

WEBFEJLESZTÉS HAGYOMÁNYOS ESZKÖZÖKKEL

2. forduló



A kategória támogatója: Training360 Kft.

Ismertető a feladathoz

Útmutató:

- A **radio button-os kérdésekre** egy helyes válasz van.
- **Ha lejár a feladatlap ideje, a rendszer AUTOMATIKUSAN** beküldi azt az addig megjelölt válaszokkal.
- Az **adatbekérős feladatokra NEM jár részpontszám**, csak a feleletválasztósakra.
- **Badge-ke**t a 4.forduló után kapsz majd először.
- Az **adatbekérős kérdéseknél** igyekeztünk minden variációt megadni (kisbetű, nagybetű, szóköz), de ha mégis eltérést tapasztalsz a megoldásokban, kérjük, jelezd felénk!

+1: Azért szólunk, hogy senkit ne a végén érjen meglepetés: a játék nem tipp-mix és csapatkategória sincs! Természetesen akinek nem inge...

Jó versenyzést kívánunk!

Felhasznált idő: 00:00/30:00

Elért pontszám: 0/24

1. feladat 0/8 pont

Mivel szereted a változatosságot, egyszerűsíteni a kódot - és mert kivételesen az idődbe is belefért - több lehetséges megoldást is készítettél az alábbi elvárt design megvalósítására.

Melyik megoldás esetében lesz a két kód (A. és B. verzió) ekvivalens egymással?

Az első 3 feladatnál csak a CSS kód különbözik, a HTML kód ugyanaz!

Kerekítésekkel, töredék pixelekkal nem kell foglalkozni!

Válasz

```
<!DOCTYPE html>
<html lang="hu">

  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Align</title>
  </head>

  <body>
    <div class="content content--center-center">
      Center of the Universe
    </div>
  </body>

</html>
```

```
/* A. verzió */
.content--center-center {
  position: absolute;
  left: 50%;
  top: 50%;
  translate: -50% -50%;
}
```

```
/* B. verzió */
.content--center-center {
  display: flex;
  align-items: center;
  justify-content: center;
}
```

```
<!DOCTYPE html>
<html lang="hu">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Aspect ratio</title>
</head>

<body>
  <div class="container">
    
  </div>
</body>

</html>
```

```
/* A. verzió */
.container {
  width: 25%;
  padding-top: 100%;
  height: 0;
  position: relative;
}

.container img {
  width: 100%;
  position: absolute;
  top: 0;
  left: 0;
}
```

```
/* B. verzió */
.container {
  width: 25%;
}

.container img {
  aspect-ratio: 1 / 1;
  width: 100%;
  object-fit: cover;
  object-position: center;
}
```

```
<!DOCTYPE html>
<html lang="hu">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Clamp</title>
</head>

<body>
  <div class="content">Shut(u) the hell...</div>
</body>

</html>
```

```
/* A. verzió */
.content {
  border: 10px solid red;
  width: clamp(240px, 50%, 960px);
}
```

```
/* B verzió */
.content {
  border: 10px solid red;
  width: 50%;
}

@media screen and (min-width: 1280px) {
  .content {
    width: 960px;
  }
}

@media screen and (max-width: 640px) {
  .content {
    width: 240px;
  }
}
```

```

<!-- A. verzió -->
<!DOCTYPE html>
<html lang="hu">

  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Table like</title>
    <style>
      .row {
        display: flex;
        flex-direction: column;
      }

      .column {
        display: flex;
      }

      .cell {
        width: 33.33%
      }
    </style>
  </head>

  <body>
    <div class="row">
      <div class="column">
        <div class="cell">content 1</div>
        <div class="cell">content 2</div>
        <div class="cell">content 3</div>
      </div>
      <div class="column">
        <div class="cell">content 4</div>
        <div class="cell">content 5</div>
        <div class="cell">content 6</div>
      </div>
    </div>
  </body>
</html>

```

```

<!-- B. verzió -->
<!DOCTYPE html>
<html lang="hu">

  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Table like</title>
    <style>
      .grid {
        display: grid;
        grid-template-columns: 1fr 1fr 1fr;
      }
    </style>
  </head>

  <body>
    <div class="grid">
      <div class="cell">content 1</div>
      <div class="cell">content 2</div>
      <div class="cell">content 3</div>
      <div class="cell">content 4</div>
      <div class="cell">content 5</div>
      <div class="cell">content 6</div>
    </div>
  </body>
</html>

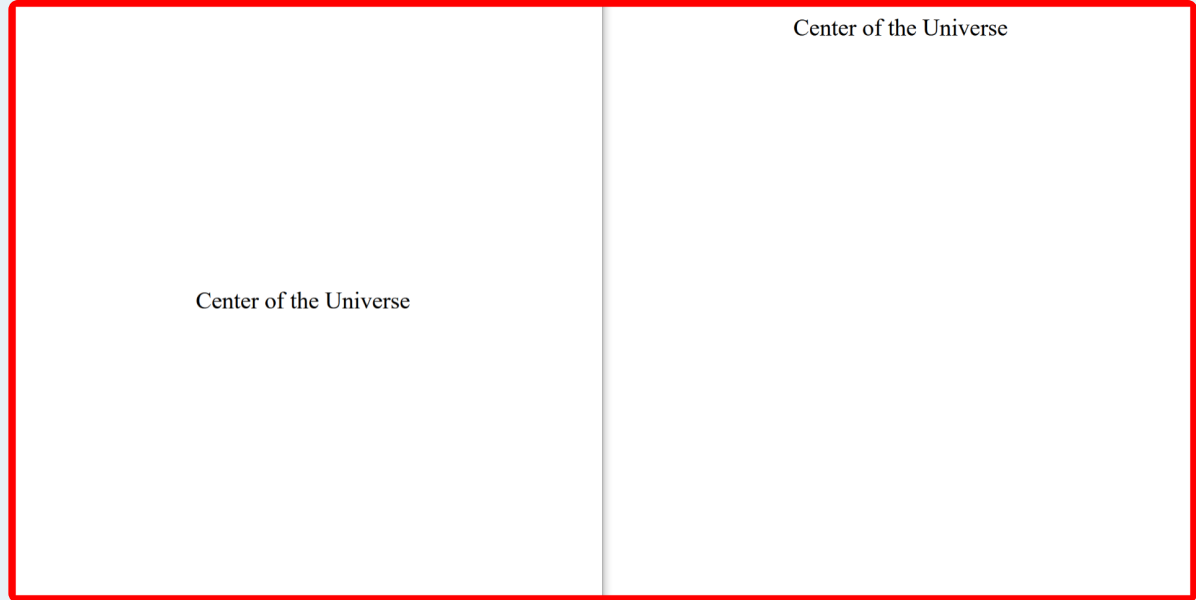
```

Ez a válasz helyes, de nem jelölted meg.

Magyarázat

Az 1. nem jó, mert a jobb oldalon nincs magasság adva az elemnek, ezért a magasság a tartalom magassága lesz, így a képernyő tetején jelenik meg a szöveg. Valamint ha precíznek akarunk lenni, a *body marginját* is ki kell nullázni.

A különbség:



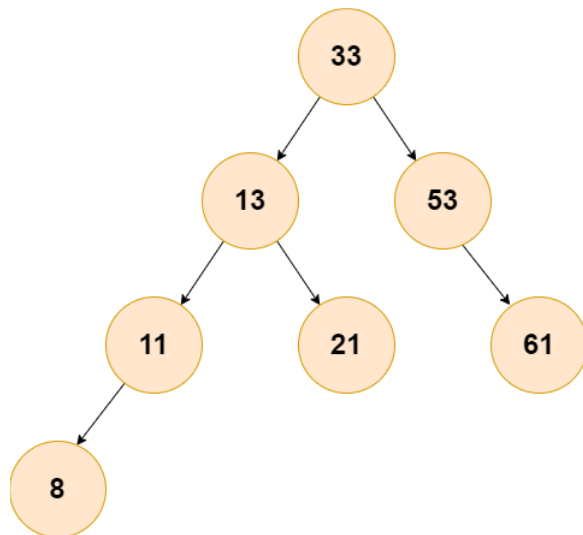
A 2. esetben szintén van különbség, mert egy kép van a div-en belül. Ennek a szélességét ha beállítom 100%-ra ki fogja tölteni a rendelkezésre álló helyet, tehát a *containert*, a magassága pedig igazodni fog ehhez, ezért a kép eredeti méretarányban jelenik meg. Persze más megközelítéssel is magyarázható lenne.

A 3. szintén nem jó, csak majdnem. Ugyanis a media query-s verzióban 1280px-től már érvényesül a 960px-es szélesség, míg a *clamp()* esetében a szélesség 50% 1280px-től és fluidan növekszik 960px-ig. Illetve a media querys töréspontoknál mindig van egy "ugrás".

A 4. megoldás jó, mert bár a 33.33% miatt lehet devtoolon néhány század/ezred pixeles eltérés, de vizuálisan ugyanazt kapjuk a két kód esetében.

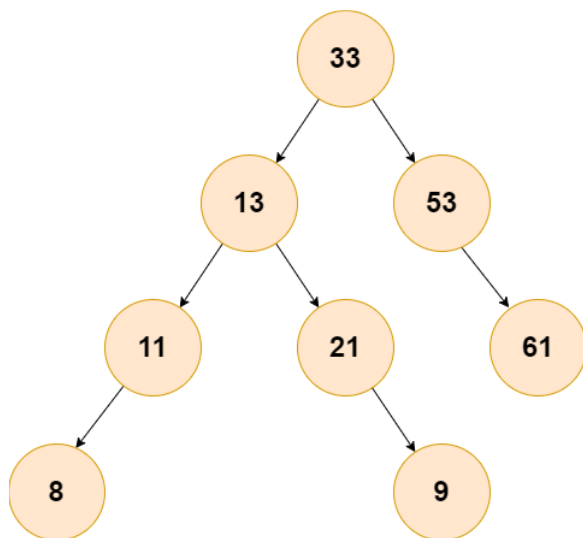
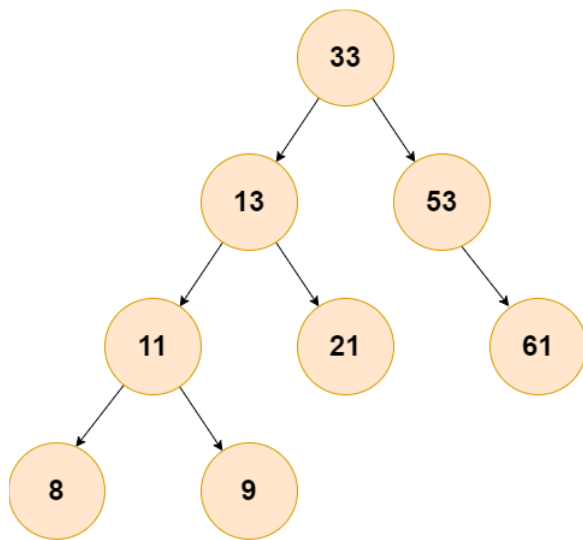
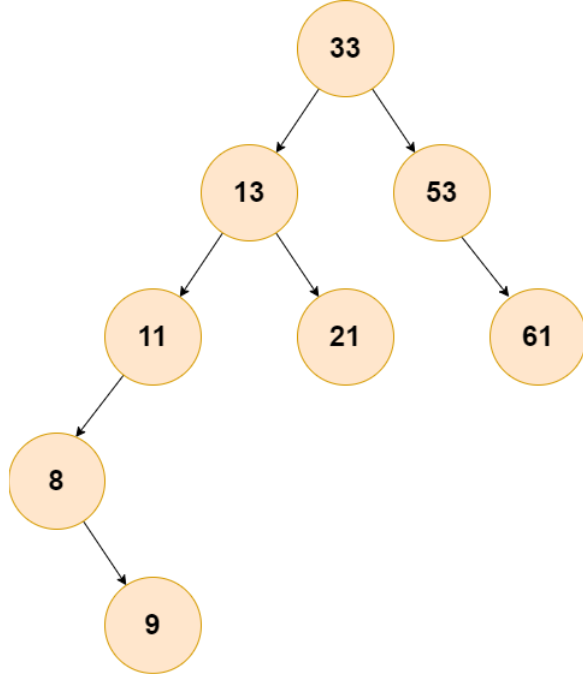
2. feladat 0/3 pont

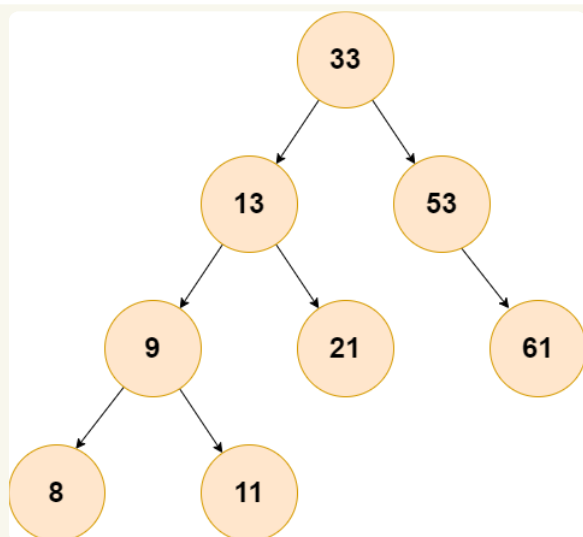
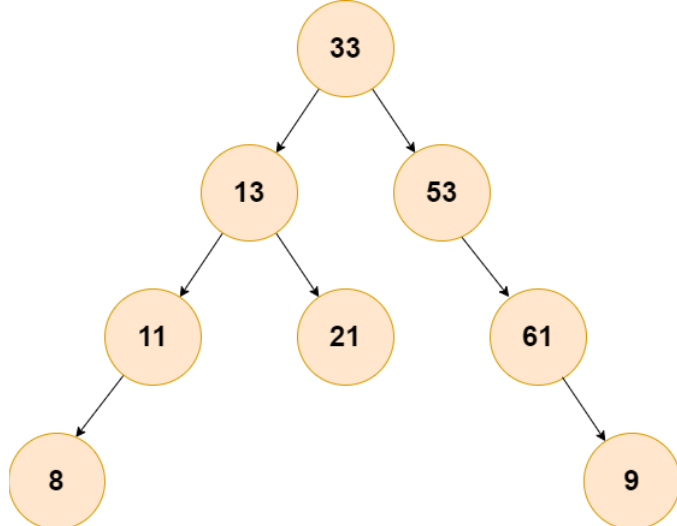
Adott a következő AVL fa:



Válaszd ki az alábbiak közül, hogy hogyan fog kinézni a fa, miután beszúrunk egy új Node-ot, a 9-et (és újra egyensúlyba kerül a fa)!

Válasz





Ez a válasz helyes, de nem jelölted meg.

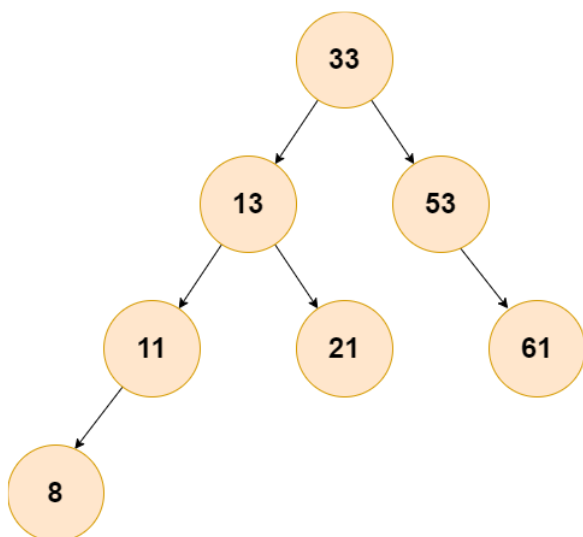
Magyarázat

A feladat az alábbi cikk alapján készült: [AVL Tree in Javascript](#)

A megoldás és a magyarázat valamint egy példakód megtekinthető a fenti linken.

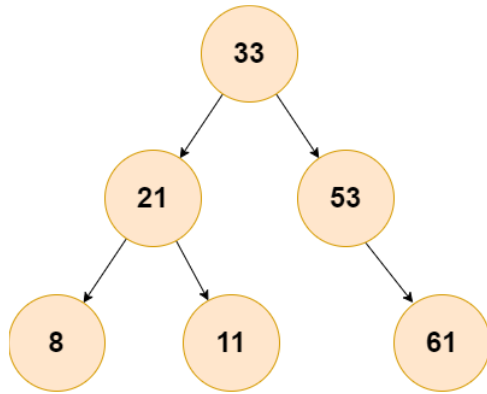
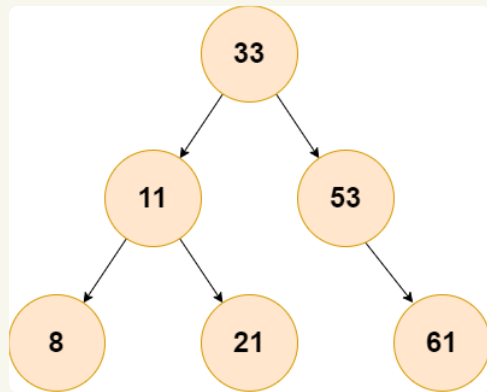
3. feladat 0/3 pont

Adott a következő AVL fa:

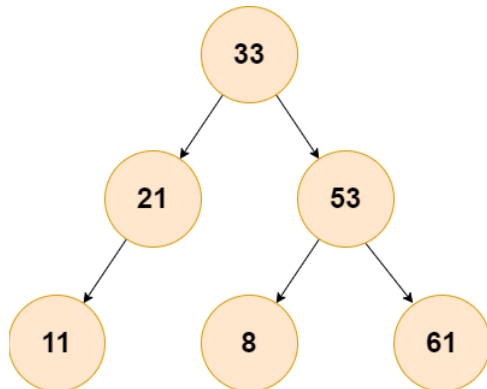
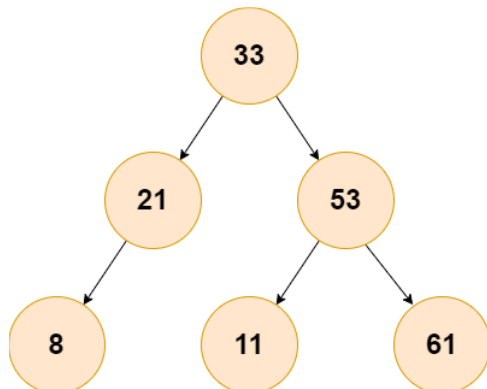
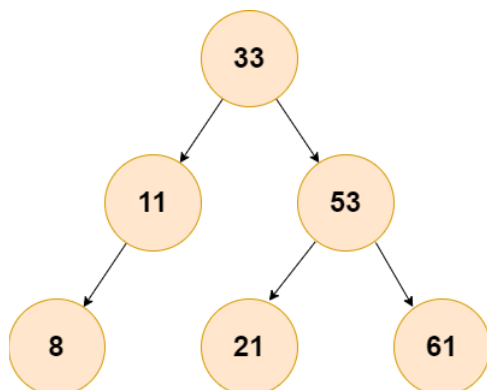


Válaszd ki az alábbiak közül, hogy hogyan fog kinézni a fa, miután töröltük a 13-as Node-ot (és újra egyensúlyba kerül a fa):

Válasz

☐☒

Ez a válasz helyes, de nem jelölted meg.

☐☐☐

Magyarázat

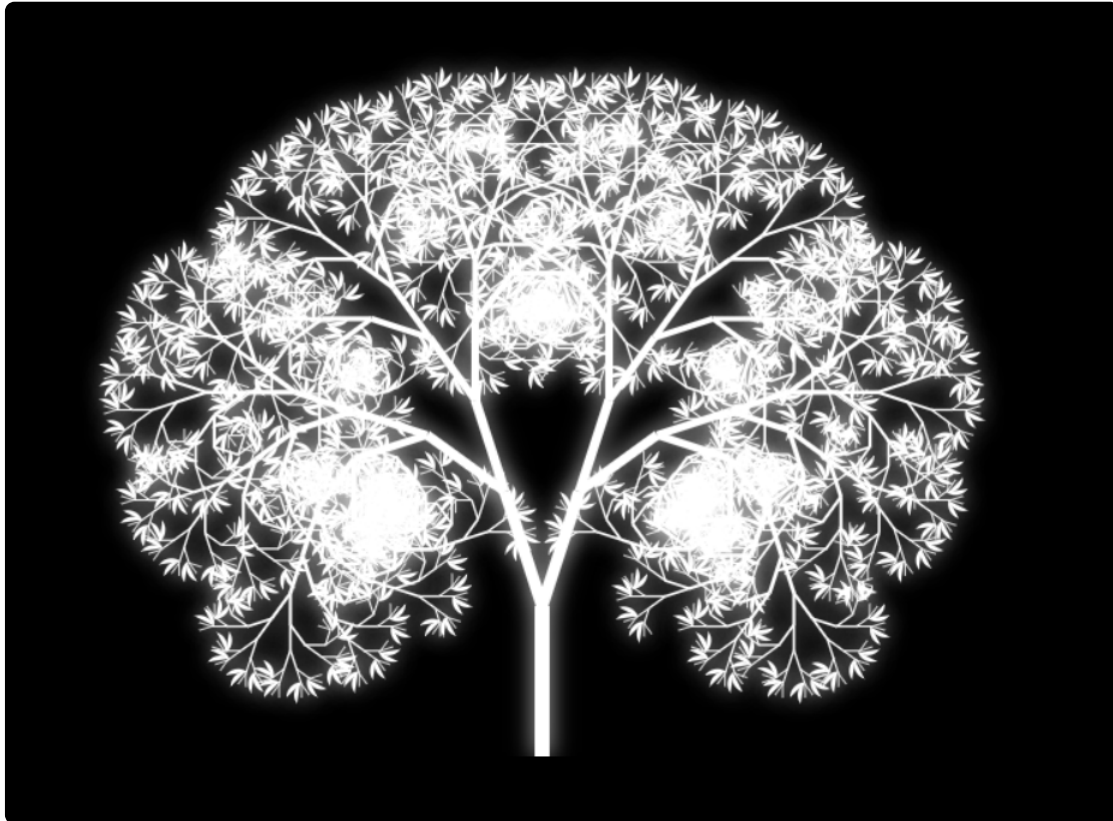
A feladat az alábbi cikk alapján készült: [AVL Tree in Javascript](#)

A megoldás és a magyarázat valamint egy példakód megtekinthető a fenti linken.

A feladatban eggyel kevesebb Node szerepelt.

4. feladat 0/10 pont

Nem működik!!! Egy fantasy map-hez kell egy speciális fraktál fát készíteni. Egy ilyen:



Kolléga felmondott, neked kell befejezni ezt a munkáját (is). Itt tartott:



Szóval egy levél. Nagyjábóóóól...

Dokumentáció nulla.

Az eddigi kód a következő:

```

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>Fractal</title>
6    <style>
7      body {
8        background: #000;
9      }
10   </style>
11 </head>
12
13 <body>
14   <canvas id="fractal-tree" width="700" height="500"></canvas>
15   <script>
16     const canvas = document.getElementById("fractal-tree");
17     const ctx = canvas.getContext("2d");
18
19     function draw(startX, startY, len, angle, branchWidth) {
20       ctx.lineWidth = branchWidth;
21       ctx.strokeStyle = '#fff';
22       ctx.fillStyle = '#fff';
23       ctx.shadowBlur = 15;
24       ctx.shadowColor = 'rgba(255,255,255,0.8)';
25
26       ctx.beginPath();
27       ctx.save();
28       ctx.translate(startX, startY);
29       ctx.rotate(angle * Math.PI / 200);
30       ctx.moveTo(0, 0);
31       ctx.lineTo(0, -len);
32
33       if (len > 10) {
34         ctx.beginPath();
35         ctx.arc(0, -len, 10, 0, Math.PI / 2);
36         ctx.fill();
37         return;
38       }
39
40       draw(startX, -len, len * 0.8, angle - 20, branchWidth * 0.8);
41       draw(startY, -len, len * 0.8, angle + 20, branchWidth * 0.8);
42     }
43
44     draw(350, 500, 100, 0, 10)
45   </script>
46 </body>

```

Milyen hibákat találsz a kódban?

Válaszok

- ☒ A if-nél pont a fordított relációs jel kell tehát < helyett >
Ez a válasz helyes, de nem jelölted meg.
- ☐ Csak egyszer kell meghívni a draw metódust, nem kétszer
- ☒ A draw metódus hívásoknál a startX és startY helyett az első paraméter 0 kell legyen
Ez a válasz helyes, de nem jelölted meg.

- ☒ Hiányzik a `ctx.restore()` az if-en belül
Ez a válasz helyes, de nem jelölted meg.
- ☒ Hiányzik a `ctx.restore()` a függvény végéről
Ez a válasz helyes, de nem jelölted meg.
- ☐ A `ctx.lineTo(0, -len)` helyett `ctx.lineTo(0, len)` kell, nem kell a mínusz jel
- ☒ A `ctx.lineTo(0, -len)` után nincs meghívva a `ctx.stroke()`
Ez a válasz helyes, de nem jelölted meg.
- ☐ A `save` metódus túl korán van meghívva

Magyarázat

A feladat az alábbi cikk alapján készült, ahol megtaláljátok a részletes leírást: <https://lutarolobo.xyz/blog/use-javascript-and-html5-to-code-a-fractal-tree/>



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

© 2023 Human Priority Kft.

KÉSZÍTETTE **cone**

Megjelenés

Világos