1.在数据库中使用UTC时区保存时间，是好的做法。避免每年因为Daylight Saving Time所引起的时间推迟提早1小时引起的错误。

USE\_TZ = True将为Django启动时区的支持。并且在存入数据库中的时候，都会带有时区。如果在使用了USE\_TZ但是试图存入naïve的datetime时候，Django会自动根据TIME\_ZONE的设置进行转化。

2.

在django.utils.timezone模块中，

is\_aware(value)方法：判断传入的datetime是否是设置时区的，True为设置时区，False为没有设置时区

is\_naive(value)方法：判断传入的datetime是否没有设置时区，True为没有设置时区，False为设置时区

now () 方法：如果USE\_TZ设置为False则返回一个naïve的datetime系统现在的时间。如果USE\_TZ设置为True, 则返回一个aware的datetime时间，该datetime设置为UTC时区（值根据当地时区转换过来）。

3.

Django项目中有default time zone以及current time zone设置。

Default time zone是通过TIME\_ZONE在settings.py中设置。

current time zone是通过django.utils.timezone.active方法来设置，如果不设置则值默认域default timezone.

4.

当USE\_TZ = True的时候，Django将使用current time zone解析form中填入的时间，并且在form.cleaned\_data中是该时区的aware datetime.

当USE\_TZ = True的时候，Django将使用current time zone来在模板中显示时间。

一个Datetime在Django中的变化流程：

我们在Django中设置USE\_TZ = True, TIME\_ZONE = “America/Los\_Angeles”

-🡪提交表单，从表单中将一个datetime放入数据库中。

用户输入2015/6/10， 8AM然后提交，Django发现我们启动了USE\_TZ = True,并且Time\_ZONE = “America/Los\_Angeles”,另外我们没有额外设置Current time zone.所以Current time zone默认也为Time\_Zone的值。当我们调用form.is\_valid的时候， Django将这个时间字符串变为aware的，并且时区为 “America/Los\_Angeles”. 然后将这个aware datetime放入数据库中（在放入数据库中的格式根据数据库有所不同）。

-🡪 从数据库中取出该时间并且显示在Django模板上

Django根据current time zone将数据库中的aware datetime变为current time zone的格式显示。

5.

UTC时间（timezone=UTC）在世界上任何地点，任何时区都是一样的。以原子时为基础，在时间计算中是最重要的基准。

Aware的datetime是设置了时区，所以可以方便的转化为UTC。而Naïve的datetime没有设置时区，所以直接变为UTC时间。