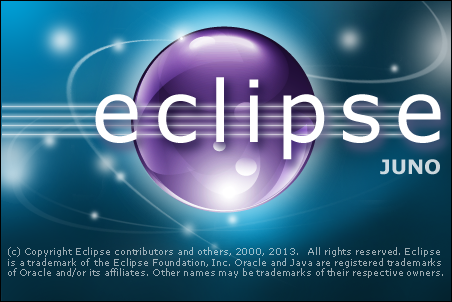
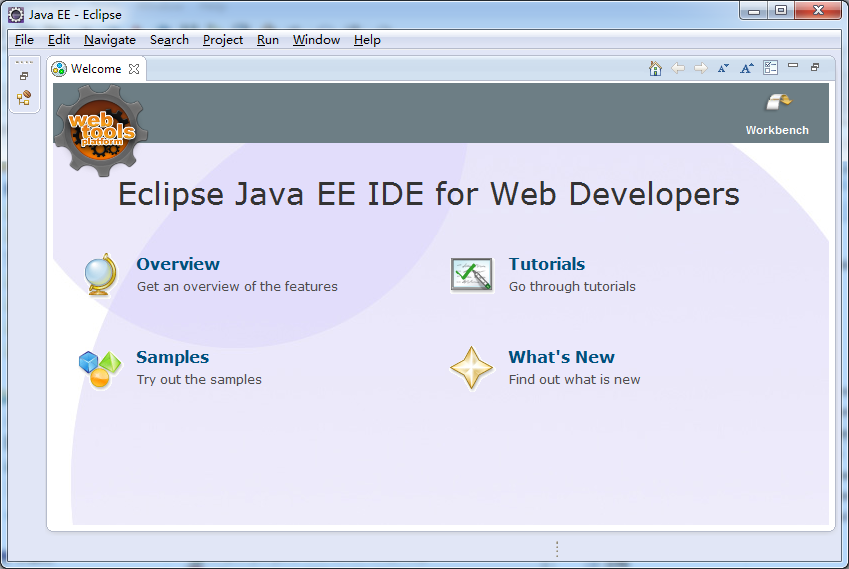
**설치**

Node.js 설치

<http://www.nodejs.org/download/>

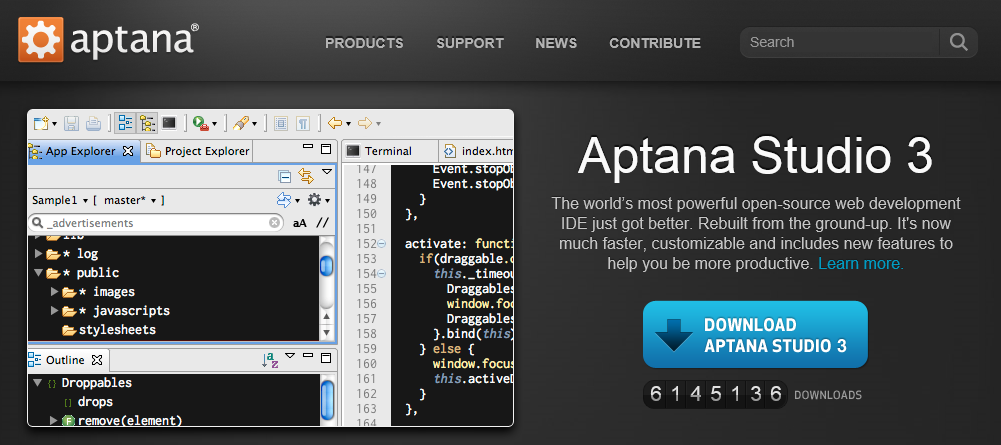


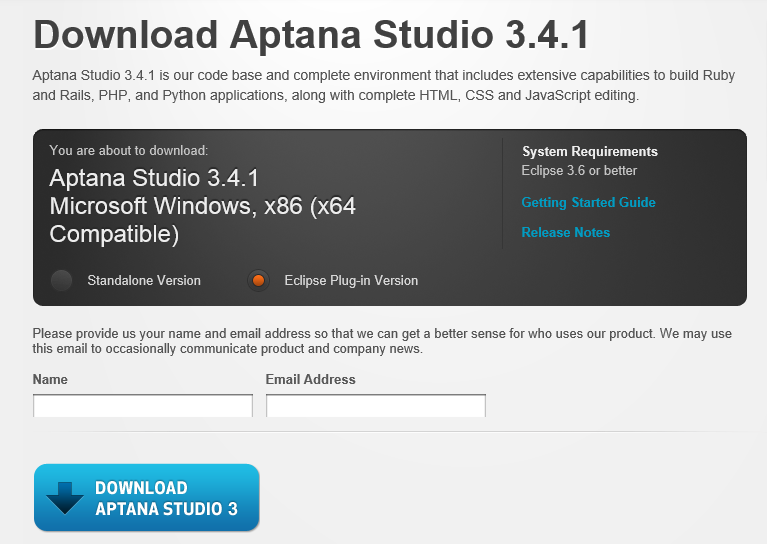
1) Get latest Eclipse from <http://www.eclipse.org/downloads/>.  
Recommended is [Eclipse IDE for Java EE Developers](http://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/junosr2) Juno (4.2) SR2, that contains Eclipse web tools.

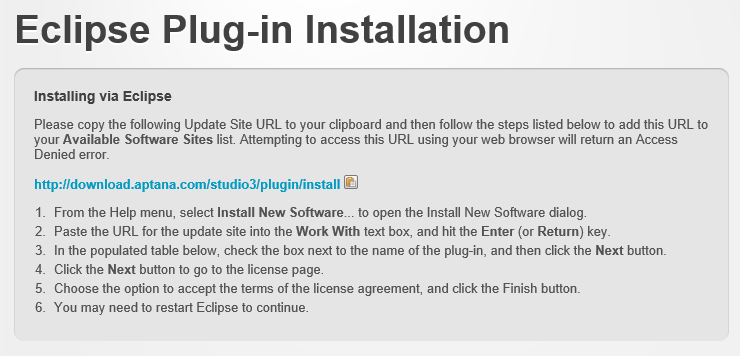
[](https://github.com/Nodeclipse/eclipse-node-ide/blob/master/Pictures/Eclipse-Welcome.png)

Aptana Studio 를 설치해야되는데 우선 <http://www.aptana.com/> 로 이동한다.

[출처] [Javascript Web Application 구현을 위한 환경 구축](http://blog.naver.com/beardless/10173140828)|작성자 [22602](http://blog.naver.com/beardless)

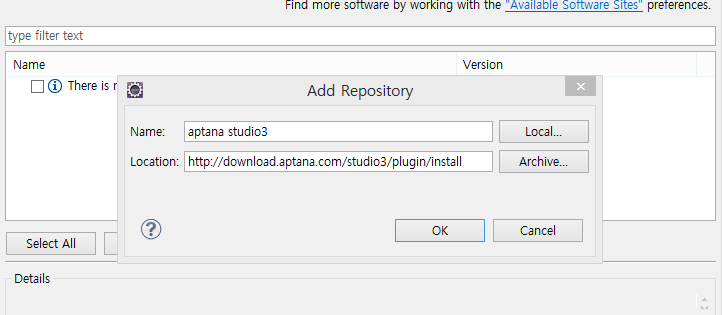


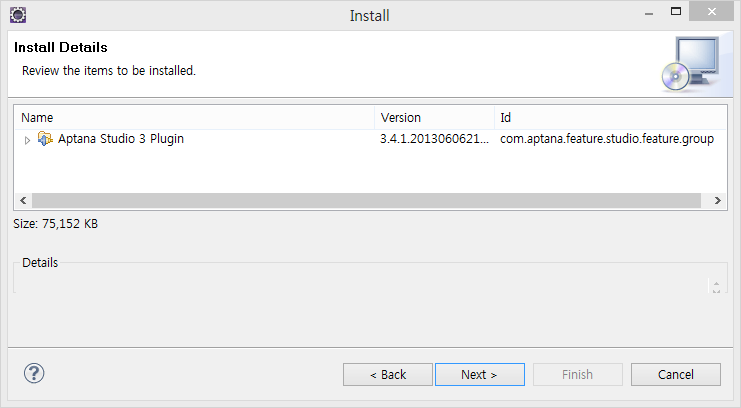


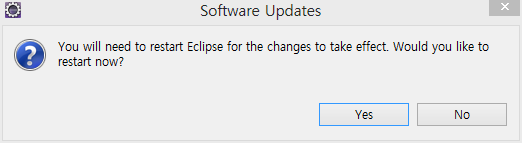


[http://download.aptana.com/studio3/plugin/install](http://www.aptana.com/downloads/start)

상기 URL을 복사해서 실제 설치는 eclipse에서 진행한다.



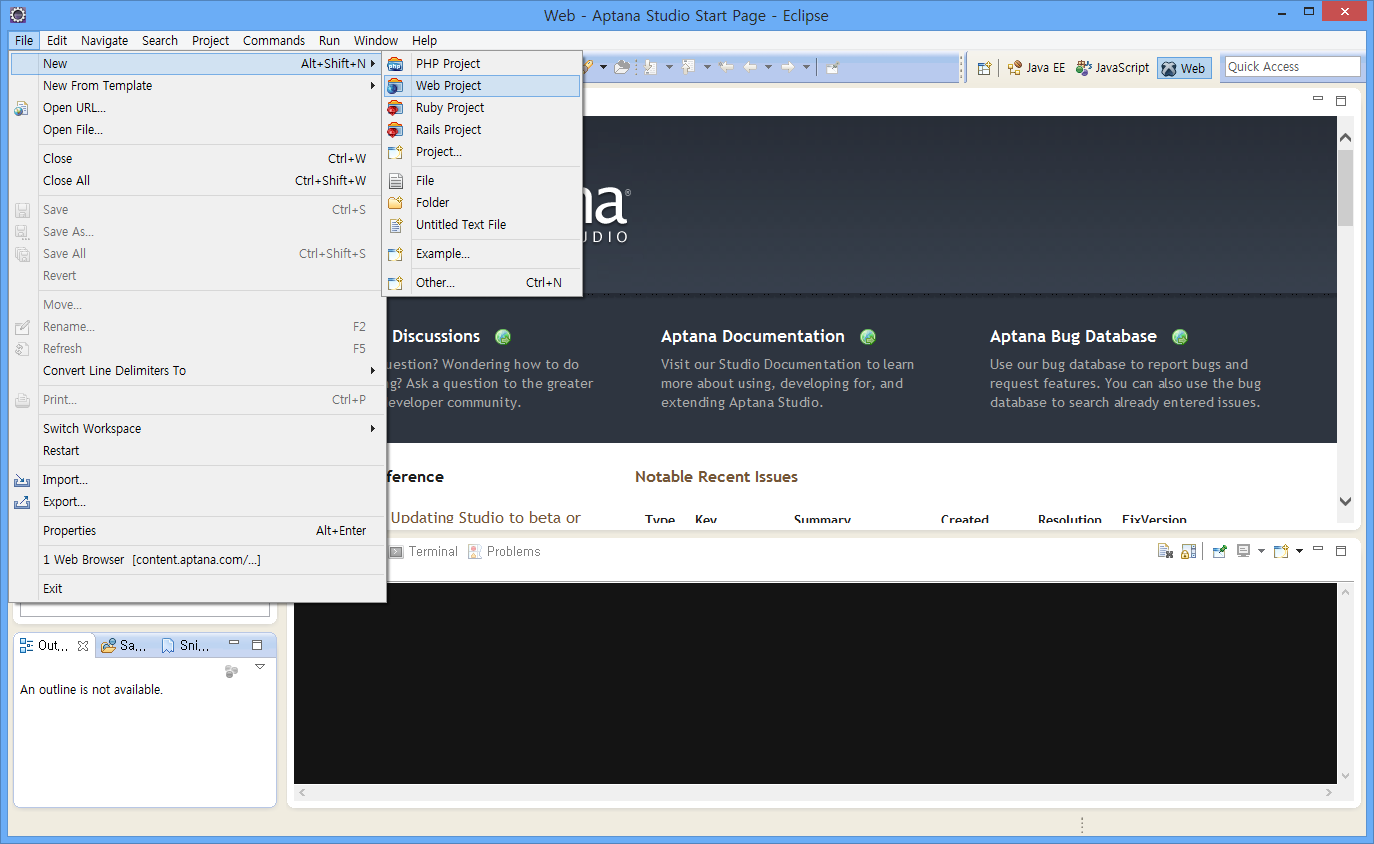


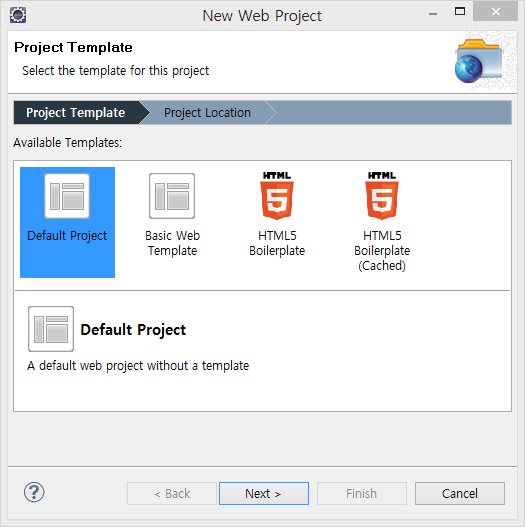


설치 완료 후 실행 시 첫 화면

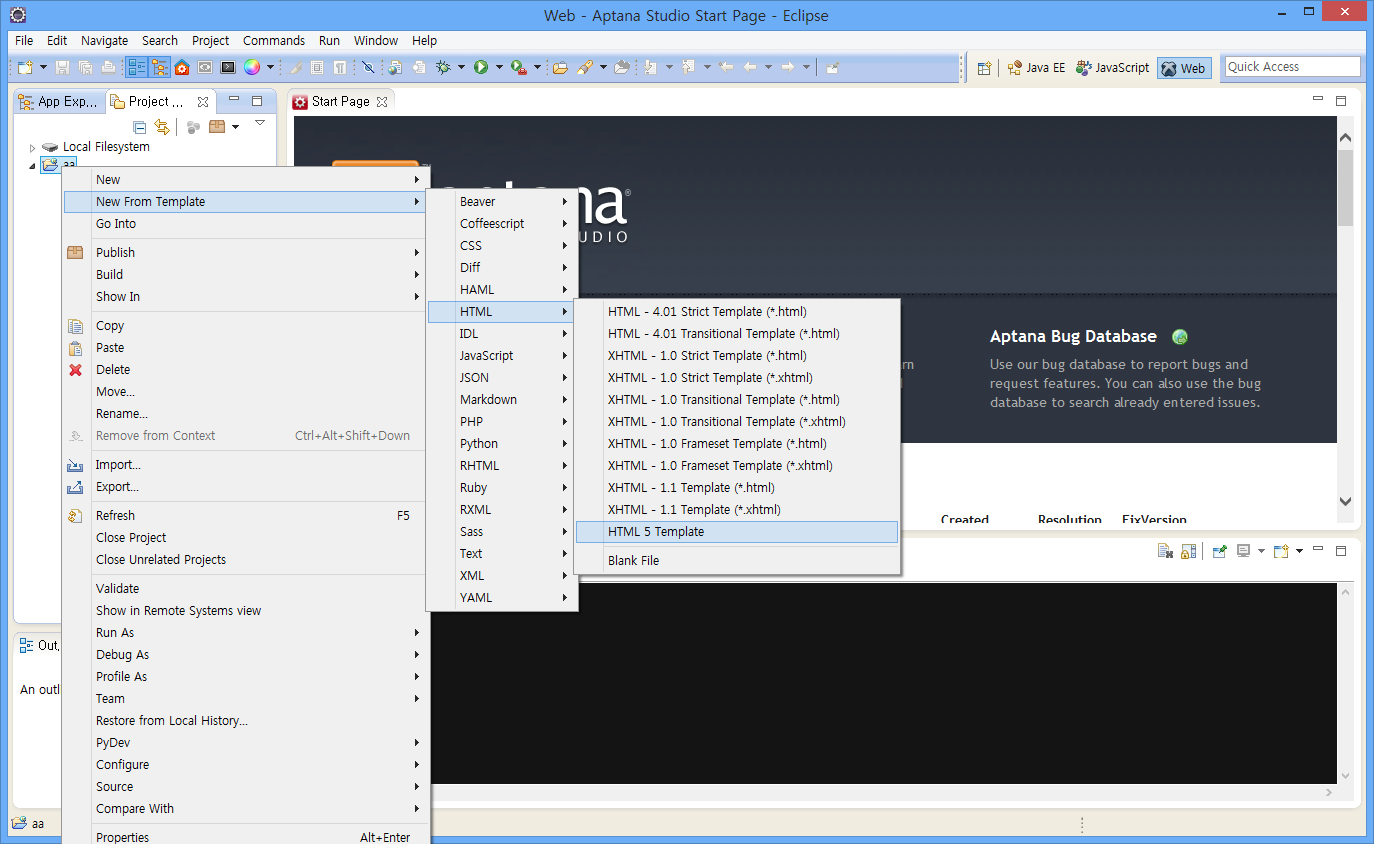


Perspective를 web으로 바꾸고 새 프로젝트 생성한다 (web Project)





10. 생성된 프로젝트에서 오른쪽 버튼을 누르면 New From Template 라는 메뉴가 있는데 여기서는 사용자가 원하는 템플릿으로 파일이 생성되어 코딩하는 데 편의를 제공한다. 우리는 HTML5 를 사용하기 때문에 HTML  메뉴에서 HTML 5 템플릿을 선택하고 파일 이름을 입력한 후 생성하면 html 파일이 프로젝트 내에 생성되어진다.



11. 만들어진 html 파일의 내용을 보면 사용자의 편의를 제공하기 위해 이미 레이아웃이 어느정도 갖춰진 상태다. 디버거 설치를 위해 바로 디버깅을 해도 상관없고,

<!DOCTYPE html>

<html lang="ko">

<head>

<meta charset="utf-8" />

<title>02-01</title>

</head>

<body>

<script type="text/javascript" src=02-01.js>

document.writeln('외부 스크립트를 사용할 때는 %lt;script&gt; 태그 아래 내용이 무시됩니다.');

</script>

<script type="text/javascript">

document.writeln('&lt;script&gt; 태그 아래의 인라인 스크립트에 의한 출력입니다.<br/>');

</script>

</body>

</html>

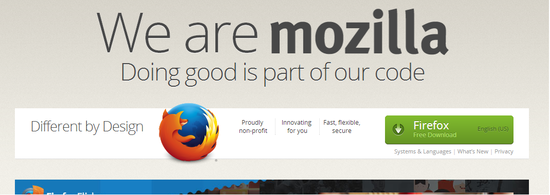
이 소스를 복사하여 html 파일에 붙여넣기 한 후 디버깅을 해도 무관하다.

[출처] [Javascript Web Application 구현을 위한 환경 구축](http://blog.naver.com/beardless/10173140828)|작성자 [22602](http://blog.naver.com/beardless)

**디버깅**

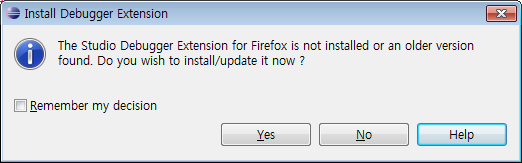
8. 다음에는 자바스크립트 애플리케이션을 구현하는 과정에 있어서 오류를 잡아내기 위한 디버거를 설치하는데 일단 그 전에 firefox 라는 브라우저가 필요하다. 현재 aptana 에서 제공하는 디버거는 파이어폭스 브라우저와만 호환이 된다. ( 이 부분은 아직 자세히 잘 모르겠다.)

파이어폭스 브라우저는 <http://www.mozilla.org/en-US/> 이 URL에 접속하게 되면 사이트 우측에 녹색 버튼이 있고 이 버튼을 클릭하면 파이어폭스 브라우저를 다운로드 할 수 있다.

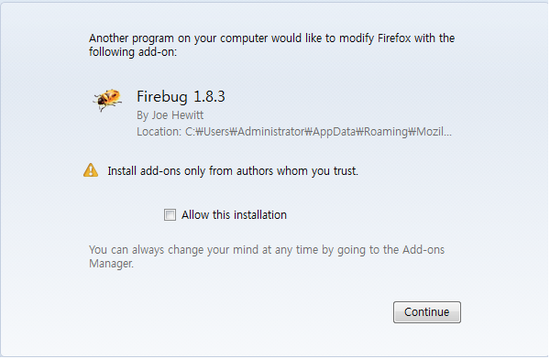


[출처] [Javascript Web Application 구현을 위한 환경 구축](http://blog.naver.com/beardless/10173140828)|작성자 [22602](http://blog.naver.com/beardless)

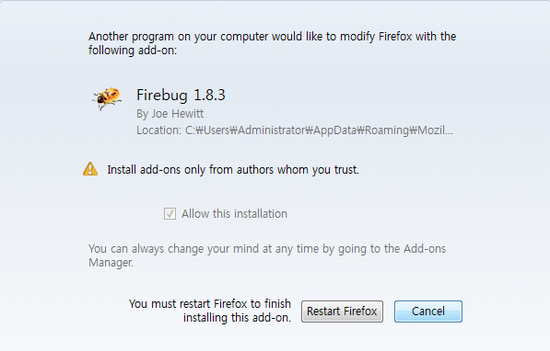
디버깅을 하기 위해서 메뉴 Run-Debug 나 F11 버튼, 혹은 벌레 모양의 아이콘http://postfiles11.naver.net/20130726_298/beardless_1374848955185trcpQ_PNG/%C1%A6%B8%F1_%BE%F8%C0%BD.png?type=w3(메뉴 바로 밑에 존재)을 누르면 디버깅이 실행된다. 현재는 디버거가 설치되어 있지 않기 때문에 디버거를 설치하겠냐는 팝업창이 출력된다.



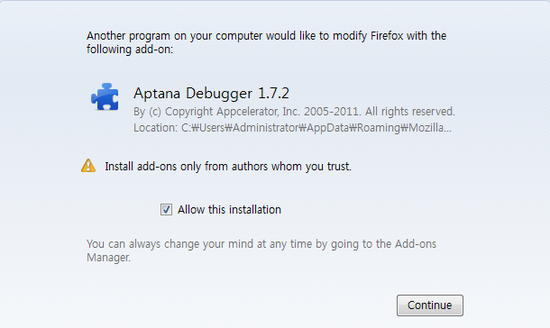
12.  위 화면에서yes를 누르면 중간에 메시지가 나오는데 OK 를 누르면 파이어폭스가 실행되면서



위 화면과 같이 Firebug 1.8.3 을 설치하는 화면이 나오게 된다. Allow this installation 을 체크하고 Continue 버튼을 누른다.

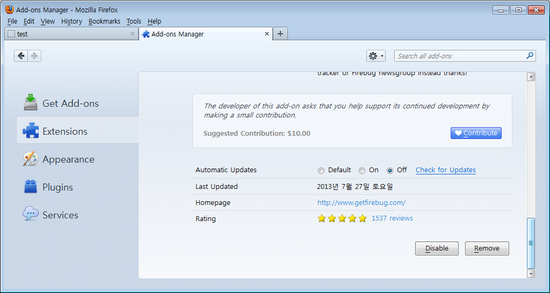


Continue 버튼을 누르면 위와 같은 화면이 출력되고 Restart Firefox 버튼을 누르면 13

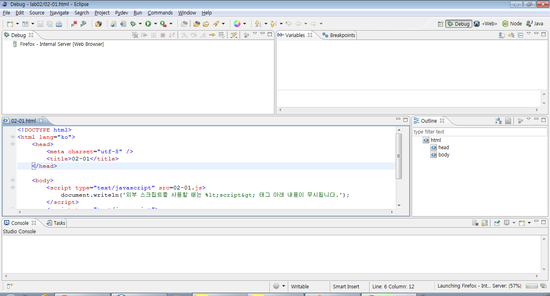


Aptana Debugger 1.7.2 를 설치하라는 화면이 출력된다. 설치 과정은 동일하므로 이하 그림 설명생략

※ 깜빡 한게 있는데 Aptana Studio가 아직 Firebug 1.8 버전까지만 지원하고 있어서, 자동 업데이트를 막아야지만 현재 버전으로 디버깅을 할 수가 있다. 즉, 업그레이드를 해버리면 버전이 너무 높아져 버려서 디버깅이 불가능하게 된다. 자동 업데이트를 막는 방법은 파이어폭스 메뉴에서 Tools - Add ons메뉴에 들어가면 좌측에 extensions 라는 항목이 있는데 누르면 Firebug 1.8.3 이 있을 것이다. 제목 옆에 more 이라는 링크가 있는데 누르면 자세한 설명이 나오고 맨 하단에 자동 업데이트 여부를 선택하는 항목이 있을 것이다. 거기서 off를 선택하면 더 이상 업데이트를 하지 않게 된다.

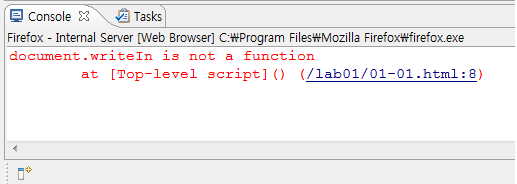


13. 다시 이클립스로 돌아와서 디버깅을 하게 되면



이런 형태로 디버깅이 가능하게 된다.

실제로 오타를 내고 실행을 하게 되면



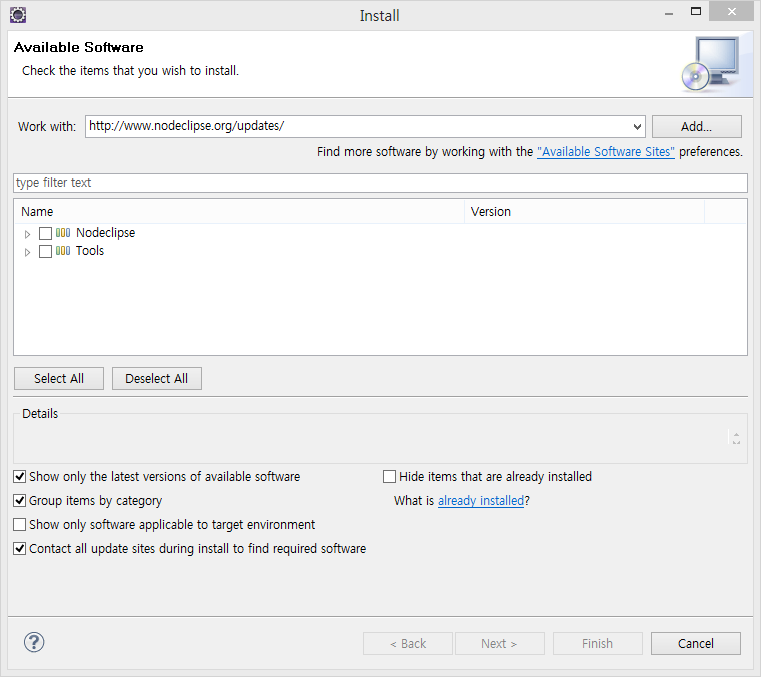
이런 식으로 콘솔에 어떤 오류가 발생했는지에 대한 정보를 출력해준다.

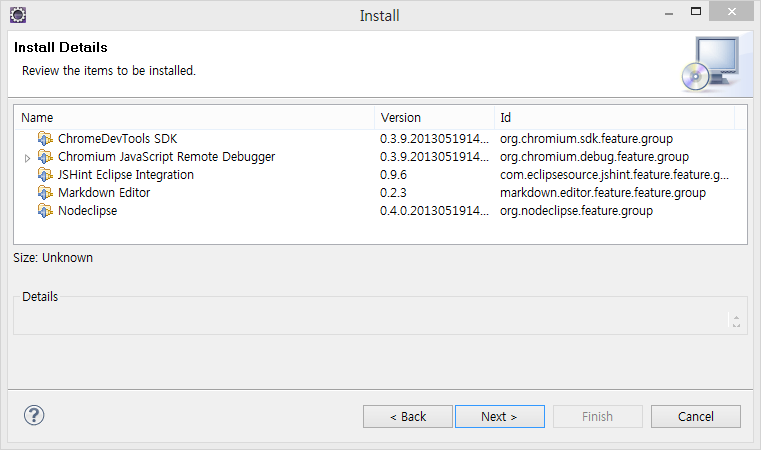
[출처] [Javascript Web Application 구현을 위한 환경 구축](http://blog.naver.com/beardless/10173140828)|작성자 [22602](http://blog.naver.com/beardless)

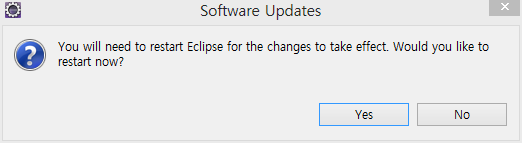
**Nodeclipse 설치**

<http://www.nodeclipse.org/>

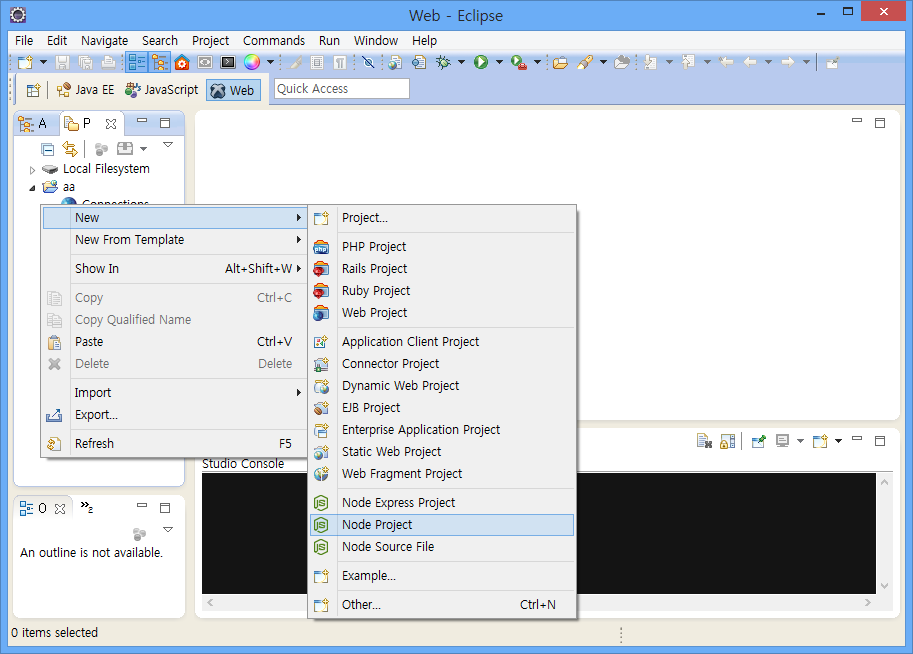
<http://www.nodeclipse.org/>updates/

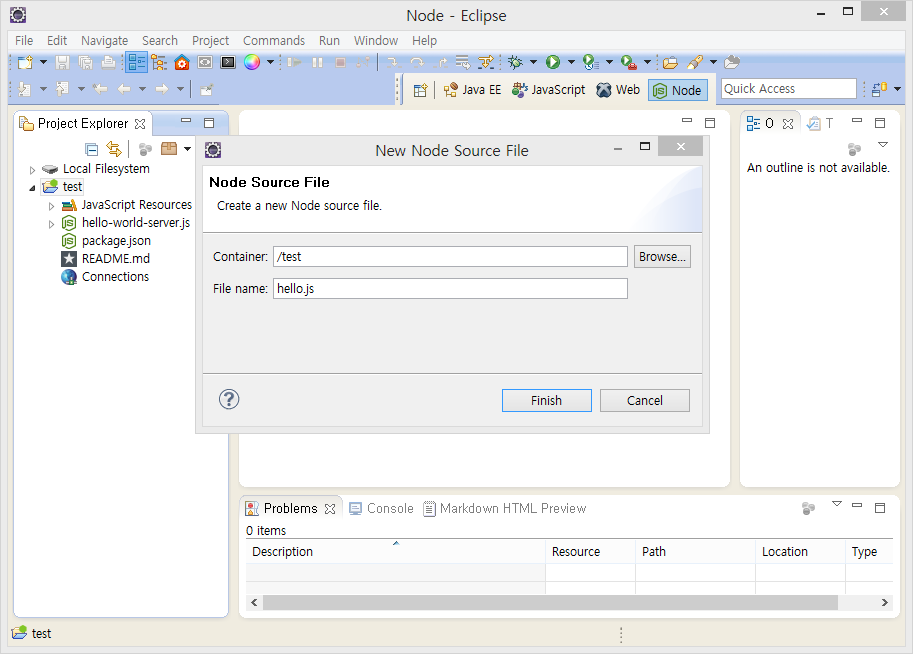




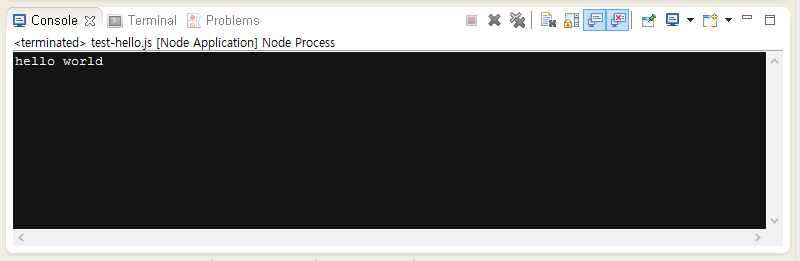


Node Project 프로젝트를 생성합니다.

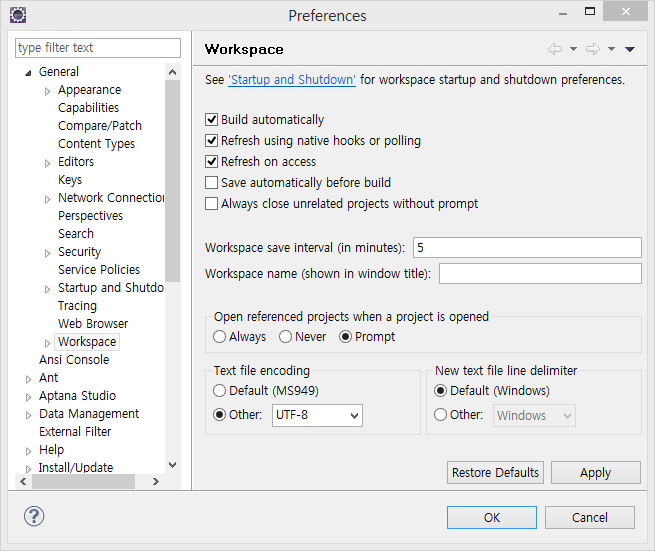




Run As > Node Application 을 선택하여 눌러 디버깅을 실행시킨다.

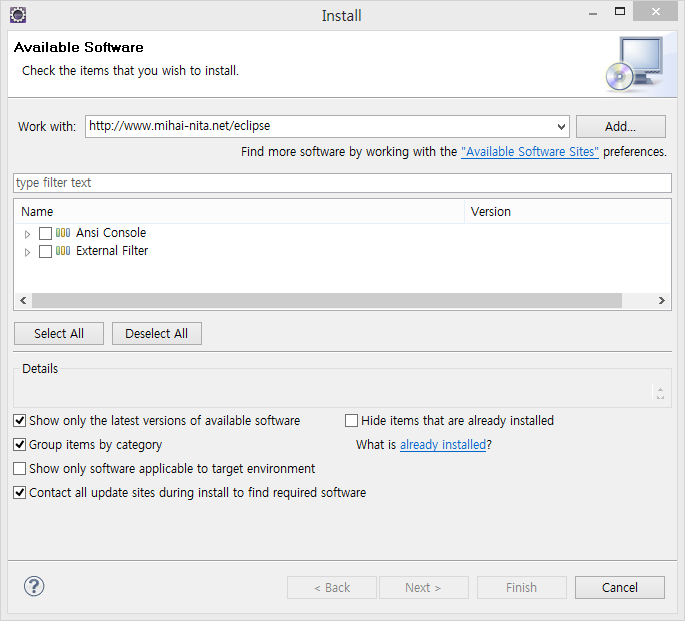


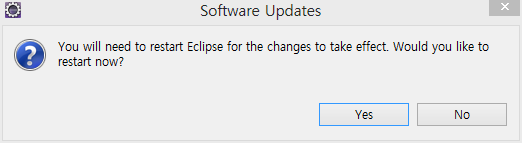
하지만 한글을 출력시켜보면 한글은 깨져서 출력된다. 문서를 UTF-8로 저장되도록 설정한다.



그래도 되지 않는다면 다음 업데이트를 실행해 본다.

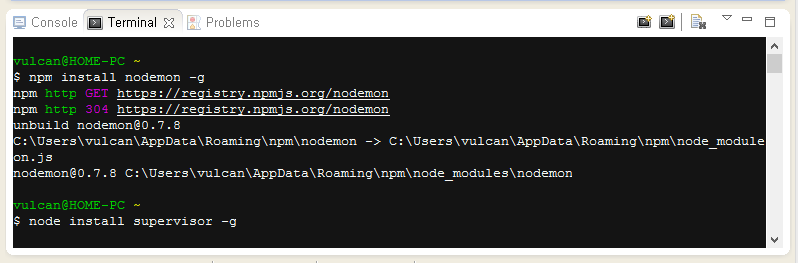
<http://www.mihai-nita.net/eclipse>





**모듈 설치**

Terminal 창에서 각 모듈을 설치할 수 있다.



console.log("프로젝트 진행 방법.js 파일을 참고하세요");

/\*

//--------------------------

// 모듈 설치

//--------------------------

\* 개발중인 app소스 자동으로 적용시켜주는 모듈

(소스 수정 후 매번 다시 실행할 필요 없도록 해줌)

\* nodemon 모듈 설치

npm install nodemon -g

nodemon "실행 js파일"

\* supervisor 모듈 설치

npm install supervisor -g

supervisor "실행 js파일"

\* 프로젝트 생성을 위한 express 프레임웍

\* express 설치

npm install express -g

//--------------------------

// express 프로젝트 생성

//--------------------------

\*원하는 경로에서 하위 폴더로 express 프로젝트 생성

\* 생성

express "프로젝트 디렉토리"

\* 필요한 의존 모듈 설치 (package.json 파일의 description 항목 참고)

cd "프로젝트 디렉토리" && npm install

\* app 실행 (웹서버 실행)

node app

\* 웹브라우져에서 http://localhost:3000 으로 접속하여 페이지 뜨는것을 확인한다.

서버를 멈추려면 콘솔창에서 Ctrl+C를 누른다.

\* 현재 하위 폴더에 설치되어 있는 모듈 확인

npm list

**Code Assist 플러그인 설치**

<http://zlukz.tistory.com/81>

<http://download.eclipse.org/vjet/updates-nightly/>

