JEGYZŐKÖNYV

Web technológia alapjai

Kutyák és kutyafajták

Készítette: **Csákó Balázs**

Neptunkód: **IOMAG5**

Dátum: **2024. Május 3.**

Miskolc, 2024.

Tartalomjegyzék

Bevezetés

A feladat leírása: 5 HTML fájlt kellet létrehozni, amelyek mind egy adott témáról szóltak. Nekem a választott témák kutyák, illetve kutyafajták voltak. A weboldalaknak emellett tartalmazni kellett adott tag-eket, metódusokat valamint eljárási módokat. A feladat 3 nagy részre volt osztható: HTML, CSS és JQuery, JSON.
A HTML részbe tartoztak a különböző tag-ek lértehozása, mint például a táblázat, űrlap elemek, media elemek, stb.
A CSS rész tartalmazta az adott tagok formázását azonosítójuk, osztályaik segítségével. Színek beállítása, elemek elrendezés valamint menü létrehozása. Hátterek és űrlap elemek, gombok formázása.
A Jquery, JSON kategória több dolgot is lefedett. Az ebbe a kategóriába tarozó feladatok megoldásához egy előre megírt függvény könyvtárat használtunk. Ide tartozott a JSON file-ok használata, akár AJAX metódussal, akár beágyazott formában. Új HTML elemek hozzáadása az append() metódussal, valamint a már meglévők módosítása. A JQuery által kínált animációk, animáló metódusok használata is szükséges volt a weblapok elkészítéséhez. Végül pedig egy saját funkció elkészítése az űrlap/form beviteli értékeinek ellenőrzésére. Emellet szorgalmi feladatként a projekt meg kellett jeleníteni a helyi szerveren NodeJs segjtségével.

I. feladat

HTML szerkezet, mappastruktúra

A project törzsmappája tartalmazza a HTML fájlokat, a feladat leírást, jegyzőkönyvet és több almappát is. Az öt HTML fájl a memory_game.html, form.html,index.html, list.html és a video.html. A weblapok között az oldalak bal oldalán található menüsáv biztosít átjárást. Mindegyik almappa és specifikus fájltípust tartalmaz. Így a Scripts mappában találhatók az oldalak által használt js fájlok, valamint a JQuery telepített változata is. A CSS mappában a közös formázásokat tartalmazó fájl van, valamint a külön memory_game scriptjét tartalmazó fájlt. A Pictures almappában a szükséges media fájlok érhetők el.

HTML elemek

Az oldalak alapjait flexelt div elem adják, amik egyes helyeken p, span van h1/h2 tartalmaznak. Képeket ay index, list és a memory_game is tartalmaz. A header-ökben a h1-ben lévő oldalcímek Vannak, míg a footerben az adott oldalhoz kapcsolódó linkek. Az egyetlen táblázat az index-ben van, ami a magyar kutyafajtákat tartalmazza.

Űrlap elemek

Az űrlap elemek nagyját a form weboldal tartalmazza. Abban megtalálhatók a beviteli mezők, az adatlista, checkboxok és jelölőnégyzetek, illetve a HTML5 dátum valamint színválasztója is. A reset és send gombok az űrlap alján találhatók.

Média elem: Videó

Az egyetlen video a video.html weblapon található. A videó maga különböző kutyafajtákat mutat be. A video mellett található a kontrol panelja is. Ha az egeret az információs ikon felé visszük, akkor a videóról néhány információ jelenik meg egy div elemben. A sebességnél a nyilakra kattintva állítható a lejátszási sebesség, amit a két nyíl közötti szám mutat. Ha arra a számra kattintunk a sebesség visszaáll az alap 1.0-ra. Valamint a nyilak folyamatos lenyomásával, folytonosan növekszik, csökken a sebesség.

II. feladat

A CSS formázások, minden weboldalon vannak internal és external formában is. A közös formázások a default.css-ben Vannak, míg az egyediek internal formában vannak. Leginkább azonosító szerint Vannak a formázások vagy tag-ek alapján. Az osztály szerinti azonosítás a list.html-ben található. A táblázatban minden középre igazított, és a cellák között térköz van hagyva. A menü bal oldalt található, flexel megoldva. A színek a barna árnyalatai, amibe a háttérszínek is beletartoznak. A linkek valamennyivel beljebb vannak a többi szíveghez képest. Míg az űrlap elemek kerete változik hibás bemenet esetén. A gombok közül a memory-game.html oldalon van megformázva.

Miskolc, 2024.

III. feladat

Form ellenőrzés

Nem minden beviteli mezőt ellenőriz, csak a kötelezőeket ellenőrzi, és hibás bevitel esetén alert() metódussal jelzi, illetve a keret pirosra változik. A név mezőbe minimum 3 karakter hosszú csak betűkből álló szöveg írható, pattern-el ellenőrizve. Az irányítószám 4 jegyű szám, születási idő nem lehet nagyobb a későbbi vagy megegyező az aznapival. Az emailnek meg kell felelnie az szabványformának és a lakcímbe is árva kell lennie valaminek.

JQuery animáció

JQuery-nek több beépített animációs függvénye is van. A memory_game a hide() és show() eljárásokat használja, míg a list.html bal felső sarkában egy kúszó kutya található, amely rákattintás után körbe körbe meg a képernyő szélén az animate() és css() segítségével.

Elemek kiválasztása, formázása

Az elemek JQuery-vel történő kiválasztása mind az azonosítójuk, az osztályuk és tag-ük alapján is történhet. Ezek közül az osztály szerinti, de leginkább az azonosító szerintit használtam.

A formázást leginkább a kiválasztott elemek osztályainak szerkeztésével értem el. Egy már előre létrehozott CSS osztállyal bővitettem az elemeket a fenti bal képen látható módon. Viszont formázni lehet a JQuery .css() utasításával is. Ez egyedül a JQuery animáció alpontban említett kutya képen használatam. Ez annyit tett, hogy amikor a kép a sarokba ért elforgattam 90 fokkal.

Új html elemek

A form.hml-ben a kutya kiválasztása után megjelennek annak információi. Ez a részben az addInformation() funkció hajtja végre, ami konkrátan hozzáadja a megfelelő adatokat tartalmazó <p> elemeket. Először létrehoz egy ideiglene változót, amiben JQuery-vel beállítja annak paramétereit, majd az append() függvény használatával hozzáadja azt.

Miskolc, 2024.

```
var place = $('<p id="place">').text('Menhely: '+dogs_data[i].place);
$('#informations').append(place);
var age = $('<p id="age">').text('Kor: '+dogs_data[i].age);
$('#informations').append(age);
var health = $('<p id="health">').text("Egészségügyi információk:"+dogs_data[i].health);
$('#informations').append(health);
var good_with = $('<p id="good_with">').text('Jól kijön:' +dogs_data[i].good_with);
$('#informations').append(good_with);
$('#informations').append("<label for='favcolor'>Nyakörv színe:</label>  <input
```

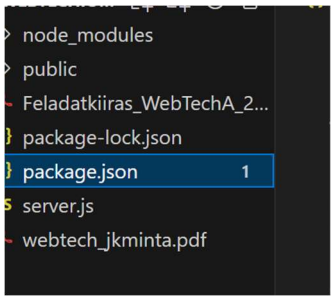
JSON file-ok, és AJAX

JSON file-okat több weboldalam is használ. Elsősorban a list.html a breeds.json- ből olvas ki konktéran minden adatot. Itt a JQuery getJSON() funkcióját használtam, amivel a fileból beolvasott adatokat egy JS Objektumként tárolom el. Azt az objektumot használja a list, ami kiolvassa a kkitya fajtájától kezdve, a kép linkjén át a tulajdonságokig mindent.

A másik az form, ahol a kutya kiválsztása után annak adatait a adopts.json-ből olvasom ki a .fetch() függvény segítségével. Mivel ez a függvény lehet eléggé időigényes a beolvasását egyszer vágzem el és az eredményt szintén JS objektumban tárolom.

Szorgalmi

A kész projekt helyi szerveren való megjelenítését a NodeJs express kiegészítőjével viszonylag gyorsan és egyszerűen meg tudtam oldani. A server.js tartalmazza a protot, valamint a fájlok elérését. A path használata is nélkülözhetetlen volt, ugyanis a szükséges css és js fájlok a Public mappa almappájában vannak elhelyezve. A node-modules mappa tartalmazza a telepített kiegészítőket. Míg a a package-lock.json a kiegészítők adatait, elérési útvonalát tartalmazza. Legvégül a package.json maga a projekt adatait tárolja.



Miskolc, 2024.