프로그래밍 언어활용

- **다음 Java 개발환경 구성에 관한 설명으로 잘못된 것은?**

1. Java 프로그램 개발은 Linux환경에서만 가능하다.

2. 개발자의 컴퓨터에는 JDK가 설치되어야 한다.

3. Eclipse를 설치하면 Java프로그램 개발하기가 더 쉬워진다.

4. Java로 만든 프로그램을 실행할 컴퓨터에는 JRE가 설치되어야 한다.

- **Java 소스를 기계어(바이트코드)로 변환하는 과정을 무엇이라 하는가?**

1. copy

2. compile

3. translate

4. int

**- .jar파일을 실행하는 명령어는 java ( ) 파일명이다.**

1. -run

2. -exe

3. -jar

4. run

- **다음 중 Java의 변수명으로 사용할 수 없는 것은?**

1. ㅋ

2. \_for

3. int\_age

4. 1test

**- 다음 중 error가 발생하는 것은?**

1. System.out.println("%.1d", 3.14);

2. System.out.println("%02d", 30);

3. System.out.print(""");

4. System.out.println("이름t:t홍길동");

- **다음 프로그램의 실행 결과는?**

public class Test {  
  
      public static void main(String[] args) {  
  
            int a;  
  
            a += 6;  
  
            System.out.print(a);  
  
      }  
  
}

1. 무반응

2. error발생

3. 6이 출력

4. 0이 출력

**- 다음 프로그램의 실행 결과는?**

public class Test {  
  
      public static void main(String[] args) {  
  
            int a = 1;  
  
            a+=2;  
  
            a--;  
  
            a\*\*;  
  
            a//;  
  
            System.out.print(a);  
  
      }  
  
}

1. 1이 출력

2. 2가 출력

3. error발생

4. 무반응

**- 다음 프로그램의 실행 결과는?**

public class Test {  
  
      public static void main(String[] args) {  
  
            int a = 10;  
  
            int b = 4;  
  
            int c = a / b;  
  
            System.out.print(c);  
  
      }  
  
}

1. 무반응

2. error발생

3. 2.5가 출력

4. 2가 출력

**- 다음 프로그램의 실행 결과는?**

public class Test {  
  
      public static void main(String[] args) {  
  
            String[] ar = {"에", "네", "르", "기", "파"};  
  
            for (int i = 0; i <= 5; i++) {  
  
                  System.out.print(ar[i]);  
  
            }  
  
      }  
  
}

1. 에네르기파 출력, exception발생

2. 에네르기파 출력

3. error발생

4. 파 출력

- **다음 프로그램에서 error가 발생하는 이유는?**

public class Test {  
  
      public void add(int i, int j){  
  
            int hab = i + j;  
  
            System.out.println(hab);  
  
      }  
  
      public static void main(String[] args) {  
  
            add(10, 20);  
  
      }      
  
}

**static인 main메소드에서 static이 아닌 add메소드를 호출**

**- 오버라이딩(Overriding)과 오버로딩(Overloading)에 대해 아는대로 설명하시오.**

**오버라이딩 : 상속받은 메소드 재정의, 오버로딩 : 메소드명 같게, 파라메터 다르게**

**- 다음 프로그램에서 error가 발생하는 이유는?**

public interface JoongDaeJang{  
  
      public void doSilmang(){  
  
            System.out.println("중대장은 너희에게 실망했다.");  
  
      }  
  
}

**인터페이스에 메소드가 구현되어있음**

**- 다음 중 객체지향프로그래밍의 상속 개념을 가장 잘 나타낸 클래스 관계는? (부모클래스 - 자식클래스)**

1. Avengers - Ironman

2. Father - Mother

3. Son - Father

4. Apple - Computer

**- 다음 프로그램의 실행 결과는?**

public class Test {  
  
      public static void main(String[] args) {  
  
            try {  
  
                  System.out.print("ㄱ");  
  
                  System.out.print(10/0);  
  
                  System.out.print("ㄴ");  
  
            } catch (Exception e) {  
  
                  System.out.print("ㄷ");  
  
                  return;  
  
            } finally {  
  
                  System.out.print("ㄹ");  
  
            }  
  
      }  
  
}

1. ㄱㄴㄷㄹ

2. ㄱㄷㄹ

3. ㄱㄷ

4. ㄱㄹ

**- Deprecated에 대해 아는 대로 설명하시오.**

**업데이트 될 예정인 기능, 다음 버전에서 없어질수도 있음, 현재는 유예기간 중**

**- 다음 설명 중 옳은 것은?**

1. 객체지향프로그래밍은 아직까지 널리 쓰이지 않는 개념이다.

2. 설계 없이 구현을 먼저 하는게 좋다.

3. 모듈간의 결합도는 줄이고 개별 모듈들의 내부 응집도를 높여야 한다.

4. 유지보수가 용이하게 설계하면 개발 시간이 짧아진다.

- **다음 설명 중 옳지 않은 것은?**

1. error란 자바 문법에 맞지 않아 컴파일이 불가능한 상태를 말한다.

2. exception이란 완성된 프로그램을 실행할 때 발생할 수 있는 예외적인 상황을 말한다.

3. 개발자는 exception이 발생할 수 있는 상황에 대비해서 프로그램 개발을 해야한다.

4. warning이란 자바 문법에 맞지 않으나 컴파일은 가능한 상태를 말한다.

- **다음 프로그램의 실행 결과는?**

public class Test{  
  
      public void main2(String[] args){  
  
            int i = 10;  
  
            String s = "a";  
  
            String s2 = s + i;  
  
            System.out.println(s2);  
  
      }  
  
}

**무반응**

**- 다음 메소드의 기능은?**

public void q19(ArrayList<Integer> al) {  
  
    int a = al.get(0);  
  
    int b = 0;  
  
    int t = 0;  
  
    for (int i = 1; i < al.size(); i++) {  
  
        t = al.get(i);  
  
        if (t > a) {  
  
            a = t;  
  
            b = i;  
  
        }  
  
    }  
  
    System.out.println(al.get(b));  
  
}

**ArrayList에서 최대값 출력**

**- 다음 메소드의 기능은?**

public void q20(int[] ar) {  
  
    int t = 0;  
  
    for (int i = 0; i < ar.length; i++) {  
  
        for (int j = 0; j < ar.length - (1 + i); j++) {  
  
            if (ar[j] > ar[j + 1]) {  
  
                t = ar[j];  
  
                ar[j] = ar[j + 1];  
  
                ar[j + 1] = t;  
  
            }  
  
        }  
  
    }  
  
    for (int i : ar) {  
  
        System.out.print(i + " ");  
  
    }  
  
}

**int[] 오름차순 정렬**

**-----------------------------------------------------------------------------------------------------------------------------------------------------**

**서버프로그램**

**- 다음 WAS에 대한 설명 중 잘못된 것은?**

1. JSP프로젝트를 실행할 때에는 WAS가 아닌 일반 웹서버 환경이어도 상관없다.

2. 일반 웹서버의 기능을 모두 포함한다.

3. 프로그램 실행이 가능한 웹서버이다.

4. Tomcat을 설치하면 컴퓨터를 WAS로 만들 수 있다.

- **Servlet에 대해 아는대로 설명하시오.**

**클라이언트의 요청에 대해 응답할 수 있는, WAS에서 실행되는 프로그램**

- **GET방식 요청과 POST방식 요청의 차이에 대해 아는대로 설명하시오.**

**GET : 요청 파라메터가 주소에, POST : 요청 파라메터가 내부적으로 전달**

- **다음 중 GET방식 요청 주소의 형태로 올바른 것은?**

1. http://IP주소/포트번호/폴더명/파일명/파라메터변수명1/파라메터변수값1/파라메터변수명2/파라메터변수값2

2. http://IP주소:포트번호/폴더명/파일명?파라메터변수명1=파라메터변수값1&파라메터변수명2=파라메터변수값2

3. http://IP주소-포트번호-폴더명-파일명-파라메터변수명1-파라메터변수값1-파라메터변수명2-파라메터변수값2

4. http://IP주소:포트번호:폴더명:파일명:파라메터변수명1:파라메터변수값1:파라메터변수명2:파라메터변수값2

- **다음 요청에 대한 설명 중 잘못된 것은?**

1. GET방식 요청할 때 포트번호를 쓰지 않으면 80번을 기본으로 사용한다.

2. GET방식 요청할 때 파일명을 쓰지 않으면 index.html을 기본으로 사용한다.

3. <a href=...></a>는 POST방식 요청이다.

4. POST방식 요청은 <form>을 통해서만 가능하다.

- **다음 한글처리에 대한 설명으로 잘못된 것은?**

1. 데이터를 보내는 쪽과 데이터를 받는 쪽의 인코딩 방식이 다르면 글자가 깨진다.

2. 파일 업로드할 때의 한글처리는 기존의 방식과 다르다.

3. POST방식 파라메터의 한글처리는 request.setCharacterEncoding("인코딩방식");으로 한다.

4. GET방식 파라메터의 한글처리는 request.setCharacterEncoding("인코딩방식");으로 한다.

- **다음 JSP파일에 대한 설명 중 잘못된 것은?**

1. <%= %> 속에 메소드를 만들 수 있다.

2. <% %> 속에 if, for등을 사용할 수 있다.

3. 사실은 Servlet이다.

4. EL, JSTL등을 사용할 수 있다.

- **다음 <input>들에 대한 설명 중 잘못된 것은?**

1. <input>의 기본 type은 text이다.

2. <input type="image">로 파일을 선택하게 할 수 있다.

3. <input type="radio">를 사용해 목록 중에 선택하게 할 수 있다.

4. <textarea>로 여러줄 text를 입력하게 할 수 있다.

- **cos.jar로 파일 업로드를 구현할 때 잘못된 것은?**

1. cos.jar파일을 프로젝트에 넣어야 한다.

2. 파일 업로드를 하는 <form>의 요청방식은 post여야 한다.

3. 파일명 중복처리는 불가능하므로, 중복되는 파일명은 클라이언트가 업로드 하지 못하게 하여야 한다.

4. 파일 업로드를 하는 <form>의 enctype은 multipart/form-data여야 한다.

**- 다음 JavaScript에 대한 설명 중 잘못된 것은?**

1. 주로 웹페이지의 이벤트 처리에 사용된다.

2. 입력 <form>의 유효성 검사에 사용하면 좋다.

3. 자바와 문법이 유사한 부분이 많다.

4. Tomcat이 실행하는 서버쪽의 컴파일러 방식 언어이다.

- **다음 화면전환에 대한 설명으로 잘못된 것은?**

1. JavaScript의 location.href = ...; 는 POST방식 요청이다.

2. <a href="..."></a>는 GET방식 요청이다.

3. 포워딩은 파라메터, 어트리뷰트 등의 값을 가지고 화면전환을 할 수 있다.

4. <jsp:include>를 사용해 다른 jsp파일 소스를 현재 jsp파일에 포함시킬 수 있다.

- **JSP Model2 방식에 대해 아는대로 설명하시오.**

**JSP프로젝트에 MVC패턴을 적용시켜, 유지보수/협업에 유리한 형태를 취하는 것**

**- 다음 설명 중 잘못된 것은?**

1. EL은 자료형 변환, import등을 자동으로 처리해준다.

2. .html에서도 사용 가능하다.

3. JSTL은 html태그 형태로 for, if, 출력형식지정 등을 가능하게 한다.

4. 실제로는 Java코드로 바뀌어 실행된다.

- **파라메터와 어트리뷰트에 관해 아는대로 설명하시오.**

**파라메터 : View에서 만든 String값, 어트리뷰트 : Model에서 만든 Object**

**- 세션과 쿠키에 대해 아는대로 설명하시오.**

**세션 : 클라이언트-서버의 연결상태, 브라우저 종료시 세션 종료. 쿠키 : 클라이언트 컴퓨터에 저장된 파일**

**-----------------------------------------------------------------------------------------------------------------------------------------------------**

SQL 활용

- **다음 OracleDB의 자료형에 관한 설명 중 잘못된 것은?**

1. varchar2 : 글자수가 정해지지 않은 글자

2. char : 글자수가 정해진 글자

3. date : 날짜

4. number : 정수형 숫자

- **다음 중 잘못된 설명은?**

1. C : 데이터를 만드는 행위 - create

2. R : 데이터를 읽는 행위 - read

3. U : 데이터를 업로드하는 행위 - upload

4. D : 데이터를 삭제하는 행위 - delete

**- 다음 SQL에 대한 설명으로 잘못된 것은?**

insert into avengers values(avengers\_seq.nextval, '아이언맨', 30, sysdate);

1. avengers테이블에 데이터를 넣는 SQL

2. avengers\_seq.nextval : avengers\_seq라는 테이블의 nextval이라는 데이터의 값을 가져오는 것

3. sysdate : 현재시간날짜

4. SQL문법상 error는 없음

- **avengers테이블의 모든 자료를 조회하는 SQL명령어는?**

**select \* from avengers;**

**- 다음 중 avengers테이블의 모든 자료들의 평균 age값을 조회하는 SQL은?**

1. select avg(age) from avengers;

2. select \* from avengers;

3. select age from avengers;

4. select \* from avengers order by age;

- **avengers테이블의 모든 자료를 name 오름차순으로 조회하는 SQL명령어는?**

**select \* from avengers order by name;**

**- avengers테이블에서 name이 ironman인 자료의 age를 30으로 수정하는 명령어는?**

**update avengers set age=30 where name='ironman'**

**- avengers테이블에서 name에 man이 들어가는 자료의 age를 1살씩 더하는 명령어는?**

**update avengers set age=age+1 where name like '%man%';**

**- avengers테이블에서 name이 hulk인 자료를 삭제하는 명령어는?**

**delete from avengers where name='hulk';**

**- avengers테이블에서 age가 평균age보다 큰 자료의 갯수를 구하는 SQL명령어는?**

**select count(\*) from avengers where age > (select avg(age) from avengers);**

**-----------------------------------------------------------------------------------------------------------------------------------------------------**

데이터베이스 구현

**- 다음 파일에 대한 설명 중 잘못된 것은?**

1. ojdbc8.jar : OracleDB연동에 사용되는 OracleDriver클래스가 존재

2. jstl-1.2.jar : 태그로 for문, if문, 출력형식 등을 지정할 수 있는 JSTL 사용에 필요

3. context.xml : 커넥션풀방식으로 DB연결할때 사용하는 톰캣 설정파일

4. cos.jar : DB서버와 연결된 프로그램을 만들때 반드시 필요한 파일

**- 다음 데이터베이스 연동에 관한 설명 중 잘못된 것은?**

1. DAO/DTO패턴을 사용하는게 좋다.

2. JSP환경에서는 커넥션풀 방식을 사용 가능하다.

3. Eclipse에 DB연결 설정을 했다면 프로젝트에는 ojdbc8.jar파일이 필요없다.

4. 커넥션풀 방식은 미리 커넥션을 맺어놓고 필요할때 바로 사용하는 방식이다.

- **다음 객체에 관한 설명 중 잘못된 것은?**

1. Connection : DB서버와 연결

2. PreparedStatement : INSERT문 수행 결과

3. ResultSet : SELECT문 수행 결과

4. String : 수행할 SQL문을 담아놓는 역할

- **자바빈에 관해 아는대로 설명하시오.**

**주로 DB에서 조회한 결과를 담아서 표시하는데 사용되는 객체, DTO라고도 함**

**- 다음 중 잘못된 설명은?**

1. SELECT문을 수행한 결과는 int이다.

2. INSERT문을 수행한 결과는 int이다.

3. DAO에서 DB관련 작업을 수행한다.

4. 작업 결과는 어트리뷰트로 세팅해서 JSP에서 사용할 수 있도록 한다.

- **DAO/DTO패턴에 대해 아는대로 설명하시오.**

**DAO에서 DB관련 작업, 그 작업 결과는 DTO객체로**

**- 다음 로그인 구현 관련 설명 중 잘못된 것은?**

1. 로그인 관련 요청은 POST방식으로 처리하는 것이 좋다.

2. DB에서 SELECT해온 결과와 사용자가 입력한 값을 비교해 로그인을 처리한다.

3. 사용자의 로그인 정보는 세션을 사용해 처리하면 좋다.

4. 쿠키는 안전하므로 사용자의 모든 정보를 저장해서 사용하면 좋다.

- **다음 게시판 구현 관련 설명 중 잘못된 것은?**

1. DAO에서 SELECT해온 데이터는 ArrayList<자바빈>으로 만들어 주는것이 좋다.

2. 사용자 개인의 검색 데이터는 세션에 담아서 처리하는것이 좋다.

3. 게시물 등록/수정/삭제 후, 전체 데이터를 조회해오는것이 좋다.

4. 전체 데이터를 한꺼번에 불러오는 행위는 자주하지 않는 편이 좋다.

- **다음 테이블을 XML로 표현해보면?**

STUDENTS

|  |  |
| --- | --- |
| s\_name | s\_age |
| 홍길동 | 20 |

홍길동20

- **다음 테이블을 JSON으로 표현해보면?**

STUDENTS

|  |  |
| --- | --- |
| S\_NAME | S\_AGE |
| 홍길동 | 20 |

**[{"s\_name":"홍길동", "s\_age":"20"}]**

**-----------------------------------------------------------------------------------------------------------------------------------------------------**

**Spring와 Mybatis 개발**

**- 다음 중 유지보수에 용이하지 않은 것은?**

1. MVC패턴 적용

2. 알아보기 쉬운 변수명 사용

3. 수정이 일어날 만한 내용은 자바소스가 아닌 .xml같은 외부 파일에 놓고 불러오는 형태로 프로그래밍

4. 모든 소스를 main()메소드 속에

- **다음 중 MyBatis의 설명으로 잘못된 것은?**

1. DB ORM Framework

2. SELECT 수행 시, 자동으로 자바빈 객체로 만들어주는 기능이 있다.

3. SQL을 .xml에 넣기 때문에 <, >등은 특수문자로 대체해야 한다.

4. 유지보수에 용이한 프로그램 개발에는 좋지 않다.

- **다음 중 Spring에 대한 설명으로 잘못된 것은?**

1. 유지보수에 용이한 개발을 지원해주는 Framework이다.

2. 자바 객체를 .xml에 만들어놓고 자바소스로 그 객체를 불러다 사용할 수 있다.

3. IoC로 DI를 지원한다.

4. 웹 개발용으로만 사용된다.

- **다음 중 SpringMVC 프로젝트에 대한 설명으로 잘못된 것은?**

1. 설정을 통해 MyBatis와 연동을 시킬 수 있다.

2. servlet-context.xml파일에 자바객체를 만들어 놓을 수 있다.

3. servlet-context.xml에 등록된 객체는 @Autowired로 자동으로 불러서 사용할 수 있다.

4. pom.xml파일에 설정을 하면 POST방식 요청의 한글 파라메터 처리가 가능해진다.

- **다음 중 SpringMVC의 Controller에 대한 설명으로 잘못된 것은?**

1. 클래스에 @Controller를 붙이면 Spring이 컨트롤러로 인식한다.

2. 메소드에 @RequestMapping을 붙이면 요청 받았을 때 그 메소드가 호출된다.

3. 메소드의 리턴을 통해 간편하게 포워딩이 가능하다.

4. 일반 JSP프로젝트보다 더 많은 컨트롤러 클래스가 필요하다.

- **다음 중 SpringMVC에서 요청 파라메터 처리에 대한 설명으로 잘못된 것은?**

1. 요청 파라메터 정보를 자바빈 객체로 자동 매핑해주는 기능이 있다.

2. 요청 파라메터명과 자바빈 객체의 멤버변수명을 같게 해주면 값이 자동으로 매핑된다.

3. cos.jar를 활용한 파일 업로드, 날짜 입력 등 자동 매핑이 되지 않는 경우도 있다.

4. 자동 매핑을 사용할 때 자바빈 객체의 클래스명은 .jsp파일명과 같아야 한다.

- **다음 중 SpringMVC와 MyBatis 연동에 관한 설명으로 잘못된 것은?**

1. 자바 interface를 통해 mapper.xml에 등록된 SQL을 사용할 수 있다.

2. DAO에 @Service를 붙여서 컨트롤러에서 @Autowired로 DAO를 불러다 사용하면 편하다.

3. SpringMVC와 MyBatis의 자동매핑 기능 덕분에 JSP보다 자바소스를 많이 줄일 수 있다.

4. pom.xml에 등록된 SqlSession을 @Autowired로 불러다 사용하면 편하다.

- **요청을 받았을 때 XML을 응답하게 @RequestMapping을 사용할 때 (?)안에 들어갈 것은?**

@RequestMapping(value = "/menu.getAll",  method = RequestMethod.GET, produces = "application/(    ?    ); charset=utf-8")

**xml**

**- 요청을 받았을 때 JSON을 응답하게 @RequestMapping을 사용할 때 (?)안에 들어갈 것은?**

@RequestMapping(value = "/student.getAll", method = RequestMethod.GET, produces = "(   ?   ); charset=utf-8")

**application/json**

**-----------------------------------------------------------------------------------------------------------------------------------------------------**

**jQuery 활용**

- **다음 JavaScript문장을 jQuery로 표현하면?**

var t = document.getElementById("test");

**var t = $("#test");**

**- 다음 jQuery문장을 해석하면?**

$("button").click(function(e){

     alert(e.pageX);

});

**HTML상의 모든 버튼중 아무거나 클릭하면 클릭한 마우스의 x좌표가 출력**

**- 다음 두 문법의 차이는?**

1) $("img")

2) $("&lt;img&gt;")

**1)은 현재 HTML상에 존재하는 모든 선택, 2)는 새로운 생성**

**- 다음 두 문장의 차이는?**

1) $(document).height();

2) $(window).height();

**1)은 HTML전체 높이, 2)는 브라우저의 높이**

- **AJAX에 대해 아는대로 쓰시오**

**비동기식으로 자바스크립트를 써서 XML파싱하는 행위. 비동기식/자바스크립트에서 DB정보를 활용할때 사용**

**- Cross-Domain AJAX문제를 해결할 수 있는 방법에 대해 아는대로 쓰시오.**

**프록시서버를 쓰거나, 서버쪽에서 설정해놓은 JSONP를 활용**

**-----------------------------------------------------------------------------------------------------------------------------------------------------**

**보안코딩**

**- 다음 중 보안코딩으로 볼 수 없는 것은?**

1. 정보를 가져오기 편하게 모든 정보를 쿠키를 담아 처리한다.

2. 로그인 유지 시간이 지났을 수 있으므로, 로그인 체크를 수시로 한다.

3. 불법적인 요청인지 체크할 수 있게 요청 파라메터값을 한번더 체크해본다.

4. 요청 헤더값을 활용해본다.

- **사이트 로그인 문제를 처리할 때 쿠키보다 세션을 활용하는 이유는?**

**쿠키는 클라이언트 컴퓨터에 저장되는 파일형태라 보안성이 낮다.**

**- 다음 중 사이트 외부에서 들어오는 불법적인 요청인지 검사할 때 취해볼 전략으로 옳지 않은 것은?**

1. 로그인 세션 체크

2. 서버주소 체크

3. 요청 헤더값 체크

4. 요청 파라메터값 체크

- **다음 중 로그인관련 보안코딩으로 잘못된 것은?**

1. 웹페이지에서 클라이언트 측의 입력을 받을 때 자바스크립트로 입력값을 유효성검사한다.

2. DAO에서 자바스크립트 유효성검사가 없다고 가정하고 한 번 더 입력값을 검사해본다.

3. DB에서 조회해온 계정정보는 따로 체크해볼 필요가 없다.

4. 수시로 세션을 체크해서 로그인이 유효한지 체크한다.

- **다음 중 보안성을 높이기 위한 프로그래밍 방법으로 적절하지 않은 것은?**

1. DB관리자를 통해, 비번은 암호화해서 DB에 저장하는게 좋다.

2. SQL 인젝션 공격이 있을 수 있으므로, 위험한 정보는 SQL로 처리한다.

3. HTTP보다는 인증서 설정을 통해 HTTPS를 사용하는게 좋다.

4. DDoS공격을 막기 위해, 짧은 시간에 여러번 요청이 들어오는것은 막아주는게 좋다.

**-----------------------------------------------------------------------------------------------------------------------------------------------------**

**빅데이터 수집**

**- 다음 중 빅데이터로 보기 어려운 것은?**

1. 어제 저녁 식단

2. 지난 5년간의 수도권 지하철 운행 정보

3. 지난 10년간의 서울 버스 운행 정보

4. 지난 20년간의 서울 권역별 미세먼지 측정 값

- **다음 중 빅데이터를 구하는 방식으로 가장 거리가 먼 것은?**

1. 아침 등교할 때 마다 직접 잰 기온들을 모아서

2. 내 근처의 치킨집 현황을 하둡을 활용해 모아서

3. 서울열린데이터광장에서 제공하는 지하철 JSON데이터를 parsing해서

4. 지인의 카드 결제 정보를 모아놓은 csv파일을 구해서

- **다음 중 Java와 Python의 차이에 대한 설명으로 잘못된 것은?**

1. Java보다 고급언어인 Python은 Java보다 개발자가 신경 쓸 부분이 적다.

2. 일반적으로 Python이 Java보다 소스가 짧다

3. Python은 변수를 만들 때 자료형을 반드시 알맞게 써줘야 한다.

4. Python은 {}를 사용하지 않고 들여쓰기로 코드 영역을 구분한다.

- **다음 중 빅데이터 프로젝트에서 Python의 역할로 가장 거리가 먼 것은?**

1. 외부 서버에서 데이터 받아오기

2. 받아온 데이터를 DB서버에 저장하기

3. DB서버에 저장되어있는 데이터를 분석하기 직전에 파일로 만들기

4. 받아온 데이터를 삭제하기

**- 다음 Python 프로그램의 실행 결과는?**

for i in range(0,5):

print(i, end='')

**에러**

**- 다음 Python소스를 해석하면?**

from urllib.parse import quote

q = quote('맛집')

**맛집을 URL인코딩으로 바꿔서 변수 q에 저장**

**- 다음 Python소스에서 loads의 기능은?**

from json import loads

kakaoBlog = loads(j)

**JSON을 Python객체(list+map)으로**