**Flask\_信号机制：**

**1.信号机制：**

大白话来说，类似于两方属于敌对关系时，某人在敌对方阵营进行交谈，一旦遇到特殊情况，某人便会发送信号，他的同伙接收（监听）到他发的信号后，同伙便会做出一系列的应对策略(杀进去|撤退)。

flask中的信号使用的是一个第三方插件，叫做blinker。通过pip list看一下，如果没有安装，通过以下命令即可安装blinker：

pip install blinker

**2.自定义信号步骤**

自定义信号可分为3步来完成。

第一是创建一个信号，第二是监听一个信号，第三是发送一个信号。

以下将对这三步进行讲解：

**1. 创建信号：**定义信号需要使用到blinker这个包的Namespace类来创建一个命名空间。比如定义一个在访问了某个视图函数的时候的信号。示例代码如下：

|  |
| --- |
| # Namespace的作用：为了防止多人开发的时候，信号名字冲突的问题  from blinker import Namespace  mysignal = Namespace()  signal1 = mysignal.signal('信号名称') |

**2. 监听信号：**监听信号使用signal1对象的connect方法，在这个方法中需要传递一个函数，用来监听到这个信号后做该做的事情。示例代码如下：

|  |
| --- |
| def func1(sender,uname):  print(sender)  print(uname)  signal1.connect(func1) |

**3. 发送信号：**发送信号使用signal1对象的send方法，这个方法可以传递一些其他参数过去。示例代码如下：

|  |
| --- |
| signal1.send(uname='momo') |

**3.代码演示：**

|  |
| --- |
| **from** flask **import** Flask  **from** blinker **import** Namespace  app = Flask(\_\_name\_\_)  #【1】信号机制 3步走  # Namespace：命名空间  #1.定义信号  sSpace = Namespace()  fire\_signal = sSpace.signal(**'发送信号火箭'**)  #2.监听信号  **def** fire\_play(sender):  print(sender)  print(**"start play"**)  fire\_signal.connect(fire\_play)  #3.发送一个信号  fire\_signal.send()  **if** \_\_name\_\_ == **'\_\_main\_\_'**:  app.run(debug=**True**) |

**4.了解Flask中文网信号：**

|  |
| --- |
| <http://docs.jinkan.org/docs/flask/signals.html> |