



Emoji Generator

NJIT 2024 – Iskola bemutató




JavaScript

CSS animáció

Hivatkozás a JavaScript fájlra

1. Nyisd meg az **index.html** fájlt!
2. Keresd meg **<!-- hivatkozás JavaScript fájlra -->** megjegyzést!
3. Alá gépeld be a képen látható kódot!
 - A **script** elemmel lehet JavaScript fájlokra hivatkozni.
 - Az **src** attribútumba állíthatod be a fájl elérési útvonalát.



```
<!-- hivatkozás JavaScript fájlra -->  
<script src="src/main.js"></script>
```



main.js fájl

1. Nyisd meg az **src** mappában a **main.js** fájlt!
2. Találsz egy kiinduló kódot:
 - A **btn** nevű változóba tároltuk el a gombhoz tartozó HTML node-t.
 - A gombhoz hozzá rendeltünk egy kattintás (**click**) eseményt.
 - Illetve hagytunk helyet az **rnd** függvény megírásához is.

```
const btn = document.querySelector('button');  
  
//emoji generálása a Generate gomb kattintás eseményére  
btn.addEventListener('click', () => {  
  
});  
  
// rnd függvény
```



main.js fájl

1. Nyisd meg az **src** mappában a **main.js** fájlt!
2. Találsz egy kiinduló kódot:
 - A **btn** nevű változóba tároltuk el a gombhoz tartozó HTML node-t.
 - A gombhoz hozzá rendeltünk egy kattintás (**click**) eseményt.
 - Illetve hagytunk helyet az **rnd** függvény megírásához is.

```
const btn = document.querySelector('button');  
  
//emoji generálása a Generate gomb kattintás eseményére  
btn.addEventListener('click', () => {  
  
});  
  
// rnd függvény
```

Az rnd függvény elkészítése

Egy véletlenszámot legeneráló függvény szeretnénk ha készítenél először.


1. Az *// rnd függvény megjegyzés* alá készítsd el a függvényt!
2. A **function** kulcsszóval tudsz függvényt definiálni.
3. Utána a függvény nevét kell megadni, ami most az **rnd** lesz.
4. A **()** zárójelek közé a függvény bemenő paraméterét adhatod meg, ami egy egész szám (**num**).
Pl. ha a **num** az **5**, akkor **1 és 5 közötti véletlenszámot** fogunk generálni.

```
function rnd(num){  
  
}
```

Az rnd függvény elkészítése

1. A függvény belsejében a **return** kulcsszóval adhatsz a függvénynek visszatérési értéket.
2. A képletet írd be a mintáról:
 - A **Math.random()** matematikai függvény egy 0 és 1 közötti valós számot generál.
 - A **Math.floor()** matematikai függvény, mindig lefele kerekíti a valós számot és egy egész számot ad vissza.

```
function rnd(num){  
    return Math.floor(Math.random()*num)+1  
}
```



A gomb kattintás eseményére kell lekódnod a következőket a { } zárójelek közé:

1. A weboldalon el kell érned a **hat** azonosítójú kép elemet (**img**).

```
const hat = document.querySelector('#hat');
```

- A **hat** változóba mentjük el a képelemhez tartozó node-t.
- A **document.querySelector()** függvénnyel kérheted le a weboldalra betöltött HTML elemeket CSS szelektor alapján.

```

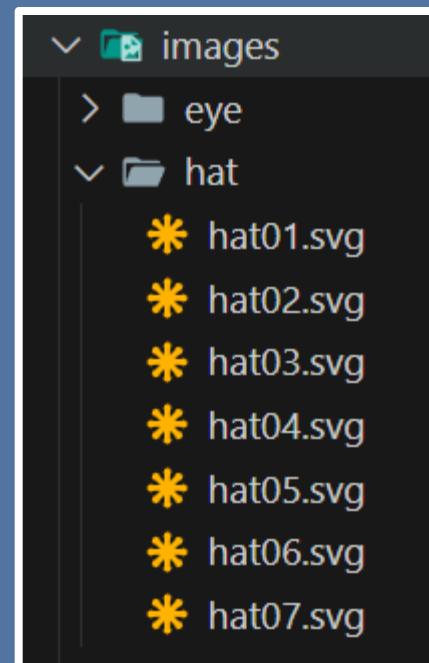
```


A kalap legenerálása 2.

2. lépésként nézd meg az **images/hat** mappában hány SVG képfájl van.

- Utána hozz létre egy **rndHat** nevű konstans változót és az **rnd()** függvény használatával generálj le egy 1 és 7 közötti egész számot.

```
const rndHat = rnd(7);
```



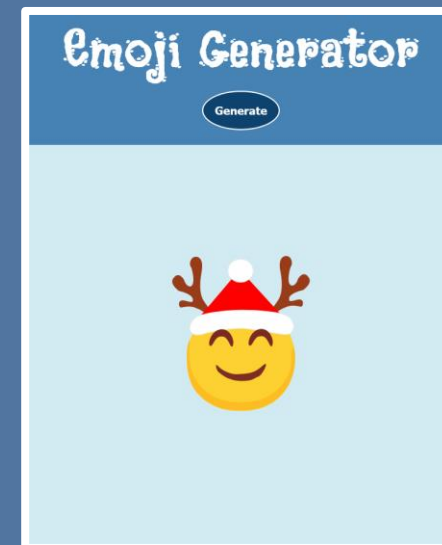
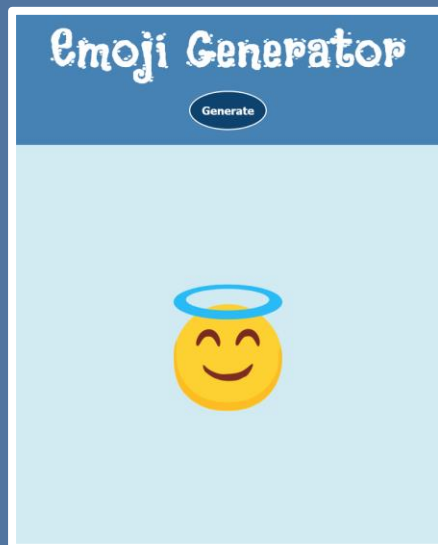
3. lépésként be kell állítanod a hat azonosítójú kép elérési útvonalát a véletlenszám segítségével.

- Az = jel bal oldalán van az hogy mit szeretnénk beállítani, azaz a **hat** kép **src** tulajdonságát.
- A `` jelek közé úgy írhat szöveget, hogy azon belül beilleszthetsz **\${}** jelek közé változókat és azok értékei szövegesen értékelődik ki.

```
hat.src = `images/hat/hat0${rndHat}.svg`;
```

A kalap tesztelése

- Böngészőbe nyisd meg az [index.html](#) fájlt!
- Nyomd meg a **Generate** gombot!
- Ha változik a kalap akkor sikeresen legeneráltad a kalapot



Az előző lépésekhez hasonlóan próbáld legenerálni az emoji többi testrészét a következő tábla alapján:



Testrész	Kép azonosítója	A következő változóval dolgozz kérlek
Fej	head	head, rndHead
Szem	eye	eye, rndEye
Száj	mouth	mouth, rndMouth





CSS animáció

1. Nyisd meg az **src/assets/css** mappában a **style.css** fájlt!
2. Keresd meg a fájlban a **/*CSS animáció*/** megjegyzést.
3. A **@keyframes** után beírtuk az animáció nevét (**flying**). A későbbiekben a névvel hivatkozatsz az itt definiált animációra.

```
/*CSS animáció*/  
@keyframes flying {  
  
}
```

1. kulcskocka létrehozása

A kulcskockák esetén %-os értékkel megadhatod, hogy a kulcskocka az animációban hol lenne. A kulcskockák közti átmenet kiszámolását már a böngészőre bízunk.

1. A 0% az animáció elejét jelenti.
2. A blokkon belül mi a mikulás szán pozícióját szeretnénk megadni, hiszen balról jobbra szeretnénk ha átrepülne a böngészőben.
3. A left CSS tulajdonsággal megadhatod a képed pozícióját vízszintes irányban.
4. A top CSS tulajdonsággal megadhatod a képed pozícióját függőleges irányban.
5. Gépeld be a mintán látható kulcskocka adatait és jöhet a következő kulcskocka.

```
/*CSS animáció*/  
@keyframes flying {  
    0%{  
        left: -300px;  
        top: 300px;  
    }  
}
```

2. kulcskocka létrehozása

1. A második kulcskocka az animáció 25%-nál legyen.
2. A szán vízszintes pozícióját módosítsd 400px-re!
3. A szán függőleges pozícióját módosítsd 150px-re!
4. Csökkentsd a szán méretét 60%-kal!
 - A **transform** CSS tulajdonsággal tudod a képet átméretezni, vagy forgatni, stb.
 - A **scale** függvénnyel tudod átméretezni a képet. Az 1-es érték jelenti a 100%-os méretét. Tehát ha szeretnénk hogy 40%-os méretűre csökkentsük akkor 0.4-et kell beírni a függvény paramétereként.

```
/*CSS animáció*/
@keyframes flying {
  0%{
    left: -300px;
    top: 300px;
  }
  25%{
    left: 400px;
    top:150px;
    transform: scale(0.4);
  }
}
```

3-5. kulcskockák

A következő táblázat alapján próbáld elkészíteni a maradék kulcskockákat:

Kulcskocka	Hol legyen az animációban	left	top	transform: scale()
3.	50%	800px	230px	1
4.	75%	1200px	200px	1.5
5.	100%	1600px	Nem módosítjuk	1



Ha elkészültél az 5 darab kulcskockával, akkor már csak alkalmaznod kell a flying animációt a képre.

1. Keresd meg a képen látható CSS stílust (kulcskocka blokk felett található).
2. A megjegyzés alá írd be a 2. képen látható kódot:

```
animation: flying 10s linear infinite forwards;
```

- Az **animation** CSS tulajdonsággal tudod meghívni az elkészített animációt.
- A **flying-al** adod meg hogy melyik animációt hajtsa végre.
- A **10s** beállítással adhatod meg, hogy az egész animáció hány másodperc alatt hajtódjon végre.
- Az **infinite** beállítással végtelenített animációt készítünk.
- A **forwards** beállítással az animáció végén a kiinduló állapotba visszaugrik.

```
body>img{  
  position: absolute;  
  top: 200px;  
  left: -200px;  
  width: 300px;  
  /*Animáció meghívása*/  
}
```



Gratulálunk! 🥰

Sikeresen elkészítetted az emoji generátort!

