http模块帮助我们实现HTTP服务器和客户端

命令行中执行

```
curl -v http://localhost:8080
 curl 是一个获取服务器资源的linux命令
```

http服务器

创建服务器并指定监听 **请求处理函数**

```
http.createServer(requestListener(request,response))
```

- request http.IncomingMessage 客户端 **请求**

服务器监听

```
server.listen(port,[host],[backlog],[callback])
```

- port 端口
- host 主机
- backlog **等待** 中的队列数量,默认值是511
- callback **请求** 到来的时候服务器调用的回调函数

珠峰培训
Everest training

http接收客户端数据

http接收客户端请求的第一个参数

为 `http.IncomingMessage` 对象,有如下的属性:

- method 客户端请求的 **方法**
- url 请求时使用的 **url** 参数字符串
- headers 请求 **头对象** ，包括客户端所有请求头信息，包括cookie
- httpVersion HTTP **版本**

常见请求头信息

从客户端发往服务器发送请求报文所使用的字段，用于补充请求的附加信息

- host 请求的服务器主机。HTTP/1.1请求 **必须** 包含主机头，否则会返回400状态码。
- connection 客户端和服务器 **连接** 选项
- accept 告诉服务器客户端能够处理的 **内容类型** 和 **优先级** q=表示权重，用分号;分隔，范围是0-1，不指定时权重默认为1
- user-agent **用户代理** ，是指浏览器,它的信息包括硬件平台、系统软件、应用软件和用户个人偏好
- accept-encoding 告诉服务器客户端支持的 **内容编码** 及内容编码的 **优先级** 顺序
- accept-language 告诉服务器能够处理的 **语言** 以及优先级

珠峰培训
Everest training

```
var urlObj = url.parse('原始的url');
```

urlObj的属性

- href 被转换的 **原URL** 字符串
- protocal 客户端请求时的 **协议**
- slashes 在协议与路径中间是否使用 **//** 分隔符
- host URL字符串中完整的 **地址及端口号** ，可能为IP也可能为主机名
- auth **认证** 部分
- hostname **主机名或IP**
- port **端口** 号
- pathname **路径** 不包含查询字符串
- query 不包含起始字符?的 **查询字符串** ,或根据该查询字符串转换而成的对象(由parse方法的 **第二个** 参数决定, **true** 就会转成 **对象**)

珠峰培训
Everest training

查询字符串

querystring用来对查询字符串进行转换

```
var queryObj = querystring.parse(str,[sep],[eq],[options]); //字符串转对象
var queryStr = querystring.stringify(obj,[sep],[eq]); //对象转字符串
```

- str 需要被转换的 **查询字符串**
- sep 查询字符串中的 **分割** 字符,默认为 **&**
- eq 查询字符串中的 **分配** 字符，默认参数值为 **=**
- options 为对象参数，可以指定maxKeys属性指写转换后的 **属性个数**,0为不限定

```
response.writeHead(statusCode,[reasonPhrase],[headers]);
```

- statusCode **状态码**
- reasonPhrase 状态码 **描述** 信息
- headers 响应头对象
  - content-type **内容类型**
  - location **重定向** 到的URL地址
  - content-disposition 下载的 **文件名**
  - content-length 响应内容的 **字节数**
  - set-cookie 写入客户端 **cookie**
  - content-encoding 响应内容的 **编码** 方式
  - Cache-Control **缓存**
  - Expires 指定缓存 **过期时间**
  - Etag 服务器响应 **内容没有变化** 时不重新下载数据
  - connection 默认是keep-alive **保持连接** ，想断开连接用close

珠峰培训
Everest training

设置响应头

- setHeader方法可以单独设置响应头

```
response.setHeader(name,value);
```

- 如果多个响应头的话可以使用数组

```
response.setHeader('Set-Cookie',['name=zfpx','age=6']);
```

其它响应设置

- getHeader **获取** 响应头
- removeHeader **移除** 响应头
- headerSent 响应头是否 **已经发送**
- sendDate 是否发送 **响应时间**
- statusCode 设置 **响应码**

创建http客户端

request方法可以向其它网站请求数据

- options
  - host **域名**或目标主机IP
  - hostname **域名**或目标主机IP,优先级比host高
  - port **端口**号
  - method 请求**方法**
  - path 请求的**路径**,默认为 /
  - headers 客户端请求**头对象**
  - auth **认证**信息 如 "username:password"
- callback = function(response){} 当**获取**到目标网站所返回的**响应流**时调用的回调函数
  - `response`是一个**http.IncomingMessage**对象,可以从中**读**

珠峰培训

写入请求并发送请求

- write方法向目标服务器 **发送** 数据,write方法可以多次调用

```
request.write(chunk,[encoding]);
```

  - chunk 要发送的 **数据** ,可以是Buffer或字符串
  - encoding **编码** ,不指定时默认是utf8

- end方法用来 **结束** 本次请求

```
request.end(chunk,[encoding]);
```

用HTML5上传文件

需求

- 选中文件后显示上传的文件信息，比如 **文件名** 、 **类型** 、 **尺寸**
- 一个能够显示 **真实** 进度的 **进度条**
- 上传的 **速度**
- **剩余** 时间的估算
- **已上传** 的数据量
- 上传结束后服务器返回上传后保存的图片并在页面中 **显示** 出来