

Az e-volúció zászlóshajói

© Kiskapu Kft. Minden jog fenntartva

Azt gondolom, 2006 az internet éve. Habár a szükséges technológiák jó része már másfél évtizede velünk van, mégis most jutottunk el oda, hogy minden kiforrottan összeálljon. Gyökeres változások tanúi vagyunk. Az internet a szemünk láttára alakul át platformmá. Az interakció megnő. A weboldalak átalakulnak.

Lassan eltűnik a felesleges oldal újratöltődés, a mehet gombok nyomogatása is a múlté. Minden oldal úgy fog fog működni, mint akármelyik asztali alkalmazásunk. Interaktívan és gyorsan. A web programozása is szemlátomást változik. Új, friss megoldásokban nincs hiány! Rövid összefoglaló következik három új szereplőről, amelyek népszerű szkriptnyelvekre, a *Pythonra* és *Rubyra* építve gyökeresen átalakítják a webet.

Ruby on Rails



Alig másfél éve indult hódító útjára az új stílusú webprogramozás zászlóshajója a *Ruby on Rails* (röviden *Rails*). Egy chichagói fiatal programozó, *David Heinemeier Hansson* alkotta meg az alapkoncepciót *basecamp* nevű fejlesztése közben. A fejlesztőeszközt rövidesen nyílt forrásúvá tette. Köszönhetően az egyre növekvő közösségnek és a fejlesztésbe

bekapcsolódó számos programozónak, a *Rails* 2005. december 13-án, alig másfél év alatt elérte az 1.0-ás verziót. *David* a rugalmas és rendkívül gyors kódolást lehetővé tevő *Ruby* nyelvet használta fel a fejlesztéshez. A főként *Japánban* és immár *Amerikában* ismert *Ruby* nagyon hasonlít a *Pythonhoz*. Fejlett szintaxisával és teljesen objektum orientált megközelítésével villámgyors kódolást tesz lehetővé. Általánosságban elmondhatjuk, hogy egy *Ruby* vagy *Python* program hossza ötöde vagy tizede egy hasonló funkciójú *Java* programnak.

A Rails legfőbb jellemzői

Minden igényt kielégítő fejlesztőeszköz, amely integráltan tartalmaz mindent ami adatbázis alapú weboldalak létrehozásához szükséges. A villámgyors munkát segíti az is hogy az alkalmazásainkat az asztali alkalmazások fejlesztésénél megszokott *Model-View-Controller* séma felhasználásával fejleszthetjük. Három önálló részre bontjuk szét az alkalmazásainkat, ezáltal leegyszerűsítve a progra-

mozást, a hibakeresést és növelve az átláthatóságot. A *model* hivatott kezelni az adatainkat, a *view* felel a megjelenítésért és a *controller* pedig megmondja, hogy a felhasználó mely interakciójára mit is csináljon a program. Az egész fejlesztőeszköz *Ruby* alapú, nekünk csak egy adatbázis-kezelőre van szükségünk hogy nekikezdhessünk a fejlesztésnek. A *Rails* kiemelt figyelmet szentel a *DRY (Don't Repeat Yourself)*, vagyis a ne ismételd magad elv betartásának. Ezért nincsenek hosszú beállításfájlok, ismétlődő programozói feladatok. Például egy új programnál elég beállítani az adatbázis paramétereit és írni tíz sor kódot, hogy működő, látható eredményt kapjunk. A Rails népszerűségéhez nagyban hozzájárul, hogy a remek leírások mellett sok videó is elérhető, melyeken végignézhethetjük miként is dolgozik egy profi. Az alábbi kép egy olyan videó záró képkockája, mely azt mutatja be, miként illeszthetjük a *Flickr.com* által kezelt fotóinkat az oldalunkba mindössze öt perc alatt.



A *Rails* kiemelten támogatja az *AJAX*-ot (*Asynchronous JavaScript and XML*), vagyis a legfontosabb Javascript kód-könyvtárak illetve vannak a keretrendszerhez, ami által úgy használhatjuk őket, hogy végig csak *Ruby* kódot írunk. Így könnyen készíthetünk az oldal újratöltődése nélkül frissülő menüket, animált szövegmezőket és még sok más. Akit a téma bővebben érdekel, nézzon szét a script.aculo.us oldalon. Mindent amit ott láthatunk szervesen integráltak a keretrendszerbe. (wiki.script.aculo.us/scriptaculous/show/Demos)

Azt gondolhatnánk hogy a gyors és interaktív fejlesztés nehezen eredményezhet hibamentes kódot. A *Rails* ebben is a segítségünkre van, hiszen alkalmazásaink minden porcióját letesztelhetjük a segítségével, sőt lehetőséget ad, hogy a teszt alapú fejlesztés gyakorlatát kövessük. Ez által professzionális munkára is ideális választás lehet.

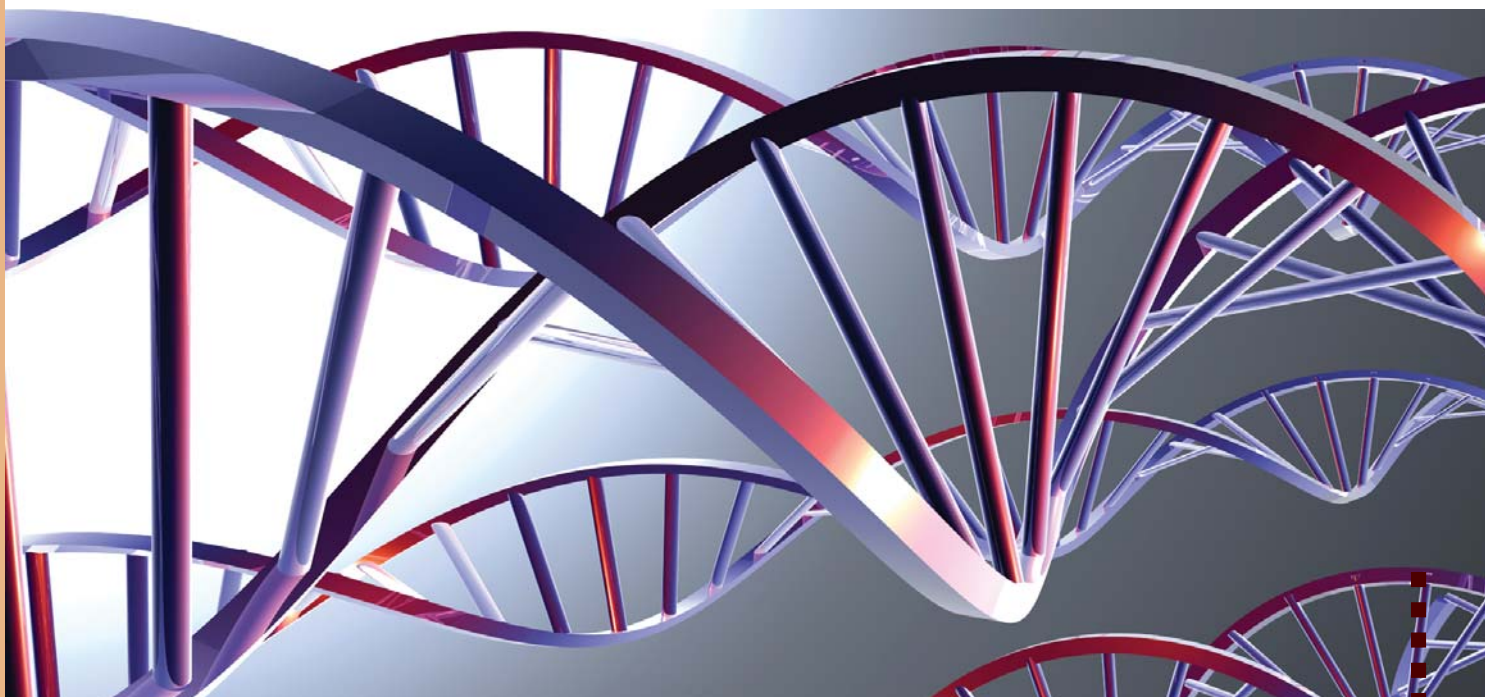
TurboGears



A mindössze fél éves *TurboGears* megpróbálja megvalósítani a *Rails* koncepcióját a *Python* programozók számára. A *Kevin Dangoor* által indított fejlesztés összeboronált számos meglévő *Python* programot, így alkotva meg egy könnyen kezelhető keretrendszert. Honlapja a turbogears.org címen található. A *Rails*hoz hasonlóan itt is a *Model-View-Controller* séma alapján oszthatjuk részekre a programjainkat. Az ábrán jól látható az egyes rétegek egymásra épülése. Az *SQLObject* segít könnyedén kezelni az adatbázisainkat, tehát ez felel meg a modellnek, a *CherryPy* a controller, mely a különböző kérésekre eldönti milyen választ kell visszaadni a böngészőnek. A *Kid* egy könnyen tanulható sablon rendszer, mely a *Mochikit* nevű *Javascript* kód-könyvtár segítségével hihetetlenül interaktív felhasználói felületeket képes generálni. Természetesen a *TurboGears* is a kezünk alá dolgozik, ha programjaink minden sorát tesztekkel szeretnénk ellenőrizni.



Akárcsak a *Rails* körül, a *TurboGears* körül is nagy közösség szerveződött. Fiatalabb rendszer lévén még van néhány gyermekbetegsége. Voltak például problémák az egyes kódlapok használatával, de a hamarosan megjelenő 1.0-ás verzió már a tervek szerint teljesen használható lesz éles rendszerekben is. A megjelenő stabil kiadás a tervek szerint tartalmaz majd egy on-line *Python* héjat (shell) is, ami azért valljuk be hasznosnak tűnik távoli fejlesztéshez vagy hibakereséshez. Összességében elmondhatjuk, hogy a *Python* közösség egy olyan



A három keretrendszer összehasonlítása

Rails

Előny: A Ruby programozási nyelv ereje, nagy közösség (magyar közösség is van már), jó dokumentáció, könnyű integráció más rendszerekkel és technológiákkal, kipróbált, skálázható, webalkalmazásokhoz ideális.

Hátrány: A többnyelvű alkalmazások készítése még nem teljesen átgondolt, bár jól használható.

Értékelés: Mindenkinnek nagyon ajánlom!

TurboGears

Előny: Pythonban íródott, gyorsan fejlődik, könnyen integrálható más technológiákkal

Hátrány: Még nem teljesen stabil, a dokumentáció néhol hiányos.

Értékelés: Kisebb munkákra már most is kitűnően használható. Várhatóan fél éven belül a Rails Pythonban írt társát ünnepelhetjük benne.

Django

Előny: Kipróbált, skálázható, stabil, jól dokumentált, könnyen integrálható más rendszerekkel, portálok kialakításához ideális.

Hátrány: Csak PostgreSQL adatbázis kezelőt támogat (ezt a stabilitás miatt akár előnyként is felfoghatjuk)

Értékelés: Portálok kialakítására ideális választás.

keretrendszerrel lett gazdagabb, amely remek eszközeivel gyors webalkalmazás fejlesztést tesz lehetővé. Minden Python rajongónak csak ajánlani tudom.

Django



Ki nem hallott még a robusztus és mindenre használható Zope alkalmazás kiszolgálóról? Nos, a kihívó megérkezett. A Django-t sokan csak így nevezik: „Zope killer”. Hogy miért? Pillanatokon belül kiderül. Készítői újságok weboldalának kialakításához használták. Ebben íródott többek között a Lawrence.com és az Ljworld.com is. Összetett és hatalmas oldalak kialakítására tervezték, mégpedig úgy, hogy miközben követi egy napilap gyorsan változó és szövegményes tartalmát, a lehető legkevesebb energiát igényelje a fejlesztőktől. Lássunk néhány eszközt, amelyekkel megkönnyíti a munkánkat! Hogy ne fáradjunk többet az SQL parancsok írásával, a Django segítségével adatbázisaink tartalmát mint objektumokat érhetjük el (Object-relational mapper). Azt hiszem ez a funkció alapvető minden komoly keretrendszer-nél. Láthattuk, hogy a Rails és a TurboGears is tartalmaz hasonló megoldást. Természetesen végrehajthatunk SQL kódot ha szükséges, de a pizkos munkát leveszi a vállunkról.

Szükségünk lenne pár adminisztrátori felületre? Semmi izgalom, a Django generálja majd őket helyettünk! Segítségével nagyon egyszerűen tudnak majd a felhasználóink tartalmakat hozzáadni és módosítani. És ez a funkció a fejlesztők szerint éles környezetben is hibamentesen használható. Érdeklődőknek ajánlom figyelmébe a Django dokumentációját. Nagyon, nagyon figyelemreméltó ez az eszköz, bár ezt csak az érezheti át igazán aki már alakított ki pár felhasználói felületet.

A modern web elvárásainak megfelelően a keretrendszer segítségével könnyen definiálhatunk állandó és rövid web címet oldalaink számára. Kézbe kapunk egy könnyen tanulható sablon rendszert is, mely úgy lett tervezve, hogy a grafikusaink is szívesen használják majd.

Megbizonyosodhatunk róla, hogy a Django professzionálisan lett kialakítva. Ugyanis beépítetten megtalálható benne egy rugalmas Cache rendszer, vagyis bizonyos tartalmakat nem renderel le minden lekéréskor, hanem megpróbálja amennyire lehet csökkenteni a terhelést azok gyorsítótárazásával. Végül megemlítendő, hogy a nemzetközi oldalak készítése is problémamentes.

Mire is lehetne még szükségünk? Aki tehát nagy és heterogén oldalak kialakítását tervezi, gondolja át a tervekét és használja a Django-t! A Django honlapja a következő címen található: www.djangoproject.com

Összefoglalás

Három kitűnő keretrendszert láthatunk, melyek segítségével a web programozása egyszerűbb mint valaha. Ismét megtapasztalhattuk a nyílt forráskód és a fejlesztők óriási közösségének teremtő erejét, hiszen mindhárom rendszer friss és naprakész. Röviden, kulcsszavakban összefoglalom most mindhárom rendszer előnyeit és gyengeségeit.



Juhász Attila

(rabszolga@goraffe.hu)
Az Információ Technológiai Kar hallgatója a Pázmány Péter Katolikus Egyetemen.

Érdeklődik a bioinformatika és a neurális hálózatok iránt. A fotózás és a tánc mellett öt éve foglalkozik webgrafikákkal. A linux terjesztések közül a Gentoo és az Ubuntu áll legközelebb a szívéhez. Fotós oldala a <http://people.goraffe.com/attila> címen található.

KAPCSOLÓDÓ CÍMEK

Ruby on Rails weblap

➔ rubyonrails.org

Magyar Rails oldal

➔ rubyonrails.hu

Javascript effektek könyvtára:

➔ script.aculo.us

Radrails, egy Eclipse alapú integrált fejlesztői környezet (IDE)

➔ radrails.org