

최신 .NET과 C#

2016. 01. 14

NHN Next

최흥배

<https://github.com/jacking75/choiHeungbae>

WHOW?

2003년 PC 온라인 게임 서버 개발 시작
보드 게임,
MMORPG,
P2P 액션 캐주얼 게임

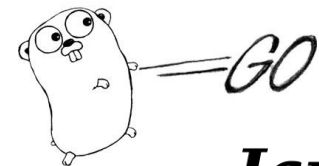


2013년 여름 모바일 게임 서버 개발 시작
2개의 미들 코어 모바일 게임 서버 아키텍처 설계
구현 및 기술 지원,



시뮬레이션 장르 게임 개발(취소),
PC/Mobile 동시 플레이 스포츠 시뮬레이션 게임
개발(취소)

2014년 NHN Next 게임학과 겸임 교수



JavaScript

[home](#)
[tag](#)
[media](#)
[location](#)
[guestbook](#)
[admin](#)
[write](#)

Programmer J

welcome to the blog.

2013/09/04 08:00
프로그래밍, GameDev | Posted by 홍배

Ubuntu에서 Boost 라이브러리 빌드 테스트 해 보기

Boost 라이브러리는 home에서 아래의 디렉토리에 설치

boost_1_53_0

컴퓨터

내 폴더

바탕 화면

다운로드

문서

비디오

사진

음악

파일 시스템

휴지통

네트워크

네트워크 찾아...

내 폴더

Dev

boost_1_53_0

bin.v2

boost

doc

more

stage

status

See secti

INSTALL

Jamroot

LICENSE_1_0.txt

boost

Boost 라이브러리를 빌드하여 debug, release 각각 만든다(정적 라이브러리)
빌드 방법은 다음 <http://jacking.tistory.com/1070>

lib

컴퓨터

내 폴더

바탕 화면

내 폴더

Dev

boost_1_53_0

stage

lib

홍배

현업 온라인 게임 서버 프로그래머입니다. 이 블로그는 프로그래밍(특히 C++과 JavaScript, HTML5), 게임제작과 관련된 글을 주제로 하고 있습니다

Category

분류 전체보기 (1047)

프로그래밍, GameDev (457)

IT 번역 (346)

강연, 세미나 후기 (25)

HalfNetwork (27)

책..... (58)

일상, 기타 등등 (91)

모임, 세미나 (40)

Mac 사용하기... (2)

Recent Post

- 이번에 구매한 책들....
- 모바일, 소셜 게임에서...
- MongoDB 사용 - C#.
- MongoDB 사용 - mongo 콘솔

RSS

Tag

C/C++ Bjarne Stroustrup
CEDEC2010 C Visual C++1
1 CEDEC vc vc++ CEDEC
2010 concept C# Boost GD
C2010 VS2012 VC11 C++ C
++11 TR1 Visual Studio 20
12 Visual C++ HalfNetwork
C++0x Network Programmi
ng Thread VC++11 VSTS 2
010 STL VC++ 10 Visual C
++ 10 .net

2013.10

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25
26	27	28	29	30
31				

Archive

- 2013/10 (11)
- 2013/09 (10)
- 2013/08 (10)
- 2013/07 (14)
- 2013/06 (12)

My Link

- 온라인게임서버개발자...
- T3엔터테인먼트 프로그...

<http://jacking.tistory.com>



Visual C++ MVP 2009 ~ 2011



🔍 크게보기 | 📖 미리보기

eBook Thinking About: C++ STL 프로그래밍

최홍배 지음 | 국내서 | 2012년 12월 21일

· 파일포맷/용량 PDF/3.49 MB

· 이용안내 eBook 구매시 유의사항 및 이용안내 ?

· 상태 판매중

· 독자리뷰 총 리뷰 0개, 평균별점 ★★★★★

· 블로거리뷰 총 리뷰 0개

· 트랙백주소 http://www.hanb.co.kr/tb_ebook/9788979149937

· 페이지 : 235쪽 | ISBN : 9788979149937 | 난이도 : 초/중급 | 변환코드 : 1993



좋아요

188

🔴 eBook : 0원 → 0원

📖 장바구니 담기

✍ 리뷰쓰기

👤 오타자 등록

💬 토론 / QA

<http://www.hanbit.co.kr/ebook/look.html?isbn=9788979149937>



🔍 크게보기 | 📖 미리보기

eBook Boost.Asio를 이용한 네트워크 프로그래밍

최홍배 지음 | 국내서 | 2013년 06월 04일

- 파일포맷/용량 PDF/2.72 MB
- 이용안내 eBook 구매시 유의사항 및 이용안내 ?
- 상태 판매중
- 독자리뷰 총 리뷰 1개, 평균별점 ★★★★★
- 블로거리뷰 총 리뷰 3개
- 트랙백주소 http://www.hanbit.co.kr/tb_ebook/9788968486111

· 페이지 : 132쪽 | ISBN : 9788968486111 | 난이도 : 초/중급 | 변환코드 : 2611



- ☒ eBook : 9,900원 → **9,900원** (0% off / P 5% 지급)
- ☐ eBook+Print : 16,900원 → **16,900원** (0% off / P 5% 지급)

🔖 장바구니 담기

📝 리뷰쓰기

👤 오타자 등록

💬 토론 / QA

<http://www.hanbit.co.kr/ebook/look.html?isbn=9788968486111>



🔍 크게보기 | 📖 미리보기

eBook Thinking About C++11 STL 프로그래밍(개정2판)

최흥배 지음 | 국내서 | 2014년 06월 12일

- 파일포맷 PDF
- 이용안내 eBook 구매시 유의사항 및 이용안내 ?
- 상태 판매중
- 독자리뷰 총 리뷰 0개, 평균별점 ★★★★★
- 블로거리뷰 총 리뷰 0개
- 트랙백주소 http://www.hanbit.co.kr/tb_ebook/9788968486623

· 페이지 : 263쪽 | ISBN : 9788968486623 | 난이도 : 초/중급 | 변환코드 : 2662



- ☒ eBook : 12,000원 → **6,000원** (50% off / P 0% 지급)
- ☐ eBook+Print : 21,200원 → **21,200원** (0% off / P 5% 지급)
- ☐ Print : 14,400원 → **14,400원** (0% off / P 5% 지급)

🛒 장바구니 담기

✍ 리뷰쓰기

👤 오타자 등록

💬 토론 / QA

<http://www.hanbit.co.kr/ebook/look.html?isbn=9788968486623>



🔍 크게보기

eBook 당신의 프로그래밍에 디버깅 더하기 : Visual C++ 디버깅 기초에서 고급까지

최홍배 지음 | 국내서 | 2014년 11월 21일

- 파일포맷 PDF
- 이용안내 eBook 구매시 유의사항 및 이용안내 ?
- 상태 판매중
- 독자리뷰 총 리뷰 0개, 평균별점 ★★★★★
- 블로거리뷰 총 리뷰 0개
- 트랙백주소 http://www.hanbit.co.kr/tb_ebook/9788968486586

• 페이지 : 57쪽 | ISBN : 9788968486586 | 난이도 : 초/중급 | 변환코드 : 2658



🔴 eBook : 0원 → 0원

📖 장바구니 담기

✍ 리뷰쓰기

👤 오타자 등록

💬 토론 / QA

<http://www.hanbit.co.kr/ebook/look.html?isbn=9788968486586>



🔍 크게보기 | 📄 미리보기

eBook 모던 C++ : C++14의 핵심 기능을 중심으로

최홍배 지음 | 국내서 | 2015년 03월 31일

- 파일포맷 PDF
- 이용안내 eBook 구매시 유의사항 및 이용안내 ?
- 상태 판매중
- 독자리뷰 총 리뷰 2개, 평균별점 ★★★★★
- 블로거리뷰 총 리뷰 0개
- 트랙백주소 http://www.hanbit.co.kr/tb_ebook/9788968487460

· 페이지 : 69쪽 | ISBN : 9788968487460 | 난이도 : 초/중급 | 변환코드 : 2746



● eBook : 6,000원 → 5,400원 (10% off / P 5% 지금)

🔖 장바구니 담기

✍ 리뷰쓰기

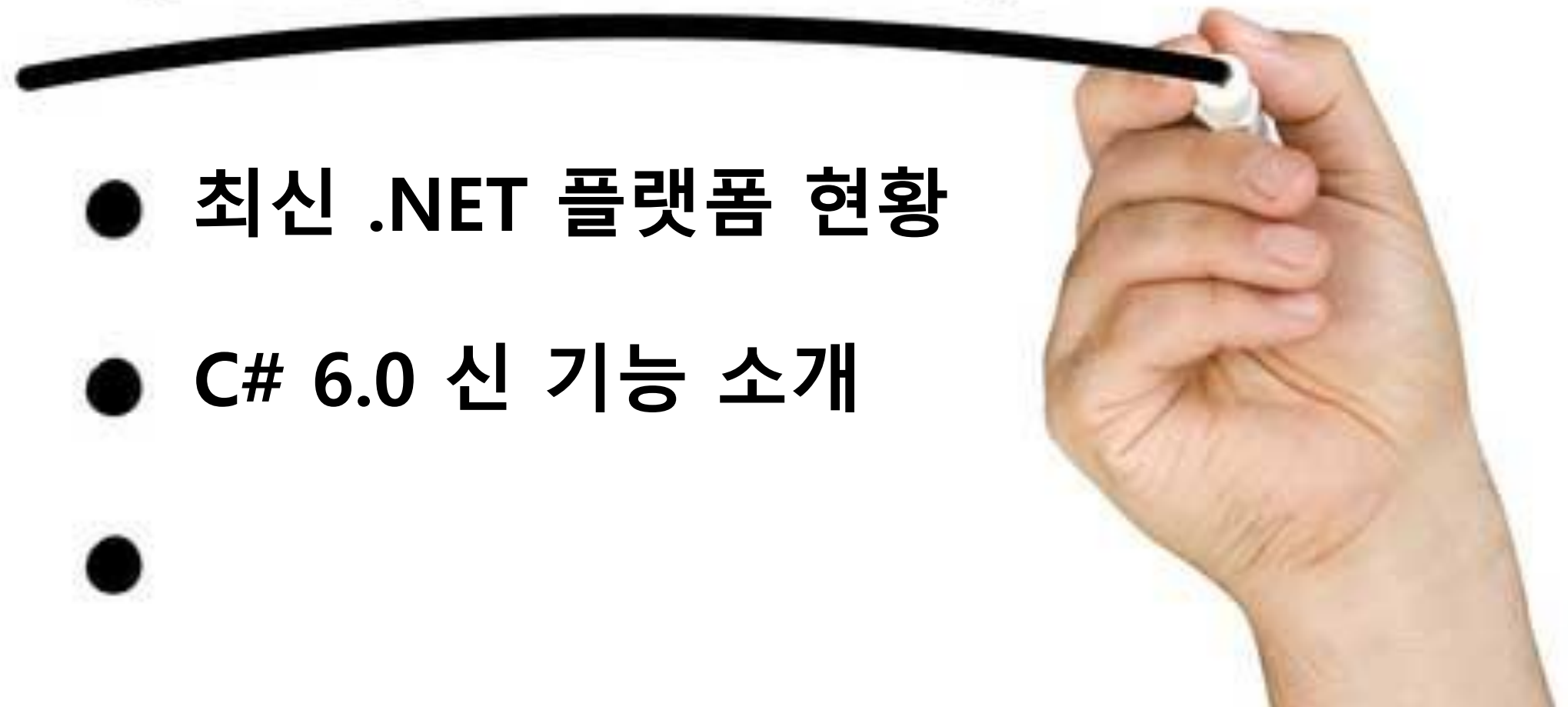
👤 오타자 등록

💬 토론 / QA

<http://www.hanbit.co.kr/ebook/look.html?isbn=9788968487460>

이 문서에 사용된 자료는 해외에서 다른 강연자가 발표한 자료에서 발취해서 정리한 것 입니다.

AGENDA



- 최신 .NET 플랫폼 현황
- C# 6.0 신 기능 소개
-

TIOBE Index for January 2016

Jan 2016	Jan 2015	Change	Programming Language	Ratings	Change
1	2	⬆	Java	21.465%	+5.94%
2	1	⬇	C	16.036%	-0.67%
3	4	⬆	C++	6.914%	+0.21%
4	5	⬆	C#	4.707%	-0.34%
5	8	⬆	Python	3.854%	+1.24%
6	6		PHP	2.706%	-1.08%
7	16	⬆	Visual Basic .NET	2.582%	+1.51%
8	7	⬇	JavaScript	2.565%	-0.71%
9	14	⬆	Assembly language	2.095%	+0.92%
10	15	⬆	Ruby	2.047%	+0.92%

.NET Framework와 Windows OS 관계

		Windows OS											
		Windows XP	Windows Server 2003 (R2含む)	Windows Vista	Windows Server 2008	Windows 7	Windows Server 2008 R2	Windows 8	Windows Server 2012	Windows 8.1	Windows Server 2012 R2	Windows 10	Windows Server 2016
.NET Framework	2.0	●	●	[3.0]	[3.0]	[3.5]	[3.5]	[3.5]	[3.5]	[3.5]	[3.5]	[3.5]	[3.5]
	3.0	●	●	◎	◎	[3.5]	[3.5]	[3.5]	[3.5]	[3.5]	[3.5]	[3.5]	[3.5]
	3.5	●	●	●	●	◎	◎	◎	◎	◎	◎	◎	◎
	4	● (SP3)	● (SP2)	● (SP1)	●	●	●	[4.5]	[4.5]	[4.5.1]	[4.5.1]	[4.6]	[4.6]
	4.5			● (SP2)	● (SP2)	● (SP1)	● (SP1)	◎	◎	[4.5.1]	[4.5.1]	[4.6]	[4.6]
	4.5.1			● (SP2)	● (SP2)	● (SP1)	● (SP1)	●	●	◎	◎	[4.6]	[4.6]
	4.5.2			● (SP2)	● (SP2)	● (SP1)	● (SP1)	●	●	●	●	[4.6]	[4.6]
	4.6			● (SP2)	● (SP2)	● (SP1)	● (SP1)	●	●	●	●	◎	◎

http://blogs.msdn.com/b/visualstudio_jpn/archive/2015/10/19/dotnet-framework-support-lifecycle-policy-summary-2015oct.aspx

◎ : OS 에 처음부터 포함 되어 있다

● : 다운로드 해서 설치 가능 ※ () 은 시스템 요건 상 필요로 하는 OS 서비스 팩

간단 .NET 4.6.1 정리

WPF

투명한 자식 창.

고 DPI 대응 개선(보더 있는 컨트롤의 레이아웃에 관해 둥글게 처리, DPI가 다른 멀티 모니터 대응 개선, 여러 해상도의 커서 화상 등).

Windows Forms

고 DPI 대응 개선(콤보 박스와 데이터 그리드 뷰 등의 컨트롤이 고 DPI 대응).

RyuJIT

버그 픽스 등.

SIMD

System.Numerics 이름 공간의 벡터형, 행렬형, 쿼터니온(Nuget에서 입수)

GC

pinned objects를 포함 컴팩션 최적화.

Gen1에서 Gen2 승격 처리 최적화.

"No GC" 구간.

WCF

TLS 1.1/1.2 대응.

HTTP 접속 분리(커넥션 그룹과 채널 팩토리)

WF

Out Of Order 실행.

ADO.NET

SQL Server 2016의 Always Encrypted(드라이버 레벨에서 항상 암호/복호화 하는 기능) 대응.

Async

System.Threading.AsyncLocal 클래스(비동기 컨텍스트 마다의 로컬 스토리지).

Task가 스레드의 컬처나 UI 컬처를 이어받음.

Task 클래스에 편리한 메소드를 추가.

네임 파이프도 비 동기 통신을 지원.

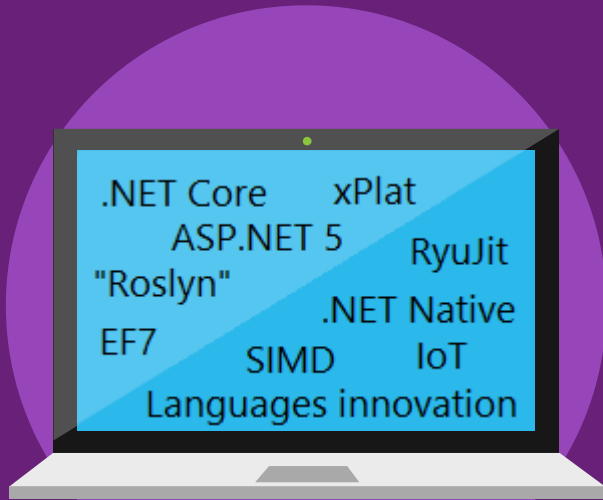
네트워크

HTTP 2 지원(Windows 10 이상에서만)

Windows 10에서 구현된 로컬 포트와 TCP 접속을 재이용하는 기구 지원.

System.Uri에서 국제화 도메인 이름/PunyCode 지원

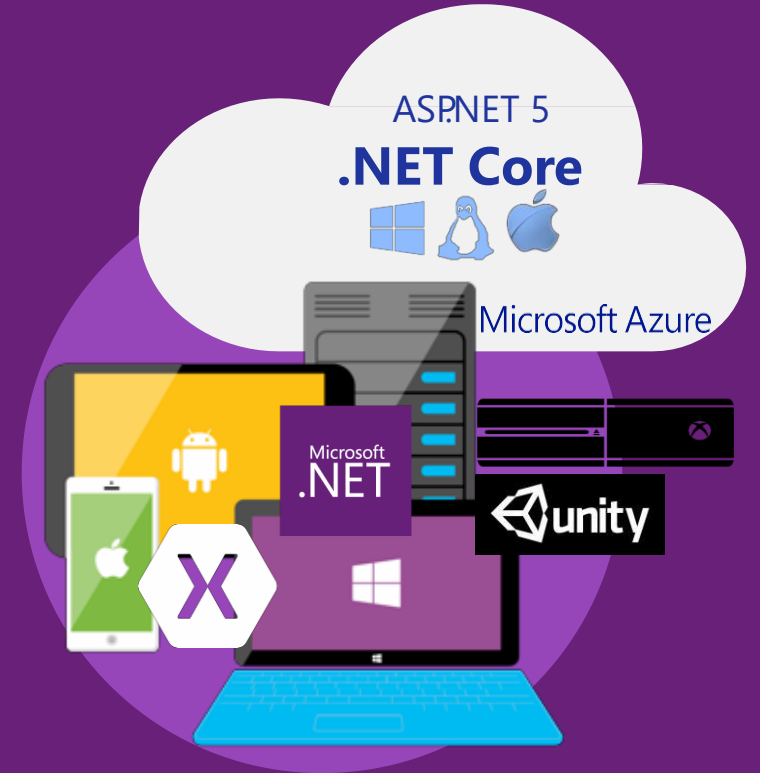
.NET의 새로운 방향성



.NET Innovation



Open Source



Cross-Platform

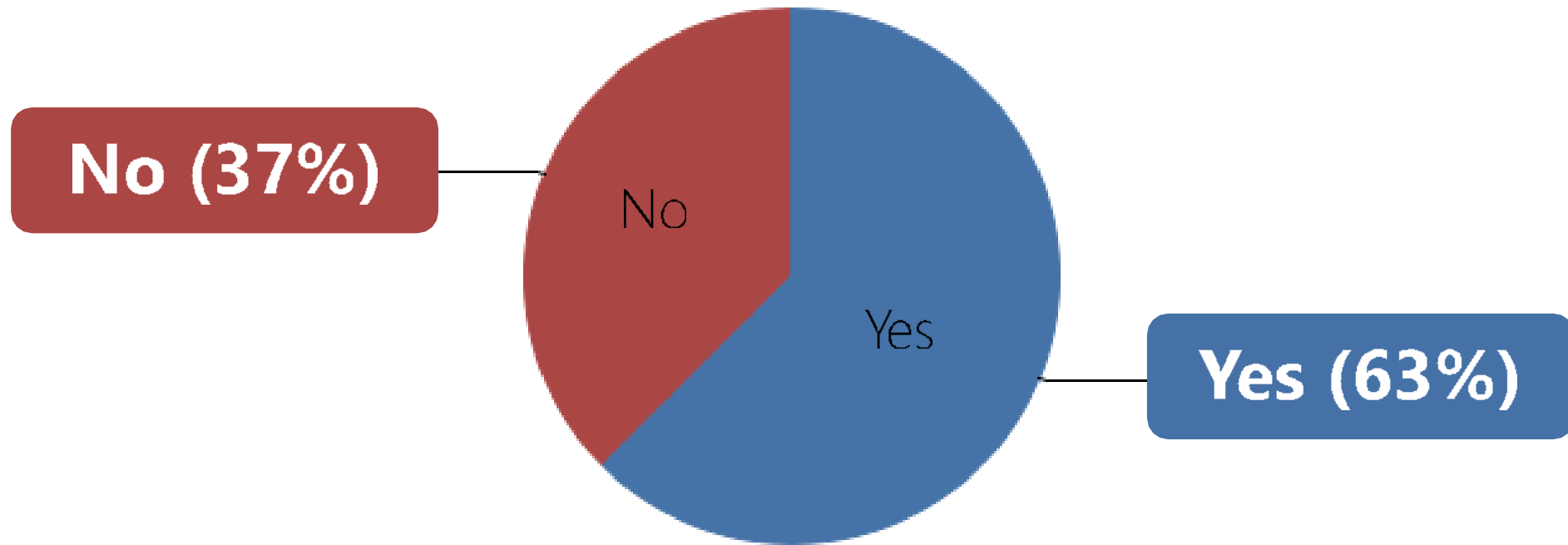
Microsoft ❤️ Linux





설문 조사 결과: <http://asp.net>

Linux 환경에서 ASP.NET이 돌아간다면 Web 애플리케이션 개발에서 ASP.NET을 채용할 기회가 증가하나요? (응답수: 2,000)



미국 마이크로소프트의 Scott Hanselman

".NET은 Linux, Mac, Windows에서 다른 언어와 같은 차원에서 경쟁하게 됐다. Ruby, PHP, Java, Python과 같은 화제 속에 .NET을 선택 사항으로서 넣고 싶다"

.NET 2015

Windows 대응의 ".NET Framework 4.6".

ASP.NET 5, ASP.NET 4.6, WPF, Windows Forms과 같은 프레임워크로 구성.
이른바 full spec의 .NET Framework.

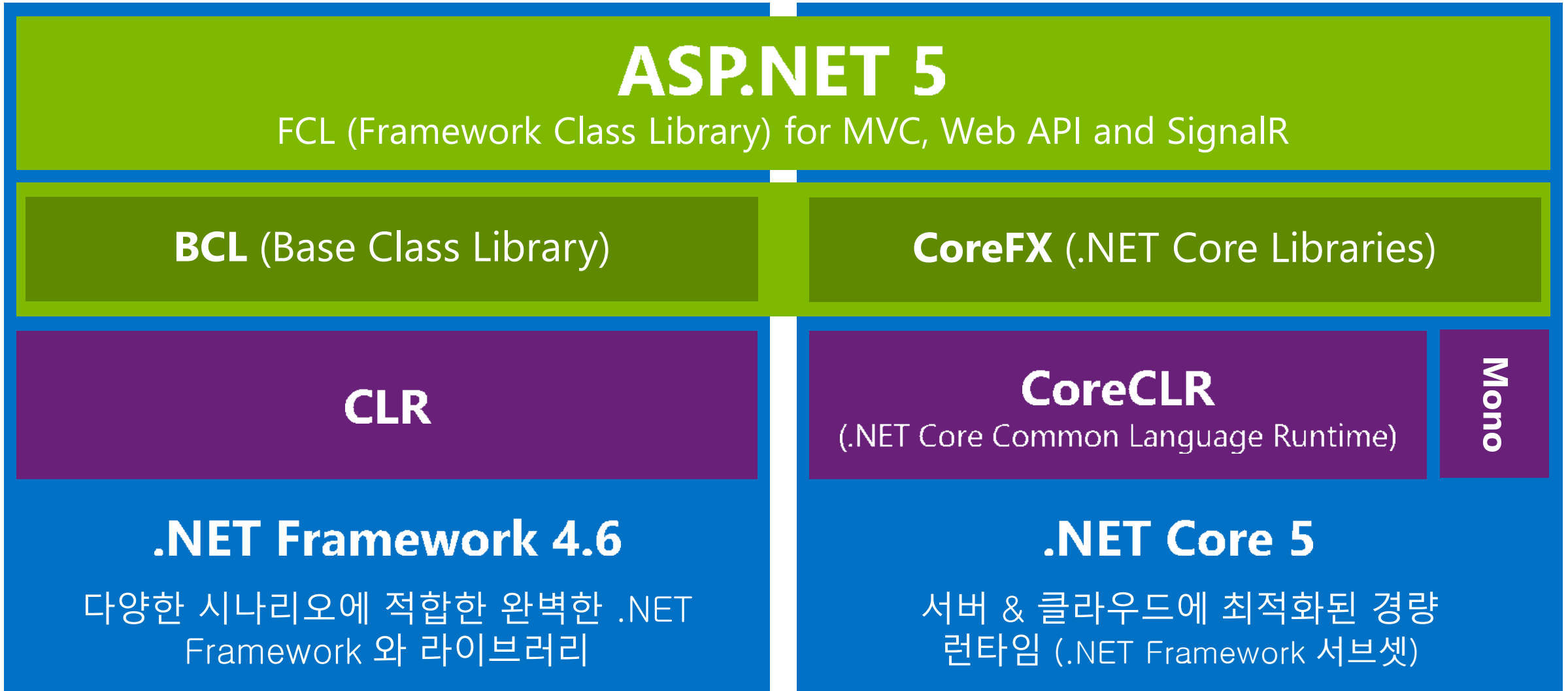


- .NET Core 5는 .NET Framework의 서브셋.
- Windows 대응의 "ASP.NET Core 5"와 ".NET Native".
- Linux 및 MacOS X 대응 "ASP.NET Core 5"로 구성.

현재, NET는 ".NET 2015"이라는 브랜드에 의해 선제를 표현하고 있다.
ASP.NET 5와 ASP.NET Core 5는 실질적으로 같은 것.

- .NET 2015의 가동 환경을 뒷받침하는 틀.
- 이 3개는 오픈 소스로 개발됐으며 Windows, Linux, MacOS X의 크로스 플랫폼 대응.

.NET Core 5 & ASP.NET 5



핵심 정리

- ASP.NET 5는 .NET 4.6 / .NET Core 양쪽에서 동작한다.
- 라이브러리는 .NET 4.6과 .NET Core에서 기본적으로 서로 호환성이 없다.
 - NuGet에서 두 개를 포함한 형태로 배포한다.
 - ASP.NET 4.6은 기존의 ASP.NET의 후계로 프로젝트 관리 방법도 기존 형식을 따른다.
 - ASP.NET 5는 차세대 ASP.NET으로 기존의 ASP.NET을 새로 만든 것이다.
- Roslyn을 사용한 실행 시 컴파일.
소스 코드를 변경해도 페이지를 Re-Load 해도 자동적으로 재 컴파일.
- NuGet으로 모두 배포
 - .NET Core의 런타임 / DNX
 - 기본 클래스 라이브러리(BCL)
 - .NET Framework 4.6 용, .NET Core 용

ASP.NET 5의 특징

- 크로스 플랫폼으로 애플리케이션의 개발/실행 가능.
- 거대한 System.Web 어셈블리에 의존하지 않고, 보다 세세한 패키지 관리 방법을 채용하여 필요한 것만 애플리케이션에서 취한다.
- IIS 이외의 웹 서버에서도 실행 가능하다(예 kestrel).
- ASP.NET Web API, ASP.NET Web Pages가 ASP.NET MVC 6에 통합된다.
- OWIN의 사상에 가까운 미들웨어에 의한 리퀘스트 파이프라인을 채용.
- IoC를 전면적으로 채용.

DNX (.NET Execution Environment) 개요

■ DNX (.NET Execution Environment)

- .NET 애플리케이션 빌드 & 실행을 위한 SDK와 런타임
- Windows, Mac OS X, Linux 에 대응

■ dnx.exe

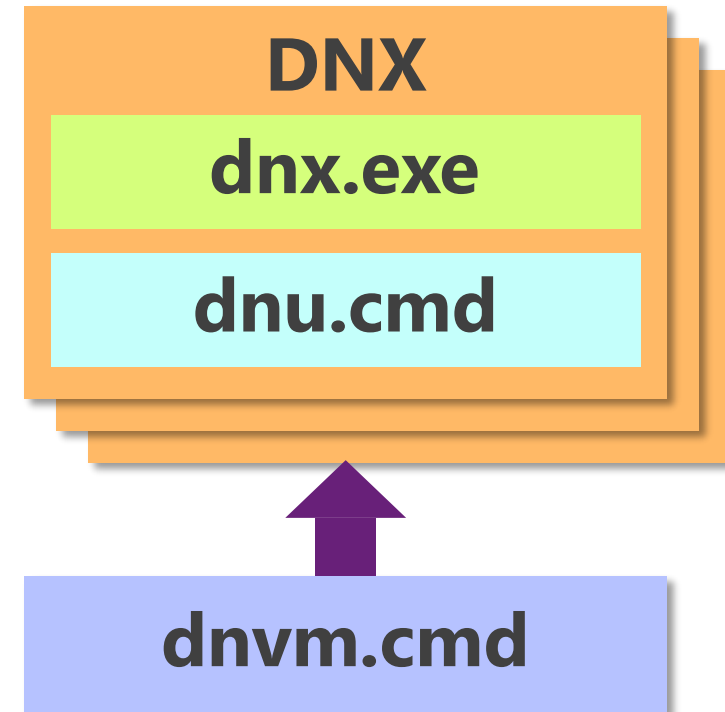
- 런타임 엔트리 포인트
 - dnx.exe <project.json 의 패스> <실행 명령어>
 - 예: "dnx . run", "dnx . web", "dnx . kestrel", ...

■ dnu 명령어 (.NET Development Utility)

- DNX 프로젝트 빌드, 패키징과 발행
- 패키지 의존 관계 해결

■ dnvm 명령어 (.NET Version Manager)

- DNX 버전 관리 : install, upgrade, list, use, alias



ASP.NET 5 문서

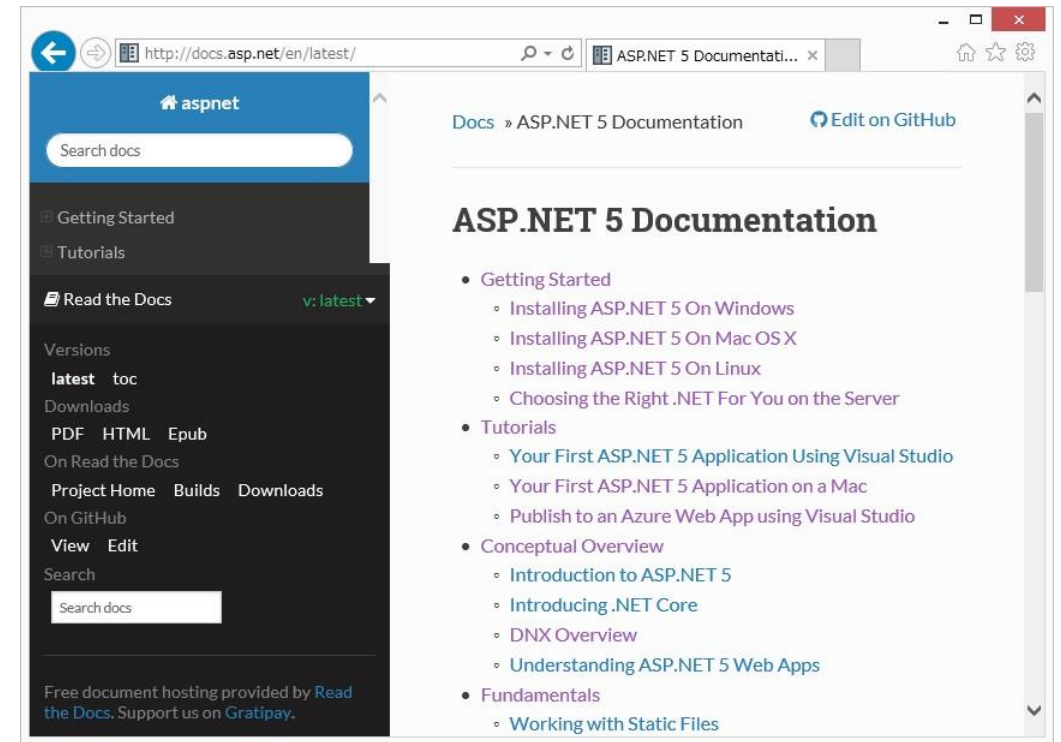
ASP.NET 5 Documentation

<http://docs.asp.net/>

- ✓ 최신판 문서를 참조 가능
- ✓ PDF / HTML / EPUB 형식 다운로드 가능
- ✓ 검색 가능

ASP.NET 5 Wiki

<https://github.com/aspnet/Home/wiki>





YEOMAN

Web 개발 클라이언트 툴

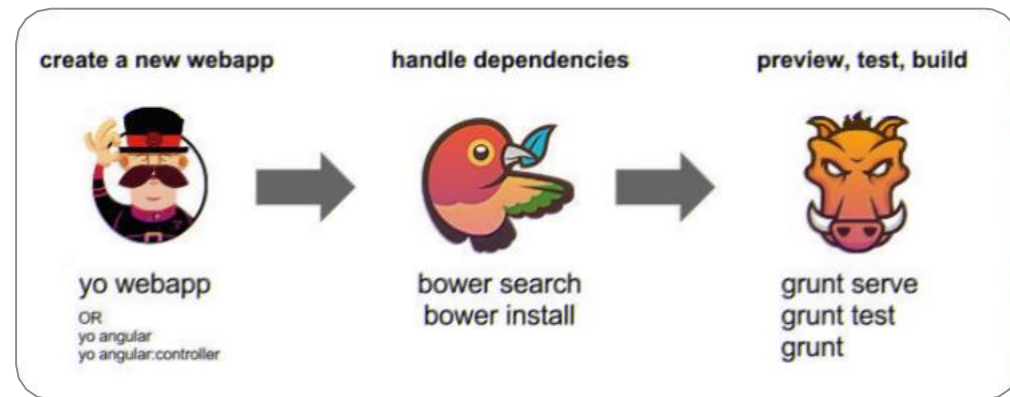
■ Yeoman (<http://yeoman.io/>)

■ 크로스 플랫폼 툴군

- Windows, Mac OS, Linux

■ 아래의 툴로 구성

- **yo** : 스캔 폴딩 툴
- **Bower, npm** : 패키지 매니저
- **Grunt, Gulp** : 태스크 런너/빌드 툴
- **yo generator for ASP.NET 5**
 - ASP.NET 5 프로젝트 생성 툴





가장 처음 설치하길 바라는 OSS 툴

- Git : <https://git-scm.com/>

- Node.js : <https://nodejs.org/>

- Gulp

```
> npm install -g gulp
```

- Grunt

```
> npm install -g grunt-cli
```

- Bower

```
> npm install -g bower
```

- Yo

```
> npm install -g yo
```

- ASP.NET 5 generator

```
> npm install -g generator-aspnet
```

- TypeScript

```
> npm install -g typescript
```

Bower 와 NuGet



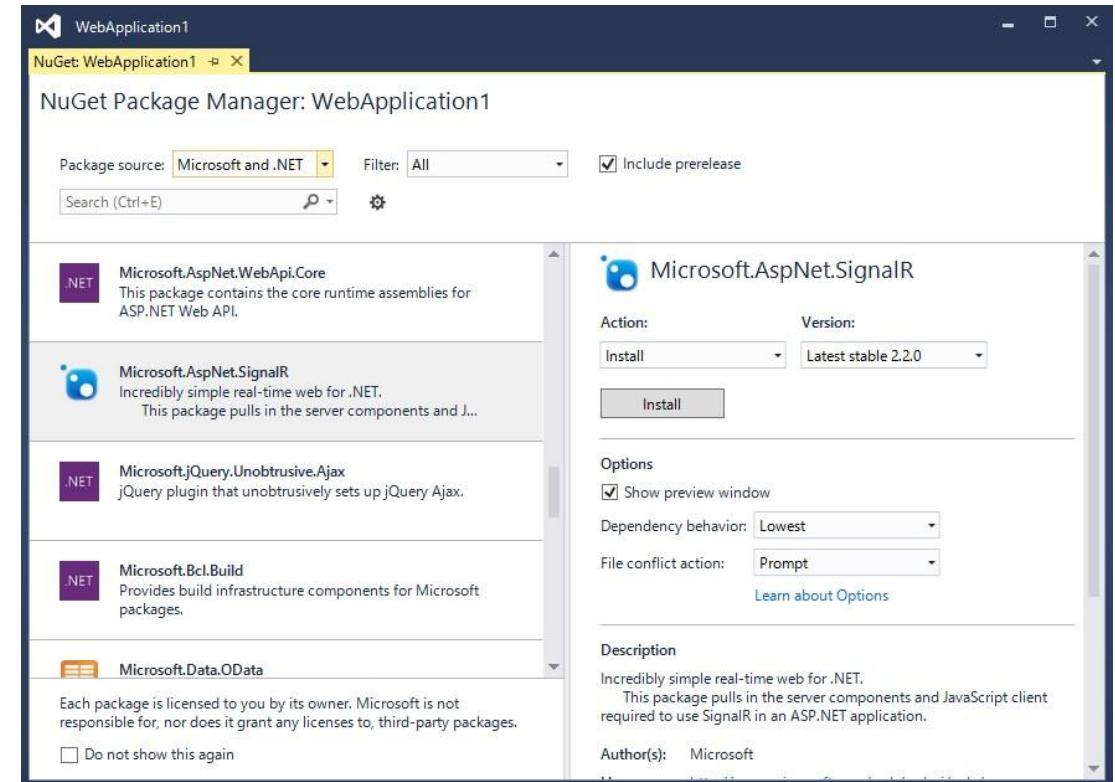
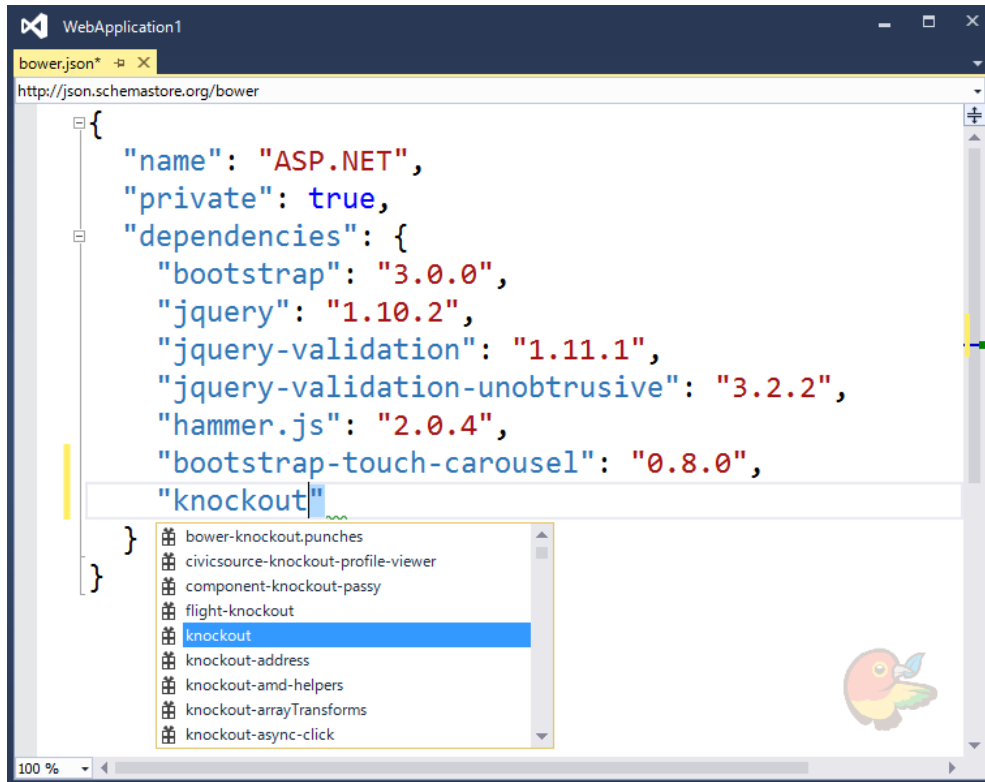
Bower

클라이언트 사이드
패키지 관리



NuGet

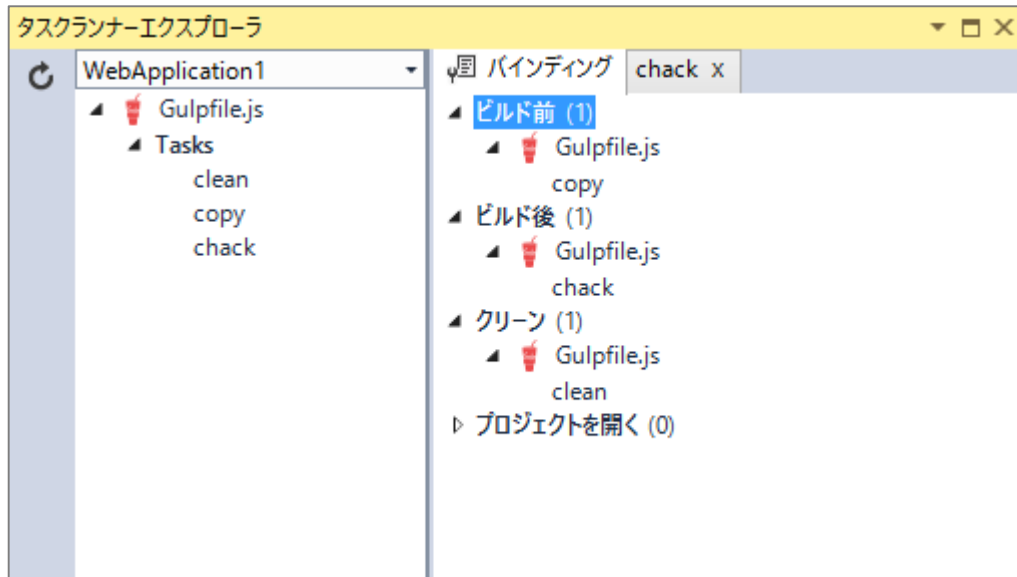
서버 사이드(.NET)
패키지 관리



태스크 런너 탐색기와 Gulp / Grunt

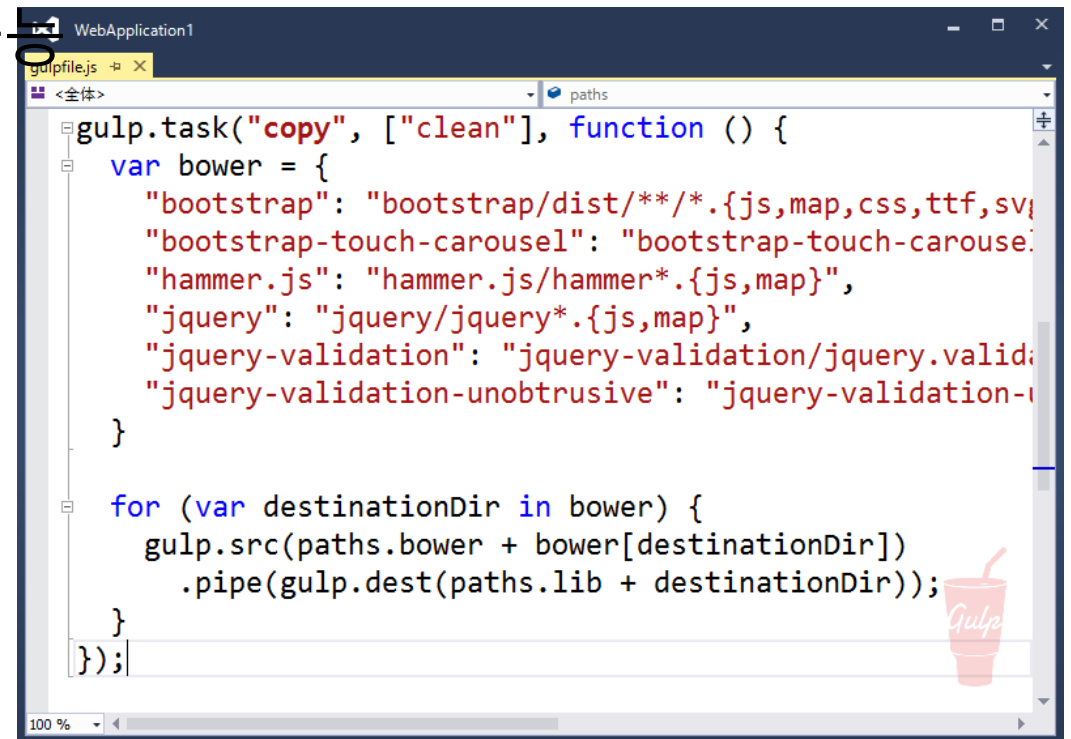
태스크 런너와 탐색기

- ✓ Grunt / Gulp
- ✓ Visual Studio 2015 – 기본 설치
- ✓ Visual Studio 2013 – 확장 기능으로 설치 가능



Gulpfile.js / Gruntfile.js

Gulp / Grunt 태스크 정의 파일
(VS 2015 の ASP.NET 5 표준은 Gulp)



복사만으로 배포 가능한 어플리케이션

- .NET 2015에는 오픈 소스화뿐만이 아니라 또 하나의 큰 변경이 있다.
- 그것은 패키지를 복사하는 것만으로 배포할 수 있다는 것이다.
- 그 동안의 .NET Framework은 모노릭으로 기본적으로 런타임 및 라이브러리나 프레임워크를 통째로 OS 위에 설치하고 그 위에서 애플리케이션을 동작시켰다.
- 그러나 .NET 2015부터는 아키텍처를 고쳐서 어플리케이션마다 런타임 및 컴포넌트를 골라 패키징 할 수 있게 된다.
- 구체적으로는 하나의 폴더 안에 애플리케이션 실행에 필요한 파일을 모두 채워 놓고, 그것을 타겟 머신에 복사하는 것만으로 배포는 완료된다.
- 즉, MacOS X나 Linux에 일일이 .NET 런타임 배포본을 설치하지 않아도 완전 새 OS 위에 애플리케이션을 폴더로 복사하면, 즉시 애플리케이션을 실행 할 수 있게 되었다.
- 클라우드에 있어서 어플리케이션의 대량 배포 등이 요구되는 장면에서 편리하게 될 것 같다.

.NET Core 앱 배치와 패키징

- 타겟 환경으로 xcopy 배치
 - 온 프레미스
 - Azure Web Apps / 가상머신
 - docker / Windows 컨테이너



- 애플리케이션과 같이 .NET Core도 배포 가능
- .NET Core의 머신 레벨 배치는 필요 없음

```
> dnu publish . -o .¥Publish
```

ASP.NET 5: Considerations for Production Linux Environments

날짜: 11 03, 2015 9:30오전-10:00오전 @ | 일 2

발표자: Glenn Condrón

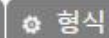
★★★★★ (1) | 290개 보기

평균: 5



```
bash: date: command not found
root@eb01d16e44c1:/# ls /bin
bash          chvt          grep          ls            open          setupcon     unicode_start
bunzip2       cp            gunzip       lsblk         openvt        sh           vdir
bzip2         cpio         gzexe       lsmmod       pidof         sh.distrib  which
bzcat         dd            gzip         mkdir         ping          sleep       whiptail
bzcmp         df            hostname     mknod         ping6         ss          ypdomainname
bzdiff        dir           ip           mktemp        plymouth      stty        zcat
bzegrep       dmesg        kbd_mode    more          plymouth-upstart-bridge su          zcmp
bzexe         dnsdomainname kill          mount         ps            sync        zdiff
bzfgrep       domainname   kmod         mountpoint   pwd           tailf       zegrep
bzip2         dumpkeys    less         mt            readlink     tar          zfgrep
bzip2recover  echo         lessecho    mt-gnu       rmdir        tempfile    zforce
bzless        egrep        lessfile    mv           run-parts    touch       zgrep
bzmore        false        lesskey     nc           running-in-container true        zless
cat           fgconsole   lesspipe    nc.openbsd  sed          udevadm     zmore
chgrp         fgrep        ln           netcat       setfont      umount      znew
chmod         findmnt     loadkeys    netstat      uname
chown         login       nisdomainname
root@eb01d16e44c1:/# echo
root@eb01d16e44c1:/# rm /bin/echo
root@eb01d16e44c1:/# echo
root@eb01d16e44c1:/# ls
bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
root@eb01d16e44c1:/# rm /bin/
```

47분, 49초



ASP.NET 5 Roadmap

Milestone	Release week
Beta6	27 Jul 2015
Beta7	2 Sep 2015
Beta8	15 Oct 2015
RC1	Nov 2015
RC2	Feb 2016
1.0.0	Q1* 2016

- Beta 6에서는 현지화, 서비스화, DNX에서 .NET Framewrk 4.6대응, HTTP.SYS 사용 및 SQL Server 분산 캐시 사용에 의한 고속화가 목표.
- Beta 7에서는 클래스 플랫폼, .NET Core의 Mac, Linux에 대한 출시가 목표.
- Beta 8에서는 기능 완성이 목표.
- RC에서는 안정성 향상이 목표.
- 1.0.0 이후의 목표는 VB지원, SignalR 3, Web Pages 4.

.NET OSS 커뮤니티의 성장

dotnetfoundation.org
dotnet.github.io
@dotnetfdn



{ 103 repositories
14,222 forks
2,501 contributors }

Growing daily...

<http://dotnet.github.io/>



dotnet / corefx

Watch

1,382

Star

8,889

Fork

2,045

Code

Issues 783

Pull requests 60

Wiki

Pulse

Graphs

This repo contains the .NET Core foundational libraries, called CoreFX. It includes classes for collections, file systems, console, XML, async and many others. <http://dotnet.github.io/>

6,513 commits

5 branches

3 releases

206 contributors

Branch: master

New pull request

New file

Find file

HTTPS

<https://github.com/dotnet/corefx>



Download ZIP



tmat Update PE-COFF.md

Latest commit 387ae53 an hour ago



Documentation

Fix spelling error in doc

2 hours ago



pkg

Initial integration of packaging support

22 days ago



src

Update PE-COFF.md

an hour ago



.editorconfig

Adding a clang-format file and a script to format every cpp and h fil...

4 months ago



.gitattributes

Add a gitattribute/editorconfig rule for .sh files

5 months ago



.gitignore

Change all project.json files to validate. Remove lockfiles from sour...

28 days ago



.gitmirrorall

Sync changes from TFS as of 2015/02/16 9:30:00 AM

11 months ago



CONTRIBUTING.md

Add folders to CoreFX Documentation

7 months ago



CoreFX-Build-1.1

Add CoreFX-Build-1.1 file which generates builds and commit default...

4 days ago

<https://github.com/dotnet/corefx>

Microsoft Reference Source

Project	Namespaces	Code
<ul style="list-style-type: none"> () System <ul style="list-style-type: none"> () Net <ul style="list-style-type: none"> () PeerToPeer () Sockets <ul style="list-style-type: none"> FormatMessageFlags INetworkProgress IPEndPointCollection IUnsafeWebRequestCreate Logging NetworkProgressChangedEvent P2PTraceSource SafeLoadLibrary UiSynchronizationContext UnsafeSystemNativeMethods 		<pre> 23 [Obsolete("This API supports the .NET Framework 3.5 or earlier")] 24 [EditorBrowsable(EditorBrowsableState.Never)] 25 event EventHandler<NetworkProgressChangedEventArgs> NetworkProgressChanged; 26 } 27 28 //introduced for supporting design-time loading of 29 [Obsolete("This API supports the .NET Framework in 3.5 or earlier")] 30 [EditorBrowsable(EditorBrowsableState.Never)] 31 public class NetworkProgressChangedEventArgs : ProgressEventArgs 32 { 33 //introduced for supporting design-time loading of 34 [Obsolete("This API supports the .NET Framework in 3.5 or earlier")] 35 [EditorBrowsable(EditorBrowsableState.Never)] 36 public NetworkProgressChangedEventArgs(int percentage) 37 { 38 // Properties 39 //introduced for supporting design-time loading of 40 [Obsolete("This API supports the .NET Framework in 3.5 or earlier")] 41 [EditorBrowsable(EditorBrowsableState.Never)] 42 public int ProcessedBytes { get { throw new PlatformNotSupportedException(); } } 43 44 //introduced for supporting design-time loading of 45 [Obsolete("This API supports the .NET Framework in 3.5 or earlier")] 46 [EditorBrowsable(EditorBrowsableState.Never)] </pre>

common target platform

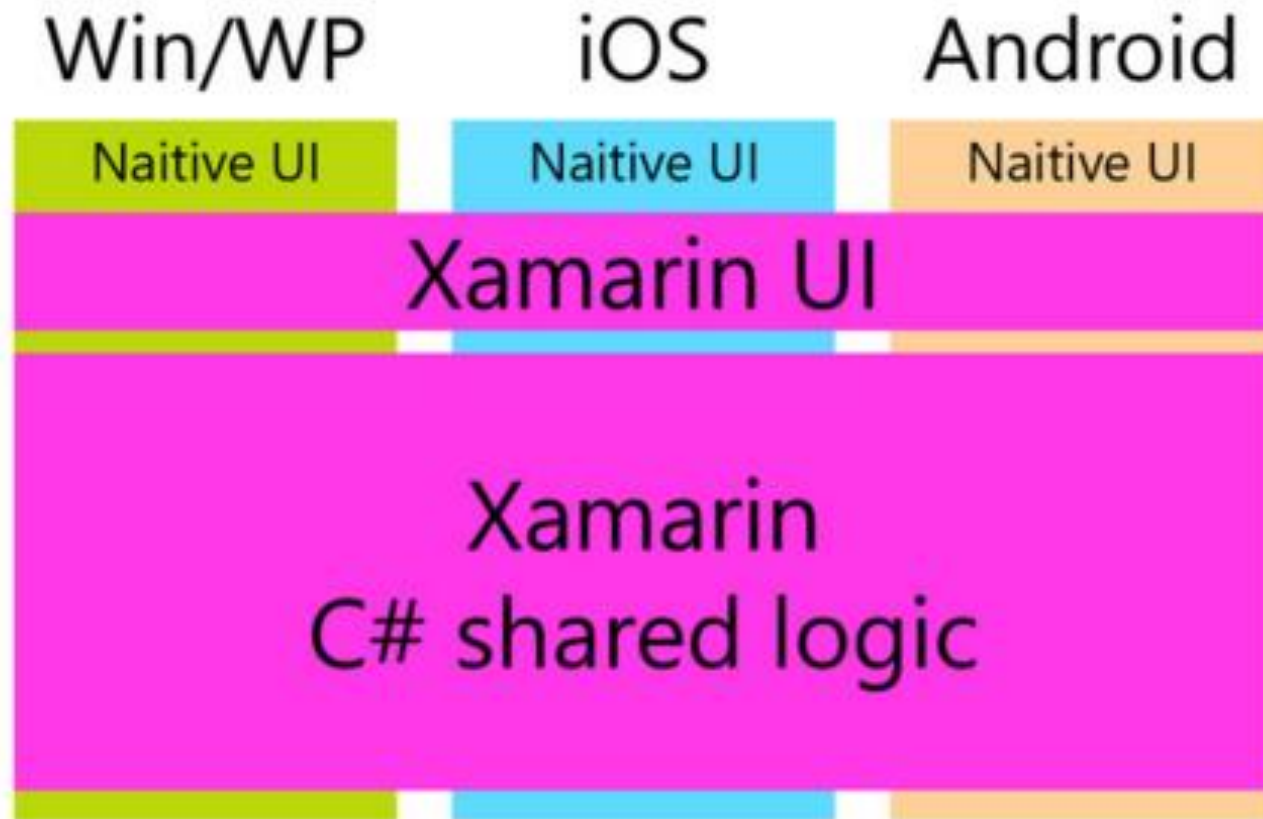
Languages

- C#
- C++
- HTML5

Solutions

- Xamarin
- Native
- Apache Cordova

C# Model : Cross-platform Apps



C++ Model : Cross-platform Apps

Win/WP

iOS

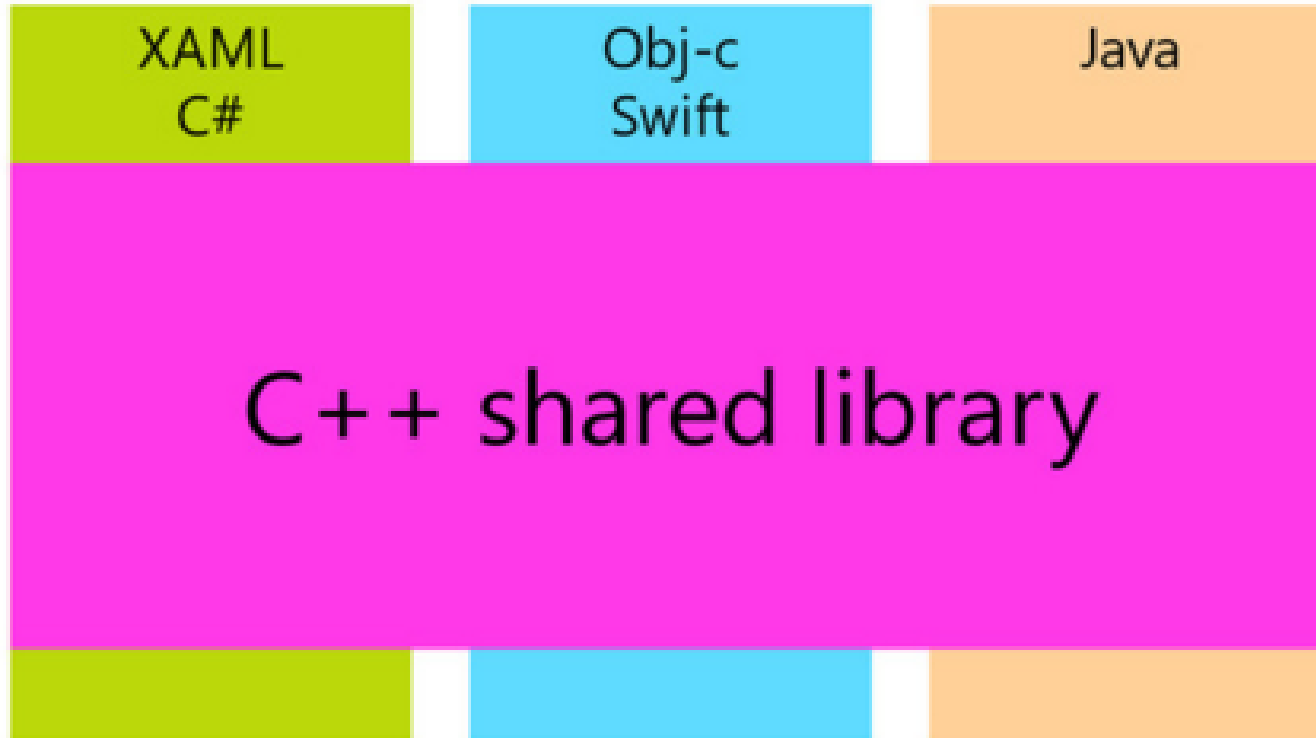
Android

XAML
C#

Obj-c
Swift

Java

C++ shared library



HTML5 Model : Cross-platform Apps

Win/WP

iOS

Android

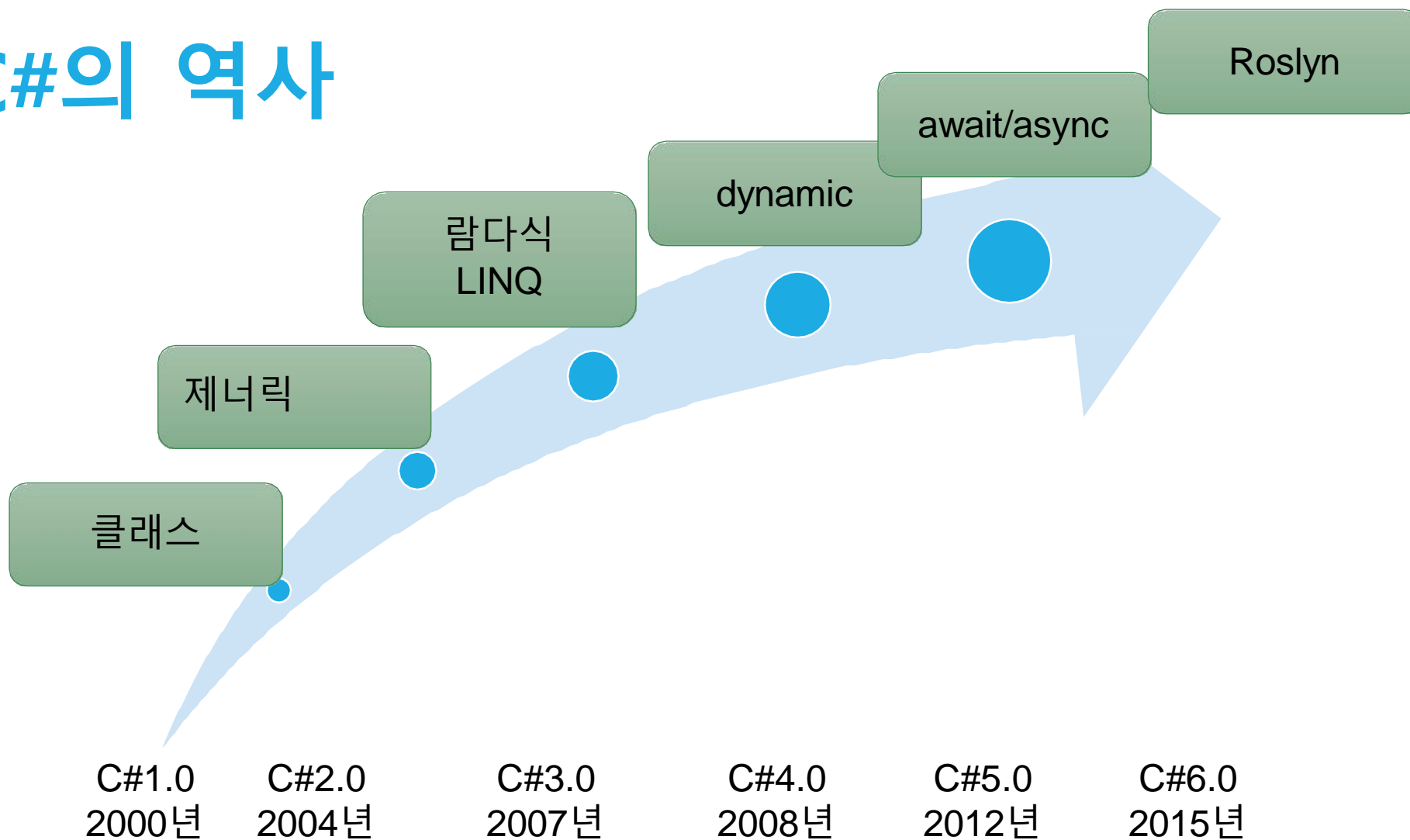
Apache CORADOVA
HTML, CSS, JavaScript



C# 6.0

What's new?

C#의 역사



광범위한 C# 이용



C# 6.0

- Visual Studio 2015에서 사용 할 수 있다.
- Roslyn \neq C# 6.0
- 큰 변경은 없고, 세세한 수정이 대 부분.

Roslyn ?

- .NET Framework의 차세대 컴파일러 플랫폼
- Compiler APIs - 컴파일러
- Feature APIs – Code Fix나 리팩토링 등
- Workspace APIs – 프로젝트 파일이나 참조 관계

C# 6.0 구현 현황

Feature	Example
Auto-property initializers	<code>public int X { get; set; } = x;</code>
Getter-only auto-properties	<code>public int Y { get; } = y;</code>
Ctor assignment to getter-only autoprops	<code>Y = 15</code>
Parameterless struct ctors	<code>Structure S : Sub New() : End Sub : End Structure</code>
Using static members	<code>using System.Console; ... Write(4);</code>
Dictionary initializer	<code>new JObject { ["x"] = 3, ["y"] = 7 }</code>
Await in catch/finally	<code>try ... catch { await ... } finally { await ... }</code>
Exception filters	<code>catch(E e) if (e.Count > 5) { ... }</code>
Expression-bodied members	<code>public double Dist => Sqrt(X * X + Y * Y);</code>
Null propagation	<code>customer?.Orders?[5]?.\$price</code>
String interpolation	<code>(\$"{p.First} {p.Last} is {p.Age} years old."</code>
nameof operator	<code>string s = nameof(Console.Write);</code>
#pragma	<code>#Disable Warning BC40008</code>
Extension Add in collection initializers	
Improved overload resolution	

Auto-property initializers

자동 구현 프로퍼티의 초기화를 생성자에서 가능!

Before

```
class Person
{
    public int Age { get;set; }

    public Person()
    {
        Age = 20;
    }
}
```

Next

```
class Person
{
    public int Age { get;set; } = 20;
}
```

Getter-only auto-properties

Getter만의 자동 구현 프로퍼티 초기화도 가능

Before

```
class Person
{
    public int Age { get { return this._age; } }
    private int _age = 20;
}
```

Next

```
class Person
{
    public int Age { get; } = 20;
}
```

Using static members

Before

```
class Person
{
    public void Say()
    {
        System.Console.WriteLine("Name is " + this.Name);
    }
}
```

After

```
using System.Console;

class Person
{
    public void Say()
    {
        Console.WriteLine("Name is " + this.Name);
    }
}
```

Await in catch/finally

Catch문과 Finally 문에서 await를 사용할 수 있다

Before

```
Exception exception = null;  
try  
{  
    //HodeHoge  
}  
catch (Exception e)  
{  
    exception = e;  
}  
if (exception != null) awa  
    it Task.Delay(2000);
```

After

```
try  
{  
    //HodeHoge  
}  
catch (Exception e)  
{  
    await Task.Delay(2000);  
}
```

Exception filters

Catch문에 조건 식을 부여할 수 있다

Before

```
try
{
    //HodeHoge
}
catch (Exception e)
{
    if (e.Data.Count > 10)
    {
    }
}
```

After

```
try
{
    //HodeHoge
}
catch (Exception e) if
(e.Data.Count > 10)
{
    // 예외 처리
}
```

Null propagation

Null 조사를 간단하게 기술

Before

```
public int? GetNameLength(Person person)
{
    if (person == null) return null;
    if (person.Name == null) return null;
    return person.Name.Length;
}
```

After

```
public int? GetNameLength(Person person)
{
    return person?.Name?.Length;
}
```

nameof operator

멤버나 변수 이름을 돌려주는 연산자

Before

```
class Person
{
    public string Name { get; set; }

    public Person(string name)
    {
        if (string.IsNullOrEmpty(name))
            throw new ArgumentException("name" + " is NullOrEmpty");
        this.Name = name;
    }
}
```

After

```
public string Name { get; set; }

public Person(string name)
{
    if (string.IsNullOrEmpty(name))
        throw new ArgumentException(nameof(name) + "is NullOrEmpty");
    this.Name = name;
}
```


String interpolation

문자열 보완을 할 때 변수를 직접 지정할 수 있다

```
int a = 3;  
int b = 4;  
int result = a * b;
```

Before

```
Console.WriteLine("{0} × {1:2D} = {2:0.0}", a, b, result);
```

After

```
Console.WriteLine($"{a} × {b:2D} = {result:0.0}");
```

C# 6.0 in Unity



We believe that Unity developers deserve to have access to the newest development tools. After looking at the specification of C# 6.0 it was so amazing that I couldn't wait but look to integrate it into Unity!

Unfortunately Unity uses a very old version of .NET framework, and an old version of C# compiler. However the great news is that C# language is independent from the platform it runs on, and we can generate .NET 3.5 assemblies from C# 6.0 code.

<http://www.unityninjas.com/new-technology/c-6-in-unity/>

The VS Code [December release](#) is now available along with hundreds of cool [extensions](#) and [themes](#)!

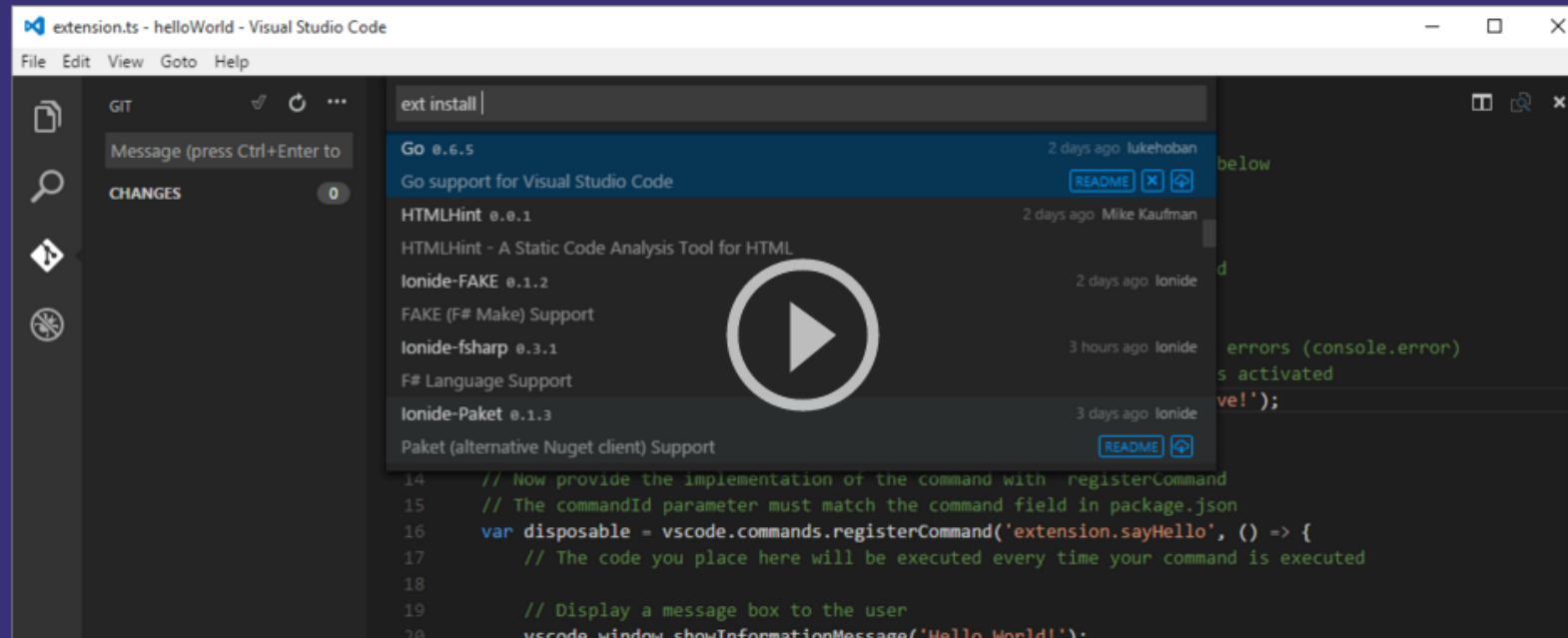
Code Editing. Redefined.

Build and debug modern web and cloud applications. Code is free and available on your favorite platform - Linux, Mac OSX, and Windows.

Download Code for Windows

Available on other platforms

By installing Visual Studio Code, you agree to the terms and data collection described [here](#)

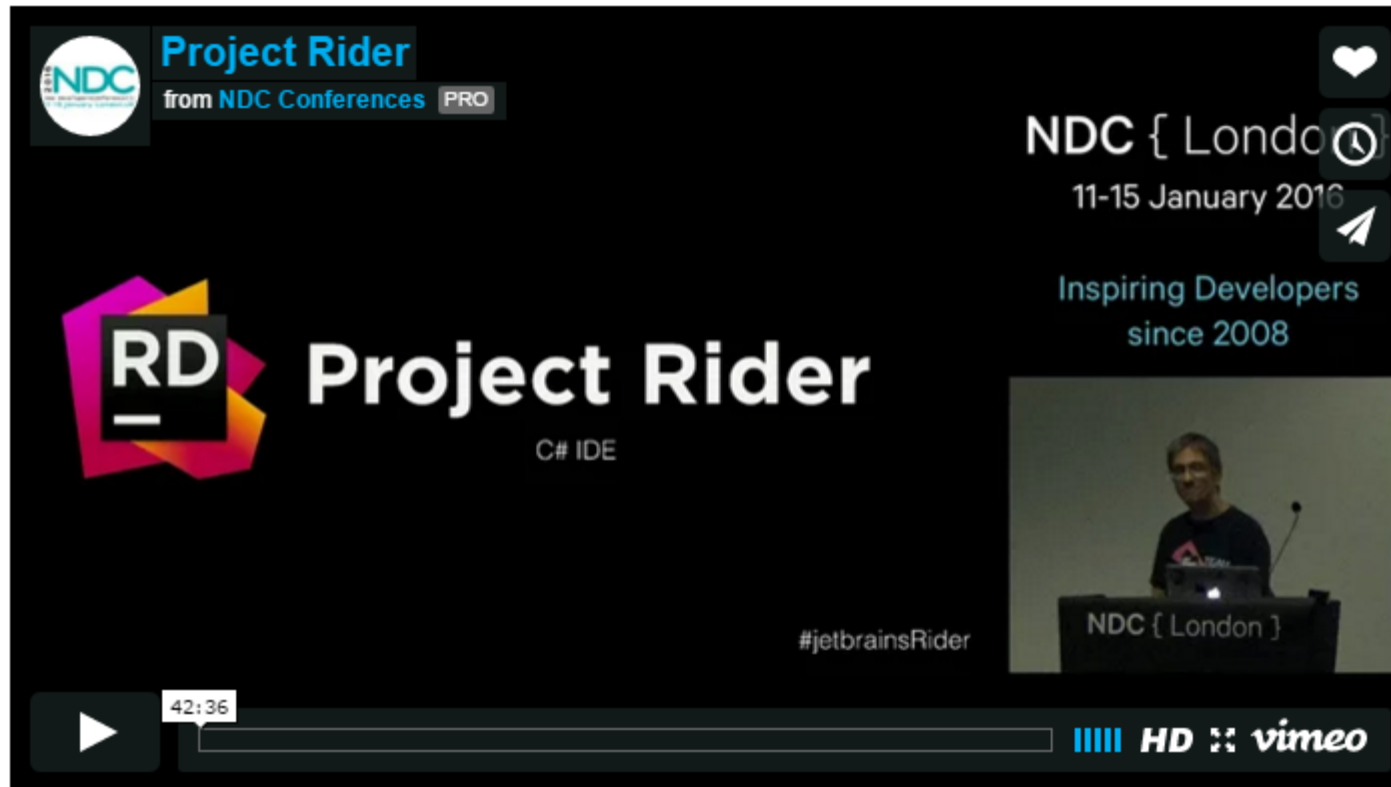


<https://code.visualstudio.com/>

Project Rider – A C# IDE

Posted on January 13, 2016 by Matt Ellis

UPDATE: NDC London organizers have kindly uploaded the recording of Project Ride presentation. Enjoy!



<http://blog.jetbrains.com/dotnet/2016/01/13/project-rider-a-csharp-ide/>

참고

Announcing . NET Framework 4.6-. NET Blog

<http://blogs.msdn.com/b/dotnet/archive/2015/07/20/announcing-net-framework-4-6.aspx>

ASP.NET 5 Schedule and Roadmap

<https://github.com/aspnet/Home/wiki/Roadmap>