Programozási nyelvek

*Ebben a dokumentumban híresebb programozási*

*nyelvekről lesz szó. A project célja, hogy kicsit jobban megismerkedjetek a benne szereplő nyelvekkel.*

*Egyaránt szó lesz a régi, manapság nem használt és a jelenleg legelterjedtebb nyelvekről is.*



*Reméljük hasznos lesz és tetszeni fog!*

Python

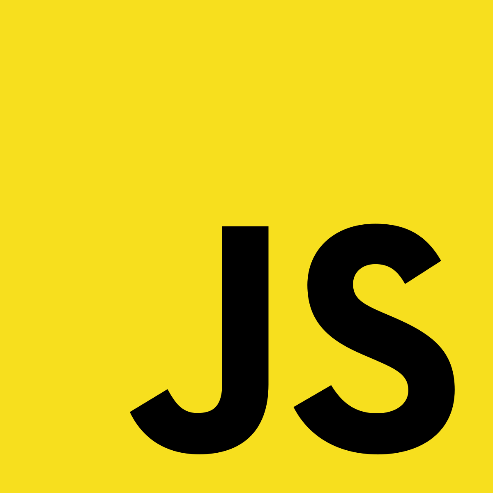
[**Tervező**](https://www.google.com/search?rlz=1C1GCEU_enHU970HU970&q=python+tervez%C5%91&stick=H4sIAAAAAAAAAOPgE-LQz9U3MK0yjNcyyCi30k_Oz8lJTS7JzM_TLyjKTy9KzM3NzEuPz0nMSy9NTE8ttkpJLc5Mz0tNUUiqXMTKX1BZkpGfp1CSWlSWWnV0IgBRujUuUAAAAA&sa=X&ved=2ahUKEwjB_vbP6vPyAhUAhf0HHR67BWMQ6BMoAHoECDQQAg)**:**[Guido van Rossum](https://www.google.com/search?rlz=1C1GCEU_enHU970HU970&q=Guido+van+Rossum&stick=H4sIAAAAAAAAAOPgE-LQz9U3MK0yjFfiBLEMMwxMk7UMMsqt9JPzc3JSk0sy8_P0C4ry04sSc3Mz89LjcxLz0ksT01OLrVJSizPT81JTFJIqF7EKuJdmpuQrlCXmKQTlFxeX5u5gZQQAviYLF18AAAA&sa=X&ved=2ahUKEwjB_vbP6vPyAhUAhf0HHR67BWMQmxMoAXoECDQQAw)

*Története:* A Python alapötlete az 1980-as évek végén született meg. A fejlesztést 1989 decemberében kezdte el Guido van Rossum a CWI-n. A CWI (**C**entrum **W**iskunde & **I**nformatica, magyarul Matematikai és Informatikai Központ) egy kutatóintézet Amszterdamban. A nyelv a nevét a Monthy Python csoportról kapta. 1991 februárjában jelent meg az első nyilvános változat (0.9.0 verzió néven) az alt.sources hírcsoportban. 1994-ben jött létre a comp lang phyton hírcsoport, ami egy jelentős mérföldkő volt a nyelv fejlődésében. Szintén 1994-ben látott napvilágot az 1.0 verzió, amit az ezredfordulón, 2000 októberében követett a Python 2.0, majd pedig 2008-ban a Python 3.0.

*Python-ra épülő alkalmazások:*

* a [Zope](http://www.zope.org/) alkalmazáskiszolgáló
* ill. a Zope-ra épülő [Plone](http://plone.org/) tartalomkezelő-rendszer ([CMS](https://hu.wikipedia.org/wiki/Tartalomkezel%C5%91_rendszer))
* [Mailman](http://www.gnu.org/software/mailman) levelezési listakezelő
* [Trac](http://trac.edgewall.com/) projekt-kezelő rendszer
* a kéretlen reklámleveleket kiszűrő [SpamBayes](http://spambayes.sourceforge.net/)
* az egyik leghíresebb fájlcserélő szoftver, a [BitTorrent](https://hu.wikipedia.org/wiki/BitTorrent" \o "BitTorrent) eredeti implementációja
* A [Blender](http://www.blender.org/) 3D modellező-animációs program, ami ezért GE-ként is működik, Pythonból scriptelhető

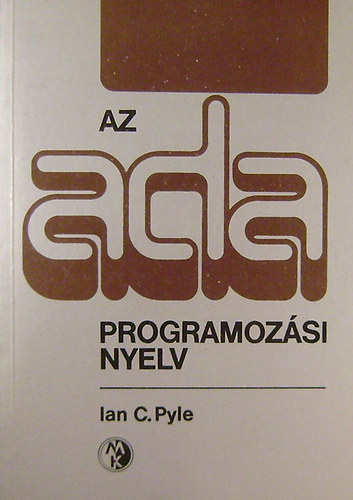
JavaScript

[**Tervező**](https://www.google.com/search?rlz=1C1GCEU_enHU970HU970&q=javascript+tervez%C5%91&stick=H4sIAAAAAAAAAOPgE-LQz9U3MCqwNNcyyCi30k_Oz8lJTS7JzM_TLyjKTy9KzM3NzEuPz0nMSy9NTE8ttkpJLc5Mz0tNUUiqXMQqnJVYllicXJRZUKJQklpUllp1dCIAfT6iKVQAAAA&sa=X&ved=2ahUKEwiZ4Pn96vPyAhUt_rsIHcaDAK8Q6BMoAHoECDQQAg)**:**[Netscape](https://www.google.com/search?rlz=1C1GCEU_enHU970HU970&q=Netscape&stick=H4sIAAAAAAAAAOPgE-LQz9U3MCqwNFcCs0yzjMu0DDLKrfST83NyUpNLMvPz9AuK8tOLEnNzM_PS43MS89JLE9NTi61SUosz0_NSUxSSKhexcvillhQnJxak7mBlBAABeaRSVgAAAA&sa=X&ved=2ahUKEwiZ4Pn96vPyAhUt_rsIHcaDAK8QmxMoAXoECDQQAw), [Brendan Eich](https://www.google.com/search?rlz=1C1GCEU_enHU970HU970&q=Brendan+Eich&stick=H4sIAAAAAAAAAOPgE-LQz9U3MCqwNFfiBLFMjI2KLbUMMsqt9JPzc3JSk0sy8_P0C4ry04sSc3Mz89LjcxLz0ksT01OLrVJSizPT81JTFJIqF7HyOBWl5qUk5im4ZiZn7GBlBAB5-llAWwAAAA&sa=X&ved=2ahUKEwiZ4Pn96vPyAhUt_rsIHcaDAK8QmxMoAnoECDQQBA)

*Kialakulása:* A JavaScript, mint neve is mutatja, egy script nyelv, amit legelőször a Netscape Navigator 2.0 támogatott. A Javascript a WWW-ért született. Jelenleg nem is létezik más implementációja, mint ami a böngészőkben és a webszerverekben fut. A Sun és a Netscape által kifejlesztett parancsnyelvnek két változata létezik. Az ún. LiveWire lehetővé teszi, hogy szerver oldali alkalmazásokat készítsünk, ami hasonló a CGi-hez (Common Gateway interface). A másik a kliens oldali, ami lehetővé teszi egyszerűbb programozási feladatok elkészítését a Weben (pl. mozgó HTML elemek, interaktív funkciók). A kliens és a szerver oldali rész egy külön világ, szintakszisuk ugyan megegyezik, de a programozói interfész teljesen más. A kliens oldali JavaScript a böngészőkön keresztül jelenik meg számunkra. Mégpedig úgy, hogy a JavaScript kódot az általános HTML kód közé ékeljük. Amikor a böngésző megjeleníti a HTML kódot, akkor a <script> és </script> közé írt kódrészleteket nem megjeleníti, hanem értelmezi, és futtatja, ezáltal dinamikus viselkedést kölcsönöz az oldalnak.

|  |  |
| --- | --- |
| **JavaScript** | **Java** |
| A kliens értelmezi, nem fordítja a kódot. | A már lefordított kódot tölti le a kliens, majd futtatja. |
| Objektum-alapú. Csak egyedi objektumokat lehet létrehozni, kiegészíteni és kezelni, osztályokat nem. | Objektum-orientált. A programok osztályokból állnak, melyeket példányosítani lehet. Osztályokat lehet származtatni. |
| A kód a HTML-be integrált, a kliens onnan olvassa ki, és értelmezi. | A programba a HTML-ből hivatkozunk, de a tényleges program attól külön található. |
| Gyenge típusosság. A változók típusát az aktuális értéke határozza meg. | Erős típusosság. Minden változó meghatározott típussal rendelkezik. |

Ada



[**Tervező**](https://www.google.com/search?rlz=1C1GCEU_enHU970HU970&q=javascript+tervez%C5%91&stick=H4sIAAAAAAAAAOPgE-LQz9U3MCqwNNcyyCi30k_Oz8lJTS7JzM_TLyjKTy9KzM3NzEuPz0nMSy9NTE8ttkpJLc5Mz0tNUUiqXMQqnJVYllicXJRZUKJQklpUllp1dCIAfT6iKVQAAAA&sa=X&ved=2ahUKEwiZ4Pn96vPyAhUt_rsIHcaDAK8Q6BMoAHoECDQQAg)**:**Jean Ichbia

Az **Ada** egy strukturált, típusos, blok kszerkezetes, imperatív, [általános célú](https://hu.wikipedia.org/wiki/%C3%81ltal%C3%A1nos_c%C3%A9l%C3%BA_programoz%C3%A1si_nyelv) [programozási nyelv](https://hu.wikipedia.org/wiki/Programoz%C3%A1si_nyelv), amelyet a Jean Ichbiah (a francia illetőségű CII Honeywell Bull munkatársa) vezette csapat alkotott meg 1977–1983 között. Nem hasonlít a C illetve C++ nyelvekre, és az egyik legbiztonságosabb, típusokon alapuló programozási nyelv.

*Leírása:* Az Ada egy strukturált [programozási nyelv](https://hu.wikipedia.org/wiki/Programoz%C3%A1si_nyelv) sok [objektumorientált](https://hu.wikipedia.org/wiki/Objektumorient%C3%A1lt) lehetőséggel. Eredetileg az [Amerikai Egyesült Államok](https://hu.wikipedia.org/wiki/Amerikai_Egyes%C3%BClt_%C3%81llamok) Védelmi Minisztériuma számára fejlesztették ki azzal a céllal, hogy elősegítse nagyméretű programok létrehozását, azok stabilitásának növelését. Néhány tulajdonsága a teljesség igénye nélkül: erős típusok, multitaszkolás, általános típusok, hibakezelés, absztrakt adattípusok.

Az Ada által támogatott egyszerű adattípusok között van többféle szám-típus, igaz-hamis típusok, karakterek, hivatkozások és felsorolások. A tömbök, rekordok (struktúrák) és karakterláncok az Ada összetett típusai. A biztonságon levő hangsúllyal nem meglepő, hogy az Ada egy erősen típusos nyelv: minden adatelemet előre definiálni kell egy bizonyos típusú vagy altípusúként, valamint folyamatos a típusellenőrzés a modulok között és azokon belül is.

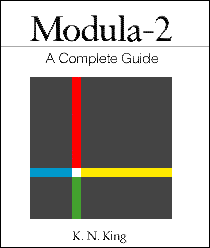
Teljes körű támogatást nyújt a szekvenciális vezérlőstruktúrákhoz, valamint a minden részletre kiterjedő hibakezeléshez.

Az Ada egy szigorúan szabályzott és jól dokumentált nyelv. A fordítók szigorú alkalmassági teszteken kell hogy átmenjenek. Legalább egy ingyenes, és több nagyon jó fizetős fordító elérhető.

A nevét [Lady Ada Lovelace](https://hu.wikipedia.org/wiki/Ada_Lovelace) (1815–1852) után kapta, aki [Charles Babbage](https://hu.wikipedia.org/wiki/Charles_Babbage) barátja és bizalmasa volt.

Az Adát nemzetközileg szabványosították 1983-ban, amit felváltott egy új 1995-ben. Ezt a két verziót ismerjük ma úgy, mint Ada 83 és Ada 95.

Modula-2

[**Tervező**](https://www.google.com/search?rlz=1C1GCEU_enHU970HU970&q=javascript+tervez%C5%91&stick=H4sIAAAAAAAAAOPgE-LQz9U3MCqwNNcyyCi30k_Oz8lJTS7JzM_TLyjKTy9KzM3NzEuPz0nMSy9NTE8ttkpJLc5Mz0tNUUiqXMQqnJVYllicXJRZUKJQklpUllp1dCIAfT6iKVQAAAA&sa=X&ved=2ahUKEwiZ4Pn96vPyAhUt_rsIHcaDAK8Q6BMoAHoECDQQAg)**:**Niklaus Writh

*Története:* Az első, és legnépszerűbb strukturált programnyelvek között szerepelt a [Pascal](https://hu.wikipedia.org/wiki/Pascal_(programoz%C3%A1si_nyelv)), amelyet Niklaus Wirth professzor (ETH – Zürich) dolgozott ki, és valósított meg. Könnyen elsajátítható programnyelvnek bizonyult, ezen kívül előnyös volt algoritmus és adatszervezés szempontjából. Sok jó tulajdonsága mellett csak strukturáltságot tartalmazott, ami idővel kevésnek bizonyult.

Legnagyobb hátrányai: nem tartalmazza a moduláris programozás tulajdonságait. [Alacsony szintű programozási nyelv](https://hu.wikipedia.org/wiki/Alacsony_szint%C5%B1_programoz%C3%A1si_nyelv) formájának hiánya, nehéz hozzáférés az [operációs rendszer](https://hu.wikipedia.org/wiki/Oper%C3%A1ci%C3%B3s_rendszer) függvényeihez.

Wirth 1975-ben kifejlesztette a Modula nevű moduláris programnyelvet. 1977-ben kezdett dolgozni a Modula-2 nevű programnyelven. Az első megvalósított verzió futtatása egy [PDP-11](https://hu.wikipedia.org/wiki/PDP-11)-es számítógépen történt 1979-ben. Wirth a Pascal előnyeit a Modula-2 -be szintén bevezette, ezen kívül kibővítette a moduláris és alacsony szintű programozás lehetőségeivel. A Modula-2 egy strukturált, moduláris programozási nyelv – [magas szintű programozási nyelv](https://hu.wikipedia.org/wiki/Magas_szint%C5%B1_programoz%C3%A1si_nyelv).

Cobol



[**Tervező**](https://www.google.com/search?rlz=1C1GCEU_enHU970HU970&q=javascript+tervez%C5%91&stick=H4sIAAAAAAAAAOPgE-LQz9U3MCqwNNcyyCi30k_Oz8lJTS7JzM_TLyjKTy9KzM3NzEuPz0nMSy9NTE8ttkpJLc5Mz0tNUUiqXMQqnJVYllicXJRZUKJQklpUllp1dCIAfT6iKVQAAAA&sa=X&ved=2ahUKEwiZ4Pn96vPyAhUt_rsIHcaDAK8Q6BMoAHoECDQQAg)**:**Grace Hopper

A célként meghatározott üzleti- és pénzügyi alkalmazások miatt az adatok megjelenítésére kiemelten nagy hangsúlyt fektetett a nyelv. Az adatok formáját a "kép" jelentésű angol Picture szó rövidítését jelentő PIC utasítás írja le. A COBOL program kötött [szintaxissal](https://hu.wikipedia.org/wiki/Szintaxis_(programoz%C3%A1si_nyelvek)) rendelkező, úgynevezett COBOL mondatok (Sentences) sorozatából áll, amelyek különböző igékből (ADD, SUBTRACT, PERFORM, READ stb.) és változónevekből épülnek fel. Minden mondatot pont zár le. A mondatok összetartozó csoportjai paragrafusokba (Paragraph) gyűjthetők, amelyek egyben vezérlési szerkezetként is kezelhetők.

A mondatokban használt igék, a kulcsszavak között vannak olyanok, amelyek csak az angol nyelv sajátosságai miatt kellenek (ezeket nem kötelező használni). Egy COBOL program valóban olvasható és érthető egy angolul értő számára, ennek azonban az az ára, hogy néhány COBOL mondat szintaxisa szinte áttekinthetetlenül bonyolult és terjedelmes, és nagyon komoly gyakorlat kell a teljes nyelvi eszközrendszer nyújtotta lehetőségek kihasználásához.

Egyes COBOL megvalósítások annyira meghatározták a forrásprogram írásának szabályait, hogy a nyelvet használó cégek előnyomtatott lapok alkalmazásával segítették a programozókat a formai követelmények betartásában. Az is előírás volt, hogy a programot csak nagybetűvel lehetett írni, ezért programozói körökben elterjedt vicc volt, hogy a COBOL az a programnyelv, amely megtanít a nagybetűk írására.

*Cobol program felépítése:*

1. Azonosítási főrész, IDENTIFICATION DIVISION.

2. Környezetleíró főrész, ENVIRONMENT DIVISION.

3. Adatleíró főrész, DATA DIVISION.

4. Eljárási főrész, PROCEDURE DIVISION.

PHP

[**Tervező**](https://www.google.com/search?rlz=1C1GCEU_enHU970HU970&q=javascript+tervez%C5%91&stick=H4sIAAAAAAAAAOPgE-LQz9U3MCqwNNcyyCi30k_Oz8lJTS7JzM_TLyjKTy9KzM3NzEuPz0nMSy9NTE8ttkpJLc5Mz0tNUUiqXMQqnJVYllicXJRZUKJQklpUllp1dCIAfT6iKVQAAAA&sa=X&ved=2ahUKEwiZ4Pn96vPyAhUt_rsIHcaDAK8Q6BMoAHoECDQQAg)**:**Rasmus Lerdorf

*Története:* A PHP fejlődése kezdetén csak CGI-programok halmaza volt. Ezeket Rasmus Lerdorf néhány Perl szkript lecserélésére írta, amelyeket honlapjának karbantartására (például önéletrajzának megjelenítésére és a látogatottság mérésére) használt. Később ezeket a programokat kombinálta a szintén általa írt *Form Interpreter* (űrlap-értelmező) alkalmazással - így jött létre a *PHP/FI*, ami már jóval szélesebb funkcionalitással bírt. Az új, C nyelven megírt változat képes volt adatbázisokhoz kapcsolódni és segítségével egyszerű dinamikus weboldalakat is létre lehetett hozni. Lerdorf [1995](https://hu.wikipedia.org/wiki/1995). [június 8-án](https://hu.wikipedia.org/wiki/J%C3%BAnius_8.) adta ki a PHP első nyilvános változatát, hogy külső segítséggel gyorsabbá tegye a hibák megtalálását és a kód további fejlesztését. Az új verziót *PHP 2*-nek nevezte el és már megtalálhatóak voltak benne a mai PHP alapvető tulajdonságai: a Perl-éhez hasonló változók, az űrlapok kezelése és a [HTML](https://hu.wikipedia.org/wiki/HTML)-kód beszúrásának lehetősége. A PHP szintaktikája is hasonló volt a Perl-éhez, de annál jóval korlátoltabb, egyszerűbb és kevésbé egységes volt.

*Működése:* A PHP oldalak elkészítésénél a [HTML](https://hu.wikipedia.org/wiki/HTML)-t gyakorlatilag csak mint formázást használják. Ezen lapok teljes funkcionalitása a PHP-re épül. Amikor egy PHP-ben megírt oldalt akarunk elérni, a [kiszolgáló](https://hu.wikipedia.org/wiki/Kiszolg%C3%A1l%C3%B3) először feldolgozza a PHP utasításokat, és csak a kész (HTML) kimenetet küldi el a [böngészőnek](https://hu.wikipedia.org/wiki/B%C3%B6ng%C3%A9sz%C5%91). Így [kliensoldalról](https://hu.wikipedia.org/wiki/Kliens) nem látható a programkód. A feldolgozáshoz egy ún. interpretert (értelmezőt) használ, amely általában a [webszerver](https://hu.wikipedia.org/wiki/Webszerver) egy külső modulja. Ebben hasonlít más szerver oldali szkript nyelvekre, mint a Microsoft ASP.NET-je, a Sun Microsystems JavaServer Pages, és a mod\_perl. Az 5.4-es verzió óta a nyelv tartalmaz egy egyszerű, parancssorban beállítható webszervert.

Ruby



[**Tervező**](https://www.google.com/search?rlz=1C1GCEU_enHU970HU970&q=javascript+tervez%C5%91&stick=H4sIAAAAAAAAAOPgE-LQz9U3MCqwNNcyyCi30k_Oz8lJTS7JzM_TLyjKTy9KzM3NzEuPz0nMSy9NTE8ttkpJLc5Mz0tNUUiqXMQqnJVYllicXJRZUKJQklpUllp1dCIAfT6iKVQAAAA&sa=X&ved=2ahUKEwiZ4Pn96vPyAhUt_rsIHcaDAK8Q6BMoAHoECDQQAg)**:**Macumoto Jukihiro

***A Ruby***[***nyílt forráskódú***](https://hu.wikipedia.org/wiki/Ny%C3%ADlt_forr%C3%A1sk%C3%B3d%C3%BA_szoftver)***, teljesen***[***objektumorientált***](https://hu.wikipedia.org/wiki/Objektumorient%C3%A1lt_programoz%C3%A1s)***, interpretált,***[***általános célú***](https://hu.wikipedia.org/wiki/%C3%81ltal%C3%A1nos_c%C3%A9l%C3%BA_programoz%C3%A1si_nyelv)***programozási nyelv. Macumoto Jukihiro kezdte el megalkotni a nyelvet az 1990-es évek közepén. A fejlesztésbe később többen bekapcsolódtak.***

***A Ruby nyelv egyszerre több***[***programozási paradigmát***](https://hu.wikipedia.org/wiki/Programoz%C3%A1si_paradigma)***valósít meg, így a funkcionális, objektumorientált, imperatív és reflektív paradigmáknak is megfelel. Legfontosabb jellemzői a dinamikus típusosság és az automatikus memóriakezelés. A dinamikus [szkriptnyelvek](https://hu.wikipedia.org/wiki/Interpreteres_nyelvek" \o "Interpreteres nyelvek) családjába tartozik, a***[***Python***](https://hu.wikipedia.org/wiki/Python_(programoz%C3%A1si_nyelv))***, [Perl](https://hu.wikipedia.org/wiki/Perl_(programoz%C3%A1si_nyelv)" \o "Perl (programozási nyelv)), [Lisp](https://hu.wikipedia.org/wiki/Lisp_(programoz%C3%A1si_nyelv)" \o "Lisp (programozási nyelv)),***[***Dylan***](https://hu.wikipedia.org/w/index.php?title=Dylan&action=edit&redlink=1)***,***[***Pike***](https://hu.wikipedia.org/w/index.php?title=Pike&action=edit&redlink=1)***vagy***[***CLU***](https://hu.wikipedia.org/w/index.php?title=CLU&action=edit&redlink=1)***nyelvekhez hasonlóan.***

***Sok platformon működik [interpreterének](https://hu.wikipedia.org/wiki/%C3%89rtelmez%C5%91_(informatika)" \o "Értelmező (informatika)) telepítése után. Egyebek mellett a [Ruby on Rails](https://hu.wikipedia.org/wiki/Ruby_on_Rails" \o "Ruby on Rails) keretrendszer révén webes alkalmazások fejlesztésére is alkalmas.***

***A nyelvvel szemben felmerült főbb érvek:***

* *A nyelvnek nincs formális definíciója.*
* *A Ruby nyelv 1.8 és 1.9 verziójában inkompatibilis módon megváltozott a nyelv szintaxisa. Ez azt jelzi, hogy a Ruby nyelv tervezés híján, ad hoc módon fejlődik.*
* *A nyelv és az interpreter nem kiforrott, túl fiatal, sok hibát hordozhat (mindkettő).*
* *A Ruby interpreter a leglassúbbak közé tartozik.*[*[4]*](https://hu.wikipedia.org/wiki/Ruby_(programoz%C3%A1si_nyelv)#cite_note-4)[*[5]*](https://hu.wikipedia.org/wiki/Ruby_(programoz%C3%A1si_nyelv)#cite_note-5)
* *A Ruby egyszerűen egy divatjelenség (hype).*[*[6]*](https://hu.wikipedia.org/wiki/Ruby_(programoz%C3%A1si_nyelv)#cite_note-6)
* *Kicsi a felhasználói tábor, kevés a konzultációs lehetőség és a dokumentáció, a támogatás hiánya.*

Swift

[**Tervező**](https://www.google.com/search?rlz=1C1GCEU_enHU970HU970&q=javascript+tervez%C5%91&stick=H4sIAAAAAAAAAOPgE-LQz9U3MCqwNNcyyCi30k_Oz8lJTS7JzM_TLyjKTy9KzM3NzEuPz0nMSy9NTE8ttkpJLc5Mz0tNUUiqXMQqnJVYllicXJRZUKJQklpUllp1dCIAfT6iKVQAAAA&sa=X&ved=2ahUKEwiZ4Pn96vPyAhUt_rsIHcaDAK8Q6BMoAHoECDQQAg)**:**Chris Lattner

A Swift egy magas szintű, általános célú, többparadigmás [rendszer- és alkalmazásprogramozási nyelv](https://hu.wikipedia.org/wiki/Programoz%C3%A1si_nyelv), amelyet az [Apple](https://hu.wikipedia.org/wiki/Apple_Inc.) cég fejleszt.

Az eleinte tulajdonolt (proprietary) szoftverként terjesztett, [zárt forráskódú](https://hu.wikipedia.org/wiki/Z%C3%A1rt_forr%C3%A1sk%C3%B3d%C3%BA_szoftver) Swiftet elsősorban az [iOS](https://hu.wikipedia.org/wiki/IOS), [OS X](https://hu.wikipedia.org/wiki/MacOS), [watchOS](https://hu.wikipedia.org/wiki/WatchOS" \o "WatchOS) és tvOS platformokra való alkalmazásfejlesztés céljából hozták létre. 2015. december 3. óta a Swift [fordítóprogram](https://hu.wikipedia.org/wiki/Ford%C3%ADt%C3%B3program) (compiler), a Swift standard library, illetve néhány más kiegészítő program és könyvtár is [szabad szoftverként](https://hu.wikipedia.org/wiki/Szabad_szoftver) terjeszthető, nyílt forráskódú, a fejlesztés a GitHub-on, nyilvánosan folyik.[[2]](https://hu.wikipedia.org/wiki/Swift_(programoz%C3%A1si_nyelv)#cite_note-2) A szabad szoftverré tétellel egyidőben az Apple kiadta a Swift fejlesztői környezet [GNU/Linux disztribúciókon](https://hu.wikipedia.org/wiki/Linux-disztrib%C3%BAci%C3%B3) futtatható bináris csomagjait is.[[3]](https://hu.wikipedia.org/wiki/Swift_(programoz%C3%A1si_nyelv)#cite_note-3) Ez a kiadás természetesen nem tartalmazza az [Xcode](https://hu.wikipedia.org/wiki/Xcode" \o "Xcode) IDE-t és a Cocoa valamint Cocoa Touch API-kat, csak a parancssoros interfésszel (CLI) rendelkező fordítóprogramot, REPL-t és a standard könyvtárat.

*Története:* A Swift fejlesztését még 2010-ben kezdte el [Chris Lattner](https://en.wikipedia.org/wiki/Chris_Lattner), az [LLVM](https://hu.wikipedia.org/wiki/Low_Level_Virtual_Machine) projekt elsődleges szerzője és vezetője. A nyelv eleinte még Apple-ös mércével mérve is szinte teljes titokban készült. A Swiftet az Apple évente megrendezett fejlesztői konferenciáján, a WWDC-n mutatták be 2014 június 2-án. A nyelv bétaverzióját ekkor rögtön elérhetővé is tették a regisztrált fejlesztőknek, csakúgy, mint a hozzá tartozó bevezető leírást, amelynek a címe: ["The Swift Programming Language"](https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/) (A Swift programozási nyelv).

A Swift 1.0 verziója 2014. szeptember 9-én jelent meg, az Xcode [fejlesztői környezet](https://hu.wikipedia.org/wiki/Fejleszt%C5%91i_k%C3%B6rnyezet) 6.0 GM kiadásával együtt. Ezt rövid idő múlva, 2014. október 22-én követte az 1.1 változat az Xcode 6.1 részeként, majd az 1.2, 2015. április 8-án, az Xcode 6.3 számú változatával együtt.