

Számos webfejlesztő számára, aki csak HTML, CSS és JavaScript ismerettel rendelkezik, a natív mobilalkalmazás fejlesztése ismeretlen terület lehet. Technikailag az Android, az iOS és a Windows Phone mobilalkalmazások teljesen különböző programozási nyelvek használatával készülnek. Egy Android alkalmazás Java-t használ, az iOS-alkalmazás az Objective-C programot használja, míg a Windows Phone alkalmazás a .NET-t használja.

De manapság bárki, aki tisztességes ismeretekkel rendelkezik a HTML, a CSS és a JavaScript használatáról, képes mobilalkalmazást létrehozni. A webes technológia használatának egyik legfontosabb előnye az alkalmazás felépítéséhez a hordozhatóság. Egy csomagoló / fordító, például a PhoneGap használatával képes lesz hordozni és telepíteni az alkalmazást számos különböző platformon.

A Hypertext Markup Language (HTML) a weboldalak és webalkalmazások létrehozásához szükséges szabványos jelölőnyelv. A HTML beágyazhat olyan programokat, amelyek egy szkriptnyelvben vannak írva, mint például a JavaScript, amelyek befolyásolják a weboldalak viselkedését és tartalmát. A CSS beillesztése határozza meg a tartalom megjelenését és elrendezését. A World Wide Web Consortium (W3C), mind a HTML, mind a CSS szabványok karbantartója 1997 óta ösztönzi a CSS használatát az explicit prezentációs HTML-en keresztül.

A HTML verziói:

- **1995. november 24.** - A HTML 2.0- IETF RF 1866-ként tették közzé. Lehetőségként RFC-kel kiegészítve:
 - 1995. november 25.: RFC 1867 (űrlap-alapú fájlfeltöltés)
 - 1996. május: RFC 1942 (táblázatok)
 - 1996 augusztus: RFC 1980 (kliensoldali kép térképek)
 - 1997 január: RFC 2070 (nemzetközivé válás)
- **1997. január 14.** - A HTML 3.2 W3C Recommendation-ként jelent meg. Ez volt az első olyan verzió, amelyet a W3C fejlesztett ki és szabványosított, mivel az IETF 1996. szeptember 12.-én felszámolta a HTML munkacsoportját. A HTML 3.2 teljes mértékben lemondta a matematikai képleteket, összeegyeztette az átfedéseket a különféle kiterjesztések között és elfogadta a Netscape vizuális jelölő címkéinek nagy részét. A matematikai képletekhez szükséges jelölés a HTML-ben nem volt szabványosított, csak 14 hónappal később a MathML-ben.
- **1997. december 18.** - A HTML 4.0 W3C Recommendation-ként jelent meg. Három variációt kínál:
 - Szigorú, ahol az elavult elemek tiltottak
 - Átmeneti, melyben az elavult elemek tiltottak
 - Keretrendszer, melyben leginkább csak kerethez kapcsolódó elemek

megengedettek

A HTML 4.0 számos böngészőspecifikus elemtípust és attribútumot fogadott el, ugyanakkor a Netscape vizuális jelölési funkcióinak fokozatos megszüntetésére törekedett a stíluslapok elutasításával. A HTML 4 egy SGML alkalmazás, amely megfelel az ISO 8879 – SGML szabványnak.

- **1998. április 24.** - A HTML 4.0 új verzióját adták ki kisebb változásokkal anélkül, hogy növelték volna a verziószámot.
- **1999. december 24.** - A HTML 4.01 W3C Recommendation-ként jelent meg. Ugyanazt a három változatot kínálja, mint a HTML 4.0 és az utolsó hibajegyzék 2001. május 12.-én jelentek meg.
- **2000. május** – ISO/IEC 15445: 2000 („ISO HTML”, a HTML 4.01 Strict alapján) ISO/IEC nemzetközi szabványként tették közzé. Az ISO-ban ez a szabvány az ISO/IEC JTC1/SC34 (ISO/IES Közös Technikai Bizottság 1, 34. albizottság – Dokumentumleírás és feldolgozási nyelvek) tartományába tartozik. A HTML 4.01 után nem volt új HTML verzió sok éven át, hiszen a párhuzamos, XML alapú XHTML nyelvek fejlesztése a 2000-es évek elején és közepén munkát adott a W3C HTML munkacsoportjának.
- **2014. október 28.** - A HTML 5 W3C Recommendation-ként jelent meg.
- **2016. november 1.** - A HTML 5.1 W3C Recommendation-ként jelent meg.

A JavaScript, gyakran JS-ként rövidítve, egy magas szintű, dinamikus, típusatlan, objektum-alapú, több paradigmás és értelmezett programozási nyelv. A HTML és a CSS mellett a JavaScript egyike a világhálós tartalomkészítés három fő technológiájának. A webhelyek többsége ezt alkalmazza és minden modern webböngésző beépített JavaScript-eszközzel támogatja anélkül, hogy plugin-ekre lenne szüksége. A JavaScript-eszközök mindegyike a JavaScript eltérő végrehajtását jelképezi, mindez az ECMAScript specifikáción alapul, mivel egyes eszközök nem támogatják a specifikációt és számos eszköz támogatja az ECMA-n túli további funkciókat.

A JavaScript-nek 3 fő típusa van:

- Kliensoldali JavaScript (CSJS) - a JavaScript kiterjesztett változata, amely lehetővé teszi weboldalak és kliensböngészők fejlesztését és manipulálását.
- Szerveroldali JavaScript (SSJS) - a JavaScript kiterjesztett változata, amely lehetővé teszi az adatbázisokhoz, fájlrendszerekhez és szerverekhez való back-end hozzáférést
- Core JavaScript - az alap JavaScript nyelv

A kliensoldali JavaScript (CSJS) és a szerveroldali JavaScript (SSJS) függ a JavaScript alapjától, és nem tud működni nélküle.

Kliensoldali JavaScriptként a Vue.js-t használtam az alkalmazásom fejlesztése során.

A Vue.js egy nyílt forráskódú, progresszív JavaScript keretrendszer a felhasználói felületek felépítéséhez. Az olyan projektekbe való integráció, melyek más JavaScript könyvtárakat használnak, könnyebbé válik a Vue-val, mivel ez úgy van kialakítva, hogy fokozatosan alkalmazható legyen. A Vue egy olyan webes alkalmazási keretként is működhet, mely fejlett, egyoldalas alkalmazásokra képes.

A Vue.js egy olyan eszköz, amely kihasználja az MVVM architektúra használatát.

A projekt az internetes felhasználói felület fejlesztés (komponensek, deklaratív felhasználói felületek, közvetlen újratöltés, idő-utazásos hibakeresés, stb.) ötletesebb megközelítésére fókuszál. A meglévő projektekhez egyszerű interaktivitást adhatunk vagy akár jQuery és más sablonnyelveket és keretrendszereket teljesen mértékben helyettesíthetünk.

A Vue.js egy JavaScript könyvtár a felhasználói interfészek (nézetek) létrehozásához alapul szolgáló adatmodellek alapján. Ahelyett, hogy csak a DOM-ban lévő felhasználói adatokat frissítené, egy adatmodell szinkronban fog megjelenni a változások tükrében. A Vue a reaktív adatmegkötésen keresztül működik. Ez azt jelenti, hogy például egy űrlapmező frissítése után az alapul szolgáló adatmodell automatikusan frissül. Ha a weboldal egyéb része is ugyanazon adatmodellhez van kötve, annak tartalma azonnal frissítésre kerül.

A Vue-t Evan You hozta létre 2014-ben, miután a Google-nél az AngularJs-en dolgozott. Később összefoglalta gondolkodási folyamatát: „Rájöttem, hogy mi lenne, ha kivonhatnám csak azt a részét, amelyre igazán szerettem volna használni az Angular-t és valami nagyon könnyűt csinálnék minden további elképzelés nélkül?”