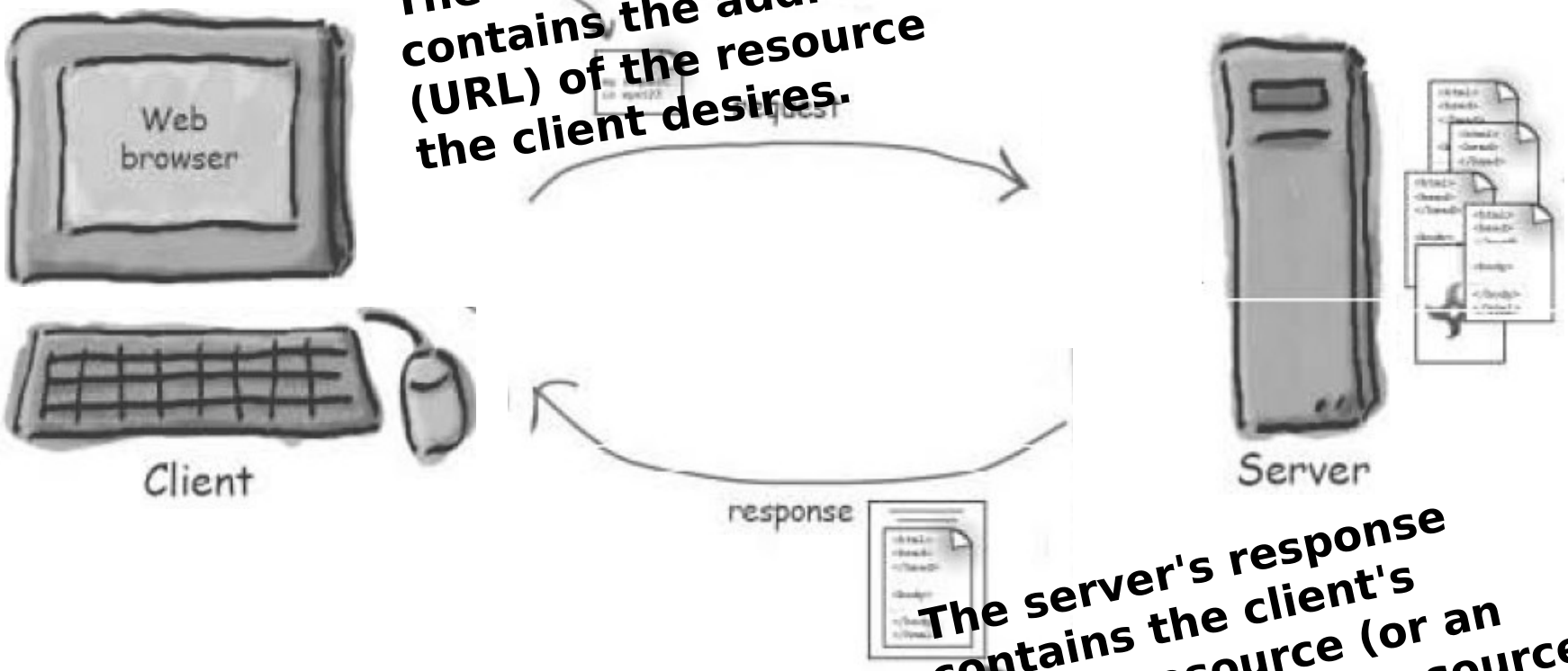
Object Oriented Programming II (JAVA ADVANCED)

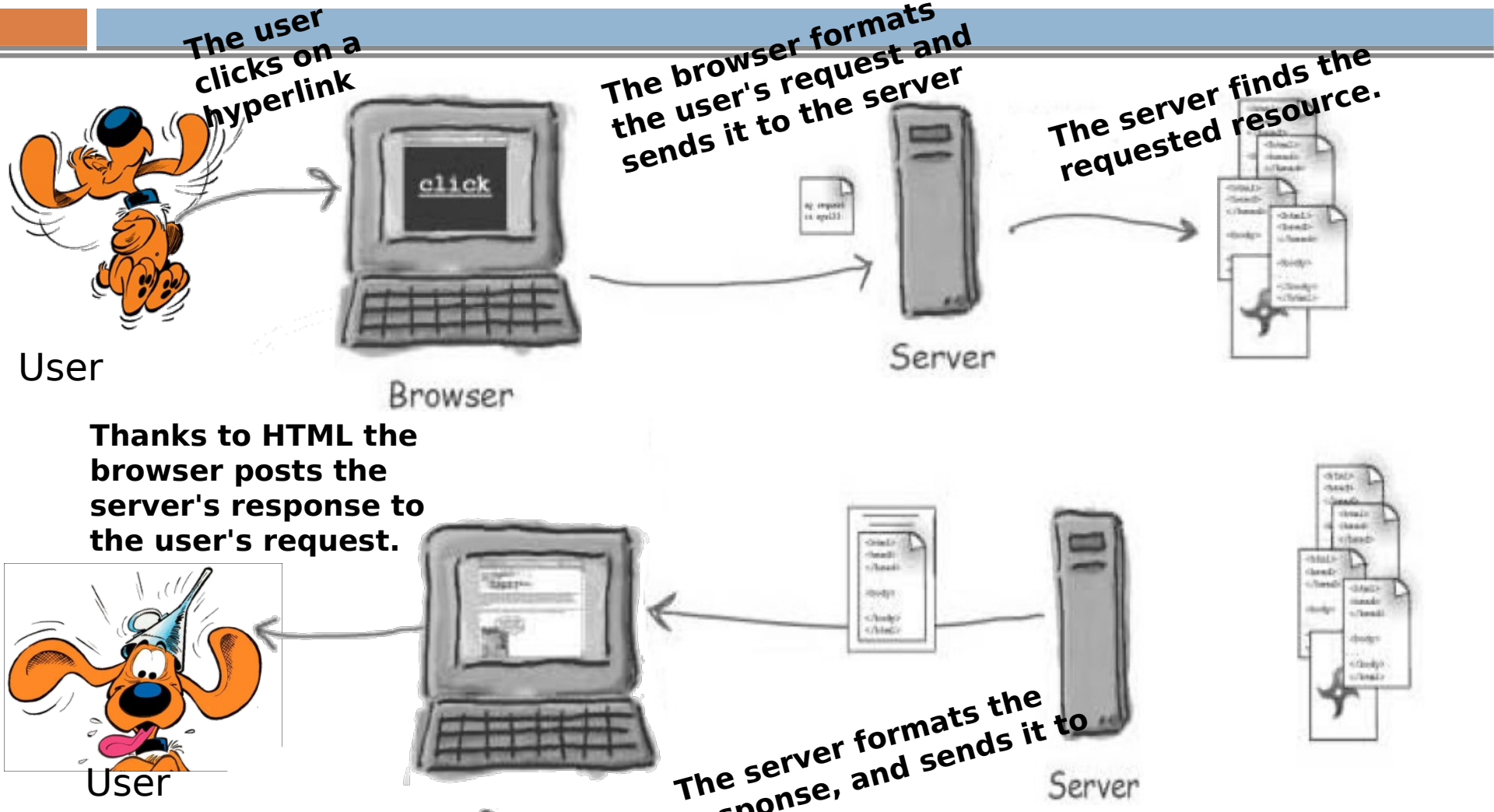
420-P33-SU

Hypertext Transfer Protocol (HTTP)

The Role of the Server



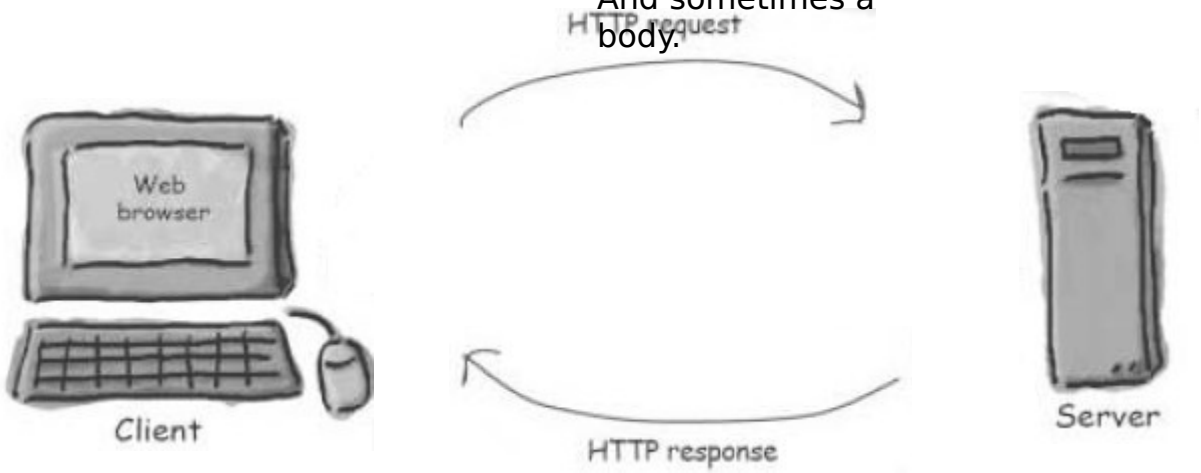
The Role of the Client



The Role of HTTP

Http requests are composed of:

- A request
- Headers
- And sometimes a body.



An Http response is composed of:

- A status
- Zero or more headers
- And generally a body.

Commands
GET
HEAD
POST
PUT
DELETE

Remarks:

*After the response is sent to the client, **the HTTP connection that was established between the server and the client is cut off**.*
HTTP is a stateless protocol, that is to say without the use of memory.

A new HTTP connection between the user and the server is only established if the client makes a new request to the server.

The GET Command

- GET is the most direct way to request a resource (web page, PDF document, image ...) without having an effect on that resource.
- GET is used by default for hyperlinks and forms.

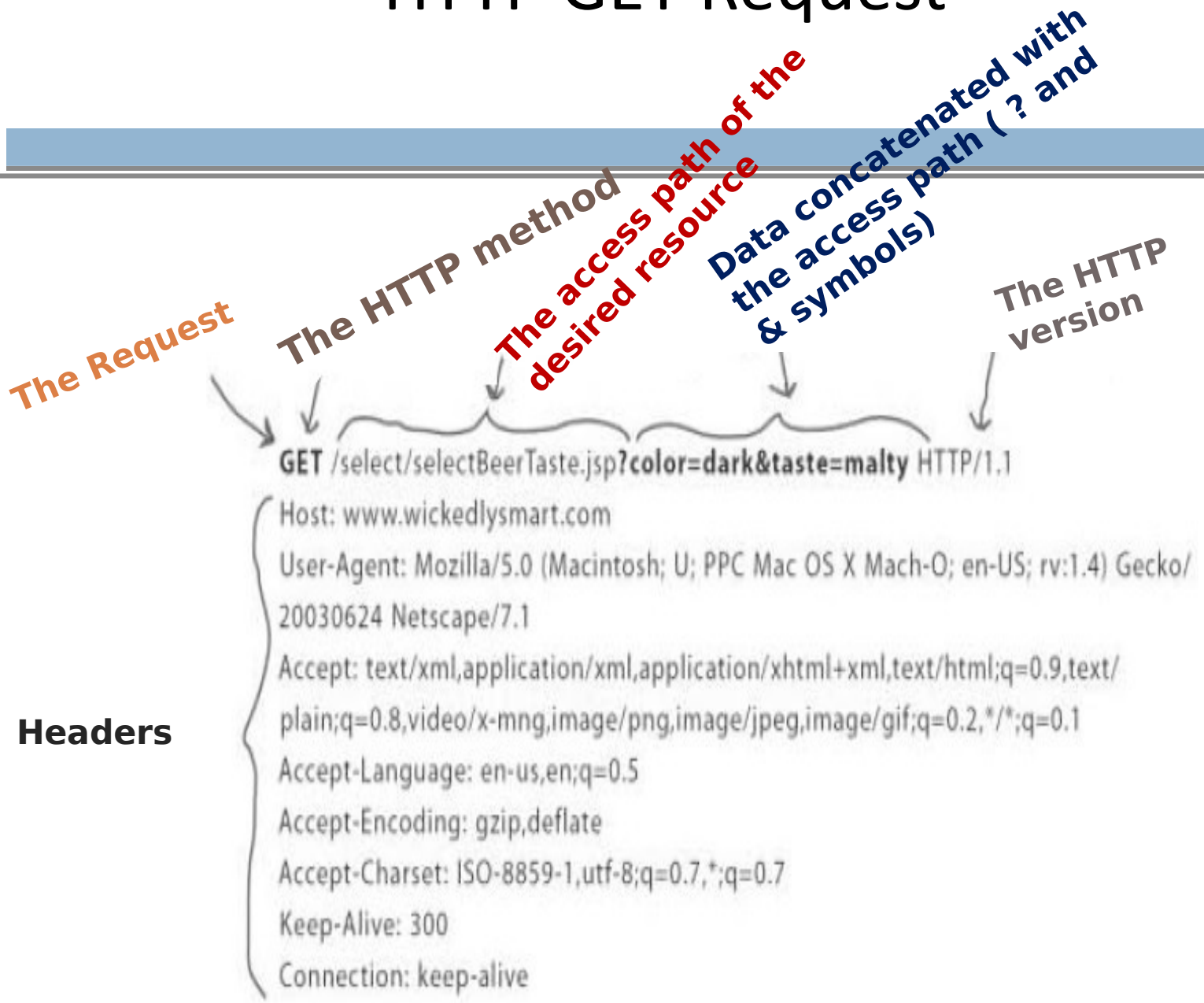
- The GET method can also send data to a web server.

Transfer data to server :

1. Limited size (is dependent on the web server used).
2. Data is concatenated with the URL and will appear on the address bar of the browser.

"In HTTP, a method is a Command that specifies a type of request, that is, it asks the server to perform an action. Generally, the action is for a resource identified by the URL that follows the name of the method." (Wikipedia)

HTTP GET Request



The POST Command

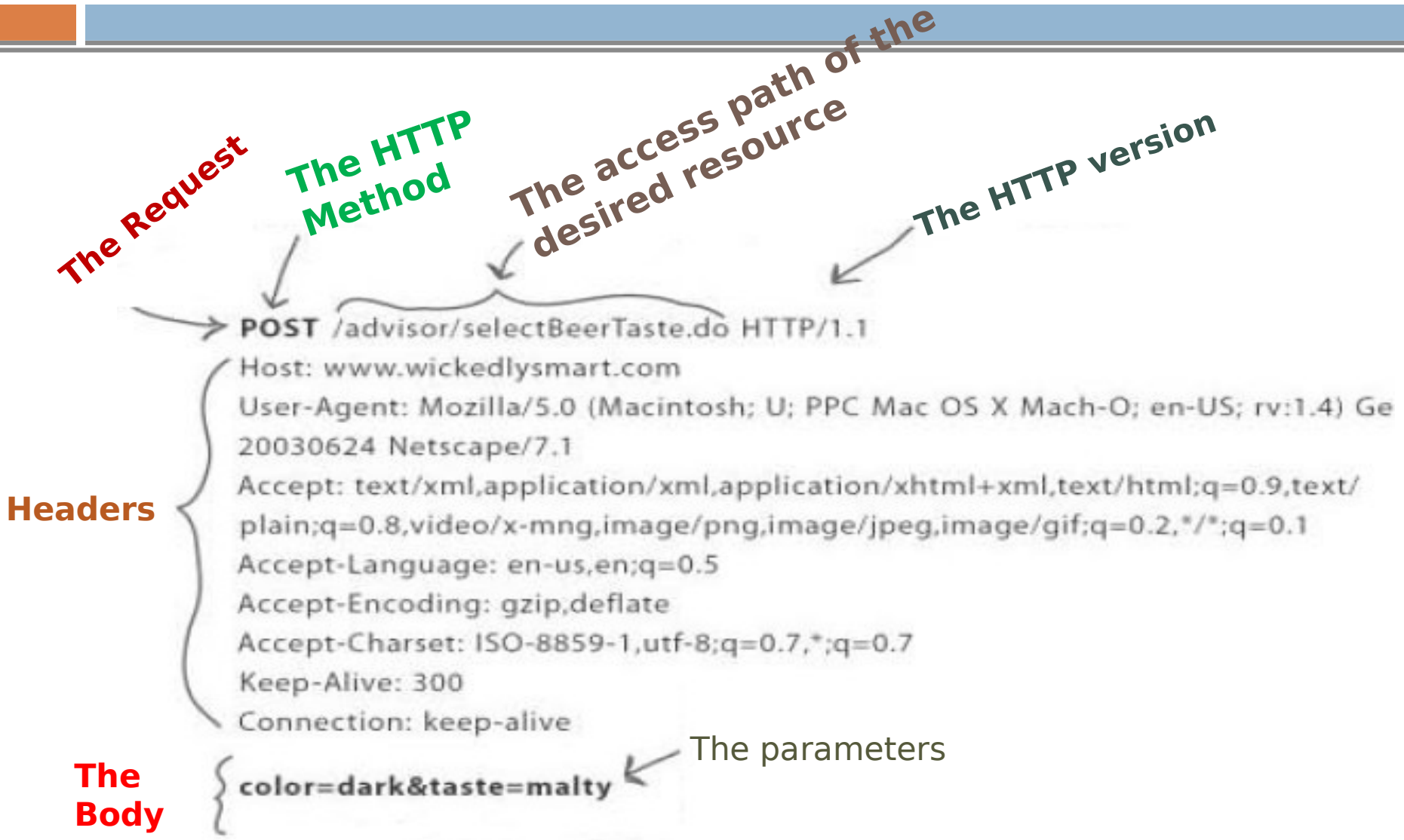
- The POST method is another HTTP command used to make requests to a server.

- The POST method, like GET, can also transfer data. In fact, POST is preferred for this purpose because:

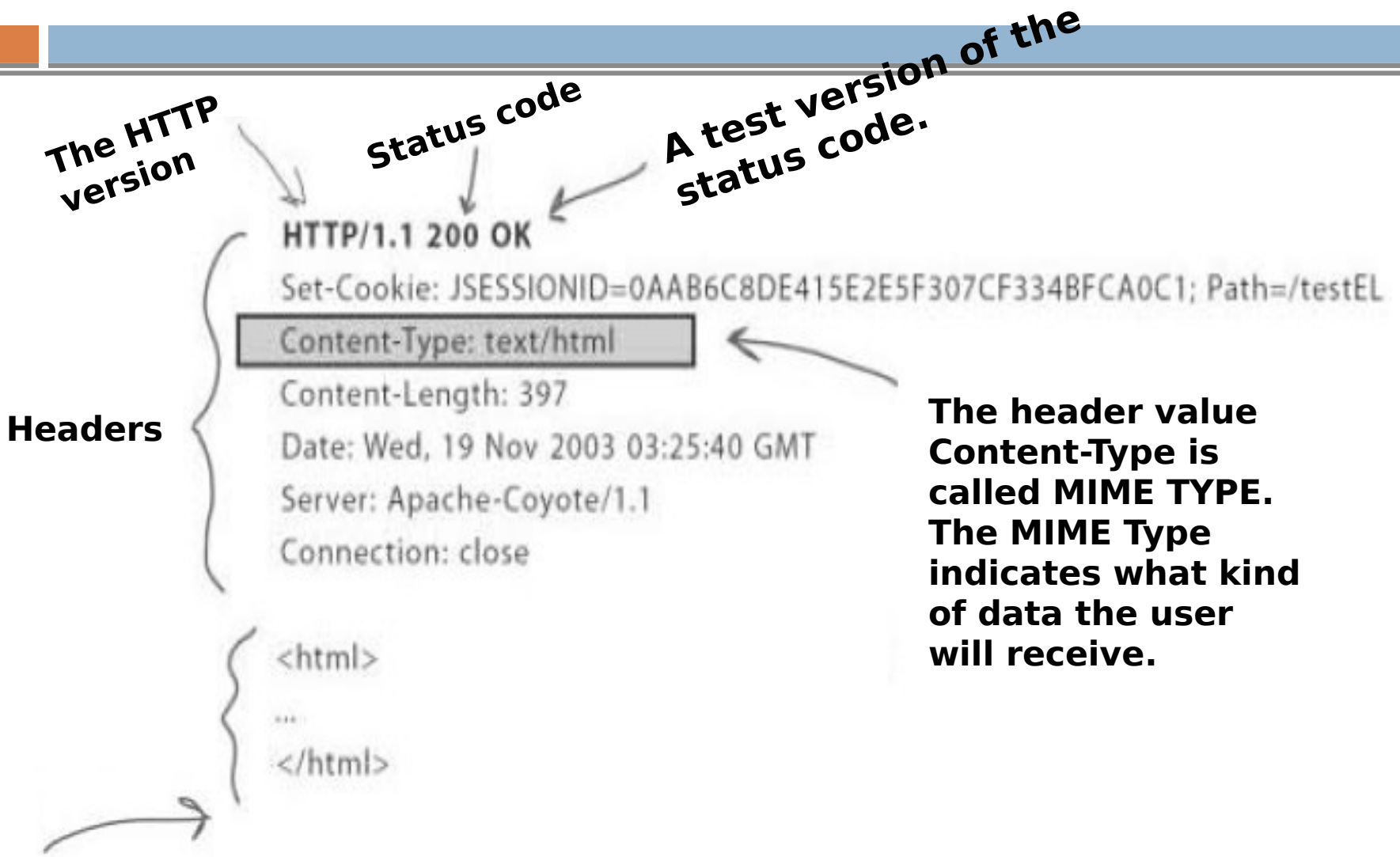
  - POST has no limit to the data transmitted.

  - POST is more secure, as the data is packaged in the body of the request (rather than in the address bar).

HTTP POST Request



HTTP Response



Summary of request / response cycle:

The user enters a URL in their browser.

The browser launches a get HTTP request.

The request is sent to the server.

The server finds the page.

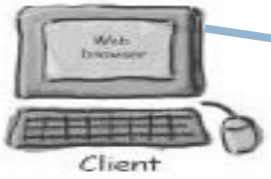
Beer1.html

```
<html><body>
<h1 align=center>Beer Login Page</h1>
<form>
  Select a beer type or buy beer
  making supplies?<p>
  <input type=radio name=select
    value=Select> Select a beer<br>
  <input type=radio name=select
    value=Buy> Buy supplies<br><br>
  <center>
    <input type=SUBMIT>
  </center>
</form>
</body></html>
```

And generates a response.

```
HTTP/1.1 200 OK
Set-Cookie: . . .
. . .
<html><body>
<h1 align=center>Beer Login Page</h1>
<form>
  Select a beer type or buy beer . . .
```

The response is sent to the client, and the connection is closed.



The browser displays the page

The user is ready to order their beer!