**[ 1 ] 웹 프로그래밍 및 개발환경 설정**

1. 웹 프로그래밍이랑?
   * 웹프로그래밍이란, 웹어플리케이션(웹에서 돌아가는 프로그램)을 구현하는 행위 입니다.
   * **웹어플리케이션이란, 웹을 기반으로 작동되는 프로그램**입니다.
   * 웹이란, 1개 이상의 사이트가 연결되어있는 인터넷 서비스의 한가지 형태를 말합니다.

…

* + 인터넷이란, 1개 이상의 네트워크가 연결되어 있는 형태를 말합니다.
  + 서로 서로 컴퓨터가 연결된 것을 네트워크가 연결되어 있다고 하고 이 네트워크를 통해 서비스 하는 것을 우리는 인터넷 서비스라 한다. 인터넷 서비스에는 종류가 굉장히 다양합니다. 그 다양한 서비스 종류 중에 web이 있죠. [www.~](http://www.~)을 통해서 사이트에 방문합니다. 즉 웹이란 컴퓨터가 여러 대 네트워크 상에 연결되어 있고 연결된 컴퓨터끼리 인터넷 서비스가 이루어집니다. 그 인터넷 서비스 중에 하나가 웹. 그럼 웹 어플리케이션을 만든다는 것은 한 컴퓨터에서만 구동되는 프로그램이 아니고, 만들기는 한 컴퓨터에서 만들었지만, 웹 인터넷 서비스를 통해 여러 군데 어플리케이션을 서비스 해 줄 수 있는 것을 만드는 것입니다. 아래의 내용을 좀 더 살펴봅시다. 단 JSP나 Servlet에서 아래의 내용을 매번 따지면서 구현하지 않으니 안심하자. 하지만 웹 프로그래밍을 개발한다하면 아래의 사전 지식을 꼭 갖기 바라는 마음에서 추가하였습니다.
    1. 프로토콜(Protocol) : 네트워크상에서 약속한 통신규약 (Http;hypertext transfer protocol 받은 정보를 화면에 출력하기를 원하는 프로토콜, FTP;file전송protocol, SMTP;simple mail transfer protocol, POP;Post Office Protocol, DHCP;Dynamc Host Configuration Protocol)
    2. IP : 네트워크상에서 컴퓨터를 식별할 수 있는 주소
    3. DNS : IP주소를 인간이 쉽게 외우도록 맵핑한 문자열로 맵핑
    4. Port : IP주소가 컴퓨터를 식별할 수 있게 해준다면, Port번호는 해당컴퓨터의 구동되고 있는 프로그램을 구분할 수 있는 번호
    5. URL의 일반적인 구성

<http://www.11st.co.kr:80/html/category/1.jsp?xzone=ctgr1>

프로토콜. 서버주소. 포트번호. 경로(웹페이지상세주소)?쿼리문자열(추가로 서버에 보내는 데이터. 같은 경로라 해도 입력한 값에 따라 다른 결과를 보여줘야 할 때 쿼리 문자열을 사용한다)

1. Java 웹
   * JAVA플랫폼(J2SE, J2EE)중에서 J2EE를 이용한 웹프로그래밍

J2EE -> 컨테이너(컴포넌트 관리는 Tomcat이 함) -> 웹 컨테이너안에는 JSP, Servlet, HTML이 담겨 있음. 이 JSP나 Servlet이나 HTML 하나 하나를 컴포넌트라하고 , 이 컴포넌트를 하나하나 완성해 가는 것을 웹 프로그래밍이라 한다

* + 컴포넌트 : JSP, Servlet, HTML 등의 웹어플리케이션을 구현하기 위한 구성요소
  + JSP(Java Server Page) ; HTML 파일 내에 java 언어를 삽입한 문서
  + Servlet(Server Applet) ; Java 언어로 이루어진 웹 프로그래밍 문서

웹컨테이너

[ JSP ]

[ Servlet ]

[ HTML ]

Tomcat

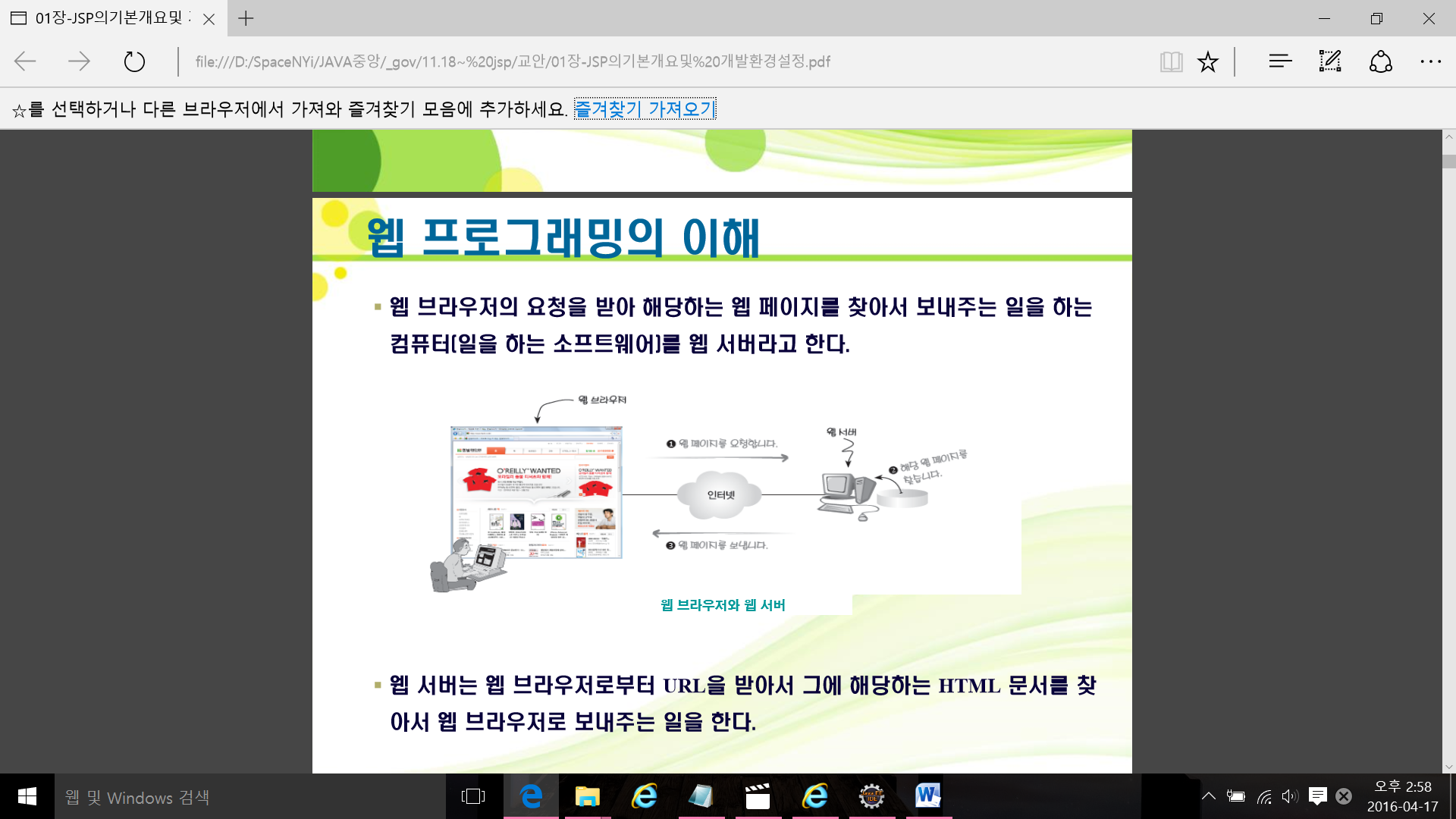
(컴포넌트관리)

J2EE

1. 웹 프로그램의 동작
   * 웹서버 : 클라이언트의 요청에 의해 정보를 제공해 주는 서버 (Aphach, IIS)

별도의 구현이 필요한 로직이 있을 경우 웹어플리케이션 서버에 요청.

* + 웹브라우저 : 웹서버에 정보를 요청하고, 웹서로부터 정보를 받는 매개체. 이때 HTTP 프로토콜을 사용함





* 웹프로그래밍 언어의 종류
  + ASP : Visual Basic 언어 기반. ActiveX개발 기능 제공. 특정 웹 서버와 OS에 동작한다는 단점. Windows 플랫폼에서 웹 서버로 IIS(Internet Information Server)만 사용
  + PHP : ASP와 달리 특정 영역에서만 동작하지 않고, C언어와 유사.
  + Servlet : Java 기반. 동적 컨텐츠 생성 기술. Java 코드 안에 HTML 태그가 혼재.
  + JSP : Java 기반. 동적 컨텐츠 생성 기술. 사용자가 직접 태그를 정의해서 사용할 수 있는 사용자 정의 태그 지정 가능. JSP 코드를 만들면 Servlet 자동 생성
* JSP와 Java의 관계
  + JSP는 Java Server Pages의 약자로 Sun Microsystems사(현재는 오라클)의 자바 서블릿(Servlet) 기술을 확장시킨 웹 환경 상에서 100% 순수한 자바만으로 서버 사이드 모듈을 개발하기 위한 기술
  + JSP는 DBMS와 같은 백 엔드 서버(Back-end Server)와 연동하여 이들 백엔드 서버의 데이터를 가공하여 웹 상의 최종적 사용자에게 디스플레이 할 수 있고, 여러 조건에 따라 디스플레이할 수 있는 내용들을 동적으로 처리할 수 있는 기능을 제공하고 있다.
  + JSP는 자바를 기반으로 만들어진 것이며, 객체 지향적, 플랫폼 독립적, 네트워크 지향적, 뛰어난 보안성, 멀티스레드 기능, 친근한 코드 등의 장점을 가지고 있다.
* HTTP 프로토콜과 Servlet 동작원리
  + HTTP 프로토콜 : HTTP은 비 연결(Connectionless: 클라이언트의 요청에 응답한 후 바로 연결을 끊음)과 비 상태(Stateless: 서버의 상태가 어떤지 간에 상관없이 요청을 함)의 특징을 가짐
  + HTTP method
    1. Get -http header에 정보를 실어 보냄 , default , Start-line 의 url뒤 에 붙음, 전달속도 빠름 , 256byte 가 한계 , 적은 양의 데이터를 전송 시 좋다
    2. Post - http의 body에 정보를 실어 보냄, 데이터 사이즈의 제한이 없음, 보안에 좋음

1. 필요한 학습
   * JAVA : JAVA웹어플리케이션을 구현하기 위한 선행 학습 필요
   * HTML : 웹 페이지를 구현하기 위한 기본 언어
   * JavaScript : 클라이언트 기능을 구현하기 위한 언어
   * Jquery : JavaScript의 대표적인 라이브러리로써, 클라이언트 사이드 스크립트 언어를 단순화 할 수 있다.
   * CSS : 웹어플리케이션의 레이아웃 및 스타일을 지정하는 언어
2. 개발 환경 설정
   * JDK 설치

JSP및 Servlet은 JAVA를 기본언어로 사용됩니다. JAVA언어로 작성한 프로그램을 컴파일하기 위해서는 JDK(Java Development Kit)가 필요 합니다

bin/javac.exe : .java 파일을 JVM이 받아들일 수 있는 .class 파일로 변환 시켜주는 프로그램. 매번 해당 경로에 들어가서 실행하기가 번거로우므로, 환경 변수 설정을 통해 어디서나 실행 가능하게 함.

* + path 설정

환경변수 PATH에 javac.exe를 포함시켜 어느 디렉토리에서나 javac.exe가 실행될 수 있도록 설정 합니다.

* + 이클립스 다운로드

[www.eclipse.org](http://www.eclipse.org)에 접속하여 Eclipse IDE for EE Developers

* + - Java Resource-src 폴더: 서블릿과 자바 소스 파일이 위치해야 하는 디렉토리
    - WebContent 폴더: 서블릿과 자바 소스 파일을 제외한 모든 파일이 위치해야 하는 디렉토리
    - WebContent-META-INF 폴더: 리소스 설정에 관련된 파일(context.xml)이 위치
    - WebContent-WEB-INF 폴더: 웹 애플리케이션 설정에 관련된 파일(web.xml)이 위치
    - WebContent-WEB-INF-classes: 컴파일 된 서블릿 클래스나 자바 클래스들이 위치
    - WebContent-WEB-INF-lib: 참조해야 하는 라이브러리(jar) 파일들이 위치
  + 톰캣설치

<http://tomcat.apache.org>에 접속하여 Tomcat 8.0 zip 파일 다운받아 압축을 푼다

bin : 톰캣을 실행하고 종료시키는 스크립트 파일

conf : server.xml을 포함한 설정 파일

lib ; 톰캣을 실행하는데 필요한 라이브러리(.jar) 파일위치

logs : 톰캣이 실행되는 동안 임시 파일이 위치함

temp : 톰캣이 실행되는 동안 임시파일이 위치

webapps ; 웹 어플리케이션이 위치

work : 톰캣이 실행되는 동안 사용되는 작업 파일이 위

* + 톰캣 환경 설정

이클립스 연동 : No server are available. Click this link to….. 이용하여 server 생성

* Tomcat v8.5 Server at localhost [Stopped, Republish] 생성
* Tomcat v8.5 Server at localhost [Stopped, Republish] 더블클릭
* Server Locations - use Tomcat (takes control to Tomcat installation)
* Server Options – Publish module contexts to separate XML files도 체크
* Server Options – Modules auto reload by default도 체크
* Ports
  + http/1.1 Port number 8181
  + 톰캣 서버 start
  + 서버 구동 확인 <http://localhost:8181> or http://127.0.0.1:8181
  + 서버 stop
  + 이클립스 한글 코드 설정
    - General –Workspace 선택 후 utf-8로
    - Web – CSS files 선택 후 Encoding에 utf-8
    - Web - HTML 선택 후 Encoding에 utf-8
    - Web – JSP 선택 후 Encoding에 utf-8