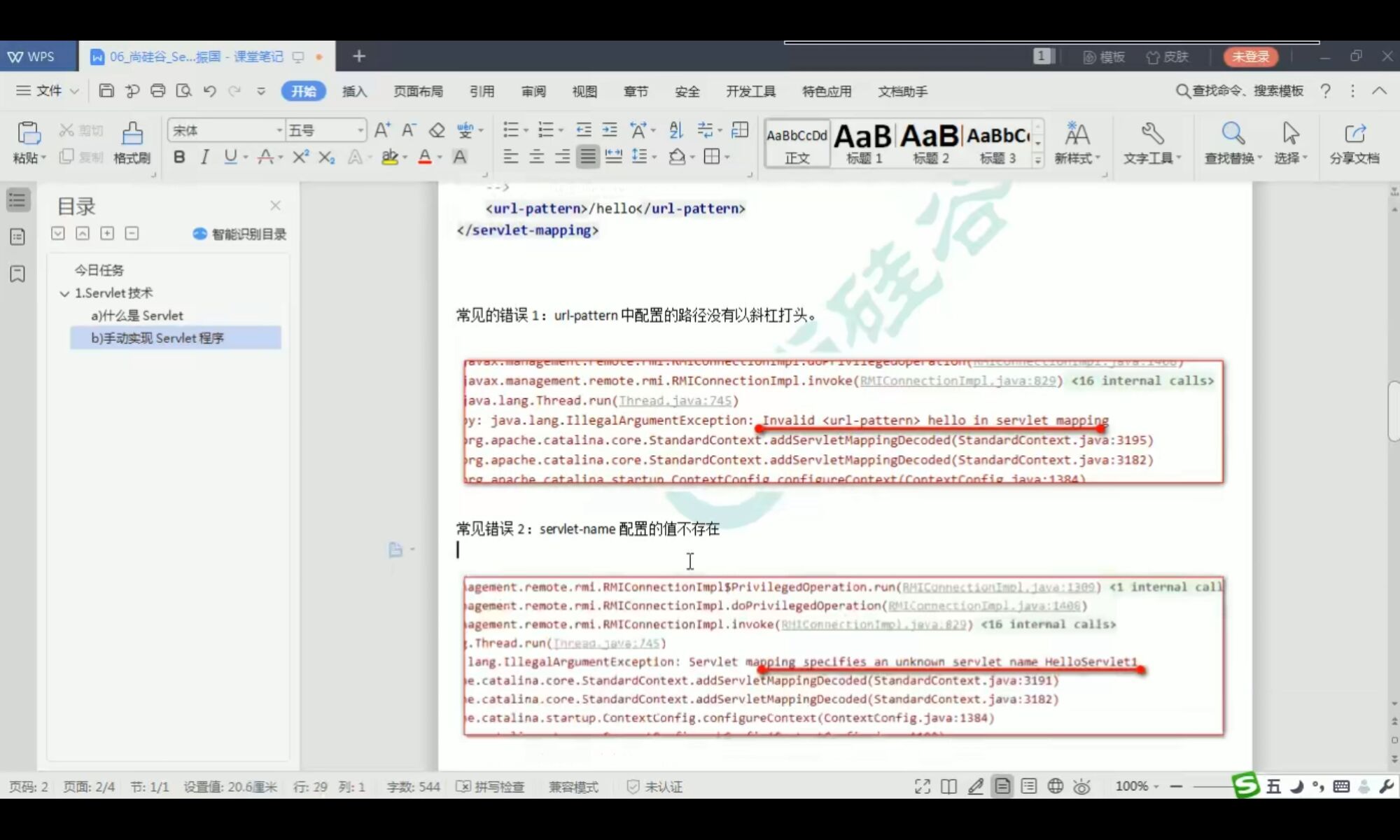
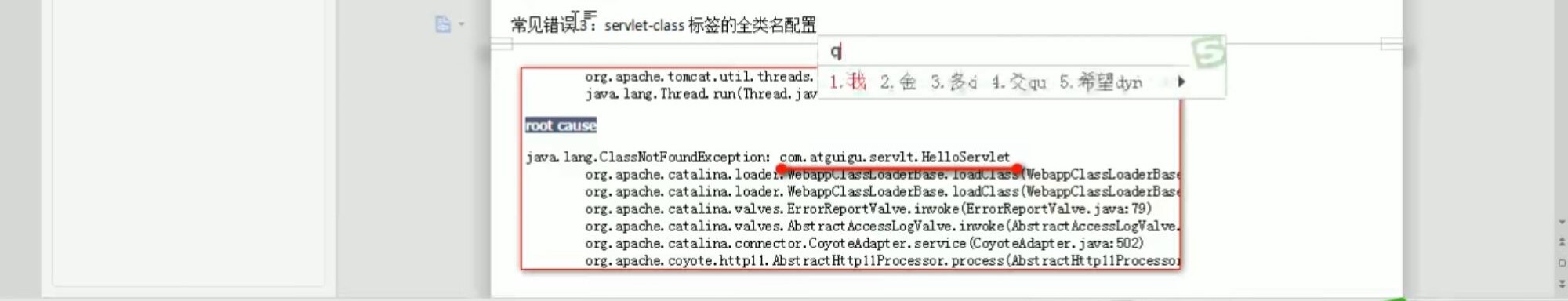
1. Servlet技术
2. 什么是servlet
3. servlet是javaEE规范之一。规范就是接口
4. servlet就是Javaweb三大组件之一。三大组件分别是servlet程序，filter过滤器，listener监听器
5. servlet是在运行服务器上的一个java小程序，它可以接受客户端发送过来的请求，并响应数据给客户端
6. 手动实现servlet程序
7. 编写一个类去实现servlet接口
8. 实现servlet方法，处理请求，并响应数据
9. 到web.xml中去配置servlet程序的访问地址
10. <**servlet-mapping**>  
     *<!--servlet-name作用是告诉服务器，我当前配置的地址给哪个serlvet程序使用-->* <**servlet-name**>helloservlet</**servlet-name**>  
     *<!--url-pattern标签配置访问地址  
     /斜杠在服务器解析的时候，表示地址为：http://ip:port/工程路径  
     /hello表示地址为：http://ip:port/工程路径/hello 可以自定义-->* <**url-pattern**>/hello</**url-pattern**>  
    </**servlet-mapping**>





1. servlet的生命周期
2. 执行servlet构造器方法
3. 执行init初始化方法
4. 执行service方法
5. 执行destroy销毁方法
6. 通过继承Httpservlet实现Servlet程序
7. 编写一个类去继承Httpservlet类
8. 根据业务需要重写doGet或doPost方法
9. 到web.xml中配置servlet程序的访问地址
10. servletConfig类

servletConfig类从类名上看，就知道是Servlet程序的配置信息类

servletConfig程序和servlet对象都是由tomcat负责创建，我们负责使用的

servlet程序默认是第一次访问的时候创建，servletConfig是每个servlet程序创建时，就创建一个对应的servletConfig对象

servletConfig类的三大作用：

1. 可以获取servlet程序的别名servlet-name的值
2. 获取初始化参数init-param
3. 获取servletContext对象
4. servletContext类
5. 什么是servletContext？
6. servletContext是一个接口，它表示servlet上下文对象
7. 一个web工程只有一个servlet对象实例
8. servletContext对象是一个域对象
9. servletContext是在web工程部署启动的时候创建的。在web工程停止的时候销毁

域对象：是可以想map一样存取数据的对象。

这里的域指的是存取数据的操作范围，整个web工程

存数据 取数据 删除数据

Map put() get() remove()

域对象 setAttribute() getAttribute() removeAttribute()

2) servletContext类的四个作用

1. 获取web.xml中配置的上下文参数ciontext-pram

2. 获取当前的工作路径，格式：/工程路径

3. 获取工作部署后在服务器硬盘的绝对路径

4. 像Map一样存取数据

2.Http协议

a) 什么是HTTP协议

什么是协议？

协议是指双方或多方，互相约定好，大家都需要遵守的规则，叫做协议

所谓HTTP协议，就是指，客户端和服务器之间通信时，发送的数据，需要遵守的规则，叫HTTP协议，HTTP协议中的数据又叫报文

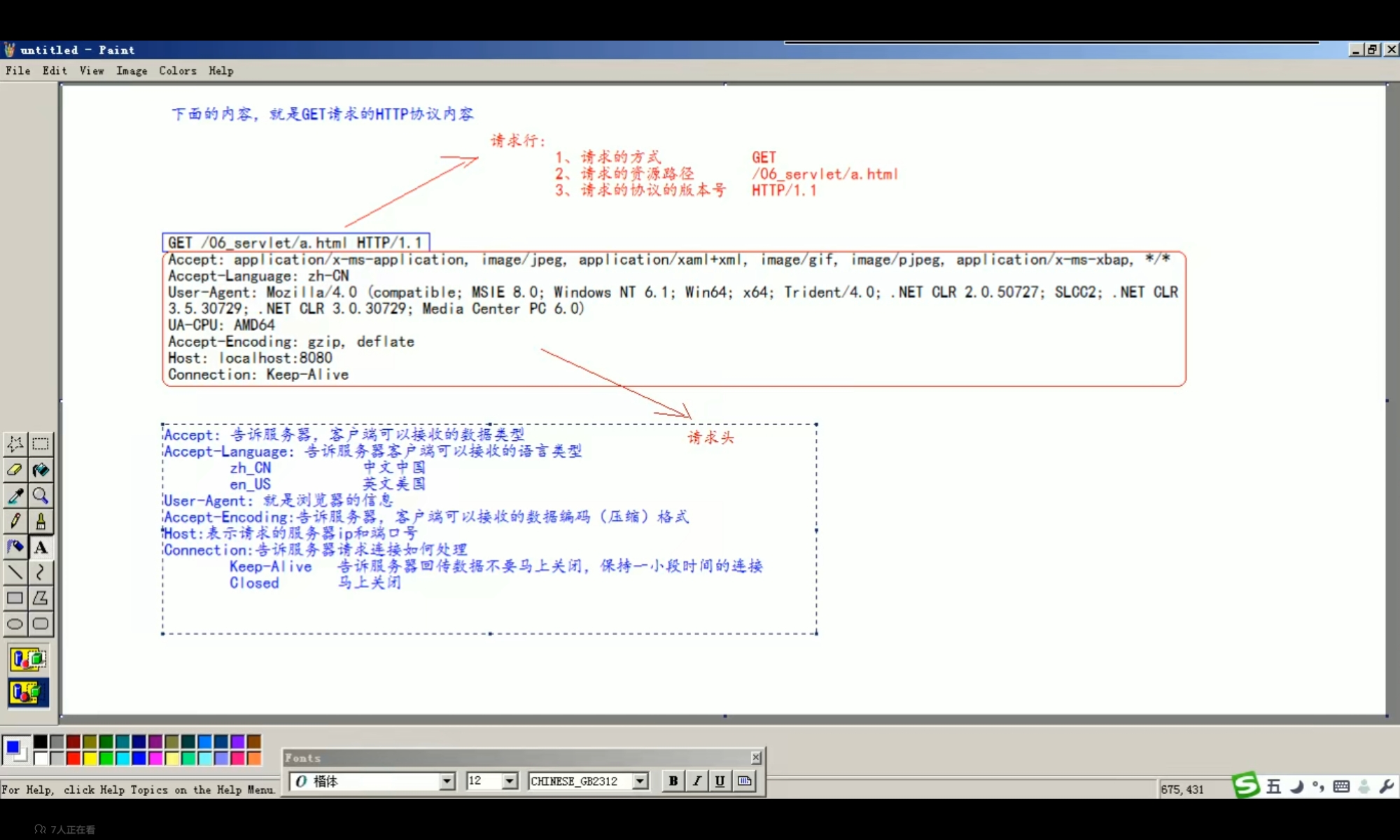
b)请求的HTTP协议格式

客户端给服务器发送数据叫请求

服务器给客户端回传数据叫响应

请求又分为GET请求和POST请求两种

1. GET请求
2. 请求行
3. 请求的方式 GET
4. 请求的资源路径[+?+请求参数]
5. 请求的协议的版本号 HTTP/1.1
6. 请求头

Key:value 组成 不同的键值对，表示不同的含义

1. POST请求

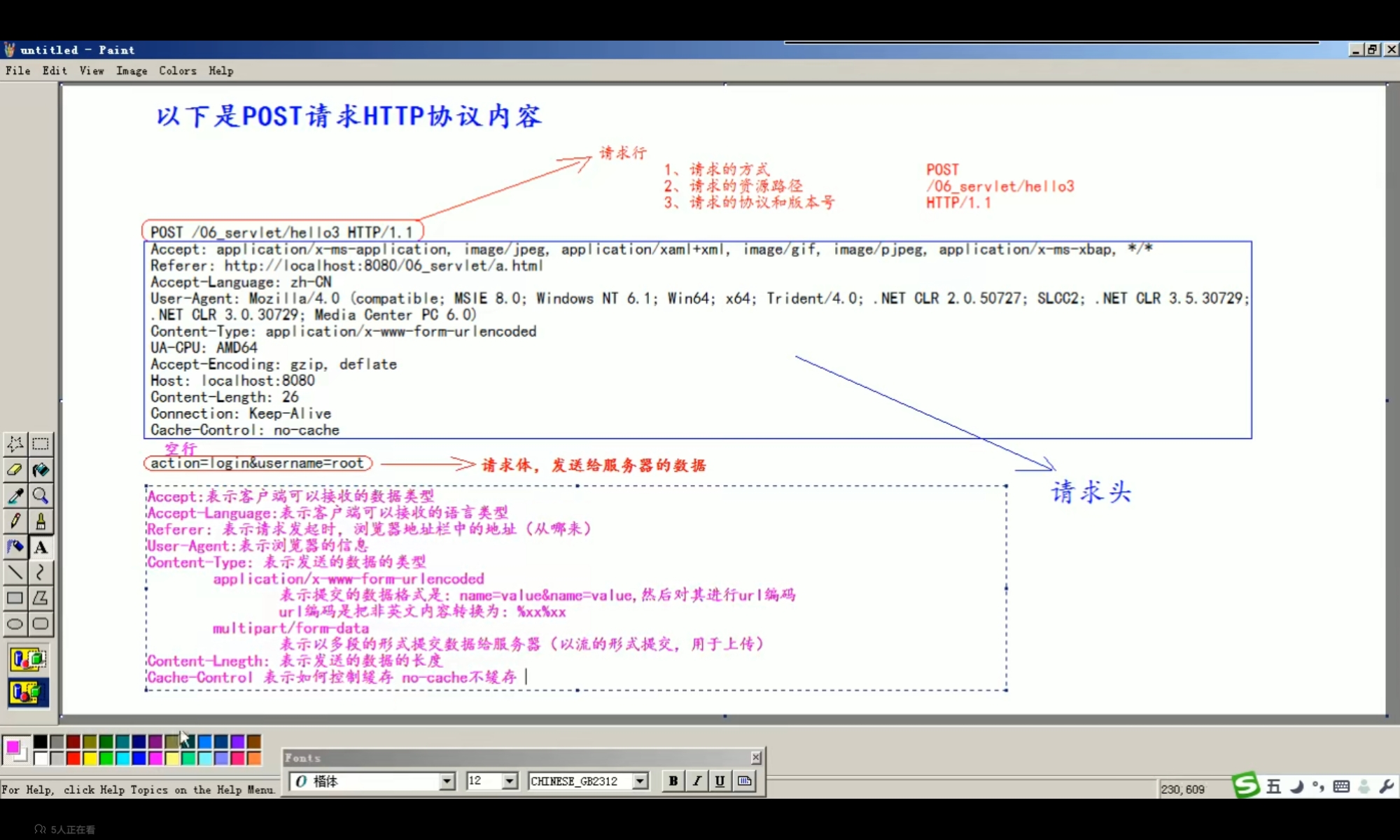
1.请求行

* 1. 请求的方式 POST
  2. 请求的资源路径[+?+请求参数]
  3. 请求的协议的版本号 HTTP/1.1

2.请求头

Key:value 组成 不同的键值对，表示不同的含义

空行

1. 请求体：==》发送给服务器的数据
2. 常用请求头的说明

Accept: 表示客户端可以接收的数据类型

Accept-Language：表示客户端可以接收的语言类型

User-Agent：表示当前客户端浏览器的信息

Host：表示请求时服务器ip和端口号

1. 哪些是GET请求，哪些是POST请求

GET请求：

1. form标签 methed=get
2. a标签
3. link标签引入css
4. Script标签引入js文件
5. img标签引入图片
6. iframe引入html页面
7. 在浏览器地址栏中输入地址后敲回车

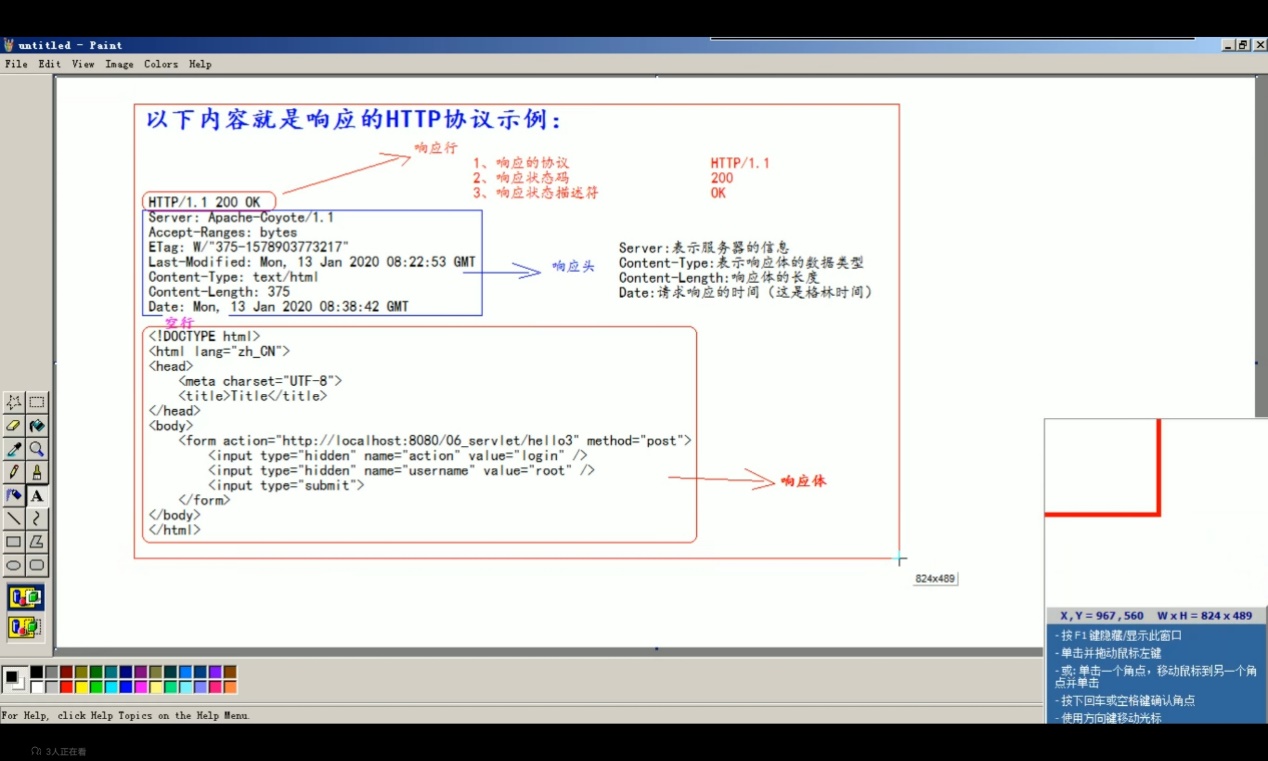
POST请求：

1. form标签 methed=post
2. 响应的HTTP协议格式
3. 响应行
   1. 响应的协议和版本号
   2. 响应状态码
   3. 响应状态描述
4. 响应头

Key:value 不同的响应头，有其不同的含义

空行

1. 响应体 ==》回传给客户端的数据

c)常见的响应码说明

200 表示请求成功

302 表示请求重新定向

404 表示请求服务器已经收到了，但是你要的数据不存在

500 表示请求服务器已经收到了，但是服务器内部错误（代码错误）

e)MIME类型说明

MIME是HTTP协议中数据类型

MIME类型的格式是“大类型/小类型“，并与某一种文件的扩展名相对应



3.HTTPServletRequest类

a) HTTPServletRequest类的作用：

每次只要有请求进入TomCat服务器，Tomcat服务器就会把请求过来的HTTP协议信息解析好封装到Request对象中，然后传递到service方法（doGet和doPost）中给我们使用。我们可以通过HttpServiceRequest对象，获取到所有请求的信息

b) HTTPServletRequest类的常用方法

i. getRequestURI() 获取请求的资源路径

ii. getRequestURL() 获取请求的统一资源定位符（绝对路径）

iii. getRemoteHost() 获取客户端的ip地址

iv. getHeader() 获取请求头

v getParameter() 获取请求的参数

vi getParameterValues() 获取请求的参数（多个值的时候使用）

vii getMethod() 获取请求的方式GET和POST

viii setAttribute(key) 设置数据域数据

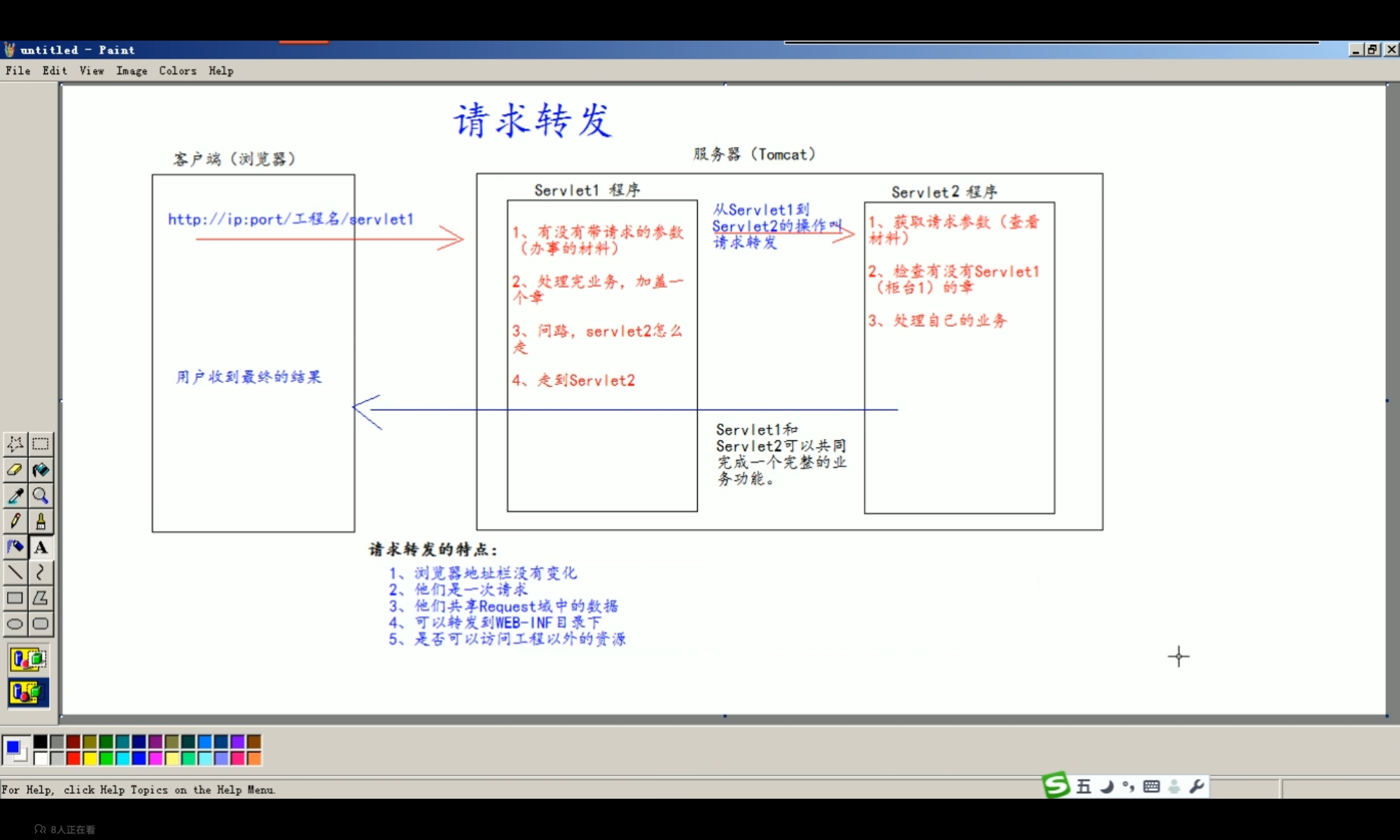
ix getAttribute(key) 获取数据域数据

x getRequestDispatcher() 获取请求转发对象

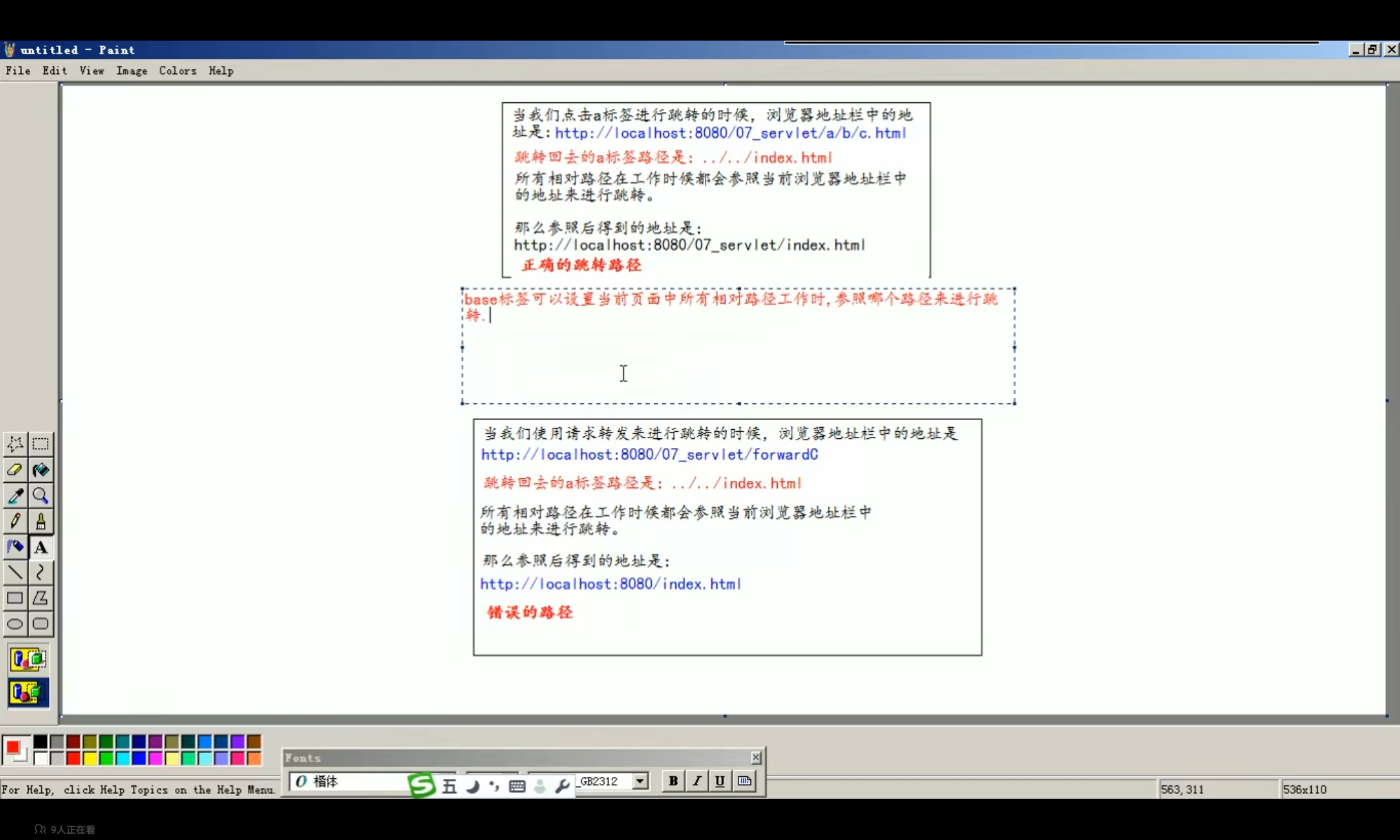
c) 请求和转发

什么是请求的转发？

请求的转发是指服务器收到请求后，从一次资源跳转到另一个服务器资源的操作叫做请求转发



d)base标签



**d)web中的相对路径和绝对路径**

相对路径：

. 表示当前目录

.. 表示上一级目录

资源名 表示当前目录

绝对路径：

<http://ip:port/工程名/>资源路径

1. web中/斜杠的不同意义

在web中/是一种绝对路径。

/斜杠如果被浏览器解析，得到的地址是：<http://ip:port/>

<a href=”/”>斜杠</a>

/斜杠如果被服务器解析，得到的地址是：<http://ip:port/>工程路径

* + 1. <**url-pattern**>/forwardC</**url-pattern**>
    2. servletContext.getRealPath(“/”)
    3. request.getRequestDispatcher(“/”)

特殊情况：response.sendRediect(“/”)把斜杠发送给浏览器解析得<http://ip:port/>

4.HTTPServletResponse类

a)HttpServletResponse类的作用

HttpServletResponse类和HttpServletRequest类一样。每次请求进来，Tomcat服务器都会创建一个Response对象传递给Servlet程序去使用， HttpServletRequest表示请求过来的信息，HttpServletResponse表示所有响应的信息，我们如果需要设置返回给客户端的信息，都可以通过HttpServletResponse对象来设置

b)两个输出流的说明

字节流 getOutputStream() 常用于下载（传递二进制数据）

字符流 getWriter() 常用于回传字符串（常用）

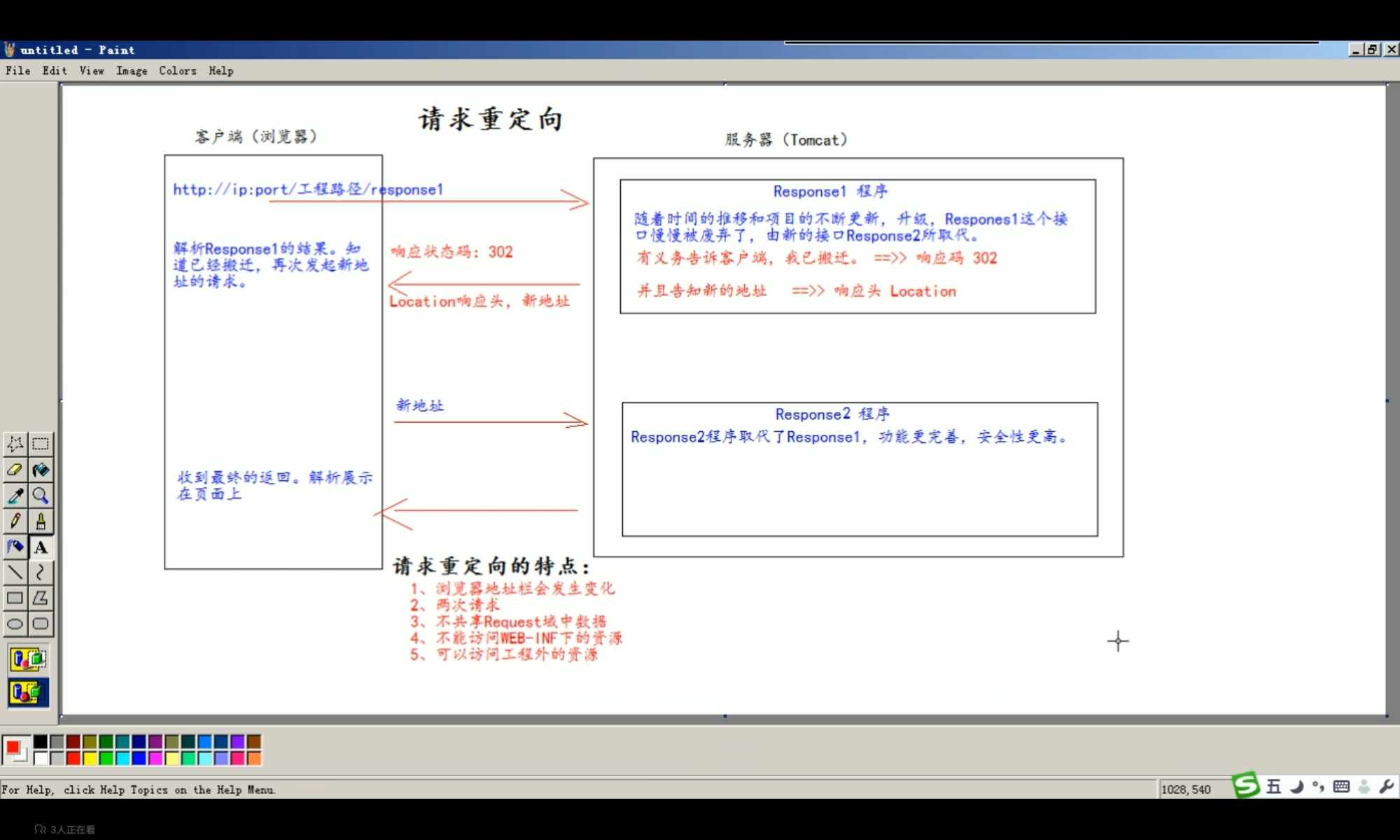
两个流同时只能使用一个

c)如何往客户端回传数据

见代码

d)请求重定向

请求重定向是指客户端给服务器发送请求，然后服务器告诉客户端说，我给你一些地址，你去新地址访问。叫做请求重定向（因为之前的地址可能已经被废弃）



请求重定向的第一种方法：

*//设置响应状态码302，表示重定向（已搬迁）*resp.setStatus(302);  
 *//设置响应头，说明新的地址在哪* resp.setHeader(**"location"**,**"http://localhost:8080/07\_servlet/response2"**);

请求重定向的第二种方法（推荐）

*//使用sendRedirect方法*resp.sendRedirect(**"http://localhost:8080/07\_servlet/response2"**);