**性能测试-Jmeter参数化应用**

|  |
| --- |
| - 掌握Jmeter环境搭建  - 理解Jmeter的主要元件及相关的作用域  - 理解线程组、HTTP请求的配置使用  - 掌握Jmeter参数化的应用 |

**性能测试工具**

**JMeter和Loadrunner工具对比：**

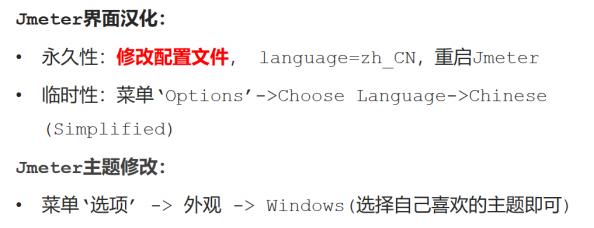


**jmeter的功能概要：**

**jmeter文件目录结构：**

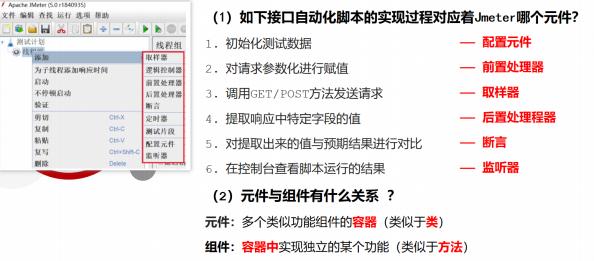


**基本配置**



**jmeter元件及基本作用域（理解）：**

**基本元件：**



**作用域的原则：**

Ø取样器：核心，没有作用域

Ø逻辑控制器：只对其子节点中的取样器和逻辑控制器起作用

Ø其他元件：

•如果是某个取样器的子节点，则该元件只对其父节点起作用

•如果其父节点不是取样器，则其作用域是该元件父节点下的其他所有后代节点（包括子节点，子节点的 子节点等）

**元件执行顺序：**

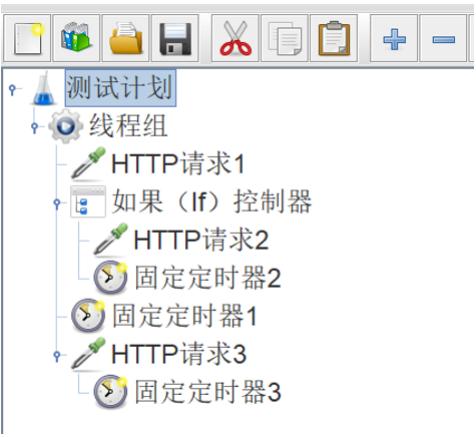
**在同一个作用域（目录/级别/缩进）的不同元件的执行顺序：**

配置元件 - 前置处理程序 - 定时器 - 取样器 - 后置处理程序 - 断言 - 监听器

**在同一个作用域（目录/级别/缩进）的相同元件的执行顺序：**

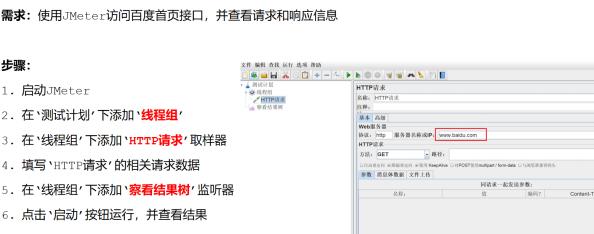
从上到下的顺序依次执行

**案例：执行顺序：**



正确：定时器1 -请求1 -定时器1 -定时器2 -请求2 -定时器1 -定时器3 -请求3

**Jmeter第一个案例：**



**Jmeter三个重要组件（重点）：**

**线程组的介绍：**

**特点：**

•控制Jmeter用于执行测试的一组用户

**线程组的分类：**

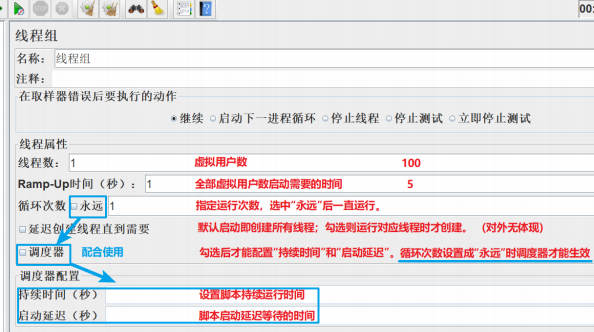
•Setup线程组：预测试操作，所有脚本之前执行

•**普通线程组：** 执行测试用例，可以有1个或者多个（并行/串行）

•Teardown线程组：测试后操作，所有脚本之后执行

**线程组的属性：**

**参数介绍：**





**案例分析：**

（5）使用1个线程组，添加HTTP请求（百度）

 配置线程数为2，循环次数为3时，运行观察结果

 配置线程数为3，循环次数为2时，运行观察结果，对比不同

分析：

 线程数代表虚拟用户数，用户数越多，负载越大

 循环次数代表运行时间，次数越多，运行时间越长

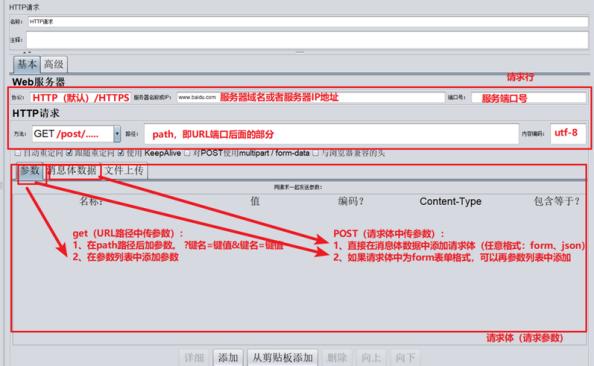
**HTTP请求：**

**参数介绍：**

**作用：** 向服务器发送http及https请求

**位置**：选中线程组->右键->添加->取样器->HTTP请求

**参数**：



案例一（使用HTTP请求路径来传递get请求参数）：

（1）使用1个线程组，添加HTTP请求（百度），路径为： /S

 使用HTTP协议， GET请求方法，路径中添加参数为: wd = test，运行观察结果

案例二（使用参数列表来传递get请求的参数）：

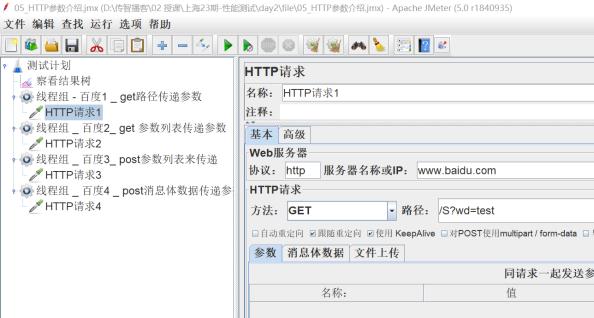
 使用HTTPS协议， GET请求方法，路径中添加参数为: wd = test，运行观察结果

案例三（使用参数列表来传递POST请求的form格式参数）：

 使用HTTPS协议， POST请求方法，消息体数据中添加参数为： wd = test，运行观察结果

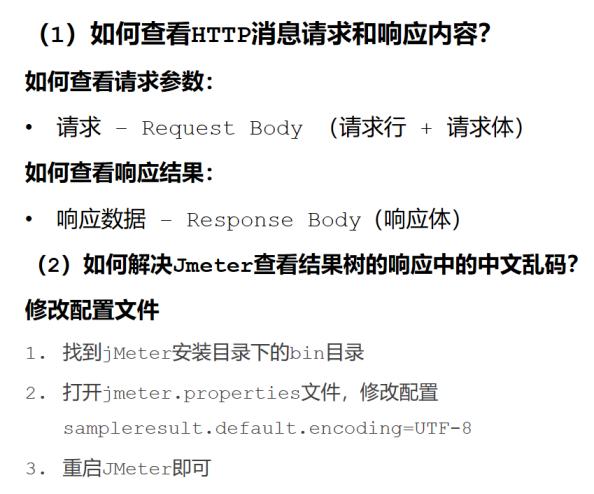
案例四：

 使用HTTP协议， POST请求方法，消息体数据中添加参数为: wd = test，运行观察结果



**查看结果树：**

**各部分的含义：**



**Jmeter参数化（重点）**

定义：使用不同的测试数据，调用相同的测试方法进行测试

本质：实现测试数据与测试方法的分离。

实现方式：

 用户定义的变量 —— 全局变量

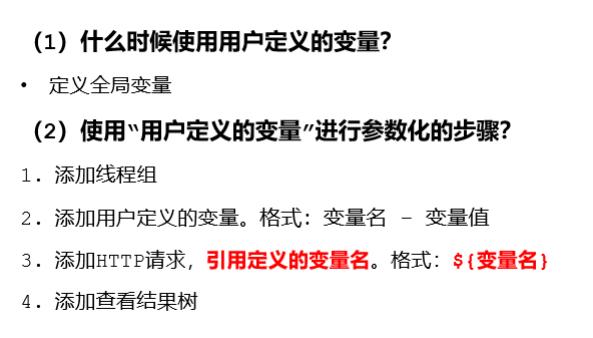
 用户参数 —— 为每个用户分配不同的参数值

 CSV数据文件设置 —— 文件方式参数化

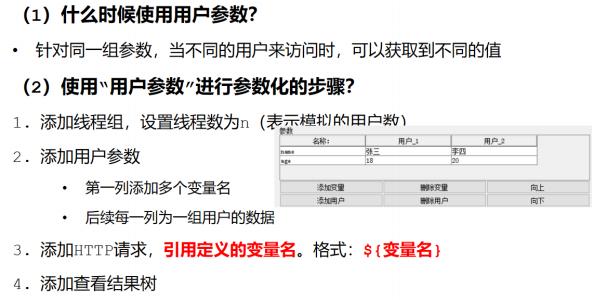
 函数 —— 随机数据

 数据库

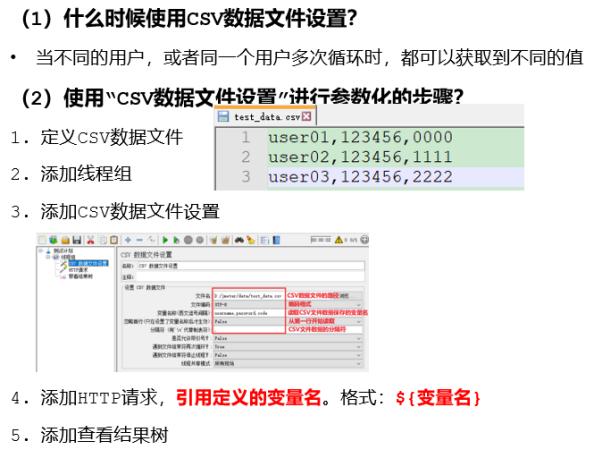
**用户定义的变量：**



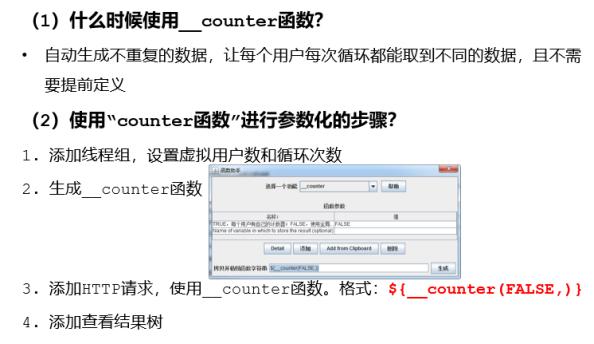
**用户参数：**



**CSV数据文件设置：**



**函数：**



**四种参数化方式的对比：**

