

자바 쪽 보

1) 변수란?

- 단 하나의 값을 저장할 수 있는 공간

2) 변수의 타입

- 기본형(primitive type) : boolean, char, byte, short, int, long, float, double

- 참조형(reference type) : 기본형을 제외한 나머지 타입, 객체의 주소를 저장

3) 형변환

- 변수 또는 리터럴타입을 다른 타입으로 변환하는 것

4) 배열

- 같은 타입의 여러 데이터를 하나의 묶음으로 다루는 것

5) 객체 지향이란?

- 상속, 캡슐화, 추상화 개념

5) 클래스란?

- 객체를 정의해 놓은 것

6) 클래스의 사용 용도

- 객체를 생성하는데 사용

7) 객체란?

- 사물 또는 개념

8) 객체 용도

- 객체가 가지고 있는 기능과 속성에 따라 다름

9) 객체의 구성

- 기능과 속성(변수와 메서드)

10) 지역변수와 클래스 변수의 차이점

- 지역변수는 클래스 전체에서 사용이 가능

- 지역변수는 메서드 내부에서만 사용이 가능

11) 오버로딩과 오버라이딩의 차이점

- 오버로딩은 상속개념이 포함되지 않고, 메서드 이름은 같지만 각기 다른 매개변수타입을 가지고 있음.

- 오버라이딩은 상속의 개념이 포함되어 있고, 부모로부터 상속받은 메서드를 자식에서 재정의 하는 개념

12) 생성자

- 인스턴스가 생성될 때 호출되는 '인스턴스 초기화 메서드' 이다.

13) 상속이란?

- 클래스 다시 만들 필요없이 재사용하는 의미 extends를 사용하여 정의

- 자식 클래스는 조상클래스보다 같거나 많다.

- is - a 상속 : 나 자신은 사람이다.

- has - a 상속 : 나는 연필을 가지고 있다.

14) 접근 제어자

- private: 같은 클래스 내부에서 접근 가능

- default: 같은 패키지 내부에서 접근 가능

- protected: 같은 패키지 내부에서 또는 다른 패키지의 자손 클래스에서 접근이 가능

- public: 접근 제한이 전혀 없음

15) 캡슐화

- 데이터를 외부로부터 보호하기 위한 기법

- 주로 private을 이용한 방법 또한 캡슐화

- setter/getter를 이용하여 접근하게 하는 방법

16) 다형성

- 조상클래스라면 당연히 자손클래스를 참조가 가능

- 반대로 자손클래스는 조상클래스를 참조 못함

17) 추상메서드

- 물체가 존재하고 상속을 통해 구현한다.

- 일종의 클래스와 같음.

18) 인터페이스

- 일종의 추상클래스이지만 물체가 존재하지 않음.

- 오직 상속과 추상메서드로만 구성되어짐.

- 상수는 public static final 로 정의 (생략가능)

- 메서드는 public abstract 로 정의 (생략가능)

- 인터페이스끼리 상속이 가능하면, 다중상속을 허용

- 구현부는 implement로 구현한다.

19) 인터페이스의 장점

- 개발시간 단축

- 표준화가 가능

- 관계없는 클래스끼리 관계를 형성

- 독립적인 프로그래밍이 가능

20) Wrapper 클래스

- 87가지의 primitive type에 대한 객체화를 위한 것

21) 컬렉션

- list, map, set

- list: 중복을 허용하고 저장순서가 유지됨(arraylist, vector, linkedlist)

- set: 중복을 허용하지 않고 저장순서가 유지되지 않음(hashset, SortedSet, TreeSet)

- map: 키와 값이 쌍으로 존재하고, 키는 중복을 허용하지 않고, 값은 중복을 허용(hashmap, HashMap, SortedMap)

22) 동기화

- 하나의 객체를 여러개의 스레드가 동시에 접근할때에 데이터의 일관성을 위해 필요함

23) 제너릭

- 객체를 매개변수로 받고 결과를 특정한 클래스 타입으로 지정함을 제너릭스라고 함.

24) 프로세스

- 프로세스란 실행중인 상태를 말함.

- 스레드란 프로세스의 구성중 하나를 말하며 메모리, 자원등도 포함한다

- 단일 스레드와 멀티스레드로 나뉨

- 멀티스레드의 장점 : 자원을 효율적으로 사용할 수 있고 CPU의 사용률을 향상, 사용자에 대한 응답성이 향상, 코드

가 간결

25)

Override (덮어쓰기)

상속관계가 있을때, 상속받은 클래스에서 부모클래스의 함수를 재 정의해서 사용하는 것

Overload (함수 매개변수만 바꾸어쓰기) : 미시거 통을 바꾸는 것

하나의 클래스에서 같은 함수의 이름으로 서로 다른 변수를 사용해 함수를 사용하는 것

상속이무 = 용도별로 달라서

- 상속이무-

최적화

다형성

확장

오라클

Database란?

-영속적(지속적)으로 유지 관리해야 할 유용한 데이터의 집합

DBMS의 연어

-데이터 정의어(DDL, Data Definition Language) : CREATE, ALTER, DROP, TRUNCATE

-데이터 제어어(DCL, Data Control Language) : GRANT, REVOKE

-데이터 조작어(DML, Data Manipulation Language) : UPDATE, INSERT, DELETE, MERGE

RDBMS란?

-관계형 데이터베이스 관리 시스템

오라클 데이터 타입의 종류

-Char(size) : size 크기의 고정길이 문자 데이터 타입

-Varchar(size) : size 크기의 가변길이 문자 데이터 타입

-Nvarchar(size) : 국가별 문자집합에 대한 size 크기의 문자 또는 바이트의 가변길이

-Number(p,s) : 정밀도(p)와 스케일(s)로 표현되는 숫자 데이터 타입

-DATE : 날짜 형식을 저장하기 위한 데이터 타입

-ROWID : 테이블내 행의 고유주소를 가지는 64진수 문자타입, 해당 6바이트 또는 10 바이트

-BLOB : 대용량의 바이너리 데이터를 저장하기 위한 데이터 타입으로 최대크기는 4GB

-CLOB : 대용량의 텍스트 데이터를 저장하기 위한 데이터 타입으로 최대크기는 4GB

-BFILE : 대용량의 바이너리 데이터를 파일 형태로 저장하기 위한 데이터 타입 최대크기 4GB

-TIMESTAMP(n) : DATE 데이터 타입의 확장된 형태 n은 Milli second 자리수로 최대 9자리까지 가능

-INTERVAR YEAR TO MONTH : 년과 월을 이용하여 기간을 저장

-INTERVAR DAY TO SECOND : 일, 시, 분, 초를 이용하여 기간을 저장 두 날짜간의 정확한 차이를

표현 함

오라클 일반 함수에 대해서 설명

대소문자 변환함수

INITCAP : 문자열의 첫 번째 문자만 대문자로 변환

LOWER : 문자열 전체를 소문자로 변환

UPPER : 문자열 전체를 대문자로 변환

문자열 길이 반환 함수

LENGTH : 문자열의 길이를 반환

LENGTHB : 문자열의 바이트 수를 반환

문자 조작 함수

CONCAT : 두 문자열을 결합, || 와 동일확 구성을 함

SUBSTR : 특정 문자 또는 문자열 일부를 추출

INSTR : 특정 문자가 출현하는 첫 번째 위치 반환

LPAD : 오른쪽 정렬 후 왼쪽에 지정확 문자 삽입

RPAD : 왼쪽 정렬 후 오른쪽에 지정확 문자 삽입

LTRIM : 왼쪽의 지정 문자를 삭제

RTRIM : 오른쪽의 지정 문자를 삭제

숫자 함수

ROUND : 지정 확 자리 이하에서 반올림 CALMMASSTISTORY.COM

TRUNC : 지정 확 자리 이하에서 잘라

MOD : m을 n으로 나눈 나머지 값

CEIL : 지정확 값 보다 큰 수 중에서 가장 작은 정수

FLOOR : 지정확 값 보다 작은 수 중에서 가장 큰 정수

오라클 그룹 함수에 대해서 설명

-COUNT : 행의 개수 출력

-MAX : NULL을 제외한 모두 행에서 최대값 출력

-MIN : NULL을 제외한 모두 행에서 최소값 출력

-SUM : NULL을 제외한 모두 행의 합계

-AVG : NULL을 제외한 모두 행의 평균 값

-STDDEV : NULL을 제외한 모두 행의 표준편차

-VARIANCE : NULL을 제외한 모두 행의 분산 값

-GROUPING : 해당 칼럼이 그룹에 사용되었는지 여부를 1 또는 0으로 변환

-GROUPING SET : 확변의 질의로 여러 개의 그룹화 가능

집합 연산자에 대해서 설명

-UNION : 합집합을 구해줍니다.(중복 자료 배제하기 때문에 성능 좋음)

-UNION ALL : 합집합을 구해줍니다.(중복 자료를 포함 하기 때문에 성능 나쁨)

-NINUS : 차집합을 구해줍니다.

-INTERSECT : 교집합을 구해줍니다.

조인에 대해서 설명

조인(JOIN)은 두 개 이상의 테이블이나 DATASET을 서로 연결하는 데이터를 검색 하고자 할 때 사용하는 방법

CARTESIAN PRODUCT, EQUI JOIN, NON-EQUI JOIN, OUTER JOIN, SELF JOIN 등으로 나뉘져있습니다

SUB QUERY에 대해서 설명

하나의 SQL 명령문의 처리 결과를 다른 SQL 명령문에 전달하기 위해서 두 개 이상의 SQL문을 하나의 SQL 명령문으로 연결

오라클 데이터 무결성 제약조건에 대해서 설명

-NOT NULL : 해당 칼럼 값은 NULL을 포함 할 수 없습니다.

-Unique Key(고유키) : 테이블에서 해당 칼럼 값은 반드시 유일 해야 합니다.

-Primary Key(기본키) : 해당 칼럼 값은 반드시 존재해야 하며, 유일 해야 합니다 Unique Key, NOT NULL 제약 조건이 결합된 형태

-CHECK : 칼럼에서 허용 가능한 데이터의 범위나 조건을 지정하기 위한 제약 조건

오라클 동의어(Synonym)란?

다른 유저의 테이블을 조회할 경우 테이블소유자:테이블이름으로 조회하는 값을 자신의 테이블과 같이 조회할 수 있도록 만들어 주는 데이터베이스 오브젝트입니다.

오라클 시퀀스(Sequence)란?

순차적으로 번호를 할당하는 오라클 오브젝트의 한 종류라고 생각 하시면 됩니다.

spring

1. 빈 (bean)
 - 빈 또는 빈 오브젝트는 스프링 IOC 방식으로 관리하는 오브젝트라는 뜻이다.
 - 스프링이 직접 그 생성과 제어를 담당하는 오브젝트만을 빈이라 한다.
2. 빈 팩토리 (bean factory)
 - IOC를 담당하는 핵심 컨테이너를 지칭한다.
 - 빈을 등록, 생성, 조회하고 돌려주며, 그 외 부가적인 빈을 관리하는 기능을 담당한다.
 - 빈 팩토리라는 바로 사용하지 않고 이를 확장 애플리케이션 컨텍스트를 이용한다.
3. 애플리케이션 컨텍스트 (application context)
 - 빈 팩토리를 확장한 IOC 컨테이너이다.
 - 빈을 등록 관리하는 기본기능은 빈팩토리나 동일하나 스프링이 제공하는 각종 부가 서비스를 추가로 지원한다.
 - ApplicationContext는 BeanFactory를 상속한다.
4. 설정정보/설정 메타정보 (configuration metadata)
 - 애플리케이션 컨텍스트 또는 빈 팩토리 IOC를 적용하기 위해 사용하는 메타정보
5. 컨테이너 (container) 또는 IOC 컨테이너
 - 애플리케이션 컨텍스트나 빈 팩토리를 컨테이너 또는 IOC컨테이너라고도 한다.
 - 주로 빈 팩토리를 IOC 컨테이너라고 하면 그냥 컨테이너 또는 스프링 컨테이너는 애플리케이션 컨텍스트를 가리키는 것으로 보면 된다.
6. 스프링 프레임워크
 - 스프링 프레임워크는 IOC 컨테이너, 애플리케이션 컨텍스트를 포함해서 스프링이 제공하는 모든 기능을 통틀어 말할 때 주로 사용한다.
7. Maven이란 무엇인가?
 - 메이븐은 Make나 Ant와 같은 툴을 단순히 항상시킴이 아니라 소프트웨어 프로젝트와 소프트웨어 개발에 관련된 새로운 시맨틱(Semantics)들을 포괄하는 플랫폼이다.
8. Maven 특징
 - 1) 디펜던시(dependency)관리 : 그룹식별자, 산출물 식별자와 버전으로 구성된 중복되지 않은 코디네이트들을 프로젝트에 정의한다.
 - 2) 원격저장소 : 의존성 관리와 관련하여, 메이븐 산출물들의 저장소를 생성하기 위하여 메이븐 프로젝트 객체 모델(Project Object Model, POM)안에 정의된 코디네이트를 사용할 수 있다.
 - 3) 빌드 로직의 지역적인 재사용 : 플러그인은 POM에 따라 실행될 수 있도록 개발되었는데, 플러그인의 동작 시 어떠한 외부 파일도 참조하지 못하도록 설계되어 있다.
 - 4) 틀에 대한 이식성과 통합 : 이클립스나 넷빈즈 그리고 IntelliJ와 같은 툴은 프로젝트에 대한 정보를 찾을 수 있는 공통 공간을 가지고 있다.
 - 5) 프로젝트 산출물들의 쉬운 검색과 필터링 : 넥서스와 같은 툴은 POM안의 정보 저장소를 사용하여 저장소의 내역들을 인덱스하고 검색하는 것을 허용한다.

9. DI란?

객체 간의 의존관계를 객체 자신이 아닌 외부의 조립기가 수행해 준다는 개념

10. AOP란?

공통의 관심사항을 적용해서 발생하는 의존관계의 복잡성과 코드 중복을 해소해 주는 프로그래밍 기법

11. IoC/DI (Inversion of Control / Dependency Injection)

IoC/DI 는 스프링의 가장 기반의 되는 기술이자 스프링의 핵심 개발원칙이다.

나머지 두가지 기술인 AOP 와 PSAD IoC/DI에 바탕을 두고 있으며,

3패키지는 아니지만 자주 등장하는 템플릿/유클 패턴의 적용도 IoC/DI가 핵심원리다.

12. POJO (Plain Old Java Object) ?

특정 규약(contract)에 종속되지 않는다.

Java 언어와 꼭 필요한 API 외에 종속되지 않는다. EJB2 의 EntityBean 상속이나 Struts 1 의 ActionForm 상속등 규약에 종속되지 않아야 한다.

특정 환경에 종속되지 않는다.

EJB3 의 JNDI 서버 서비스의 의존이나 특정 벤더의 서버나 기업프라이머임에서만 동작 가능한 코드가 아니다.

객체지향원리에 충실해야 한다.

특정 기술규약과 환경에 종속되지 않은 Java Object 가 모두 POJO라 할수는 없다. 객체지향 개념이 남아있지 않은 것만 POJO 이다.

13. @Resource

Service형태의 객체변수 위에 사용하며 어노테이션 선언 시 name값에 해당하는 bean class를 찾아서@Service 로 선언된 클래스를 찾음) 해당 클래스로 객체변수에 인스턴스를 할당해준다. (별도의 new XXX) 코드가 필요 없다는 뜻)

14. @RequestMapping

Controller 자바파일 내의 함수 위에 사용하며 View단에서의 form action 속성값에 매핑되어 해당함수가 실행되어진다.

Ex) @RequestMapping(value="/sample/egovSampleList.do")

15. @Controller

특정 클래스에 @Controller annotation을 적용하면 다른 클래스를 상속받거나 Servlet API를 사용하지 않아도 해당 클래스가 컨트롤러 역할을 수행하도록 해준다

16. @SessionAttributes

17. @ModelAttribute

이 어노테이션은 2가지 방식으로 사용된다.

1. 비즈니스로직에서 처리한 결과데이터를 ModelMap에 저장하는 방식
2. View단에서 넘어온 파라미터와 Model을 바인딩하는 방식

해당 어노테이션이 적용된 객체가 자동으로 View에서 컨트롤러로 전달된다. View단에서 <form path="멤버변수명"> 형식으로 지정을 하면 자바단에서 [해당객체명.멤버변수명] 으로 바인딩된 값을 참조 할 수 있다.

- * 사전에 form태그의 commandName 속성값으로 객체명을 지정해야 함
- * 태그는 Stripping태그를 사용

18 @Repository

이 어노테이션이 적용된 DAO클래스는 DB Exception Translation을 자동으로 해준다.

19 @Qualifier

목적 : @Autowired의 목적에서 동일 타입의 빈객체가 존재시 특정빈을 삽입할 수 있게 설정한다.
설정위치 : @Autowired 어노테이션과 함께 사용된다.

추가설정 : 동일타입의 빈객체 설정에서 <qualifier value="alias명" />를 추가하여 준다.

옵션 : name - alias명

20 @Required

목적 : 필수 프로퍼티를 지정

설정 위치 : setter메소드

추가설정 : RequiredAnnotationBeanPostProcessor 클래스를 빈으로 등록시켜줘야 한다. 해당 설정 대신에 <context:annotation-config> 태그를 사용해도 된다.

21 @Autowired

목적 : 의존관계를 자동설정할 때 사용하며 타입을 이용하여 의존하는 객체를 삽입해 준다. 그러므로 해당 타입의 빈객체가 존재하지 않거나 또는 2개 이상 존재할 경우 스프링은 예외를 발생시키게 된다.

설정 위치 : 생성자, 필드, 메소드(굳이 setter메소드가 아니어도 된다)

추가설정 : AutowiredAnnotationBeanPostProcessor 클래스를 빈으로 등록시켜줘야 한다. 해당 설정 대신에 <context:annotation-config> 태그를 사용해도 된다.

옵션 : required - @Autowired어노테이션을 적용한 프로퍼티에 대해 굳이 설정할 필요가 없는 경우에 false값을 주며 이때 해당 프로퍼티가 존재하지 않더라도 스프링은 예외를 발생시키지 않는다. 다들 모르는 true

spring AOP(Aspect oriented Programming)의 주요 용어

AOP: 여러 객체에 공통적으로 적용할 수 있는 기능을 구분하여 재사용성을 높여주는 프로그래밍 기법
=> 관점 지향 프로그래밍(관심 지향 프로그래밍)

Aspect: 여러 객체에 공통적으로 적용되는 기능, 트랜잭션이나 보안 등이 좋은 예 Advice 정보 +Pointcut 정보
Advice: 언제 공통 관심 기능을 핵심로직에 적용할지 정의함 예)메서드를 호출하기 전(인제)에 트랜잭션 시작(공통기능)기능을 적용한다. '라고 정의함

JoinPoint: Advice를 적용 가능한 지점을 의미함 메서드 호출, 필드값 변경 등 스프링에서는 메소드 호출에 대한 JoinPoint만 지원

PointCut: JoinPoint의 부분집합으로 실제로 Advice가 적용되는 JoinPoint를 나타냄 스프링에서는 정규식이나 Aspect의 문법을 이용하여 정의

Weaving: Advice를 핵심 로직 코드에 적용하는 것

Advice의 종류

before:대상 객체의 메서드 호출 전에 공통 기능 실행
after returning: 대상 객체의 메서드가 Exception 없이 실행된 이후에 공통기능 실행

after throwing: 대상 객체의 메서드를 실행하는 도중 Exception이 발생한 경우 공통 기능 실행
after : 대상 객체의 메서드를 실행하는 도중 객체가 발생했는지 여부에 상관없이
메서드 실행후 공통 기능 실행(try-catch-finally의 finally와 비슷)

around: 대상 객체의 메서드 실행전, 후 또는 Exception 발생 시점에 공통 기능 실행

JSP 속보

웹 서버(Web Server)

- 클라이언트의 요청을 받아 HTML이나 오브젝트를 HTTP 프로토콜
- 웹 서버만 구축된 서버는 웹 페이지, 이미지 등 정적인 페이지를 생성하지만, JSP 컨테이너가 탑재되어 있는 WAS는 JSP 페이지를 컴파일 해 동적인 페이지를 생성한다.

웹 어플리케이션 서버(Web Application Server / WAS)

- 웹 서버 + 웹 컨테이너.
- 웹상에서 사용하는 컴포넌트들을 올리놓고 사용하게 되는 서버
- EJB와 같은 빈들이 올라가게 되며, 서버에 따라 웹에 필요한 많은 기능들을 포함하고 있다.
- J2EE 스펙을 구현한 서버(JSP / Servlet Container와 EJB Container 로서의 기능)
- 가장 많이 사용되는 WAS는 BEA사의 Web Logic, IBM사의 Web Sphere, T-max사의 Jaus, Tomcat, Redhot사의 JBoss 등이 있다.)
- Tomcat은 JSP / Servlet Container의 기능을 구현했으나 EJB Container로서의 기능은 없다. 따라서 Tomcat은 WAS가 아니라는 사람들도 있다.

컨테이너(Container)

- JSP와 서블릿을 이용*는 웹 응용 프로그램은 자바 언어를 이용해서 작성할 수 있는데, JSP와 서블릿을 실행시킬 수 있는 소프트웨어를 웹 컨테이너(Web Container) 혹은 서블릿 컨테이너(Servlet Container)라고 한다.
- 웹 서버에서 JSP를 요청하면 플랫폼에서는 JSP 파일을 서블릿으로 변환하여 컴파일을 수행하고, 서블릿의 수행결과를 웹 서버에서 전달하게 된다.

웹 서버와 WAS의 일반적인 구성

- 사용자가 브라우저에서 요청을 하게 되면 다음과 같이 웹 서버와 WAS 서버를 거쳐 응답이 돌아옴
사용자 요청(웹 브라우저) -> 웹 서버 -> WAS(동적 처리) -> 웹 서버 -> 사용자 응답 매세지

▶서블릿(Servlet) : 자바 클래스 형태로 구현되는 웹 어플리케이션 프로그램.

- 서블릿 클래스는 javax.servlet.HttpServlet 클래스를 상속하도록 만든다.
- doGet 또는 doPost 미서드 안에 웹 브라우저로부터 요청이 있을 때 해야 할 일을 기술.
- HTML 문서는 doGet, doPost 메서드의 두 번째 파라미터(response)를 이용해서 출력

▶ jsp(java server pages) : 웹 애플리케이션을 구현할 때 작성하는 코드

- 자바를 서버환경에서 사용하는 스크립트 방식의 언어로 단일 스레드로 클라이언트의 요청에

서버스 한다.

- MVC model 1 : jsp로만 구성...배를 개발 가능....유지보수 힘들.

- MVC model 2 : model : 비즈니스 로직(java)

view : 사용자에게 보여주는 로직(jsp)

control : 처리 흐름 제어(serverlet)

위의 3단계로 나누어 개발.. 설계가 복잡하지만 유지보수가 편리

▶ 스크립팅 요소

- 스크립틀릿(scriptlet) : <% %> 사이에 자바 명령문이 들어갈 수 있다.

웹 브라우저로 전송되는 것이 아니라 웹 서버 쪽에서 실행.

- 익스프레션(expression) : <%= %> 사이에 자바 식이 들어갈 수 있다.

웹 서버에서 실행되고 웹브라우저로 결과 값만을 전송.

- 선언문(declaration) : <%! %> 사이에 변수 선언이나 메서드 선언.

▶ 지시자(directive) : jsp의 다른 문법들(스크립팅 요소, 익스프레션 언어, 액션)과는 다른 목적으로

사용되는 문법. 웹 컨테이너가 jsp 페이지를 서블릿 클래스로 변환할 때 필요한 여러 가지

정보들을 기술하기 위해 사용하는 문법. <%@ %> 의 형태로 사용

- page 지시자 : jsp 페이지 전체에 적용되는 정보를 기술하기 위해 사용

- include 지시자 :

- taglib 지시자 :

▶ 내장변수(implicit variable) : 선언하지 않고도 사용할 수 있는 변수.

- request : getParameter 메소드와 사용하면 웹 브라우저를 통해 입력 값을 가져올 수 있다.

- response : 웹 브라우저에 보내는 응답 정보를 담는다.

- out : 웹 브라우저로 HTML 코드를 출력하는 기능.

- application : jsp 페이지가 속하는 웹 애플리케이션에 관련된 기능.

- config : jsp 페이지의 구성 정보를 가져오는 기능.

- pageContext : jsp 페이지 범위 내에서 사용할 수 있는 데이터 저장 기능 등.

- session : 세션에 관련된 기능.

- page : jsp 페이지로부터 생성된 서블릿.

- exception : 익셉션 객체.

▶ forward 메서드와 sendRedirect 메서드의 차이점

- sendRedirect 메서드는 URL을 웹 브라우저로 보내서 간접적으로 웹 자원을 호출하지만

forward 메서드는 웹 서버 쪽에서 직접 호출한다.

- sendRedirect 를 이용하면 다른 웹 서버에 있는 웹 자원도 호출할 수 있지만, forward를

이용하면 같은 웹서버, 같은 웹 애플리케이션 디렉터리 내에 있는 웹 자원만 호출

- sendRedirect 메서드를 이용하면 텍스트 데이터만 전달 할 수 있지만

forward 메서드를 이용하면 객체 형태의 모든 데이터를 전달 할 수 있다.

- sendRedirect 메서드를 통해 한글 데이터를 전달하기 위해서는 URL 인코딩을 해야 하지만

forward 메서드에서는 그런 작업이 필요치 않다.

- ▶ 쿠키 : 웹 컴포넌트 간의 데이터 전달을 위해 사용. 웹브라우저 쪽에 데이터를 저장했다가 다시 읽어오는 방식을 취한다.

세션 : 여러 웹 컴포넌트들이 협력 작업을 시작해서 끝내기까지의 기간을 의미

HTML이란?

HTML은 Hypertext Markup Language의 약자로 웹에서 하이퍼텍스트를 작성하고 인식하는데 사용되는 표준언어이다.

웹문서는 일반적으로 HTML로 작성되며, 파일의 확장자로는 .html 이나 .htm을 사용한다. 대부분의 HTML 문법은 < >와 </>쌍으로 구성되어 있다.

CSS란 무엇인가?

스타일 시트의 정의

- 아래 아 한글이나 워드와 같은 워드프로세서에서 사용하는 스타일과 같은 것
- 웹 페이지에서 사용하는 여러 가지 속성들(문자 종류와 크기, 여백, 글자 색, 배경색, 정렬 등을 원하는 대로 정의해 놓고 일괄적으로 적용할 수 있도록 만들어 놓은 것
- 홈 페이지를 일관성 있게 유지하는데 필요하다.

스타일 시트의 장점

- ① 다양한 기능의 확장
- ? HTML에서 지원하지 않는 다양한 글자 크기의 지정도 가능하다.
- ② 통일된 문서 양식 제공
- ? HTML은 각 태그마다 속성을 일일이 지정해 주어야 하지만 CSS를 사용하면 한 번의 속성 정의로 여러 문서에서 동시에 다양하게 적용할 수 있다.
- ③ 사용자 환경과 상관없는 독립된 문서 제작 가능
- ? 브라우저에 따라서 HTML의 결과물이 다르게 나타나는데, CSS를 통해서 이를 방지하고 의도대로 결과를 작성할 수 있다.
- ④ 문서 형식의 다양화
- ? 사용법을 쉽게 배우는 반면 기능은 단순하다. CSS를 사용해서 다양한 양식의 HTML 문서를 만들 수 있다.
- ⑤ 로딩시간 단축
- ? 통일된 문서 양식을 적용하였기 때문에 브라우저가 읽어야 하는 문서의 수가 작아져 로딩 시간이 단축된다.

CSS 선택자

CLASS 선택자

- CLASS 선택자는 태그에 대해 상황에 따라 다른 스타일 속성을 적용하기 위한 스타일 구분이다.

ID 선택자

- ID 선택자도 태그에 대해 상황에 따라 다른 스타일 속성을 적용할 수 있게 스타일을 구분하는데 사용한다.

CSS 사용방법

- 인라인 방식
- 임베디드방식
- 링크 방식

지바스크립트란 무엇인가?

지바스크립트는 스크립트 언어다

지바스크립트(Javascript)는 웹을 위한 인터프리터 언어이자 스크립트 언어다.

지바스크립트로 작성된 프로그램을 스크립트라고 하며, 컴파일 필요하지 않다. 그냥 HTML 웹 페이지에 스크립트를 삽입하기만 하면 동작하며 최신 웹 브라우저에서 모두 동작한다.

지바스크립트를 주로 클라이언트 측 지바스크립트라고 하는데, 이는 스크립트가 웹 서버가 아닌 클라이언트 컴퓨터에 설치된 브라우저에서 실행된다는 의미다.

지바스크립트의 역할

지바스크립트의 일반적인 용도는 웹 페이지에 기능을 더해 HTML 웹 페이지를 동적이고 살아 있게 만드는 것이다.

지바스크립트는 다음과 같은 일을 할 수 있다.

- ?HTML 페이지 변경 및 HTML 엘리먼트와 콘텐츠의 추가나 제거
- ?CSS 및 HTML 엘리먼트의 스타일 변경
- ?사용자와의 상호작용, 폼의 유효성 검증
- ?마우스와 키보드 이벤트에 대한 스크립트 실행
- ?웹 브라우저 제어, 쿠키 등의 설정과 조회
- ?AJAX 기술을 이용한 웹 서버와의 통신

지바스크립트의 한계

지바스크립트는 웹 브라우저에서 실행되기 때문에 일부 보안상의 제약이 있으며, 브라우저에서 웹 페이지를 열 때 안전하고 위험에 처하지 않도록 보장해야 한다.

- ?지바스크립트는 OS에 직접 접근할 수 없으며, 하드디스크를 읽거나 쓸 수 없고, 다른 프로그램도 호출할 수 없다.
- ?지바스크립트는 도메인이 동일한 두 웹/원도우를 제외하고 웹/원도우 간에 통신을 수행할 수 없다.
- ?일반적으로 지바스크립트는 자체 도메인에 대해서만 제한없이 네트워크 요청을 보낼 수 있다.

정리

지바스크립트는 전 세계에서 가장 인기 있는 프로그래밍 언어이며, 웹 개발자를 위한 강력한 언어다.

제이쿼리란 무엇인가?

jQuery 는 자바스크립트와 HTML 사이의 상호작용을 강조하는 경량화 된 웹 애플리케이션 프레임 워크이다.
존 레시그에 의해, 2006년 뉴욕 시 바캠프에서 릴리즈 되었다.

jQuery는 MIT라이선스와 GNU 일반 공중 사용 허가서의 듀얼 라이선스 하의 자유 오픈 소프트웨어이다.

기능

DOM 엘리먼트 선택

DOM 트래버설 및 수정(CSS 1-3지원, 기본적인 Xpath 를 플러그인 형태로 지원)

이벤트

CSS조작.

특수효과 및 애니메이션

Ajax

확장성

유틸리티 - 브라우저 버전, 'each' 함수

자바스크립트 플러그인

Ajax란 무엇인가?

ajax란 Asynchronous Javascript And Xml(비동기식 자바스크립트와 xml)이며

브라우저가 가지고있는 XMLHttpRequest 객체를 이용해서 화면의 전체 갱신없이 내부에서 서버와 통신을 하는것을 말합니다.

통신 후 리턴받는 데이터는 xml 또는 json 데이터타입이 되며 받은데이터로 javascript에서 ui조작을 하여 전체갱신없이 ui의 일부분만 동적인 변경이 가능하게 됩니다

이는 페이지요청후 화면이 전체갱신되면서 html이 렌더링되는 일반적인 웹프로래밍 방법과는 다르며, 윈도우 클라이언트 프로그램과 더 비슷한 방식이기도 합니다.

장점 : ajax통신후 받은 데이터를 갖고 화면의 전체갱신없이 부분적인 ui변경이 가능합니다

html 전체코드가 아닌 최소한의 필요한 데이터(xml 또는 json)만 받으므로 통신량이 줄어듭니다.

단점 : javascript 코딩량이 많아지게 됩니다.

jquery를 이용해서 ajax를 사용하는 이유 : 일반 javascript로 ajax를 하게되면 코딩량도 많아지고 브라우저별로 구현방법이 다른 단점이 있는데 jquery를 이용하면 더 적은 코딩량과 동일한 코딩방법으로 대부분의 브라우저에서 같은 동작을 할 수 있게 됩니다.

한마디로 크로스브라우저가 쉬워집니다.