# Alapfogalmak megismerése

## Protokol

Ha valamit közölni akarunk egy másik személlyel a körülöttünk lévő világ minden részére szavakat – kódokat – használunk. A számítógépes kommunikációban ugyanúgy, mint a beszédben a továbbítandó mondanivalót kódolnunk kell. Ezeket a kódolási szabályokat nevezzük az informatikában protokolloknak.

### HTTP, HTTPS

Az internet leggyakrabban használt protokolljai. Multimédiás dokumentumok leírására használják az 1990-es évek elejétől. Úgy lett megalkotva, hogy a számítógép hardverétől és operációs rendszerétől függetlenül ugyanabban az elrendezésben tudja megjeleníteni a szöveget, képet, mozgóképet és hangokat tartalmazható dokumentumokat, mint ahogyan a készítő meghatározta. Minden internetes dokumentumnak rendelkeznie kell egyedi azonosítóval, ezt nevezzük **URI** *(Uniform Resource Identifier-nek, vagyis egységes erőforrás-azonosító*-nak). A **H**yper**T**ext a dokumentumnak azon szöveges részei, amelyek egy másik internetes dokumentumra való utalást tartalmaznak.

## Portok

Minden kommunikáció kódolás – továbbítás – dekódolás lépésekből áll. Mivel nagyon sok kódolási forma – protokoll – használatos, ezért a fogadó eszközt „tolmáccsal” kell ellátni. Egy tolmács csak egyetlen egy nyelven tud beszélni. Ezek a „tolmácsok” csak egy adott kapun keresztül érhetőek el, amelyeket 0 és 65 535 közötti számmal azonosítanak. A HTTP „tolmácsa” alapértelmezetten a 80-as kapun, a HTTPS „tolmácsa” a 443-as kapun várja a dekódolandó üzeneteket.

## Domain nevek

Az URI (*Uniform Resource Identifier, vagyis egységes erőforrás-azonosító*) karakterlánc, amelyet webes erőforrások azonosítására és elérésére használnak leggyakrabban URL-ek (*Uniform Resource Locator*) formájában jelenik meg, azaz webcímként.

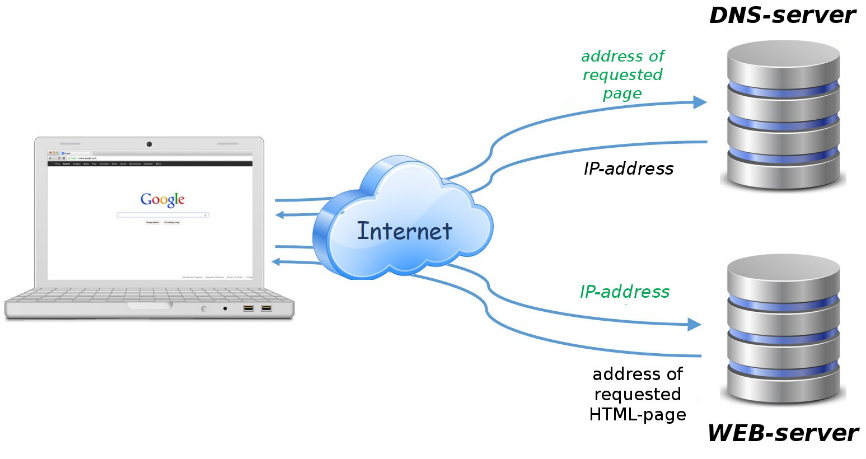
Például: https://hu.wikipedia.org/wiki/URI.

## Böngészők

A böngészők olyan szoftverek, amelyek alkalmasak arra, hogy az internetes dokumentumokat, vagyis a weblapokat a készítő szándékának megfelelő formában tudják megjeleníteni. Ilyen programból nagyon sok készült. A jelenleg legnépszerűbbek: Google Chrome, Safari, Microsoft Edge, Mozilla Firefox, Brave és Vivaldi.

## kliens-szerver modell

Ha valaki meg akar jelenni az interneten, akkor kódolja a HTML5 szabályai szerint a dokumentumát és keres egy olyan szolgáltatót, amelyen bármelyik felhasználó elérheti. Itt kap egy azonosítót, egy URL-t. Ennek az URL-nek a megadásával a böngésző lekéri a dokumentumot és megjeleníti a weblapot.



1. ábra Hogyan működik az internet?

Aki olvassa a dokumentumot az a **kliens**, ahol elérhető az a **szerver**.

### Hosting (*Tárhelyszolgáltatás / Webszolgáltatás*)

A **hosting** (*tárhelyszolgáltatás*) az a szolgáltatás, amelyet egy hosting cég nyújt, lehetővé téve magánszemélyek vagy szervezetek számára, hogy weboldalaikat vagy webes alkalmazásaikat az interneten elérhetővé tegyék. Lényegében a hosting szolgáltatók szervereket adnak bérbe, amelyekre feltöltheted a weboldalad fájljait, adatbázisait és egyéb szükséges elemeit.

# Eszközök és fejlesztési környezet

* Kódeditor használata (pl. VS Code, https://code.visualstudio.com/)
* Böngészőfejlesztői eszközök (DevTools)
* Verziókezelés: Git és GitHub alapjai

# HTML – a weboldal váza

## HTML alapstruktúra

<!**DOCTYPE** html>

<**html** lang=**"en"**>

<**head**>

<meta charset=**"UTF-8"**>

<meta name=**"viewport"** content=**"width=device-width, initial-scale=1.0"**>

<title>**Document**</title>

<link rel=**"stylesheet"** href=**"harrmadik.css"**>

</head>

<**body**>

<p>**Közlendő szöveg**</p>

</body>

</html>

## Szöveges tartalmak

### Címek

### Bekezdések

### Listák

### Képek

### Linkek

### Táblázatok

### Űrlapok

* HTML5 szemantikus elemek (header, nav, section, article, footer)

# CSS – a megjelenés kialakítása

## Alap szintaxis és szabályok

body **{**

background-color: rgb(235, 235, 104);

font-family: Arial, sans-serif;

}

szelektor **{**

tulajdonság1: érték1;

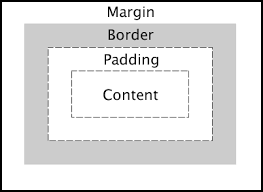
tulajdonság2: érték2;

}

## Színek

## Betűk

## Box model



2. ábra Box model

margók, padding, border

## Osztályok és azonosítók használata

## Elhelyezés: Flexbox és Grid

## Media query-k (reszponzív design alapjai)

# Gyakorlás – egyszerű weboldal készítése

* Statikus oldalak készítése HTML + CSS segítségével
* Reszponzív (mobilbarát) elrendezés készítése

# JavaScript – interaktivitás hozzáadása

* Alapfogalmak: változók, ciklusok, feltételek, függvények
* DOM manipuláció: elemek kiválasztása, eseménykezelés
* Űrlapellenőrzés
* Egyszerű animációk, események

# Modern front-end technológiák (alapszint)

* Bevezetés a CSS preprocesszorokba (pl. SCSS)
* JavaScript keretrendszerek (React, Vue, stb.)
* Build eszközök (pl. Vite, Webpack – opcionálisan)

# Weboldal publikálása

* Ingyenes hosting lehetőségek: GitHub Pages, Netlify
* Fájlok feltöltése, hibakeresés, végső tesztelés

# További tanulási irányok

* Backend (pl. Node.js, PHP, Python Flask/Django)
* Adatbázis-kezelés (SQL, MongoDB)
* API-k, REST, JSON használata