1.

Django有Form类用于支持网页的表单提交，其中包含了表单验证，表单渲染成页面等功能。

2.

表单域label关键字，用于表示当渲染成HTML页面时候的文字说明。

表单域max\_length关键字，用于表示此域可接受的最大长度。

表单域initial关键字，用于表示此域的默认值。

3.

关于表单的处理上，只会有两种HTTP请求与之相关，分别是GET和POST.

GET请求将数据以键值对的形式拼接在URL中，它不应该改变后台状态以及数据库的状态。

POST请求将数据以数据包的形式封装在POST请求体中，它将改变后台状态以及数据库状态。并且Django将会对POST请求执行CSRF验证。

4.

Form类的is\_valid() 方法用于验证表单中所有域是否满足规定要求。is\_valid()方法需要用于bound的表单，在非bound的表单上也可以认证，但是永远is\_valid()都是False并且form.errors也没有错误信息。

如果满足则返回True并且将所有合法的数据放入cleaned\_data属性中,并且cleaned\_data属性中的数据类型都根据Form定义中转变好,对应的成为数字，字符串等。

如果不满足则返回False并且将错误信息放入form实例.errors中。

Form类的is\_bound属性将返回Form实例是否是被绑定的。

5.

Django的Form自动产生的HTML不包括<form>这个元素，只包括<input>表单元素，<form>元素我们需要自己添加。

6.

Djang中unbound Form是没有数据，初始化的Form。 而bound Form是根据**用户提供的数据**以后绑定了数据的Form。

例如：

TestForm = FormA()

TestForm.fields[‘username’] = “AAA” #此TestForm是unbound的

TestForm = FormA ({}) 或者 TestForm = FormA({request.POST}) #此TestForm是bound的

TestForm = FormA (initial= {}) #此TestForm是unbound的

7.

{{Form}} 在模板当中就会根据Form的数据render成Html语言。并且Django根据Form中定义的域的类型变成相应的Html表单元素：

例如：URLField, EmailField将变为type=”url”, type=” email” 的HTML表单元素

8.

Form的Widgets

Django Form中的widget用于将一个FormField变为对应的Html表单元素。

例如CharField默认变为<input type=”text”>， 这样的行为是因为CharField有默认的Widget.我们也可以手动设置去改变这个默认的widget例如forms.CharField(widget=forms.Textarea)

9.

将Form实例变为html 时候我们可以直接在模板中使用{{form}},同时Django还提供了其他一些render方式：

{{form.as\_table}} 将把Form变为表格显示

{{form.as\_p}} 将把Form变为<p>显示

{{form.as\_ul}} 将把表单变为ul显示。

P.s 无论是{{form}}, {{form.as\_table}}, 这些render出来的html代码都不包含<form>自身这个标签，需要自己添加。

P. s 除了可以直接{{form}} 将form实例变为HTML代码以外，我们还可以使用{{form.field}}来特定将form实例中的某个域变为HTML.Django会根据该域的类型使用默认的widget生成对应的HTML代码（但是不会生成对应的Label）。

10.

在用户第一次访问Form页面的时候，我们可以让Form预先填充信息。

--我们可以在Form的定义时候，是否initial关键字，初始化某个属性的初始值。

例如name = forms.CharField(initial='Your name')

--我们也可以在创建Form的时候，使用initial关键字参数，传入字典来初始化Form.这样初始化比Form定义使用initial关键字更加动态灵活。

例如f = CommentForm(initial={'name': 'instance'}

--我们也可以创建一个Form的实例，然后逐一为Form中的属性赋值。

例如form = JiraPTSForm(initial=data)

form.fields[‘name’] = “AAA”

11.

class JiraPTSForm(forms.Form):

impact = forms.ChoiceField(label=u'Impact', choices=IMPACT\_CHOICES, required=True)

访问impact中的choices时可以 JiraPTSForm的实例.fields[‘impact’]. choices可读可写。

12.

如果我们自定义Form的类，需要实现类内的方法clean该方法用于验证表单在验证数据有效时候的行为。

13.

如果我们在Form中定义了三个数据域，但是传给Form的数据中有多余的数据域时，该Form进行验证的时候只会验证在Form中定义的数据域，如果这些数据域都符合要求，则为valid的，并且form.cleaned\_data中也只会包含Form原本定义中的数据域，不会含有多余的数据域。

14.

forms. Form和forms.ModelForm用途不一样，forms.ModelForm会有Meta指示于models.py中的哪一个表相符合用于插入数据库中，而forms.Form更像用于网页的表单提交。

15.