**Web개발\_Day46\_노트정리**

**웹 요청, 전달 흐름**

ㆍ웹 어플리케이션의 프로세스

* 웹 어플리케이션 (게시판/ 쇼핑몰/ SNS 등)

ㆍ요청의 흐름 :

* 클라이언트/사용자/브라우저 -> 요청 (Request) -> 서버
* 클라이언트로부터 요청을 받으면 서버가 클라이언트에게 전달
* 서버 -> 응답(response) -> 브라우저
* 전달된 페이지를 브라우저에서 표시

ㆍServlet, 서버가 리소스를 전달할 수 있게 코드를 변환

* Servlet, 서블릿 을 통해서 전달한다
* Random, Scanner와 같은 하나의 클래스의 일종이다.
* 결국 Java의 일종이므로 컨테이너가 돌려줘야 하는데, 서블릿 컨테이너 (=Tomcat)이 실행을 한다.
* 특징 : Java 코드인데도, 내부에 HTML을 작성할 수 있다.
* Servlet의 문제점 : Java와 HTML의 결합된 형태는 유지보수/관리에서 문제가 발생한다.
* “ “ 내부에 코드를 작성하므로 오타등을 확인하기 어렵다.
* 위의 문제점을 해결하기 위해 등장한 것이 JSP이다.

ㆍJSP : Java내부에 HTML을 작성하는 것이 Servlet이라면, JSP는 HTML코드 내부에 Java코드를 삽입한다.

* 장점 : 가독성이 향상되고, 오타 방지에 용이해진다.
* 특징 : JSP를 컴파일 하게 되면 결국에는 Servlet으로 변환되어 생성된다.

ㆍ정리 :

* 클라이언트 → 요청 → 서버(서블릿 컨테이너/톰캣) → Servlet파일 → 실행 → 내부의 순수 HTML코드 추출 → 페이지 전달 → 표시

**네트워크**

ㆍ네트워크의 중요성

* 데이터가 전다로디는 흐름을 아는데에 필요
* 문제발생시, 문제가 발생한 “지점”에 대한 이해에 중요

ㆍProtocol, 프로토콜

* 정의 : 네트워크 연결규격 통신 신호체계
* OSI 7계층

|  |  |  |
| --- | --- | --- |
| **구분** | **특징** | **예시** |
| 응용 | 개발자가 직접 구성 | 카톡 |
| 표현 | 암호화, 복호화 |
| 세선 | 동기화 |
| 전송 | TCP(빠른비암호) vs UDP(느린암호) |
| 네트워크 | 장치에서 수행되는 부분 | 데이터의 목적지 |
| 데이터링크 | IP주소(연결규격) |
| 물리 | MAC주소 (물리적 주소) |

ㆍ인터넷

* 에러 발생의 원인이 주로 인터넷에서 생김
* www웹기반으로한 SNS서비스가 많아지고 있다.
* DNS(Domain Name Service) 처리과정
* 인터넷주소 / IP주소등이 안맞을 경우의 에러
* 처리과정 : www.naver.com 요청 -> DNS -> 8.8.8.8 -> PC 표시

ㆍ웹

* URL에 대한 이해, 요청/응답에 대한 이해가 필요한 개념
* http 프로토콜을 사용
* GET방식 : URL에 정보가 보여지므로 보안에 취약, 데이터를 전송 가능하지만 길이에 제한 있음
* POST방식 : 기본보안은 되어있음, 길이제한은 없다. 데이터를 서버로 전송하기 위해 사용한다.

ㆍ서버에 접속해서 서비스를 이용 == 웹 프로그램 == 어플리케이션

* HTML : 구조를 생성
* CSS : 디자인을 지정
* JS : 클라이언트(사용자)와의 상호작용, 동작
* 서버프로그래밍 : PHP, ASP, JSP
* PHP = C언어, JSP = Java