

HTML

Feladat: Személyes oldal készítése

- Hozzunk létre egy `index.html` fájlt, ami tartalmazza:
 - A saját nevet `<h1>` címként.
 - Egy rövid bemutatkozó szöveget egy `<p>` tagben.
 - Egy felsorolást (``) a kedvenc hobbikról (legalább 3 hobbi).
 - Egy képet, alt szöveggel.
 - Egy email linket (`mailto:`) a kapcsolatfelvételhez.
-

Feladat: Recept oldal

- Készítsünk egy „Recept” oldalt, ami tartalmazza:
 - A recept nevét `<h1>`-ben.
 - Rövid leírást egy bekezdésben.
 - Hozzávalók listáját egy rendezetlen listában (``).
 - Lépéseket egy rendezett listában (``).
 - Egy táblázatot, amiben feltüntetjük a tápanyagértékeket (pl. kalória, fehérje, zsír, szénhidrát).
-

Feladat: 3 oldalas mini weboldal

- Hozzunk létre egy mini webhelyet 3 HTML oldallal:
 - **Kezdőlap** (`index.html`) – rövid üdvözlő szöveggel és linkekkel a másik két oldalra.
 - **Rólunk** (`rolunk.html`) – néhány információval az oldal „készítőiről”.
 - **Kapcsolat** (`kapcsolat.html`)
- Minden oldal tetején legyen egy navigációs menü linkekkel a három oldalhoz.
- Legyen minden oldalon legalább egy cím (`<h1>`), és egy bekezdés (`<p>`).

CSS

Feladat: Személyes oldal + CSS

HTML:

- Fejlécben a név, rövid bemutatkozás, hobbi lista, profilkép és email link.

CSS:

- Az oldal háttere legyen világos szürke (`background-color: #f0f0f0;`).
 - A fejléc (`<header>`) kapjon sötét hátteret, fehér szöveget és középre igazítást.
 - A profilkép legyen 150px széles és kerek (`border-radius: 50%`).
 - A hobbi lista elemei legyenek dőlt betűsek.
 - A linkek (`<a>`) alapértelmezésben legyenek kék, hover állapotban pirosak.
-

Feladat: Recept oldal + CSS

HTML:

- Cím, leírás, hozzávalók ``, lépések ``, és tápérték táblázat.

CSS:

- A cím (`<h1>`) legyen középre igazítva, nagyobb betűmérettel (pl. `font-size: 2em;`).
 - A hozzávalók listája (``) kapjon belső margót (`padding`) és egy halvány színű háttért.
 - A táblázat:
 - Legyen szegély minden cella körül (`border: 1px solid #ccc;`).
 - A fejléc (`<th>`) legyen félkövér és sötétebb háttérrel.
 - A sorok között legyen minimális térköz.
 - A lépések (``) számozása legyen római számok (`list-style-type: upper-roman;`).
 - A recept oldal háttere legyen világos krémszínű.
-

Feladat: 3 oldalas webhely + CSS

HTML alap:

- Három oldal (Kezdőlap, Rólunk, Kapcsolat) összekötve navigációs menüvel.

CSS követelmények:

- A navigációs menü (<nav>) legyen vízszintes sáv, világos háttérrel és középre igazított linkekkel.
- A linkek legyenek aláhúzás nélkül alapértelmezésben, hover esetén legyenek aláhúzva és más színűek.
- Az aktív oldal linkje legyen félkövér (font-weight: bold;) és más színű (pl. piros).
- A tartalom (<main>) legyen max. 800px széles, középre igazítva (margin: auto;).
- A lábléc (<footer>) legyen kisebb betűméretű és középre igazított szürke szöveggel.

JavaScript

Feladat: Üdvözlő üzenet gombnyomásra

- Készítsünk egy HTML oldalt egy gombbal: „Kattints rám!”.
 - Ha a felhasználó rákattint a gombra, jelenjen meg egy `alert()` üzenet: „Szia, üdvözlő a JavaScript!”.
-

Feladat: Összeadás két szám között

- Hozzunk létre két `<input type="number">` mezőt és egy „Számold ki” gombot.
 - Ha a gombra kattintunk, a két szám összege jelenjen meg egy `<p>` elemben az oldalon.
 - Használjuk `document.getElementById()`-t az értékek kiolvasásához.
-

Feladat: Háttérszín váltása gombnyomásra

- Készítsünk egy gombot: „Színváltás”.
 - Ha a felhasználó rákattint, az oldal háttere váltson **véletlenszerű színre**.
 - Tipp: Használjuk a `Math.random()`-ot és `Math.floor()`-t RGB érték generálásához.
-

Feladat: Szöveg megfordítása

- Készítsünk egy `<input type="text">` mezőt és egy gombot: „Fordítsd meg!”.
 - Ha a gombra kattintunk, írja ki az input szövegének **megfordított** változatát egy `<p>`-be.
 - Tipp: Szöveg megfordítása: `split('').reverse().join('')`.
-

Feladat: Számláló gomb

- Készítsünk egy gombot „+1” felirattal és egy `<p>` elemet, ami kezdetben 0-t mutat.
 - Minden kattintásnál a szám eggyel növekedjen.
 - **Extra:** Adjunk hozzá egy másik gombot „Reset” funkcióval.
-

Feladat: Lista bővítése dinamikusan

- Hozzunk létre egy üres `` listát és egy input mezőt + gombot: „Hozzáadás”.
- Ha a felhasználó beír egy szót és rákattint a gombra, az új elem kerüljön be a listába.

- **Extra:** Ha az input üres, ne történjen semmi, és írja ki: „Adj meg szöveget!” (pl. egy `<p>`-ben).

Feladat: Véletlenszám generátor

- Készítsünk egy input mezőt két számmal (min és max) és egy gombot: „Generálj számot!”.
 - Ha rákattintunk a gombra, jelenítsen meg egy **véletlen egész számot** a megadott tartományban.
 - Tipp: `Math.floor(Math.random() * (max - min + 1)) + min`.
-

Feladat: Egyszerű óra (valós idő)

- Készítsünk egy `<p>` elemet, ami folyamatosan mutatja az aktuális időt (HH:MM:SS).
 - Frissüljön minden másodpercben.
 - Tipp: `new Date()` és `setInterval()`.
-

Feladat: Karakterek számolása (input)

- Készítsünk egy szövegbeviteli mezőt, alatta egy `<p>`-et: „0 karakter”.
 - Ahogy a felhasználó gépel, frissítsük a karakterek számát.
 - **Extra:** Ha több mint 50 karakter, írjuk ki: „Túl hosszú!” piros szöveggel.
-

Feladat: Rejtett szöveg megjelenítése/elrejtése

- Legyen egy szöveg (pl. „Titkos üzenet”) és egy gomb: „Mutasd / Rejtsd”.
 - A gomb megnyomásával a szöveg váltson a látható és rejtett állapot között.
 - Tipp: Használjuk az alábbi beállítást: `style.display = 'none' / 'block'`.
-

Feladat: Képváltó galéria

- Legyen egy kép (``) és két gomb: „Előző” és „Következő”.
 - Hozzunk létre egy tömböt 3-4 képpel (URL-lel).
 - A gombokkal lehessen váltogatni a képeket előre-hátra.
 - Tipp: Kezeljük az elején és a végén a körbefordulást.
-

Feladat: Egyszerű kvíz játék

- Készítsünk egy kvíz kérdéseket pl. 3 válaszlehetőséggel (rádiógombok).

- Legyen egy „Ellenőrzés” gomb, ami kiírja: „Helyes!” vagy „Helytelen!” a választól függően.
- **Extra:** Színezzük zöldre a helyes választ és pirosra a rosszat.

Feladat: Szám kitalálós játék

- Generálj egy véletlen számot 1 és 100 között.
 - A felhasználó adjon meg egy tippet egy input mezőben, és kattintson az „Ellenőriz” gombra.
 - Írd ki a program, hogy a szám **kisebb** vagy **nagyobb** a tippnél.
 - Ha eltalálta, írd ki: „Gratulálok, eltaláltad!”.
-

Feladat: Szorzótábla generátor

- Készíts egy input mezőt, amibe a felhasználó beír egy számot (pl. 5).
 - Ha kattint egy gombra, írd ki az adott szám szorzótábláját 1-től 10-ig egy `` listába.
-

Feladat: Szöveg váltása kattintással

- Készíts egy `<p>` elemet „Helló, világ!” szöveggel és egy gombot „Váltás”.
 - Ha kattintanak, a szöveg változzon „Szia, JavaScript!”-re, újabb kattintásra vissza.
-

Feladat: Dinamikus számológép (összeadás, kivonás)

- Készíts két input mezőt és egy `<select>` elemet két opcióval: „+” és „-”.
 - Gombnyomásra számold ki az eredményt, és írd ki a `<p>`-be.
-

Feladat: Visszaszámláló időzítő

- Készíts egy input mezőt, ahova a felhasználó megad másodperceket (pl. 10).
 - Ha kattint a „Start” gombra, induljon el egy visszaszámlálás, és minden másodpercben írd ki a hátralévő időt.
 - Ha eléri a nullát, írd ki: „Idő lejárt!”.
-

Feladat: Kép megjelenítése input alapján

- Készíts egy input mezőt, ahova a felhasználó beírja egy állat nevét (pl. „kutya”, „macska”).

- Ha rákattint a „Mutasd” gombra, jelenjen meg egy kép az adott állatról (``), ha ismeretlen, írd ki: „Nincs találat”.

Feladat: Kötelező mezők ellenőrzése

- Készíts egy egyszerű űrlapot: név, email, jelszó mezőkkel.
 - Ha a felhasználó a **Küldés** gombra kattint, ellenőrizze a JavaScript, hogy minden mező ki van-e töltve.
 - Ha valamelyik üres, jelenjen meg egy `alert()`: „Töltsd ki az összes mezőt!”.
 - **Extra:** Ha minden mező ki van töltve, írd ki egy `<p>`-be: „Űrlap sikeresen elküldve!”.
-

Feladat: Email formátum ellenőrzése

- Legyen egy email mező és egy **Küldés** gomb.
 - Ha a felhasználó rákattint, ellenőrizd, hogy az email **tartalmaz-e @ jelet és pontot** a formátum szerint.
 - Ha nem megfelelő, jelenjen meg egy `` piros szöveggel: „Hibás email cím!”.
 - Ha jó a formátum, ``-ben zöld szöveg: „Email OK!”.
-

Feladat: Jelszó és jelszó megerősítés ellenőrzése

- Készíts két jelszó mezőt: Jelszó és Jelszó újra.
- A **Küldés** gomb megnyomásakor ellenőrizze a program, hogy a két mező értéke megegyezik-e.
- Ha nem egyezik, jelenjen meg piros `` üzenet: „A jelszavak nem egyeznek!”.
- Ha egyezik, jelenjen meg zöld ``: „Jelszavak egyeznek!”.

jQuery

Feladat: Szöveg elrejtése és megjelenítése

- Legyen egy bekezdés és két gomb: **Elrejt** és **Megjelenít**.
 - A gombok kattintására a szöveg elrejtődjön vagy megjelenjen jQuery segítségével.
-

Feladat: Szöveg színének váltása kattintásra

- Legyen egy `<p>` szöveg és egy gomb.
 - Ha rákattintunk a gombra, a szöveg színe pirosra változik.
 - **Extra:** Még egy gomb, ami visszaállítja az eredeti színt.
-

Feladat: Listaelemek kiemelése hover eseménnyel

- Legyen egy `` lista 5 elemmel.
 - Ha az egér egy elem fölé megy, változzon meg a háttérszíne (pl. sárgára), és ha elhagyja, térjen vissza az eredetire.
-

Feladat: Kattintásra eltűnő elem animációval

- Legyen egy kép és egy gomb „Eltüntetés”.
 - Ha a gombra kattintunk, a kép lassan eltűnjön (`fadeOut()`), újabb gombnyomásra visszajöjjön (`fadeIn()`).
-

Feladat: Form input kiolvasása és kiírása

Cél: `val()` és `text()` használata

- Legyen egy input mező és egy gomb „Kiírás”.
 - Ha kattintunk a gombra, írd ki az input tartalmát egy `<p>`-be.
-

Feladat: Dinamikus lista bővítése

Cél: `append()` használata

- Legyen egy input mező és egy gomb „Hozzáadás”.
 - Ha kattintunk a gombra, az input értéke kerüljön hozzá egy `` listához új `` elemként.
-

Feladat: Kép csere kattintásra

Cél: `attr()` használata

- Legyen egy `` és egy gomb „Kép váltása”.
 - Ha a gombra kattintunk, a kép `src` attribútuma változzon egy másik URL-re.
-

Feladat: Számláló gomb

- Legyen egy `<p>` ami 0-val indul, és egy gomb „+1”.
 - Minden kattintáskor a szám nőjön 1-gyel (`text()` függvény segítségével).
-

Feladat: Összes checkbox kijelölése és törlése

Cél: `prop()` használata

- Legyen 5 darab checkbox és két gomb: **Mind kijelöl** és **Mind töröl**.
 - Az első kattintásra minden be legyen jelölve, a másodikra minden ki legyen kapcsolva.
-

Feladat: Elemek sorrendjének felcserélése

Cél: DOM manipuláció `prepend()` és `append()` segítségével

- Legyen egy lista 3 elemmel: „Első”, „Második”, „Harmadik”.
 - Gombnyomásra a harmadik elem menjen az első helyre.
-

Feladat: Animált doboz mozgatása

Cél: `animate()` használata

- Legyen egy 100x100 px-es div, piros háttérrel.
- Gombnyomásra a doboz mozduljon jobbra 200px-re.
- **Extra:** Készítsük el a animációt is.

Feladat: Kötelező mezők ellenőrzése

Cél: `required`, `if` feltételek, `alert()` használata

- Készítsünk egy egyszerű űrlapot: név, email, jelszó mezőkkel.
 - Ha a felhasználó a **Küldés** gombra kattint, ellenőrizzük, hogy minden mező ki van-e töltve.
 - Ha valamelyik üres, jelenjen meg egy `alert()`: „Töltsd ki az összes mezőt!”.
 - **Extra:** Ha minden mező ki van töltve, írjuk ki egy `<p>`-be: „Űrlap sikeresen elküldve!”.
-

Feladat: Email formátum ellenőrzése

Cél: `Regex` és `test()` metódus használata

- Legyen egy email mező és egy **Küldés** gomb.
 - Ha a felhasználó rákattint a gombra, ellenőrizze, hogy az email **tartalmaz-e @ jelet és pontot** a formátum szerint.
 - Ha nem megfelelő, jelenjen meg egy `` piros szöveggel: „Hibás email cím!”.
 - Ha jó a formátum, ``-ben zöld szöveg: „Email OK!”.
-

Feladat: Jelszó és jelszó megerősítés ellenőrzése

- Készítsünk két jelszó mezőt: Jelszó és Jelszó újra.
- A **Küldés** gomb megnyomásakor ellenőrizzük, hogy a két mező értéke megegyezik-e.
- Ha nem egyezik, jelenjen meg piros `` üzenet: „A jelszavak nem egyeznek!”.
- Ha egyezik, jelenjen meg zöld ``: „Jelszavak egyeznek!”.

JSON

Feladat: Alap JSON objektum létrehozása

- Hozzunk létre egy JSON objektumot `szemely` néven a következő mezőkkel:
 - `nev: „Kovács Anna”`
 - `eletkor: 25`
 - `hobbi: ["olvasás", "futás", "programozás"]`
 - Írjuk ki az objektumot a konzolra.
-

Feladat: Tömb JSON objektumokkal

- Hozzunk létre egy JSON tömböt `diakok` néven, ami 3 diákot tartalmaz.
 - Minden diák egy objektum a következő mezőkkel: `nev`, `osztaly`, `jegyek` (tömb 3 számot tartalmazzon).
 - Írjuk ki a második diák nevét és jegyeit a konzolra.
-

Feladat: JSON beolvasása és módosítása JavaScript-ben

- Legyen egy JSON string:

```
'{"nev": "Kovács Anna", "eletkor": 25, "hobbi": ["olvasás", "futás"]}'
```

- Alakítsuk át JavaScript objektummá (`JSON.parse()`).
 - Adjuk hozzá a hobbi tömbhöz a `"programozás"` elemet.
 - Alakítsuk vissza JSON stringgé (`JSON.stringify()`) és írjuk ki a konzolra.
-

Feladat: Nested (beágyazott) JSON

- Hozzunk létre egy JSON objektumot `auto` néven a következő elemekkel:
 - `marka: „Hyundai”`
 - `tipus: „i20”`
 - `tulajdonos: objektum {nev: "Kovács Anna", eletkor: 25}`
 - Írjuk ki a tulajdonos nevét a konzolra.
 - Adjuk hozzá a tulajdonoshoz az `rendszam` mezőt: `„ABC-123”`.
-

Feladat: JSON tömb szűrése

Cél: Tömb és objektumok kombinálása, `filter()`.

- Hozzunk létre egy JSON tömböt `termekek` néven, 5 termékkel: minden termék objektum, az alábbi mezőkkel: `nev`, `ar`, `raktaron` (true/false).
- Írjuk ki a konzolra csak azokat a termékeket, amelyek **raktáron vannak**.
- **Extra:** Írjuk ki a raktáron lévő termékek nevét külön listába.