

# Capybara bemutató weboldal – dokumentáció

Készítette: Stremler László, AQYO8L

## Tartalomjegyzék

1. Bevezető .....	3
2. Főoldal .....	3
3. Capybara bemutató oldal.....	4
3.1. Az oldal dizájnja.....	4
4. Érdekességek/tények a Capybarákról .....	4
5. A kapcsolat oldal .....	5
6. Technikai megvalósítások .....	5
6.1. A mappa struktúra .....	5
6.2. A script fájlok.....	6
6.2.1. A script.js fájl .....	6
6.2.2. A headerColor.js fájl .....	6
6.2.3. A form.js fájl .....	6

## 1. Bevezető

Ez az oldal azért jött létre, hogy részletes információkat nyújtson a világ egyik legkedvesebb és legérdekesebb emlőséről, a Capybaráról. A weblap 4 oldalt tartalmaz:

- Főoldal
- Capybara bemutató oldal
- Érdekességek/tények a Capybaráról
- Kapcsolat

További felépítőeleme a navigációs sáv, amelyen helyet kapott a három oldal. A navigációs sávhoz tartozik egy CSS stíluslap, ami megvalósítja a 'hover' effektust. (sötétebb zöld lesz a szöveg háttere, ha felé visszük az egeret).

Ezenfelül található egy címsor az oldalon, ami az éppen aktuális oldal címét hivatott közölni.

A láblécben szerepel egy copyright karakter, valamint az oldal neve.

## 2. Főoldal

Az oldal főoldalán a felhasználót az állat rövid ismertetése fogadja, így azonnal meg lehet ismerni az állat jellemzőit és tulajdonságait. Az ismertető alatt található egy képgaléria, amelyben gyönyörű felvételek sorakoznak, és a felhasználó könnyedén navigálhat közöttük a megjelenő nyilak segítségével.

A dinamikus képgaléria működését egy Javascript függvény biztosítja. Ez a függvény egy előre definiált tömb segítségével állítja be a képek forrásait, így a felhasználó az oldal böngészése során zökkenőmentesen haladhat át a különböző szögekből fotózott rágcsálók között.

Az egyik kiemelkedő előnye ennek a megoldásnak, hogy minimális terhelést jelent az oldal betöltési idejére és teljesítményére. A képfolyam cseréje során mindig csak egyetlen kép cserélődik ki, így a felhasználók számára zökkenőmentes és gyors élményt nyújt. Amikor a felhasználó rákattint a nyilakra a képgalériában, egy gyors és látványos negyed másodperces átfedéssel animáció következik be. Ez az animáció a képek közötti sima átmenetet biztosítja, ami még élvezetesebbé teszi az oldal böngészését.

A logika az egész mögött annyi, hogy a Javascript függvény időzítő segítségével állítja be a kép áttetszőségét. Először a kép áttetszőségét 0-ra állítja, így láthatatlanná válik, majd az időzítő lejártá után 1-re állítja azt, mielőtt a következő kép megjelenik. A képgaléria rövid, de hatékony animációi segítenek abban, hogy az állatok iránt érdeklődők teljes mértékben élvezhessék az oldal böngészését és felfedezését.

### 3. Capybara bemutató oldal

A második oldalon nagyobb bepillantást lehet nyerni a Capybarák tulajdonságaiba. Wikipédia oldalról kimásolt tényekkel a felhasználó megtudhatja a rágcsáló származási helyét, illetve főbb tulajdonságait. A szöveg végén található jel elnavigálja a felhasználót a forrás oldalra.

#### 3.1. Az oldal dizájnja

Dizájn szempontjából ez az oldal is követi a többi oldal letisztult felületét, illetve színvilágát. Ugyanúgy erősen kiemelt zöld részekkel próbálja elkapni az olvasó figyelmét.

Az fejléc és a lábléc is megegyezik a többi oldaléval, egyedül az oldal címe más.

### 4. Érdekességek/tények a Capybarákról

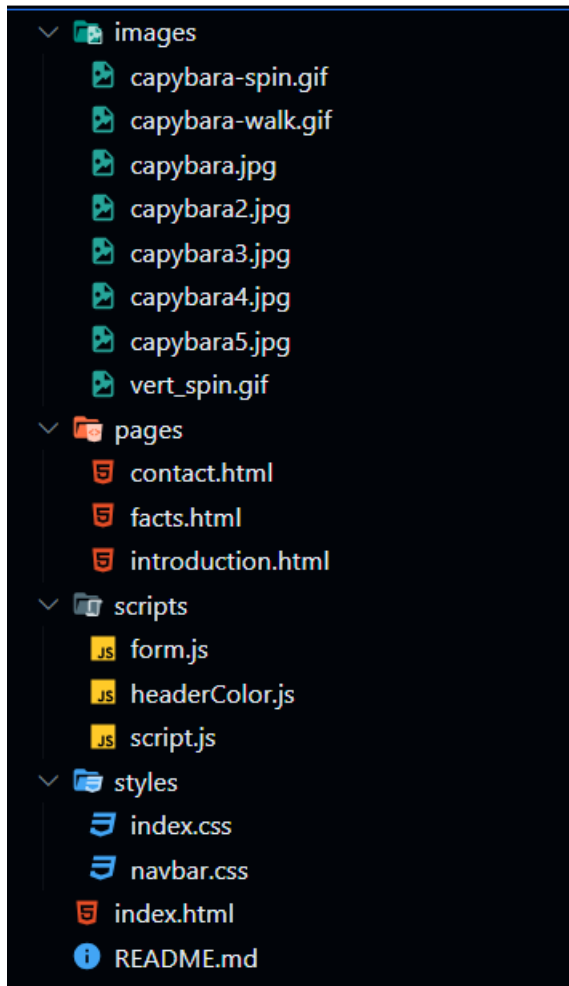
Ezen az oldalon egy lista fogadja a felhasználót, amelyben fel vannak sorolva a Capybara legfőbb érdekességei, illetve tények erről a rágcsálóról. Itt kap lehetőséget a látogató, hogy még közelebbről megismerje ezt a rendkívüli rágcsálót, és felfedezze annak valóban barátságos és szociális oldalát.

A felsorolás egy számozatlan lista elemmel lett megvalósítva, amelyre utána hasonló stílusformázás került, mint a többi szövegre az oldalon. Ez segít abban, hogy az információ könnyen olvasható és vonzó legyen, miközben megragadja a látogató figyelmét.

Az érdekességek felsorolása alatt megtalálható néhány szórakoztató és vicces GIF is, amelyek megörökítik az állat különleges személyiségét és közkedveltségét. Ezek az animációk nemcsak szórakoztatják, hanem még inkább megerősítik azt a benyomást, hogy a Capybara egy valóban egyedülálló és barátságos lény a világ állatvilágában.

## 5. A kapcsolat oldal

Ezen az oldalon egy űrlap található, amellyel lehet üzenni az oldal üzemeltetőjének. Az oldal tartalmazza a szokásos elemeket: fejléc, lábléc, színválasztó, navigációs sáv. Ezenkívül 3 beviteli mező is helyet kapott az



oldalon: Név, email, üzenet. Amennyiben a felhasználó üzenetet szeretne küldeni, akkor ezeket a mezőket ki kell töltenie. Hiányos mező esetén hibát dob az oldal és nem küldi el. Az email cím megadásánál az általános formát követni kell, nem lehet bármit megadni. A felhasználónak lehetősége van feliratkozni a hírlevél szolgáltatásra, amennyiben bejelöli a feliratkozás melletti négyzetet. Az űrlapot elküldeni, illetve ellenőrizni az „Elküldés” gombbal lehet.

## 6. Technikai megvalósítások

Ebben a részben lesz szó a beadandó technikai megvalósításairól, valamint a projekt strukturális felépítéséről is.

### 6.1. A mappa struktúra

A gyökérkönyvtár tartalmazza a főoldalt, amely az index.html nevet viseli. Ebben a mappában található az images mappa, amely a képek tárolására szolgál; a pages mappa ami a további 3 oldalnak ad helyet; a scripts mappa amiben található a galéria mozgatórugója, a script.js, a kapcsolat oldalon található űrlap feldolgozója, a form.js, valamint a színezésért felelős headerColor.js; végül van a styles mappa, amiben a stíluslapok, névszerint az index.css és a navbar.css kapott helyet.

1: Mappaszerkezet

## 6.2. A script fájlok

### 6.2.1. A script.js fájl

A script.js fájl egy függvényt tartalmaz, amely kezeli a megjelenő képeket a galériában. Ezt úgy éri el, hogy képnek megadott osztálytípus alapján a `document.getElementById()` függvény segítségével lekéri a képet az oldalon, majd ennek a képnek változtatja a `source`-ját (forrását). Ez a megoldás teljesítményoptimalizálási szempontból véleményem szerint azért előnyös, mivel egy képnek van cserélve a forrása, helyett hogy annyi kép lenne betöltve, amennyi rendelkezésre áll. A megjelenítendő képek relatív elérési útja egy tömbben van letárolva, majd ennek a tömbnek mindig egyel nagyobb indexére történik hivatkozás, ha a felhasználó képet kíván váltani. Ha elérte a tömb hosszát az index változó, akkor automatikusan újra 0 lesz az értéke.

A két gombot is az előbb említett függvénnyel kéri le a script, majd ha az elemek felhívják az `onclick` eseményüket, akkor arra reagálva a script meghívja a képváltás eljárást.

### 6.2.2. A headerColor.js fájl

A headerColor.js fájl felel a fejléc, navigációs sáv, valamint a főoldalon és a kapcsolat oldalon található szöveg átszínezéséért. Az oldalakon található színválasztó gomb lenyomásakor megnyílik a HTML-ben alapelemként beépített colorpicker input elem. A kívánt szín kiválasztásakor az oldal fejléce, valamint a szöveg megváltozik a választott színre, majd ezt a színt elmenti a böngésző helyi tárába (Local Storage). Az oldal ismételt látogatásokor ezt az elmentett értéket tölti be a script, majd beállítja az elemeket erre a színre. Ha nem találja a helyi tárbán a mentett színt, akkor az eredeti (zöld) szín állítódik be. Ez a funkció bővíthető úgy, hogy a változtatni kívánt elemnek a „changable” osztályt adjuk, mivel ezt az osztályt módosítja a script.

### 6.2.3. A form.js fájl

Ez a fájl a nevéhez híven a 4.oldalon található űrlap ellenőrzéséért felel. HTML5-ben már megadható a `required` tulajdonság az input elemeknél, azonban ezt nem minden böngésző támogatja, ezért lehet hasznos ha egy script végzi az ellenőrzést. A script lekéri a név, email, üzenet, illetve feliratkozás gomb tartalmát és validálás elé állítja. A név, üzenet mezőket csak kitöltöttségi szinten ellenőrzi, míg az email mezőt összehasonlítja a regex-szel, ami az általános email formátumot tartalmazza ([valami@valami.valami](mailto:valami@valami.valami)). Az elküldés gombra kattintva hívódik meg ebben a fájlban található függvény, amely végrehajtja ezeket az ellenőrzéseket. Ha mindent rendben talált, akkor `true`-val tér vissza a függvény, valamint kiírja a böngészőben (az `alert` függvény segítségével) a beírt adatokat. Amennyiben nem megfelelően kitöltött, vagy üresen hagyott mezőt észlel a script, akkor a mezők alatt egy piros hibaüzenetet jelenít meg, amely tartalmazza hogy melyik elem hibás. A hibaüzenet az Elküldés gomb megnyomása után frissül vagy tűnik el. A feliratkozás checkbox értékét lekéri, majd ennek megfelelően egy változónak vagy igen vagy nem értéket ad, így az űrlap elküldésekor ezt írja ki a felhasználónak.