**1. modul**

Introduction to Basics

Web basics

HTML, CSS, Javascript

# Bevezetés

Miért van szükségünk a weboldalakra? Azért, mert szeretnénk a felhasználók igényeit figyelembe véve olyan kreatív felületet nyújtani ami kielégíti a szükségleteiket. Multimédiás eszköz, mely több médiumot is magába hordoz.

# HTML

A weboldal leírója a HTML (HyperText Markup Language) nyelv, egy szabványosított programozási nyelv. A HTML fájl egy egyszerű szövegállomány, mely úgynevezett jelölő tag-eket tartalmaz. A tag-ek alapján tudja megjeleníteni az oldalt a böngésző.

A fájlok kiterjesztési nyelve .html, szerkesztéséhez elég lehet egy egyszerű szövegszerkesztő.

## HTML fájl felépítése

|  |
| --- |
| <html>  <head>  <title>az oldal címe</title>  </head>  <body>   ez a honlapom  </body> </html> |

HTML tagot a < és > karakterek veszik körül.

A tagok tulajdonságokat is tartalmazhatnak (dupla idézőjelekkel). Pl.:

|  |
| --- |
| <body bgcolor="red"></body> |

## Alapvető HTML tag-ek

### Címsorok

A h1 a legnagyobb szintű cím, a h6 a legkisebb.

|  |
| --- |
| <h1></h1>, <h2></h2>, <h3></h3>, <h4></h4>, <h5></h5>, <h6></h6> |

### Bekezdések

A HTML-ben a tördelést nem befolyásolhatjuk a kódban elhelyezett szóközökkel és ENTER-ekkel. Ezeket HTML tag-ekkel kell jelöljük.

|  |
| --- |
| <p>ez egy bekezdés</p> <br>ez már új sorban vam <hr> ez már a vízszintes vonal alatt van |

Lehetőség van block-okba zárni egy egy kódrészletet, hogy közös tulajdonságokkal ruházzuk fel őket. Méghozzá így:

|  |
| --- |
| <div>ami ezen belul van, akár tag-ek között is az össze van fogva</div> |

### Betűstílusok

A megjelenítendő szöveg formázásához:

|  |
| --- |
| <b>ez egy félkövér szöveg </b> <big>nagyobb szöveg</big> <em>szövegkiemelés</em>  <i>dőlt szöveg</i> <small>kisebb szöveg</small> <strong>félkövér szöveg</strong> |

### Linkek

Amennyiben valahova el szeretnénk navigálni az oldalunkról:

|  |
| --- |
| <a href="http://www.google.com">Ez egy link a google.com-ra.</a> <a href="http://www.google.com" target="\_blank">Link megnyitása új lapon.</a> |

### Gombok

Előfordulhat, hogy szükségünk van gombra:

|  |
| --- |
| <button type="button">Kattanj rám.</button> |

### Képek

Megjeleníthetünk képeket is:

|  |
| --- |
| <img src="hatter.jpg"> <img src="hatter.jpg" alt="ez egy kép egy hídról"> |

A html elemeket link-ként is lehet használni:

|  |
| --- |
| <a href="index.html">  <img src="fooldal.jpg" alt="vissza a főoldalra" width="40" height="40"> </a> |

### Táblázatok

A táblázatoknak megadott struktúrájuk van:

|  |
| --- |
| <table>  <tr>  <td></td>  <td></td>  <td></td>  </tr>  <tr>  <td></td>  <td></td>  <td></td>  </tr> </table> |

Ahol a *<table></table>* létrehozza magát a táblázat HTML elemet, a *<tr></tr>* tag a sorokat, a *<td></td> p*edig a sorokon belüli mezőket. Ha szeretnénk a table tag-en belüli a sorok előtt megadhatjuk *<th></th>*  tag-ek között fejlécként az oszlopok neveit.

### Listák

Ahogy egy szövegszerkesztőben, úgy a weboldalon is többféle lista lehet.

**Felsorolások**:

|  |
| --- |
| <ul>  <li>kiskutya</li>  <li>kismacska</li>  <li>kiskacsacsőrűemlős</li> </ul> |

**Számozások**:

|  |
| --- |
| <ol>  <li>kiskutya</li>  <li>kismacska</li>  <li>kiskacsacsőrűemlős</li> </ol> |

**Definíciós lista:**:

|  |
| --- |
| <dl>  <dt>kiskutya</dt>  <dd>puli,roti</dd>  <dt>kismacska</dt>  <dd>balinéz,bengáli</dd> </dl> |

### Head (fejrész, fejléc)

Ez egy olyan HTML tag, elem ami információkat tartalmaz a dokumentum egészére. Amolyan “beállítási lehetőségek” mintsem konkrét elemek találhatóak itt. Ebből a tag-ből csak egy lehet, minden “beállítási” tag ezen belül helyezkedik el.

|  |
| --- |
| <head>  <title>cím ami a böngésző tab-ján látszik</title>  <meta name="description" content="progmatic gyakorlat">  <style>body{color: blue;}</style>  <script>alert('Hellobello')</script> </head> |

Ahol a *<title>* a címet a bongésző ablaknak, a *<meta>* metaadatok beállítása, például keresőszavak a Google számára, a *<style>* a weboldal megjelenítési stílusának a beállítása. A *<script>* pedig a dinamikus, interaktív tartalom használatáról gondoskodik. Ami a fenti példában egy felugró ablak a weboldal betöltésekor. **Fontos megjegyezni**, hogy a script tag(ek)-be ágyazott kódrészlet előbb fut le, hajtódik végre mint a *<body>* tag tartalma.

HTML-ben kommentet írni a **<!--** és a **-->** tagek közé írva lehet írni.

# CSS

CSS, azaz Cascading Style Sheet. Egy olyan “adatlap”, ami meghatározza a HTML elemek megjelenését. Ezeket a stílusokat általában külön, egy .css kiterjesztésű fájlban tároljuk. Ezzel időt spórolhatunk meg, könnyebb karbantarthatóság.

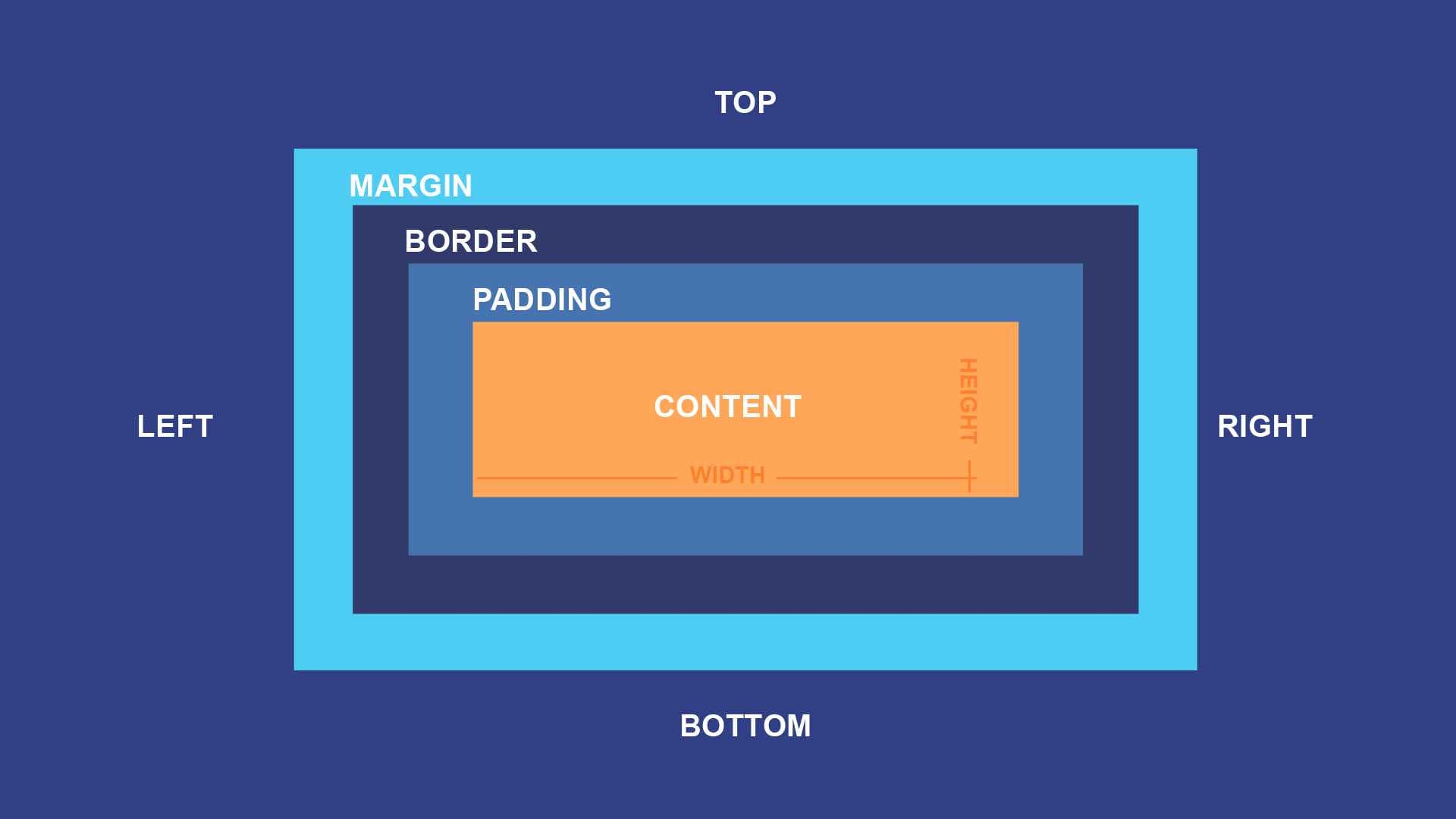
Az alábbi módon épül fel:

|  |
| --- |
| htmlelem{  tulajdonság: beállítandóérték;  tulajdonság2: beállítandóérték2;  } |

Az alábbi módon épül fel:

|  |
| --- |
| body{  color: blue;  text-align: center; */\*szöveg középre helyezése\*/* } |

**Megjegyzések**kel tűzdelhetjük kódunkat a későbbi könnyebb megérthetőség miatt. A **/\*** és a **\*/** karakterek közötti szövegre nem kódként tekint a böngésző.



## 

## Háttér

A következőkben megnézzük milyen stílus beállítási lehetőségeink vannak az egyes elemekre a teljesség igénye nélkül.

A háttér CSS beállításához a következő beállítási lehetőség közül választhatunk:

|  |
| --- |
| htmlelem{  background-color: érték; */\*háttér színe \*/*  background-image: url(a háttérkép elérése);   background-repeat: érték; */\*háttérkép ismétlődése \*/*  background-position: érték; */\*háttérkép pozíciója \*/*  background-attachment: érték; */\*háttérkép ragasztása\*/* } |

## 

## Szöveg

A szövegek CSS beállításához a következő beállítási lehetőség közül választhatunk:

|  |
| --- |
| htmlelem{  color: érték;  text-align: érték; */\*igazítás\*/*  text-decoration: érték; */\* szöveg megj: aláhúzott, áthúzott, stb.\*/*  font-family: érték; */\*betűtípus \*/*  font-size: érték; */\*betűméret \*/*  font-style: érték; */\*stílus \*/*  font-weight: érték; */\*betű vastagsága \*/* } |

## 

## Szegély

A szegélyek CSS beállításához a következő beállítási lehetőség közül választhatunk:

|  |
| --- |
| htmlelem{  border-style: érték; */\*solid, dotted, dashed, groove, double, stb. \*/*  border-color: érték; */\*keret színe\*/*  border-width: érték; */\*px-ben megadva\*/*  border-radius: érték; */\*adott html elem sarkainak lekerekítése\*/*  } |

## 

## Térköz

A térközök CSS beállításához a következő beállítási lehetőség közül választhatunk:

|  |
| --- |
| htmlelem{  margin: értékek; */\*külső margó px, cm, em-ben megadva, sorrend: fel, jobbra, le, balra \*/*  margin:0 auto; */\*középre igazítás\*/*  padding: értékek; */\*belső margó px, cm, em-ben megadva, sorrend: fel, jobbra, le, balra \*/* } |

## 

## Lista

A lista CSS beállításához a következő beállítási lehetőség közül választhatunk:

|  |
| --- |
| htmlelem{  list-style-type: érték; */\*lista jelének típusa: disc, square, circle, none számozott listák esetén: decimal, lower-roman, upper-roman, lower-alpha \*/*  list-style-image: url(elérési út); */\* "listajel" helyett kép berakása\*/*  list-style-position: érték; */\*lehet: inside, outside \*/*  list-style: értékek; */\*előbbiek közös megadása\*/* } |

## 

## 

## 

## Méret

Az elemeknek nem csak a formáját és színét hanem egyebek mellett a méretét is megadhatjuk:

|  |
| --- |
| htmlelem{  width: szélesség; */\*csak blokkszintű elemeknél: <p>,<div>,<h1>,<ul>,<ol>,<table>\*/*  height: magasság; */\*csak blokkszintű elemeknél\*/* } |

## 

## Megjelenítés

Az elemeknek nem csak a formáját és színét hanem egyebek mellett a méretét is megadhatjuk:

|  |
| --- |
| htmlelem{  display: érték; */\*lehet block, inline, none, flex, stb\*/*  float: érték; */\*left, right, stb., körbefuttatás szöveggel vagy hasábos\*/* } |

## 

## Pozicionálás

Fontos lehet, hogy az elemek hogyan helyezkednek el egymáshoz viszonyítva:

|  |
| --- |
| htmlelem{  position: érték; */\*static, relative, absolute, fixed \*/*  } |

**static**: elemek alapértelmezett tulajdonsága  
**relative**: megadható top és left tulajdonság, de az elem eredeti helye üres lesz, nem kerül oda elem  
**absolute**: hasonló mint a relative, de itt az elem nem tart fenn területet  
**fixed**: rögzített, a képernyőhöz viszonyítva állandó a pozíciója

## 

## Linkek

A linkek is különféleképpen viselkedhetnek:

|  |
| --- |
| a:link { color:érték; }  a:visited { color:érték; } a:hover { color:érték; } a:active { color:érték; } |

**link**: link alapértelmezett tulajdonsága  
**visited**: már látogatott link

**hover**: felé vittük az egeret

**active**: rákattintunk

# CSS osztályok

Eddig a CSS esetén amit láttuk példákat, mind mind az összes HTML elemre adott stílus jellemzés megadására szolgált az adott oldalon. (amennyiben a head tag-ben adtuk meg és nem inline, az adott tag sorában)

Ha a HTML elemek egy csoportjának szeretnénk egyedi stílusjellemzőt adni, akkor erre létre tudunk hozni saját stílus osztályokat. Kinézete és megadási módja nagyon hasonló a HTML elemekhez, csupán egy szembetűnő különbség van, az az hogy pont-ot teszünk a neve elé.

|  |
| --- |
| .osztalynev{  attributum: érték;  } |

Például:

|  |
| --- |
| .sajatgombom{  background-color: blue;  } |

Itt is ugyanúgy egyszerre több tulajdonságot beállíthatunk természetesen. Ha már megvolt a létrehozás, akkor be is kell állítani a kívánt elemeknél, hogy ők bizony ebbe a stílus osztályba tartoznak. Ezt a *class* attributummal tudjuk megtenni:

|  |
| --- |
| <button class="sajatgombom"> Click me!  </button> |

# 

# CSS azonosítók, id-k

Az is felmerülhet bennünk, hogy mi nem mind, de nem is csak a HTML elemek egy csoportját szeretnék egyedi stílussal felruházni hanem csak egyet, egyiket. Erre is lehetőség van. Tudunk egyedi azonosítókat definiálni, az osztályokhoz képest, egy nagyon apró különbséggel:

pont helyett kettőskereszt használatával:

|  |
| --- |
| #id{  attributum: érték;  } |

Például:

|  |
| --- |
| #sajatpirosgombom{  background-color: red;  } |

Használni pedig nagyon egyszerűen, az eddig ismert class helyet az id attribútumot használjuk.

|  |
| --- |
| <button id="sajatpirosgombom"> Click me!  </button> |

Természetesen technikailag semmi akadálya nincs hogy több ugyanannak a HTML elemnek ugyanaz legyen az id-ja, de nagyon nem szerencsés így cselekedni.

Illetve még amit érdemes megjegyezni, hogy egy HTML elemnek lehet id-ja és class-ja is egyszerre. Tehát egy példa, hogy megmondhatjuk class segítségével, hogy az összes OK gomb hogy nézzen ki az oldalon, de id segítségével a Fizetés gombot specializálhatjuk, hogy az megkapja a class beli értékeket, de id-ban meghatározzuk hogy legyen zöld.

# Formok, form elemek

A weboldalak egyik nagyon fontos lehetséges területe a formok, másnéven űrlapok és az ezekhez kapcsolódó HTML elemek, melyek a felhasználói bevitelt biztosítják.

Ezeket az elemeket a form-ok esetén form tag-ek közé szokás rakni.

|  |
| --- |
| <form> . form elements . </form> |

## **Input**

A HTML elemnek többféle típusa van. Nézzük őket sorba:

### Egyszerű beviteli mező

|  |
| --- |
| <input type=”text”> |

### 

### Radio button

|  |
| --- |
| <input type=”radio”> |

### Checkbox

|  |
| --- |
| <input type=”checkbox”> |

### Submit/elküldés gomb

|  |
| --- |
| <input type=”submit”> |

### Reset/visszaállítás gomb

|  |
| --- |
| <input type=”reset”> |

### Egyszerű gomb

|  |
| --- |
| <input type=”button”> |

### Jelszó mező

|  |
| --- |
| <input type=”password”> |

### Dátum mező

|  |
| --- |
| <input type=”date”> |

### Mennyiség választó mező

|  |
| --- |
| <input type=”number”> |

### Mennyiség csúszka mező

|  |
| --- |
| <input type=”range”> |

Most pedig nézzük meg mi más elem lehet még:

### Label / felirat

|  |
| --- |
| <label></label> |

### Dropdown / lenyíló lista

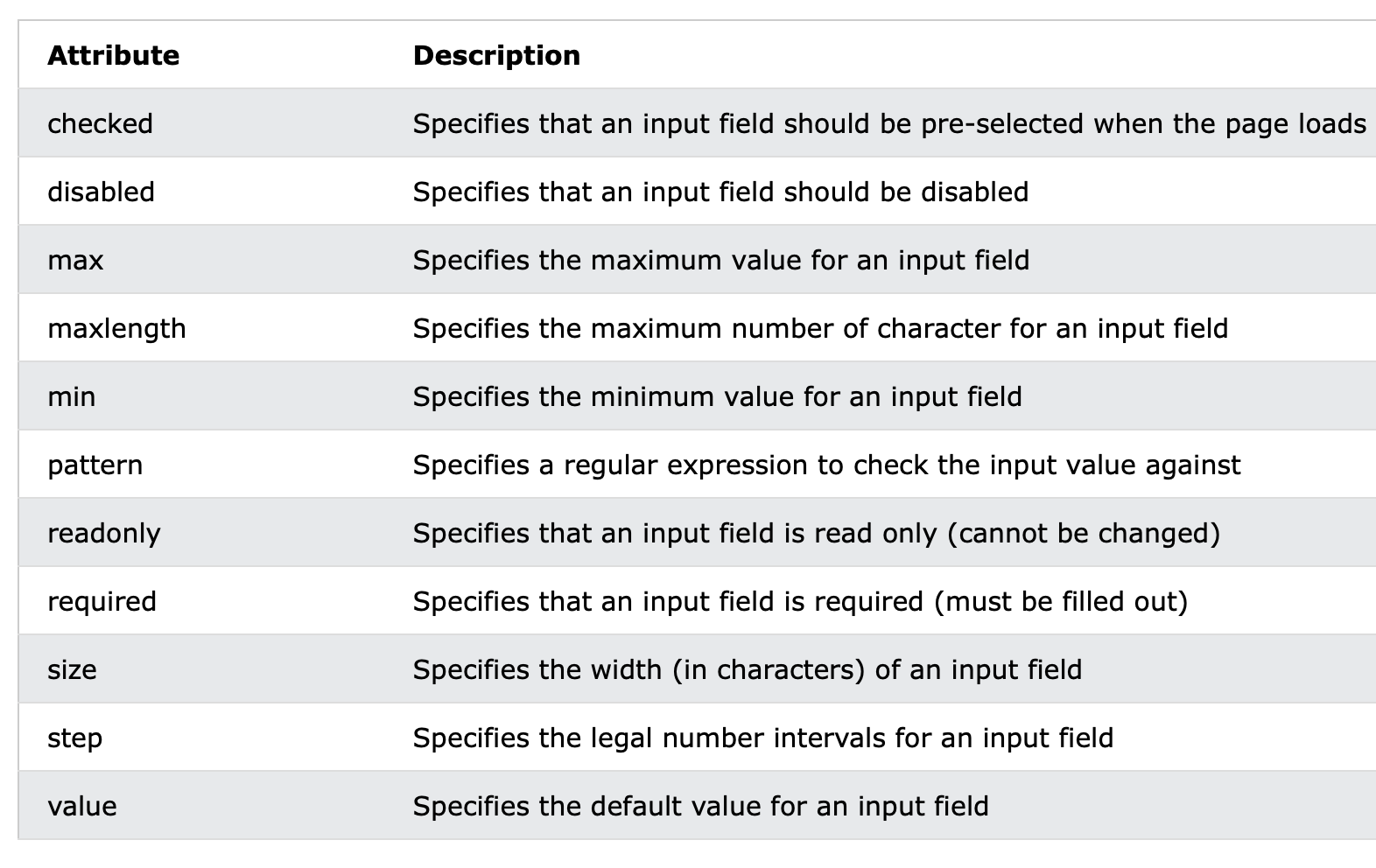
|  |
| --- |
| <select id="workdays" name="workdays">  <option value="monday">Monday</option>  <option value="tuesday">Tuesday</option>  <option value="wednesday">Wednesday</option>  <option value="thursday">Thursday</option>  <option value="friday">Friday</option>  </select> |

### Textarea / szövegdoboz

|  |
| --- |
| <textarea rows="20" cols="70"></textarea> |

### Fieldset / Mezőhalmaz

|  |
| --- |
| <fieldset></fieldset> |



Például:

|  |
| --- |
| <input type="number" id="quantity" name="quantity" min="0" max="100" step="10" value="30"> |

Bővebben: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input>

# Javascript

Ahogy fentebb néztük (alert popup), a javascript tud dinamikát adni. Tehát ők töltik meg élettel a web elemeket. Nézzük csak meg, hogy mi lenne ha beírnánk csak úgy, hogy 5+2?

|  |
| --- |
| const myHeading = document.querySelector("h1"); myHeading.textContent = "Hello world!"; |

Nézzünk is egy példát rá:

|  |
| --- |
| <button type="button" onclick="document.getElementById('demo').innerHTML = Date()"> Click me to display Date and Time.</button>  <p id="demo"></p> |

Itt ami történik, hogy a gombnak van egy ***onclick*** tulajdonsága, ami akkor hívódik meg, ha kattintás történik a gombon. Az pedig, hogy mi történjen, az utána lévő kifejezés mondja meg. Jelen esetben az utána lévő kódrészlet azt jelenti, hogy megkeresi az oldalon azt az elemet aminek az “id”-ja *demo* és beleírja a jelenlegi dátumot. *(tehát nem meglepően de létre kell hozni egy olyan id tulajdonsággal rendelkező elemet, amire szükségünk van)*

A Javascript segítségével nem csak a HTML elem tartalmát tudjuk módosítani, hanem a tulajdonságait is felül tudjuk írni. Például:

|  |
| --- |
| <button onclick="document.getElementById('myImage').src='bulb-on.gif'">Turn on the light</button>  <img id="myImage" src="bulb-off.jpg" style="width:250px"> |

Itt a *myImage* id-jú elemet megkeresi a javascript és lecseréli az src tulajdonságának értékét egy másikra.

Nem csak az src tulajdonságot tudjuk módosítani, hanem bármelyiket. Nézzük meg ha például a *style*-t manipuláljuk:

|  |
| --- |
| <button type="button" onclick="document.getElementById('demo').style.fontSize='35px'">Click Me!</button>  <p id="demo">JavaScript can change the style of an HTML element.</p> |

Kommentet írni Javascript-ben a /\* és \*/ karakterekkel lehet. Bármi ami köztük van, az kommentnek fog minősülni.

# Struktúra

Rend a lelke mindennek, ezért még mielőtt elkerüljük a zsúfoltságot a HTML elemek, stílusok és scriptek világában picit rendezzük sorainkat. Hisz nem meglepő egyik program sem egy darab fájlból szokott állni.

A CSS, stílussal kapcsolatos összes beállítást egy (vagy több), de külön fájlba szokás tárolni aminek kiterjesztése .css.

Ezt követően a használathoz csak be kell állítanunk az elérését a head tag-en belül

|  |
| --- |
| <link rel="stylesheet" href="egyedistilus.css"> |

Az *egyedistilus.css* fájl tartalma pedig:

|  |
| --- |
| body{color: blue;} |

Nézzük ugyanezt a javasript esetében, amit szintén a head tag-ek közé kell rakni:

|  |
| --- |
| <script src="sajatjavascript.js"></script> |

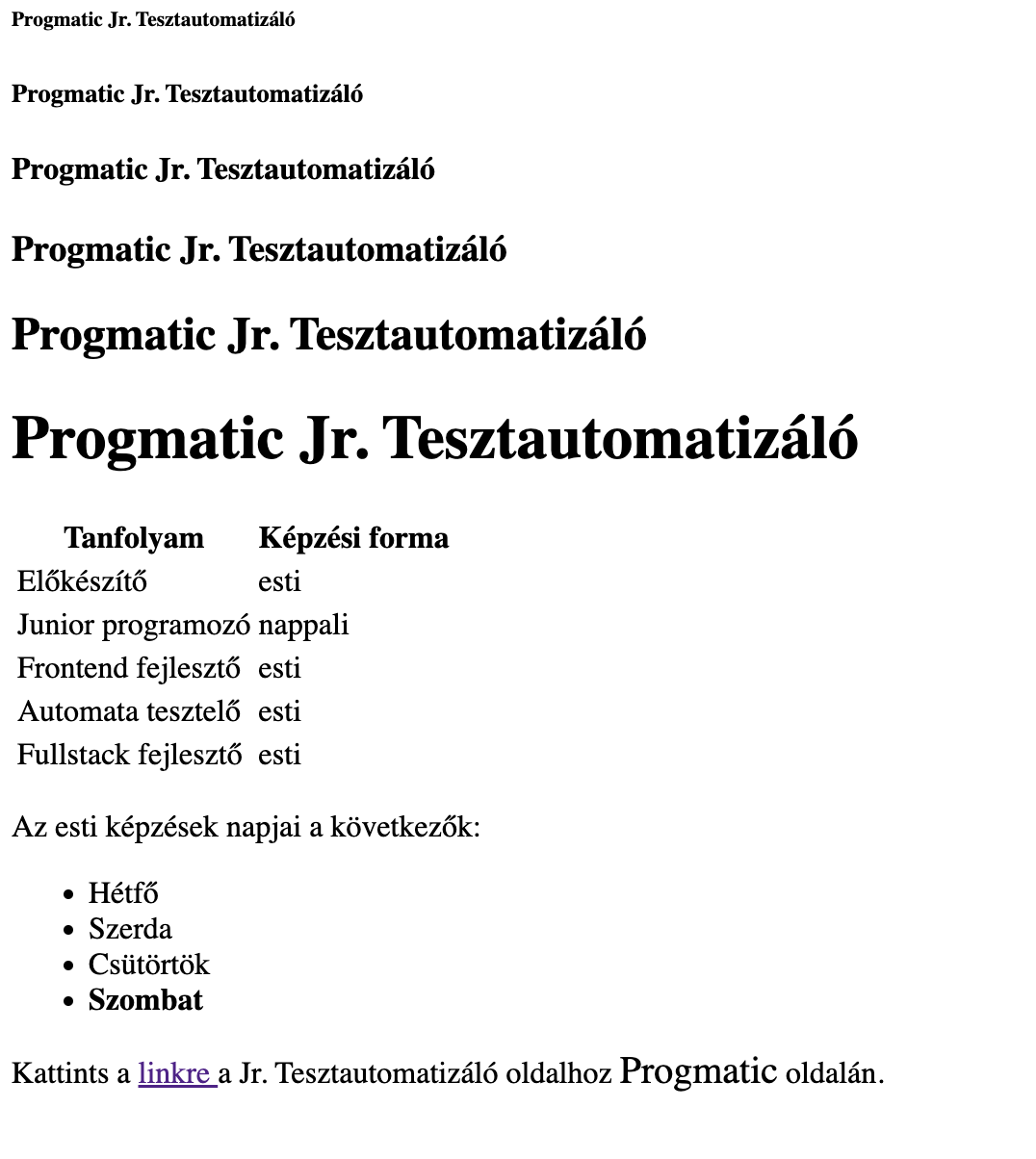
A *sajatjavascript.js* tartalma pedig:

|  |
| --- |
| alert('Hellobello') |

# 

# Feladatok

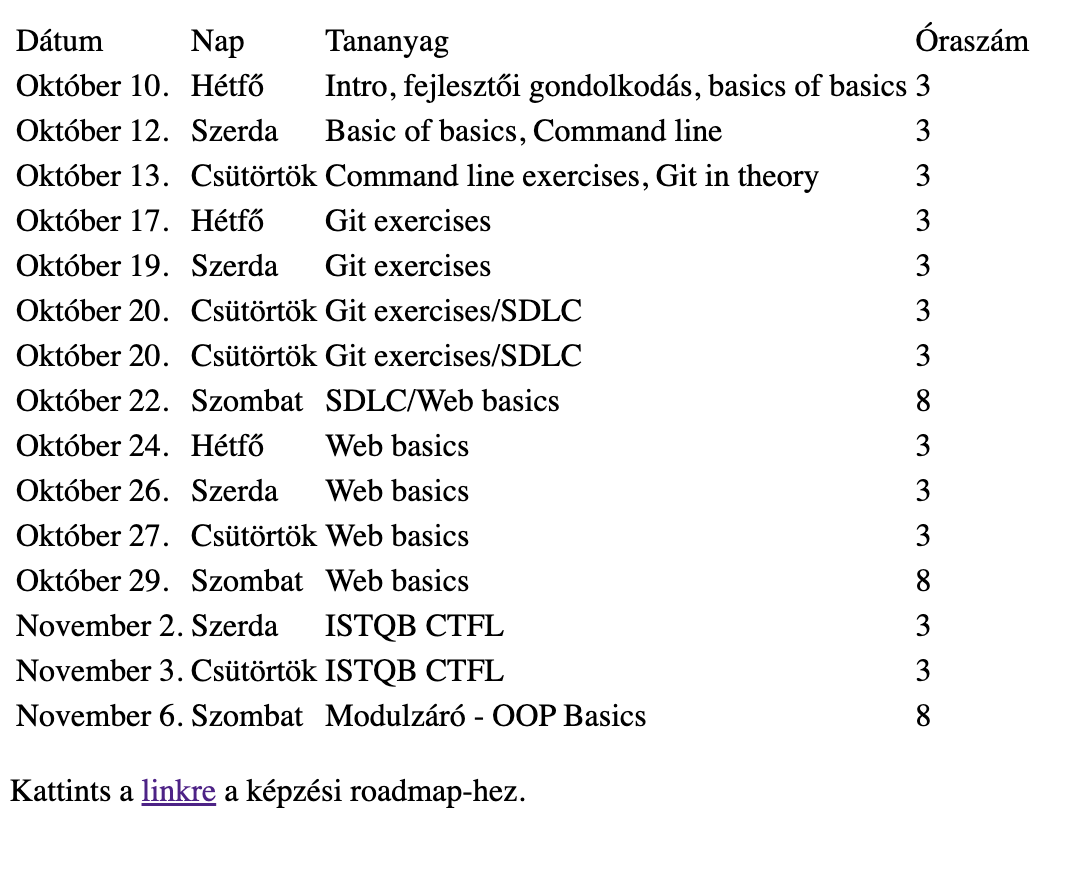
1. Készítsünk egy weboldalt, melyre kiírjuk a *Progmatic Jr. Tesztautomatizáló* szöveget h6-tól h1-es címsor méretig. Valamint legyen egy link, az iskola képzési oldalára. Egy kis táblázatba jelenítsünk meg az aktuálisan futó tanfolyamokat.  
   +: Legyen egy kis táblázat a képzésekről, az oszlopok: tanfolyam és képzési forma. Ezen felül egy listát is készítsünk, melynek elemei az esti képzések napjai.



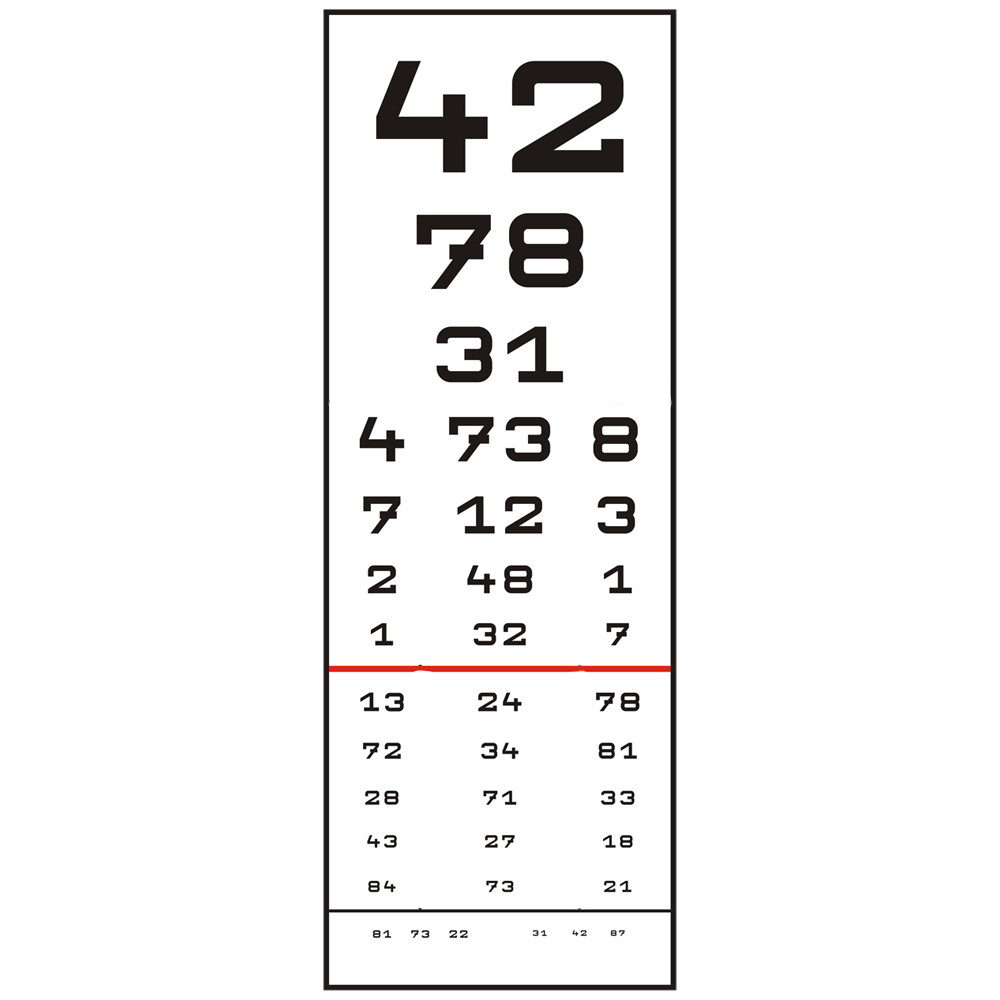
1. Készítsünk egy weboldalt, amely listába szedve összegyűjti a szerencsejatek.hu aloldalait, amin látszanak a korábban kisorsolt számok pl [ötös](https://bet.szerencsejatek.hu/jatekok/otoslotto/sorsolasok). Az oldal alján helyezzük el a cég logóját, melyet helyi könyvtárból olvasunk fel. A kép mérete legyen 400px széles és 100 px magas.  
   +: a felsorolt elemek után mindegyiknek legyen ott a kisméretű logója/képe, ami nagyjából olyasmi magas legyen, mint a szöveg maga

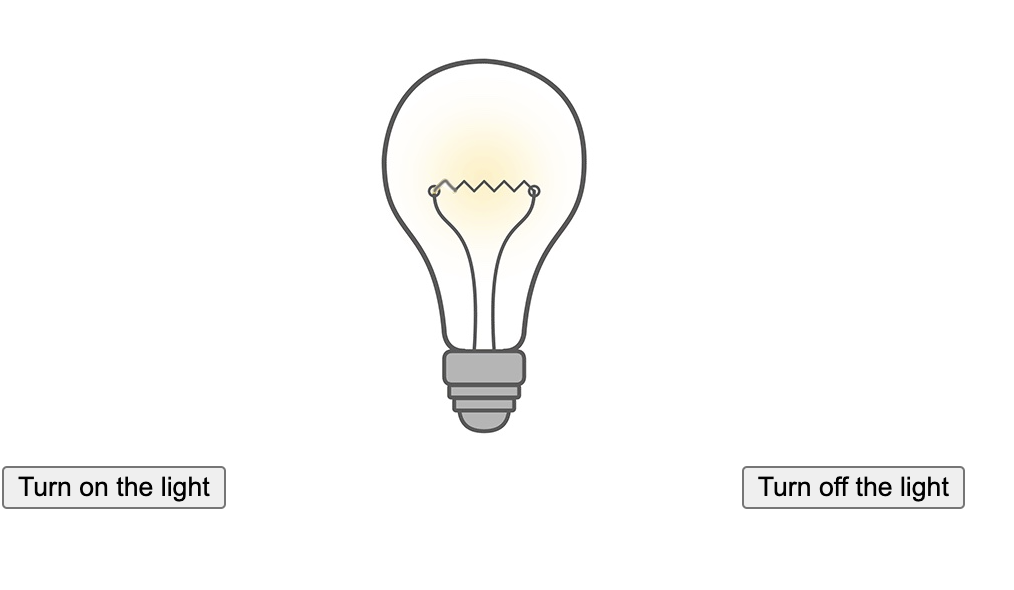


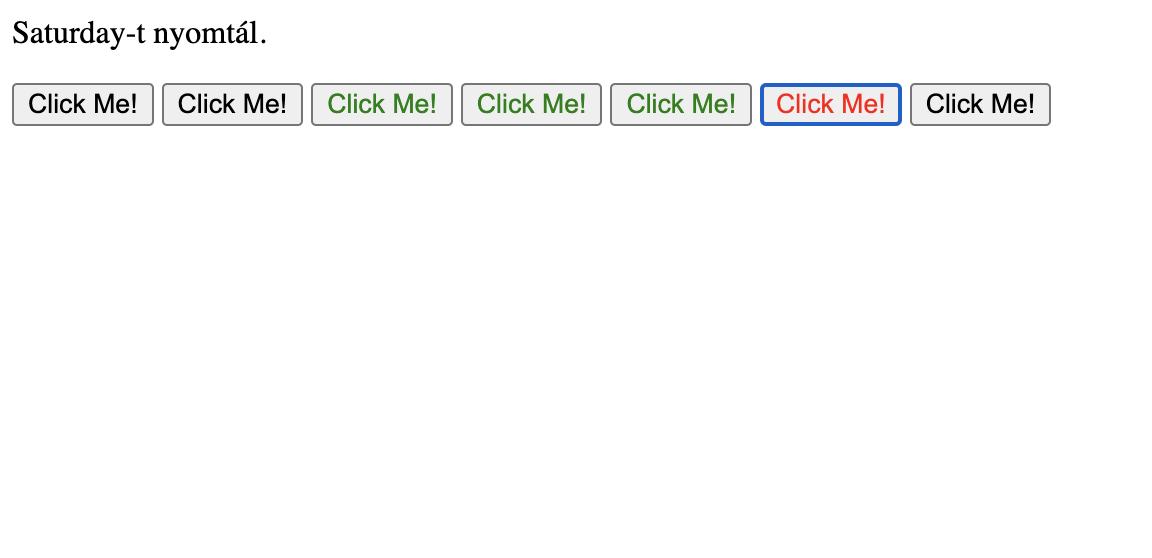
1. Készítsünk egy, az idei évre vonatkozó tanmenetről szóló egyszerű táblázatot egy weboldalra. A táblázat oszlopai: dátum, nap, tananyag, óraszám, órától-óráig. Illetve legyen benne egy link, mely a tanfolyam tanmenet táblázatra mutat. *(Elég csak az első modul képzési napjait tartalmaznia.)*

**

1. (Órai) Írjunk egy weboldalt, amely a szemészetnél/látszerésznél használatos látásvizsgálatra.



1. (Órai) Gyakoroljuk a CSS-t, készítsünk egy weboldalt, melyen egy képet a lehető legtöbb formában megjelenítjük.
2. Készíts az előző feladatok alapján egy állattablót egy menhely vagy vadaspark számára. Legyen üdvözlőszöveg, egy pár mondatos leírás a helyről. Egy kis táblázat a nyitvatartási időről és egy másik a jegyárakról. Illetve legyen minimum 7 kép az egyes állatokról, a nevükkel együtt.
3. (Órai) Nézzünk példát a gombok gyűjteményére, ahol használunk HTML elem szintű, class szintű és id szintű stílus beállítást is
4. (Órai) Írjunk egy weboldalt, amelyen van két gomb, meg egy villanykörte amit kapcsolgatni lehet.
5. (Órai) Írjunk egy weboldalt, amelyen van 7 gomb, melyek a hét napjait szimbolizálják. Ezen felül egy label (felirat), ami kiírja, hogy épp mely napra kattintottunk és színezze át a gombok feliratát, amelyikre már kattintottunk.
   1. Ha ez megvan, akkor egy feltétellel megspékelve, egy bizonyos nap esetén más színű legyen a gomb felirata, mint a többi nap.



1. Írjunk kincskereső játékot. Legyen egy 7x7-es tábla, ahol az egyes mezők gombok. Mindig írja ki a program, hogy mely gombra kattintottunk és egy “sajnos nem talált” szöveget, ha a gomb alatt van a kincs. Amennyiben az a gomb ahol a kincs van, akkor pedig egy “hurrá, megvan a kincs!” feliratot. *(Az hogy melyik gomb alatt van a kincs, az mi döntjük el a kódban, a lényeg hogy csak egy ilyen gomb legyen ami teljesíti ezt a feltételt. )*
2. (Órai) Készítsünk egy weboldalt, melyen tudunk kapcsolgatni a dark-mode és light-mode között.
3. (Órai) Készítsünk egy űrlapot, melyen minimum egy dátum van, mint beviteli mező és egy gomb, melyre kattintva egy másik oldalra navigál minket.
4. Készítsünk egy weboldalt egy hírlevél feliratkozásról, ahol meg kell adni a keresztnevet és emailcímet, illetve rá lehet kattintani a Feliratkozás gombra, melyre kattintva egy tájékoztatjuk a felhasználót, hogy sikeresen feliratkozott.
5. Készítsünk egy weboldalt, mely egy webshop megvásárolt termékei esetén kitöltendő vásárlói és szállítási/számlázási adatokat képes elkérni a végfelhasználótól. Használjunk CSS-t az elemek csinosabbá tételére, valamint esetleg rakjuk be egy képet az oldalra vagy háttérnek ami a webshop területére utal. A form gombjára való kattintást követően navigáljon el a weboldal egy másik weboldalra, melyen megköszönjük a vásárlást.
6. Készítsünk egy regisztrációs weboldalt, mely lényegében 3 oldalból áll. Az első oldalon üdvözöljük a webáruházban egy rövid leírással az árult termékekre vonatkozóan és tájékoztatjuk, hogy a vásárláshoz regisztráció szükséges. Illetve egy táblázat a lehetséges utánvételes szállítási díjakhoz. A regisztráció megkezdéséhez tegyünk ki a felületre egy gombot, amire kattintva a regisztrációs felületre navigál az oldal. Ezen a regisztrációs felületen kérjünk el minden szükséges adatot a felhasználótól. Tegyük fel, hogy a webshop szeretne különböző statisztikákat készíteni a vásárlóktól, így minél több adatot kérjünk el a vásárlóktól. A regisztrációs adatok megadása után legyen egy gomb, amire kattintva az oldal egy “köszönjük, hogy regisztrált, most már vásárolhat”.
7. Készítsünk egy olyan hiánypótló weboldalt a fentiekhez, melyen vannak kártyák a megvásárolható termékekről és minden kártyán egy gomb amivel megvásárolhatom őket. Ezeken felül az oldal alján egy gomb, mely a Kosár oldalra navigál. A Kosár oldalon tételszerűen soroljunk fel termékeket. *(fixen 3-4-5 terméket, az előző oldalon lévő Megvásárolom gomboknak nem kell működnie)* A tétel lista alatt pedig egy link vagy gomb, mely a következő oldalra navigál a szállítási és számlázási adatok megadásához. Tegyünk róla, hogy a 12-14-es feladatban elkészült oldalak valamint a mostani össze legyenek “kötve”, végig tudjon menni a felhasználó rajtuk.
8. Írj egy bemutatkozó oldalt magadról, annyi HTML, CSS és Javascript elemet felhasználva amennyit csak tudsz. *(kedvenc szabadidő, hobby, melyik városban laksz, mit ajánlasz ott mint látnivaló, mi a célod az életben, hova utaznál legszívesebben és miért…stb)*

# Linkek

Mozilla Online Web tutorial for beginners <https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web>

HTML elemekről leírás  
<https://developer.mozilla.org/en-US/docs/Web/HTML/Element>

HTML Teszt feladatok  
<https://www.w3schools.com/html/html_exercises.asp>

CSS Almanac  
<https://css-tricks.com/almanac/>

Integrált Fejlesztő Környezet (Integrated Development Environment - **IDE**)  
<https://code.visualstudio.com/>