog) user.role = 'antagonist ';
7
8 // 2. verzió
9 const job = 'hacker';
10 const hasDog = false;
11 const user = {
12   name: 'Warai Otoko',
13   ...job === 'hacker' && { it: true },
14   ...hasDog && { role: 'antagonist' },
15 }
```

Ez a válasz helyes, és meg is jelölted.



```
1 // 1. verzió (Minden argumentum objektum)
2 function objectsMerge() {
3   const object = {}
4   for (let i = 0; i < arguments.length; i += 1) {
5     for (let key in arguments[i]) {
6       object[key] = arguments[i][key];
7     }
8   }
9   return object;
10 }
11
12 // 2. verzió (Minden argumentum objektum)
13 const objectsMerge = (...objects) =>
14   objects.reduce((acc, curr) => ({ ...acc, ...curr }), {});
```

Ez a válasz helytelen, de megjelölted.



```
1 // 1. verzió
2 const compareObjects = (obj1, obj2) => {
3   const obj1Keys = Object.getOwnPropertyNames(obj1).sort();
4   const obj2Keys = Object.getOwnPropertyNames(obj2).sort();
5   return JSON.stringify(obj1Keys) === JSON.stringify(obj2Keys);
6 }
7
8 // 2. verzió
9 const compareObjects = (obj1, obj2) =>
10   Object.keys(obj1).sort().every((key, index) =>
11     key === Object.keys(obj2).sort()[index]);
12
13 // 3. verzió
14 const compareObjects = (obj1, obj2) =>
15   Object.keys(obj1).every((prop) =>
16     obj2.hasOwnProperty(prop));
```

## Magyarázat

Az A-nál:

- `String(42)` - A `String` konstruktorfüggvény visszaad egy stringet : `'42'`
- `42.0.toString()` - Egy lebegőpontos szám `toString` Metódusát meghívva szintén stringet kapunk: `'42'`, Mivel a törtész `.0` volt, ez nem fog megjelenni a stringben
- `42..toString()` - A `42..` tehát dupla ponttal helyes kifejezés, pontosan az előzővel ekvivalens. A `42.0` helyett írhatjuk azt is, hogy `42.`
- `42.0.toFixed()` - A `toFixed()` visszaadja egy szám egész részét stringként, tehát az eredmény itt is `'42'`

Az **B**-nél a spread és **&&** operátorokat kombináljuk egymással. A **&&** balról jobbra értékelődik ki. Amennyiben a bal oldali rész igaz, úgy a jobb oldali részt kapjuk vissza. Ez egy object egy propertyvel, amit a spread hozzáfűz a jelenlegi objektumhoz.

Pl.:

`job === 'hacker' && { it: true }` - ha a bal oldali rész igaz, akkor ezt: `{ it: true }` kapjuk vissza, és ez lesz összemergelve az objektumunkkal.

Az **C**-nél egy apró eltérés van. A dupla *for*-os megoldásban nincs vizsgálva, hogy az adott objektumnak a saját, vagy valamilyen öröklött tulajdonságáról van-e szó. Így ezek nem lesznek kihagyva. Hiányzik egy *if*:

```
if(Object.prototype.hasOwnProperty.call(arguments[i], key))
```

A **D** szintén nem jó. Több sebből is vérzik a dolog.

Például a legegyszerűbb szituáció:

```
const obj1 = { a: 1, b: 2 };
const obj2 = { a: 1, b: 2, c: 3 };
```

A három függvény visszatérési értéke

- *false*
- *true*
- *true*

Tehát a második kettő nem a teljes egyezést vizsgálja.



[Legfontosabb tudnivalók](#)  [Kapcsolat](#)  [Versenyszabályzat](#)  [Adatvédelem](#) 

© 2023 Human Priority Kft.

KÉSZÍTETTE 

Megjelenés

 Világos 