# 1.为什么要有yang？

Netconf需要对设备的配置和状态做操作，例如编辑配置，获取状态，因此需要一种语言来对configuration和status进行建模，建好的模型需要通过XML进行实例化。打个比方，我需要向领导请假，领导说你写个请假单，包含请假人的姓名，请假的起止时间，请假事由和代理人。于是我做了一个表格，包含了上述要求，并根据实际情况填入了真实信息。那么领导的描述，就可以理解为“建模”，而我最后提交的填好内容的表格，就是将模型实例化了。

# 2.建模需要什么呢?

## 2.1module和submodule

Module的header主要是一些描述信息，而body就是data model定义的地方。Module可以分为submodule，模块化的好处就是方便引用。

## 2.2data modeling basic

敲黑板，重点来了。这里介绍最基础的四种结构

### 2.2.1leaf Nodes

**一个leaf Nodes包含且只有一个value，**可以是数字或字符串，具体是什么，看关键字后面“type”后面跟的什么。Leaf node 下面不能挂子节点。

例如：

YANG Example:

leaf host-name {

type string;

description "Hostname for this system";

} -----------------------

NETCONF XML Example: <host-name>my.example.com</host-name>

此处YANG定义了一个名为host-name的leaf，它包含一个对自己的description，有一个string类型的值。那么当XML实例化这个leaf的时候，就需要对host-name进行具体的赋值。换句话说，YANG是挖坑的，XML是填坑的，但是XML填坑用的材料形状得和YANG定义的一样。

### 2.2.2leaf-list Nodes

可以认为leaf-list Nodes表示的是一个“数组”，“数组”中的元素的值必须与type保持一直，而且不能重复。

leaf-list domain-search {

type string;

description "List of domain names to search";

}

-----------------------

NETCONF XML Example:

<domain-search>high.example.com</domain-search>

<domain-search>low.example.com</domain-search>

<domain-search>everywhere.example.com</domain-search>

和leaf node一样，它也只定义一个value，但是可以有一系列同类型的值，例子中的domain-search的值多个，但是定义和类型都是统一的。