

# DOKUMENTÁCIÓ

Web technológiák I.

Beadandó

Készítette: Demján Csongor

Neptunkód: SZ7MGG

## Tartalomjegyzék

HTML előtag, head .....	3
Example.....	4
Style.css.....	4
Script.js.....	5
Rick And Morty.....	6
movingEyes.js .....	7
Ez a JavaScript kód a weboldal felhasználói interakcióira válaszol, az egér mozgását követi nyomon, és átméretezi a megjelenített elemeket az egér helyzetének függvényében. ....	7
Az első sorban a .eye CSS osztályú összes elemet kiválasztjuk, amelyek a szemeket jelentik. Ezután kiválasztjuk a #anchor id-jú elemet, és eltároljuk annak szélességét, magasságát és pozícióját a getClientRect() metódus segítségével. Az anchorX és anchorY változóknak eltároljuk az #anchor elem közepének pozícióját.....	7
Ezután hozzáadunk egy eseményfigyelőt az mousemove eseményhez, amely figyeli az egérmozgást. Az eseményfigyelő függvényében kiszámítjuk az egér pozícióját a clientX és clientY koordinátákból, majd a angle() függvény segítségével kiszámítjuk az egér pozíciójának és az #anchor elem középpontjának közötti szöget. ....	7
Az angleDeg változóban eltároljuk az egér pozíciója és az #anchor középpontja közötti szöget fokban kifejezve. Ezután végigmegyünk az összes .eye elemen, és beállítjuk a transform tulajdonságot, hogy forgassuk a szemet a számított szöggel. Emellett beállítjuk az anchor szűrőjét a hue-rotate() funkcióval, amely az egér pozíciója és az #anchor középpontja közötti szöggel változtatja az elem színét.....	7
Végül a angle() függvény kiszámítja a két pont közötti szöget a cx és cy koordináták (az egér pozíciója) és az ex és ey koordináták (az #anchor középpontja) alapján, és fokban kifejezve adja vissza. Ez a függvény a számított szöggel tér vissza a fenti eseményfigyelő függvényében használt angleDeg változó értékének kiszámításához. ....	7
Techno .....	7
Techno.css .....	7
Change the background.....	8
changeBg.js .....	8

## HTML előtag, head

Minden .html file a dokumentum típusának deklarációjával kezdődik `<!DOCTYPE html>`, ami azt jelzi, hogy az adott dokumentum HTML5-ös.

A HTML nyelvű dokumentum megnyitása a `<html>` tag segítségével zajlik, amelynek attribútumában angol nyelv van megadva.

A HTML fejléc a következő elemeket tartalmazza:

„meta” tag az UTF-8 karakterkódolás beállításához.

„title” tag az oldal címének megadásához.

Stílusinformációk betöltése,

- a „link” taggal valósul meg,
- a `rickAndMorty.html` és a `changeBg.html` fájlokban a head-ben deklarált a style.

body: A dokumentum háttérszínét fehérre, szövegszínét fehérre állítja, és a paddingot és margót is eltávolítja.

#header: A fejléctet fekete háttérszínnel, 10px-es paddinggal és flex-ben jeleníti meg.

nav ul: A navigációs menü listaelemek stílusát befolyásolja, amelyek listázás nélkül jelennek meg, 30px-es margóval és 0 paddinggal, valamint flex megjelenítési tulajdonsággal rendelkeznek.

nav li: A navigációs menü elemek margóját 20px-re állítja.

nav a: A navigációs menü hivatkozások színét fehérre állítja, valamint az aláhúzás stílusát eltávolítja.

nav a:hover: A navigációs menü hivatkozásokat aláhúzásra állítja, amikor a kurzor fölé kerül.

h1: A címsor stílusát befolyásolja, 3 rem-es betűmérettel, 10px-es paddinggal, középre igazítással és nagybetűs szöveggel.

A `rickAndMorty.html` dokumentumban megtalálható továbbá:

main: A fő tartalom középre igazított és a grid megjelenítési tulajdonsággal rendelkezik, ami lehetővé teszi a tartalom elemeinek rugalmas elhelyezését, középre igazítja mind függőlegesen, mind vízszintesen, a minimális magasságot 100%-ra és relatívvá állítja.

#eyes, #eyes img: Az „eyes” és az ahhoz kapcsolódó képek abszolút pozicionálással rendelkeznek.

Script tag, amelynek végrehajtása az oldal betöltődése után történik. A „defer” attribútum azt jelenti, hogy a script futtatása a dokumentum feldolgozásának végeztével történik.

## Example

A „header” azonosítójú div elem a fejléc tartalmának csoportosítására szolgál. A h1 elem fő fejlécként az „Welcome” szöveget jeleníti meg. A nav elem tartalmazza az ul elem által képviselt hivatkozások listáját, és a lista minden eleme az li elemmel van megjelölve. A linkek négy különböző HTML-fájlra mutatnak („index.html”, „rickAndMorty.html”, „techno.html” és „changeBg.html”), amelyek a webhely különböző oldalait képviselik. Az img elem egy példa képének megjelenítésére szolgál. Az src attribútum a képfájl elérési útját adja meg, az alt attribútum pedig alternatív szöveget biztosít a képhez arra az esetre, ha az nem jeleníthető meg. Összességében ez a kódblokk a webhely fejlécrészének szerkezetét képviseli, amely tartalmaz egy fő címsort, egy navigációs menüt a webhely különböző oldalaira mutató hivatkozásokkal és egy példaképet.

A tartalmi blokk egy "section1" azonosítóval ellátott szakaszból áll, amely címkével és szöveggel, gombbal, táblázattal, hivatkozással, űrlappal, számozott és pontozott listákkal rendelkezik. A szakasz tartalmaz egy HTML formot is, amelynek az "id" attribútuma nincs beállítva, és több különböző típusú űrlapmezőt tartalmaz, beleértve a "text", "email", "textarea", "checkbox", "radio", "color" és "date" típusokat. Az űrlapnak van egy "Submit" és egy "Reset" gombja is.

## Style.css

A #header elemnek zöld átlátszó hátteret, rózsaszín szöveget és 10 pixelnyi belső margót ad, flex elrendezést használva.

A body-nak kék hátteret ad.

A nav ul-nek nincs stílusoszlopa.

A nav li-nek 20 pixelnyi jobb margót ad.

A nav a linknek narancssárga színt ad és eltávolítja az aláhúzást.

A nav a: hover esetén az aláhúzás jelenik meg.

A header h1 elemnek 3 rem-es betűméretet, középre igazítást, nagybetűssé alakítást és 10 pixelnyi belső margót ad.

Az `img`-nek blokk elrendezést, maximális szélességet, automatikus magasságot, középre igazítást és olajzöld szegélyt ad.

Az `#text-input` elemnek narancssárga hátteret ad.

A `.form-element` osztálynak fekete szegélyt ad.

A táblázatnak 100%-os szélességet, 1 em-es betűméretet, 20 pixelnyi margót és összeomló határvonalat ad.

A `th` és `td` elemeknek olajzöld szegélyt, 12 pixelnyi belső margót és balra igazítást ad.

Az páros számú soroknak szürke hátteret ad, és az egérkurzor fölé helyezéskor a háttérszín kék árnyalatúvá válik.

A menünek sötét narancssárga hátteret, rejtett túlsordulást ad és a menüpontoknak fekete betűt, középre igazítást, 14 és 16 pixelnyi belső margót, nincs aláhúzást ad.

A menüpontokra mutató egérkurzor piros színnel jelzi a kiválasztott menüpontot.

Az `error` osztályú elemnek piros szegélyt ad.

A `#section1` elemnek fekete hátteret, fehér szöveget, 1,2 em-es betűméretet, 2 em-es belső margót, nagybetűsséget ad.

Az `h2` elemnek 2,5 em-es betűméretet, nagybetűsséget és 1 em-es alsó margót ad.

A `p` elemnek 1,2 em-es betűméretet, 1,5-szeres sortávolságot és 1,5 em-es alsó margót ad.

A táblázat fejlécének fehér hátteret, fekete szöveget

## Script.js

Ez a JavaScript kód egy űrlap validációt végző függvényt tartalmaz. Az űrlapot egy `form` elemmel lehet elérni, amelyhez eseménykezelőt adunk hozzá a `submit` eseményre. Az eseménykezelő funkciója meghiúsítja az alapértelmezett eseményt (azaz az űrlap elküldését), majd ellenőrzi a kitöltött mezők érvényességét.

A kódban az `getElementById()` függvény segítségével elérjük az űrlapban szereplő nevet (`nameInput`), e-mail címet (`emailInput`) és üzenetet (`messageInput`). Ha az egyik mező nincs kitöltve, akkor hibát jelezünk az adott mező mellett az `error` osztály hozzáadásával, majd egy riasztással

figyelmeztetjük a felhasználót a megfelelő kitöltésre. Ha a mező érvényes, eltávolítjuk az error osztályt.

Az `isValidEmail()` függvény egy egyszerű ellenőrzést végez az e-mail cím érvényességére egy reguláris kifejezés segítségével. Ha az e-mail cím helytelen formátumú, akkor hibát jelezünk és megjelenítünk egy riasztást.

Összességében a kód célja az űrlap kitöltési hibáinak felderítése és jelzése a felhasználónak, hogy adatait helyesen tudja beküldeni.

## Rick And Morty

A fő tartalom részben egy kép van, amelynek az "anchor" azonosítója van megadva. Ezután egy "eyes" azonosítójú tartalom rész található, amelyben négy kép található, mindegyiknek az "eye" osztályneve van megadva, és mindegyik kép stílus tulajdonságai változóak, hogy azok a megfelelő helyen legyenek elhelyezve.

## movingEyes.js

Ez a JavaScript kód a weboldal felhasználói interakcióira válaszol, az egér mozgását követi nyomon, és átméretezi a megjelenített elemeket az egér helyzetének függvényében.

Az első sorban a `.eye` CSS osztályú összes elemet kiválasztjuk, amelyek a szemeket jelentik. Ezután kiválasztjuk a `#anchor` id-jú elemet, és eltároljuk annak szélességét, magasságát és pozícióját a `getBoundingClientRect()` metódus segítségével. Az `anchorX` és `anchorY` változóknál eltároljuk az `#anchor` elem közepének pozícióját.

Ezután hozzáadunk egy eseményfigyelőt az `mousemove` eseményhez, amely figyel az egérmozgást. Az eseményfigyelő függvényében kiszámítjuk az egér pozícióját a `clientX` és `clientY` koordinátákból, majd a `angle()` függvény segítségével kiszámítjuk az egér pozíciójának és az `#anchor` elem középpontjának közötti szöget.

Az `angleDeg` változóban eltároljuk az egér pozíciója és az `#anchor` középpontja közötti szöget fokban kifejezve. Ezután végigmegyünk az összes `.eye` elemen, és beállítjuk a `transform` tulajdonságot, hogy forgassuk a szemet a számított szöggel. Emellett beállítjuk az `anchor` szűrőjét a `hue-rotate()` funkcióval, amely az egér pozíciója és az `#anchor` középpontja közötti szöggel változtatja az elem színét.

Végül a `angle()` függvény kiszámítja a két pont közötti szöget a `cx` és `cy` koordináták (az egér pozíciója) és az `ex` és `ey` koordináták (az `#anchor` középpontja) alapján, és fokban kifejezve adja vissza. Ez a függvény a számított szöggel tér vissza a fenti eseményfigyelő függvényében használt `angleDeg` változó értékének kiszámításához.

## Techno

Ez a techno különböző típusainak leírása, több alműfajra osztva. Tartalmazza az egyes alműfajok eredetét, jellemzőit és nevezetes művészeit. A tárgyalt alműfajok a Detroit Techno, az Acid Techno, a Minimal Techno, az Industrial Techno, a Dub Techno, a Tech House és a Trance. A cikk szerzői jogi megjegyzéssel zárul.

## Techno.css

A `.post-content` osztályú elemeknek szürke háttérszíne van, 5 pixeles `border-radius` tulajdonsággal, 20 pixeles belső `padding`gal és 20 pixeles alsó margóval. A doboz árnyékát a `box-shadow` tulajdonság határozza meg, amelynek értéke

0px 0px 5px 0px rgba(0,0,0,0.2). Az elemek szövegét középre igazítjuk a text-align tulajdonsággal.

A .post h2 osztályú elemeknek 24 pixeles betűméretet, 10 pixeles alsó margót, középre igazított szöveget és fehér betűszínt állítunk be.

A .post-info osztályú elemek fehér betűszínt kapnak, és középre igazítjuk a szöveget.

Az #header id-jú elemeknek szintén szürke háttérszíne van, 10 pixeles paddinggal és flex elrendezéssel rendelkeznek.

A navigációs menühöz tartozó ul elemeknek nincs lista jelölője, 30 pixeles külső margója van, 0 belső margója van, és flex elrendezéssel rendelkezik.

A navigációs menühöz tartozó li elemeknek 20 pixeles jobb margót adunk.

A navigációs menühöz tartozó az elemek fehér betűszínt kapnak és aláhúzás nélkül jelennek meg.

A navigációs menühöz tartozó a elemekre mutatra aláhúzást kapnak a text-decoration: hover tulajdonság hatására.

Az oldal fejlécének h1 elemének 3 rem méretű betűt, középre igazított szöveget, fehér betűszínt, nagybetűs szövegformázást, és 10 pixeles paddingot állítunk be.

Az oldal háttérszíne fekete (#000). Az oldal láblécének #footer id-ű p elemének fehér betűszínt adunk.

## Change the background

A negyedik oldalon csupán egy h1-es cím található a header-ön kívül.

### changeBg.js

Ez egy eseménykezelőt hoz létre, amely a dokumentumra kattintásra aktiválódik. Az esemény kezelő véletlenszerűen hoz létre egy színt a vörös, zöld és kék komponensekből, majd az új színt beállítja a dokumentum háttérének színévé. Tehát minden alkalommal, amikor valaki kattint a dokumentumban, a háttérszín véletlenszerűen változik.