一、JDBC

二、JDBC常用四个核心对象

DriverManager

getConnection();

Connection

createStatement();

prepareStatement();

Statement (PreparedStatement)

ResultSet executeQuery();

int executeUpdate();

boolean execute();

delete from users where id=?

ps.setInt(1,5);

ResultSet

next();

getInt();

getString();

getDouble();

getDate();

...

三、编写一个java应用步骤：

1、创建项目，添加jar包

2、编写操作数据库的类

```
try{
//加载驱动
Class.forName("com.mysql.jdbc.Driver");
//创建连接Connection
Connection conn =
DriverManager.getConnection("jdbc:mysql:///day06","root","abc");
//得到执行sql的statement对象
//conn.createStatement();
```

```
                PreparedStatement ps = conn.prepareStatement("select * from
users where name=? and pwd=?");
                ps.setString(1,"tom");
                ps.setString(2,"123");
                //执行语句，并返回结果
                ResultSet rs = ps.executeQuery();
                //处理结果
                while(rs.next()){
                        User u = new User();
                        u.setId(rs.getInt(1));
                        ....
                }
                }catch(Exception e){
                        e.printSt...
                }finally{
                        //关闭资源
                        if(rs!=null)
                                rs.close();
                        if(ps!=null)
                                ps.close();
                        if(conn!=null)
                                conn.close();
                }
```

xml：

XML：eXtensible Markup Language 可扩展标记语言 version="1.0"

* 可扩展：所有的标签都是自定义的。
* 功能：数据存储
    * 配置文件
    * 数据传输

* html与xml区别：
    * html语法松散，xml语法严格
    * html做页面展示，xml做数据存储
    * html所有标签都是预定义的，xml所有标签都是自定义的

W3C:word wide web consortiem 万维网联盟
xml语法：

    * 文档声明：
        * 必须写在xml文档的第一行。
        * 写法：<?xml version="1.0" ?>
        * 属性：
            * version：版本号 固定值 1.0
            * encoding:指定文档的码表。默认值为 iso-8859-1
            * standalone：指定文档是否独立 yes 或 no

    * 元素：xml文档中的标签
        ** 文档中必须有且只能有一个根元素
        * 元素需要正确闭合。<body></body>
        * 元素需要正确嵌套
        * 元素名称要遵守：
            * 元素名称区分大小写
            * 数字不能开头

    * 文本：
        * 转义字符：>
        * CDATA: 里边的数据会原样显示
            * <![CDATA[数据内容]]>

    * 属性：
        * 属性值必须用引号引起来。单双引号都行
    * 注释：
        <!-- -->

* 处理指令：现在基本不用

<?xml-stylesheet type="text/css" href="1.css"?>

xml约束：

* 约束就是xml的书写规则
* 约束的分类：

dtd：

dtd分类：

* 内部dtd：在xml内部定义dtd
* 外部dtd：在外部文件中定义dtd

* 本地dtd文件：<!DOCTYPE students SYSTEM "student.dtd">

* 网络dtd文件：<!DOCTYPE students PUBLIC "名称空间" "student.dtd">

schema：

导入xsd约束文档：

1、编写根标签

2、引入实例名称空间 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

3、引入名称空间 xsi:schemaLocation="http://www.itcast.cn/xml student.xsd"

4、引入默认的名称空间

xml构造思想：

XML解析：

* 解析xml可以做：
  * 如果xml作为配置文件：读取
  * 如果xml作为传输文件：写，读
* xml解析思想：
  * DOM：将文档加载进内存，形成一颗dom树(document对象)，将文档的各个组成部分封装为一些对象。
    * 优点：因为，在内存中会形成dom树，可以对dom树进行增删改查。

* 缺点：dom树非常占内存，解析速度慢。

Document

Element

Text

Attribute

Comment

* SAX：逐行读取，基于事件驱动
    * 优点：不占内存，速度快
    * 缺点：只能读取，不能回写

* xml常用的解析器：
    * JAXP：sun公司提供的解析。支持dom和sax。
    * JDOM：
    * DOM4J：dom for java民间方式，但是是事实方式。非常好。 支持dom
        1.导入jar包 dom4j.jar
        2.创建解析器
            SAXReader reader = new SAXReader();
        3.解析xml 获得document对象
            Document document = reader.read(url);

* XPATH：专门用于查询
        * 定义了一种规则。
        * 使用的方法：
            * selectSingleNode():
            * selectNodes():
        使用步骤：
            1、注意：要导包 jaxen...jar
            2、创建解析器
                SAXReader reader = new SAXReader();
            3、解析xml 获得document对象
                Document document = reader.read(url);

* 解析XML：

// 1、得到某个具体的节点内容:第2本书的书名--》葵花宝典

// 2、遍历所有元素节点

XPath：

// nodename 选取此节点。

// / 从根节点选取。

// // 从匹配选择的当前节点选择文档中的节点，而不考虑它们的位置。

// .. 选取当前节点的父节点。

// @ 选取属性。

// [@属性名] 属性过滤

// [标签名] 子元素过滤