# Dobok

## Bevezetés

Ebben a leckében egy olyan alkalmazást készítünk, amellyel dobolni lehet a számítógépen:



A gombokra kattintva vagy a megfelelő billentyűt lenyomva különböző dobok hangjait játsszuk le.

## Előkészületek

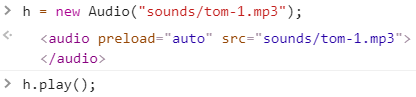
1. Nyisd meg a *dobok* mappát, és az abban lévő fájlokat a Visual Studio Code-ban!
2. Nyisd meg az oldalt a böngészőben is!
3. Nézd át a HTML kódot! Figyeld meg, hogy a gombokhoz milyen azonosítókat és osztályt rendeltünk!
4. Írj be a lábléc végére egy smiley-t! (Win+.)
5. Nyisd meg a CSS fájlt, és nézd végig a beállításokat!

## Hangok lejátszása

Más nyelvekhez hasonlóan a JavaScriptben is használunk objektumokat. Ezek tulajdonságokból és metódusokból állnak. A tulajdonságok az objektum saját változói, amelyek különböző értékeket tárolnak. A metódusok az objektum saját függvényei, amelyeket futtathatunk.

Mi is készíthetünk objektumokat, és vannak előre elkészített objektumok is. Utóbbiak közül most Audio típusú objektumot fogunk használni a hangfájlok lejátszásához.

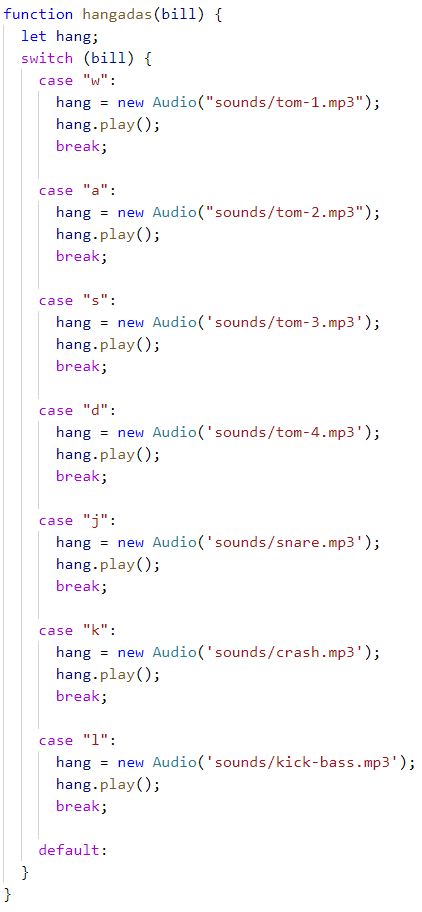
Próbáld ki a következőket a konzolon:



Az első parancsban létrehozunk egy új, Audio típusú objektumot, amely a sounds mappában lévő tom-1.mp3 fájlt tudja lejátszani.

Ezután meghívjuk az objektum play metódusát.

Készíts az index.js fájlban egy olyan metódust, amely egy paraméterként kapott betűhöz hozzárendeli és lejátssza a megfelelő hangfájlt:



Frissítsd az oldalt a böngészőben, majd konzolról próbaképpen hívd meg a függvényt:



Próbáld ki a többi betűvel is! Mi történik, ha olyan betűt adsz meg, amelyhez nincs hozzárendelve egy hangfájl?

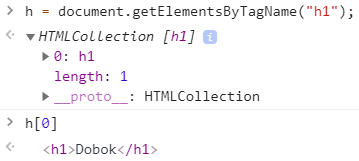
## Oldal elemeinek kiválasztása

Egy oldal egy elemének kiválasztásához eddig a document objektum (vagyis az oldal) getElementById() metódusát használtuk. Vannak azonban más lehetőségek is.

Próbáld ki az alábbiakat a konzolon!

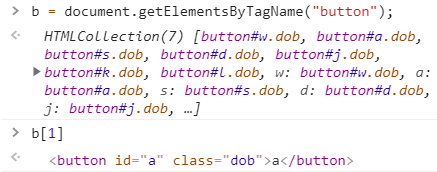
A getElementsByTagName() metódus összegyűjti egy tömbbe az összes olyan elemet, amelynek tag-neve a megadott. (Figyeld meg, hogy az element**s** többes számban van! Ez jelzi, hogy egy tömböt kapunk eredményül.)

Például:



Ez most egy 1 elemű tömb, mert csak 1 darab h1 tag van az oldalon.

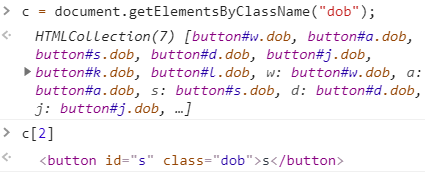
Próbáld ki a button tag-ekkel is:



A tömb hossza megadja, hogy hányszor fordul elő az elem az oldalon:



Ha azt szeretnénk, hogy egy tömbbe kerüljön az oldal összes olyan eleme, amelyhez hozzárendeltünk egy osztályt, akkor a getElementsByClassName() metódust használjuk:



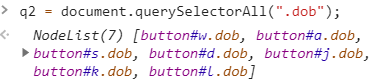
Mivel minden gombhoz hozzárendeltük a dob osztályt, ugyanazt a tömböt kaptuk, mint az előző paranccsal.

A document.querySelector() metódus kiválasztja az oldalon az első olyan elemet, amelyet a paraméterként kapott kijelölő (selector) kijelöl. Például:



Paraméterként tetszőleges CSS selectort megadhatunk.

Ha több elemet szeretnénk kiválasztani, akkor a document.querySelectorAll(); metódust használjuk:



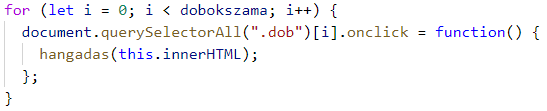
Ennek felhasználásával határozd meg a dobok számát az index.js fájl elején:



Frissítsd az oldalt, majd írasd ki a konzolon a dobokszama változó értékét:



Ezután rendeld hozzá minden egyes dobhoz a hangadas függvényt a megfelelő paraméterrel:



Nézzük meg ezt alaposabban! A for i ciklusváltozója 0-tól a dobokszáma‑1 ‑ig megy 1-esével.

Minden i értéknél kiválasztja az oldalról az i-dik dobot, és a dobra történő kattintáshoz hozzárendel egy névtelen függvényt.

A névtelen függvényben meghívja a hangadas függvényt. A this azt az objektunot (most button-t) jelöli, amelyre rákattintottunk. Ennek tartalma most az a karakter, amellyel meg kell hívni a hangadas függvényt.

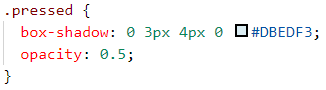
Próbáld ki a böngészőben, hogy a gombokra kattintva már szólnak a dobok!

## Animáció

Azt szeretnénk, hogy a gombok lenyomáskor két tized másodpercig átlátszóak és árnyékosak legyenek.

Ehhez először készítünk egy CSS osztályt, amely a módosítandó beállításokat tartalmazza, majd egy JavaScript függvény segítségével két tized másodpercre hozzárendeljük a lenyomott gombhoz.

Készítsd el az alábbi CSS osztályt:



A JavaScript fájlban készíts egy új függvényt az animációhoz:



A függvény paraméterként megkapja, melyik billentyűt nyomták meg (pl. „w”). Ha olyan billentyűt kap, amelyhez nem tartozik dob, akkor kilép.

Ennek vizsgálatához az includes metódust alkalmaztuk, amely azt vizsgálja, hogy a pont előtti stringben szerepel-e a paraméterként megadott string. (A „wasdjkl”-ben szerepel-e a bill tartalma.)

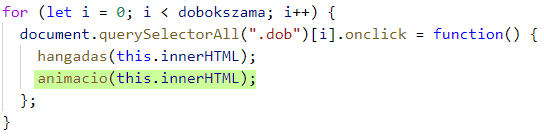
A függvényben ezután kiválasztjuk azt a gombot, amelyhez a megnyomott billentyű tartozik. Kihasználjuk, hogy a billentyű neve azonosítóként hozzá van rendelve a gombhoz.

Ha egy elem több CSS tulajdonságát szeretnénk egyszerre módosítani, akkor célszerű ezeket egy osztályba teni, és az osztályt hozzárendelni az elemhez, vagy eltávolítani róla.

Minden HTML elemnek van egy classList tulajdonsága, amely a hozzá tartozó CSS osztályok listáját tartalmazza. Az osztályokhoz az add metódussal tudunk újabb osztályt hozzáadni, a remove metódussal pedig egy osztályt eltávolítani.

Függvényünkben először hozzárendeljük a pressed nevű osztályt a kiválasztott gombhoz. Ezután megadunk egy időzítést, amely 200 ms múlva eltávolítja a pressed osztályt.

Az animacio függvényt a gombok megnyomásakor kell meghívni. Ezért a gombok eseménykezelőjét kibővítjük az animacio függvény meghívásával:



Próbáld ki!

## Billentyűzet figyelése

Szeretnénk a megfelelő billentyűk lenyomásával is működtetni az alkalmazást.

Az egérkattintásokat az oldal minden eleme figyeli, egy billentyű lenyomásakor azonban a dokumentum kapja meg az eseményt.

A billentyűzet figyelését így kell elkészíteni:

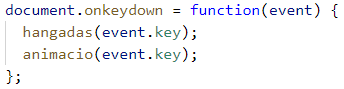
document.onkeydown = function(event) {

  // feldolgozás…

};

Az event paraméter egy objektum, amelynek key mezője megadja a lenyomott billentyű kódját.

Ez alapján írd be a következő függvényt a JavaScript kódba:



Teszteld az alkalmazást! Nyisd meg konzolt, és ellenőrizd, hogy nincs-e hibaüzenet! Találtál még hibát?