**自定义时间处理过滤器案例**

例如：操作发布新闻 与现在的时间间隔

|  |
| --- |
| **from** datetime **import** datetime  *#需求：操作发布新闻 与现在的时间间隔*  @app.template\_filter(**'handle\_time'**)  **def** handle\_time(time):  *"""*  *time距离现在的时间间隔*  *1. 如果时间间隔小于1分钟以内，那么就显示“刚刚”*  *2. 如果是大于1分钟小于1小时，那么就显示“xx分钟前”*  *3. 如果是大于1小时小于24小时，那么就显示“xx小时前”*  *4. 如果是大于24小时小于30天以内，那么就显示“xx天前”*  *5. 否则就是显示具体的时间 2018/10/20 16:15*  *"""*  **if** isinstance(time, datetime):  now = datetime.now()  timestamp = (now - time).total\_seconds()  **if** timestamp < 60:  **return "刚刚"**  **elif** timestamp >= 60 **and** timestamp < 60 \* 60:  minutes = timestamp / 60  **return "%s分钟前"** % int(minutes)  **elif** timestamp >= 60 \* 60 **and** timestamp < 60 \* 60 \* 24:  hours = timestamp / (60 \* 60)  **return '%s小时前'** % int(hours)  **elif** timestamp >= 60 \* 60 \* 24 **and** timestamp < 60 \* 60 \* 24 \* 30:  days = timestamp / (60 \* 60 \* 24)  **return "%s天前"** % int(days)  **else**:  **return** time.strftime(**'%Y/%m/%d %H:%M'**)  **else**:  **return** time |

使用：

<**p**>发布时间：{{**新闻创建时间**|**handle\_time**}}</**p**>