

HTTP

HTTP

A Brain-Friendly Guide

2nd  
Edition  
New Mock Exam  
Included

# Head First Servlets & JSP™

Passing the Sun Certified Web Component Developer Exam

Test yourself  
with more than  
200 realistic  
exam questions



**Watch it!**

Avoid deadly  
traps & gotchas  
on the 1.5 exam



Learn how Ted improved his  
appeal with dynamic attributes



Updated to cover  
the latest version of  
the SCWCD exam  
for J2EE 1.5

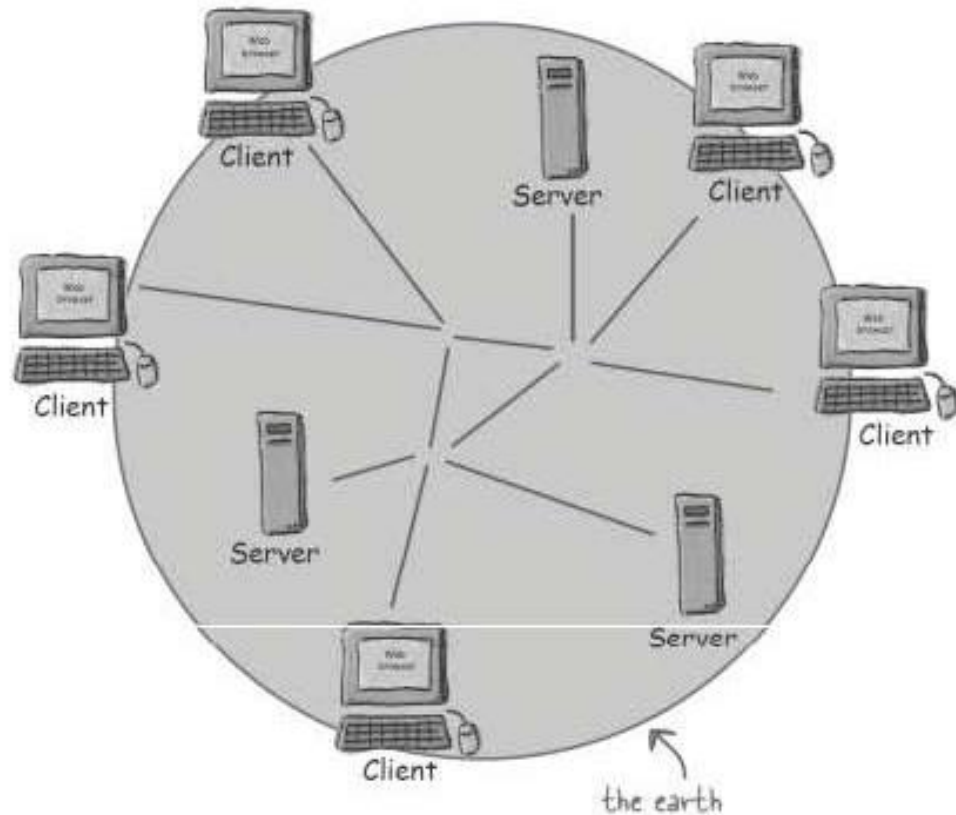


Use `c:out` to get your  
message to the world

Fool around  
in the Custom  
Tag Library

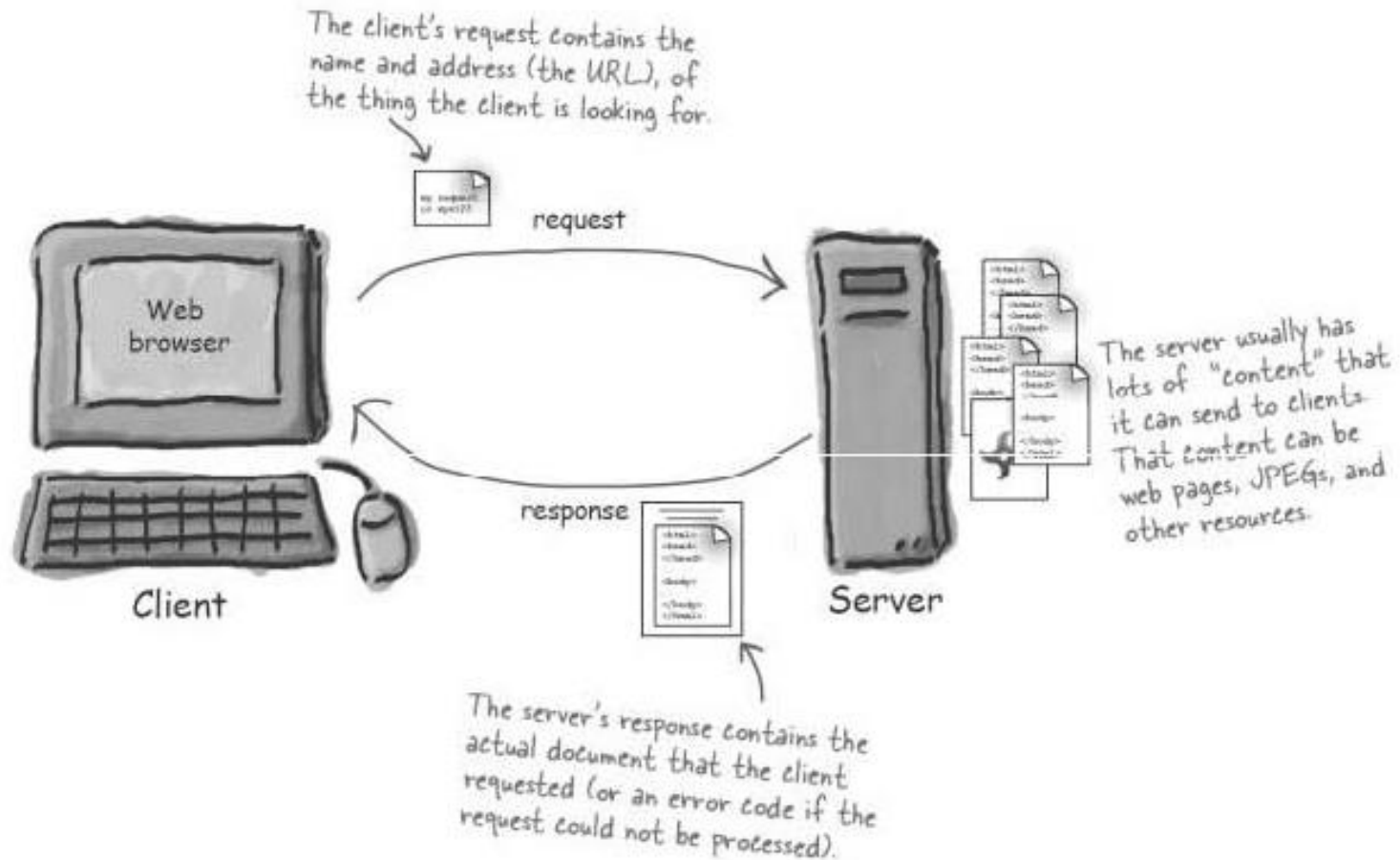


# La web



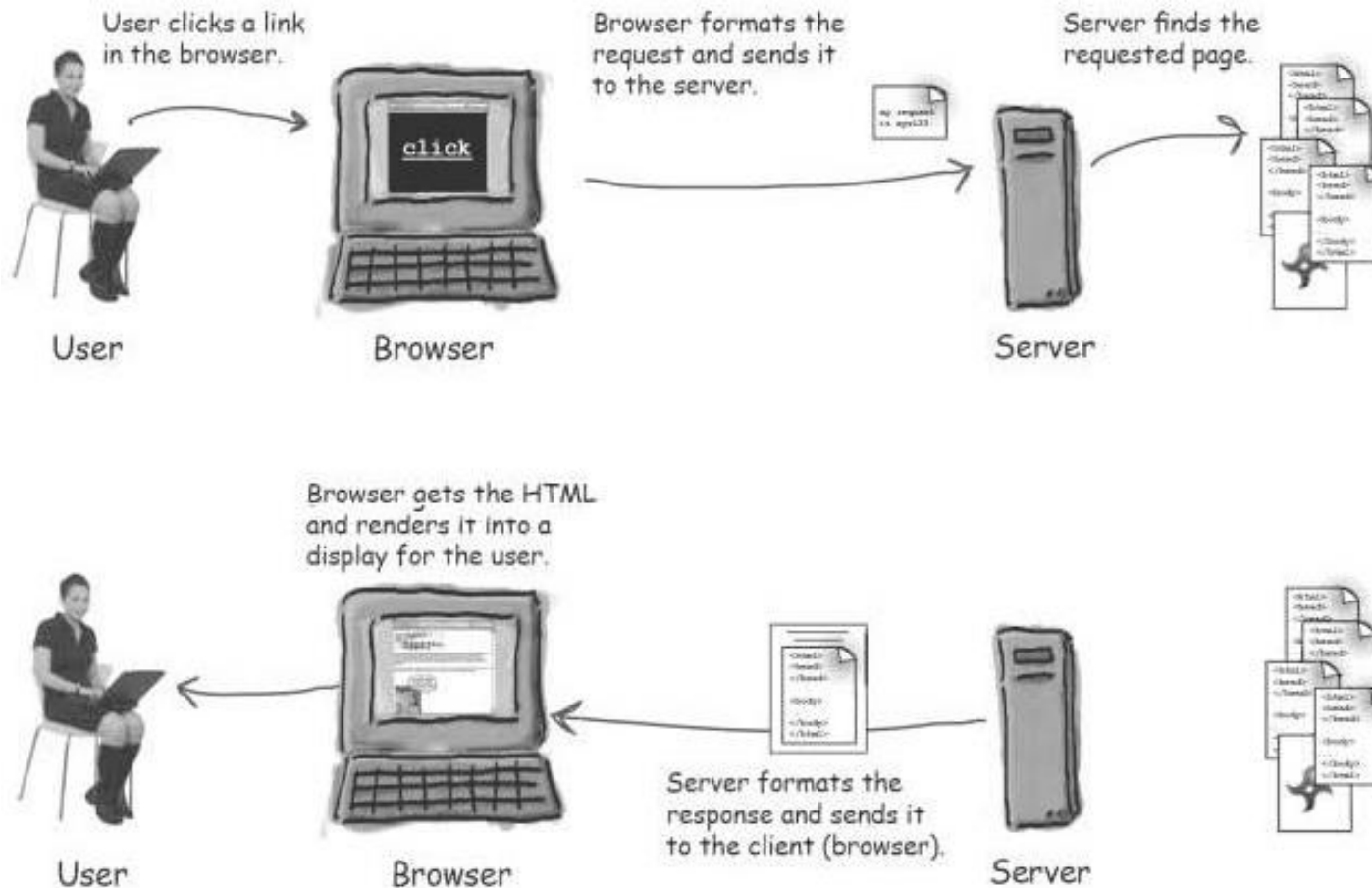
The web consists of gazillions of clients (using browsers like Mozilla or Safari) and servers (using web server apps like Apache) connected through wires and wireless networks. Our goal is to build a web application that clients around the globe can access. And to become obscenely rich.

# ¿Qué hace un servidor web?



Un servidor web toma una solicitud de un cliente y da algo de regreso al cliente

# ¿Qué hace un cliente web?



Un cliente web le permite a un usuario solicitar algo al servidor y muestra el usuario el resultado de la solicitud

# HTML y HTTP

- Los clientes y servidores conocen HTML y HTTP

# HTML

- Cuando un servidor responde una solicitud, el servidor usualmente envía algún tipo de contenido al navegador de tal forma que el navegador pueda mostrarlo

# HTML

- Los servidores envían normalmente al navegador un conjunto de instrucciones escritas en HTML (HyperText Markup Language)



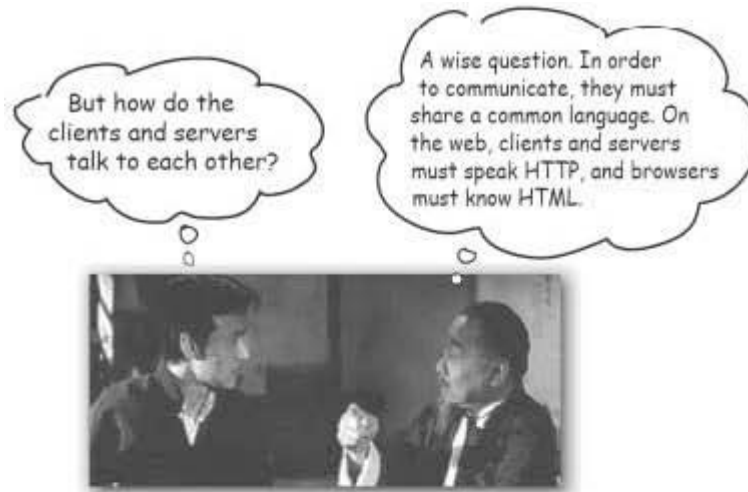
# HTML

- Los servidores envían normalmente al navegador un conjunto de instrucciones escritas en HTML (HyperText Markup Language)
  - EL HTML le dice al navegador como presentar el contenido al usuario

# HTTP

- La mayoría de las conversaciones en la web entre los clientes y los servidores son hechas utilizando el protocolo HTTP (HyperText Transfer Protocol) el cual permite conversaciones simples de solicitud-respuesta
  - Cuando el servidor web envía al cliente una página HTML, la envía utilizando HTTP

# En resumen hasta este momento ...



*HTML tells the browser how to display the content to the user.*

*HTTP is the protocol clients and servers use on the web to communicate.*

*The server uses HTTP to send HTML to the client.*

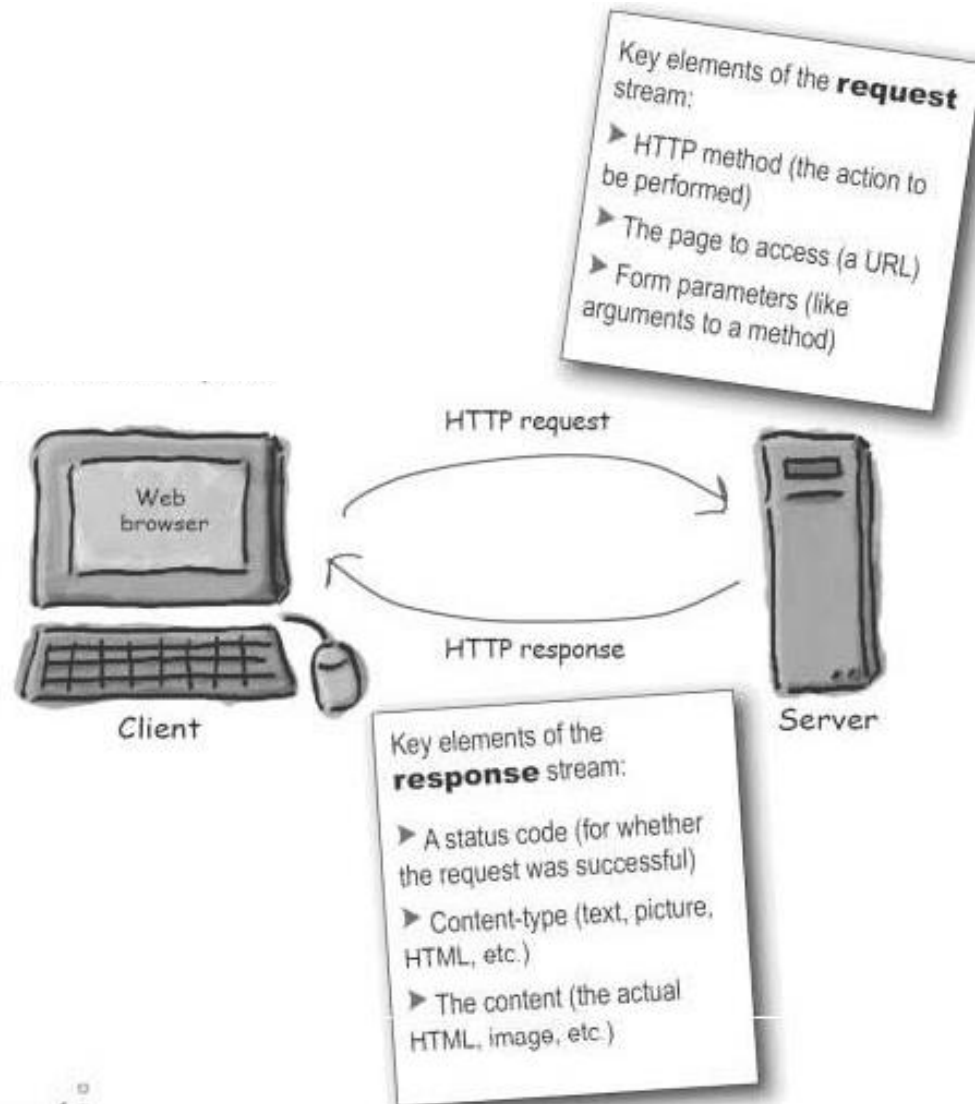
# Más del protocolo HTTP

- El protocolo HTTP corre encima de TCP/IP
- TCP es el responsable de asegurarse que un archivo enviado de un nodo de red llega al nodo destino como un archivo completo, aun aunque el archivo sea dividido en fragmentos (paquetes) cuando es enviado
- IP es el protocolo básico que mueve/enruta los paquetes de un host a otro en su camino al destino

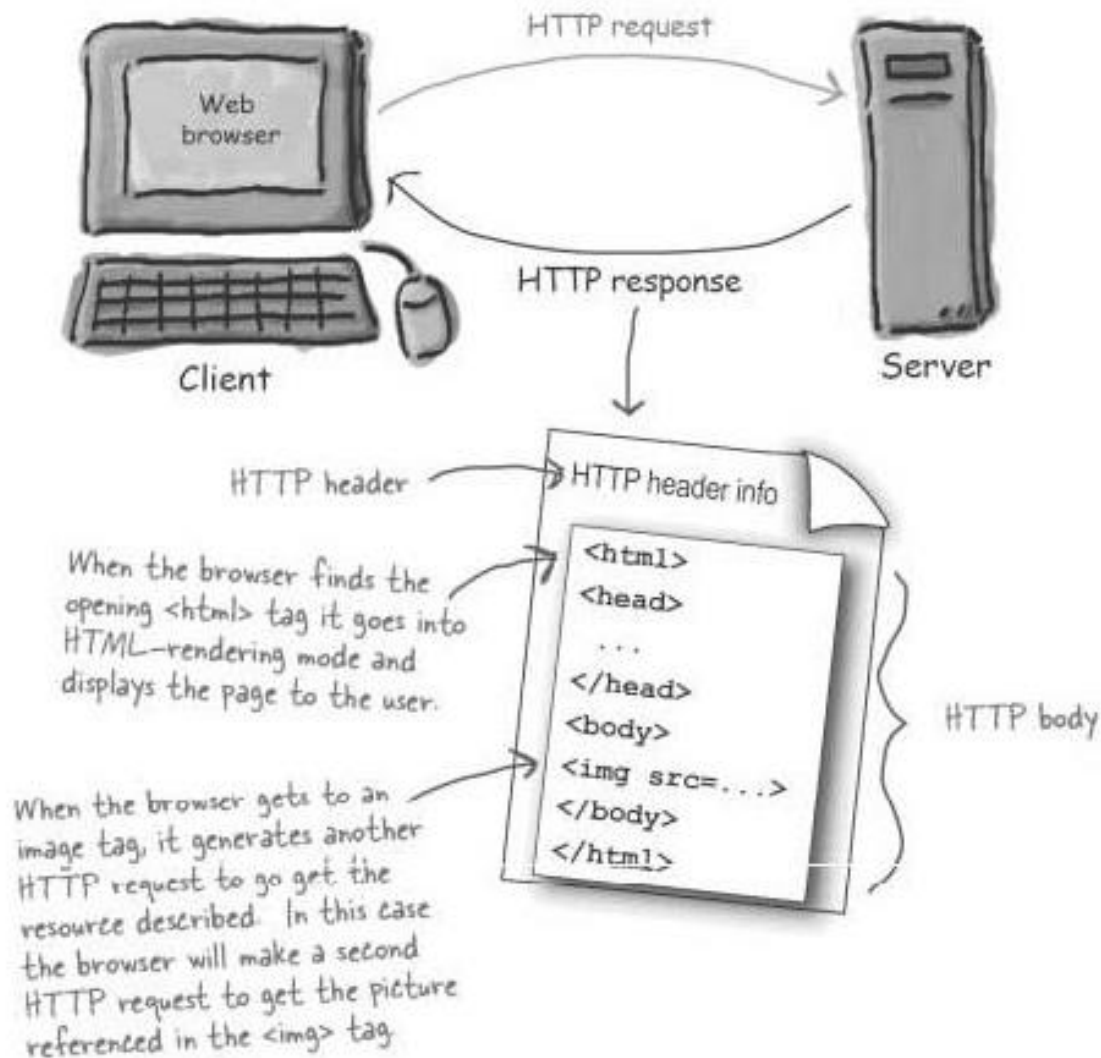
# Más del protocolo HTTP

- El protocolo HTTP, es entonces otro protocolo de red el que tiene características específicas de la web, pero depende de TCP/IP para completar la solicitud y respuesta de un lugar a otro
- La estructura de una conversación HTTP es una secuencia simple **Solicitud/Respuesta**
  - Un navegador solicita y un servidor responde

# Más del protocolo HTTP



# HTML es parte de la respuesta HTTP



# Solicitud

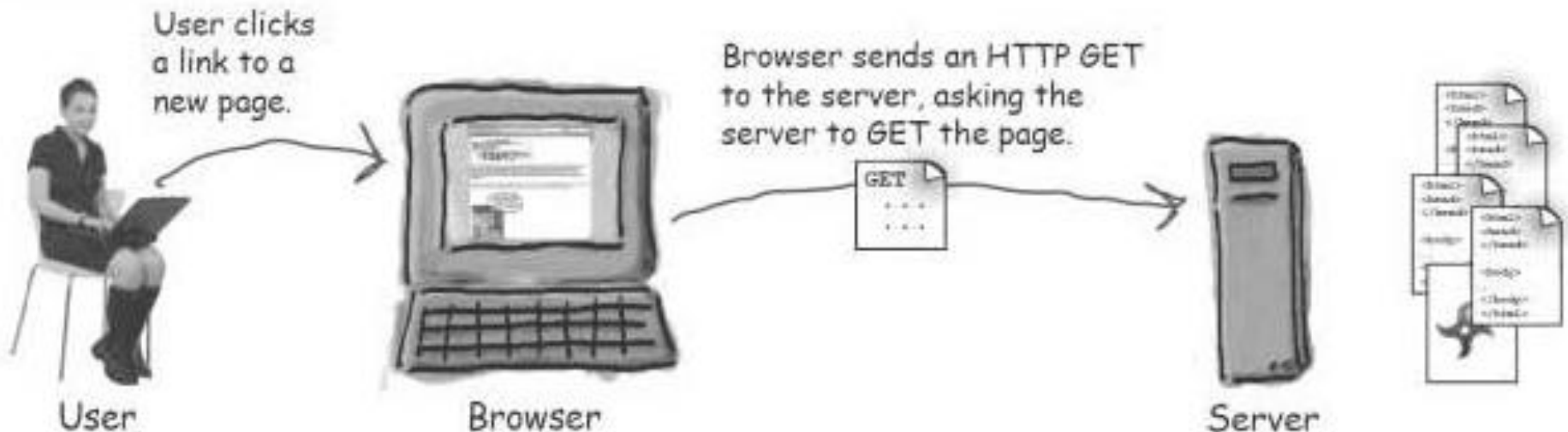
- La primera cosa que se encontrará en la solicitud es el nombre del método HTTP
- El nombre del método le indica al servidor la clase de solicitud que se está haciendo, y como se formateará el resto del mensaje
- El protocolo HTTP tiene varios métodos, pero los que normalmente se utilizan son los métodos GET y POST



# Método GET

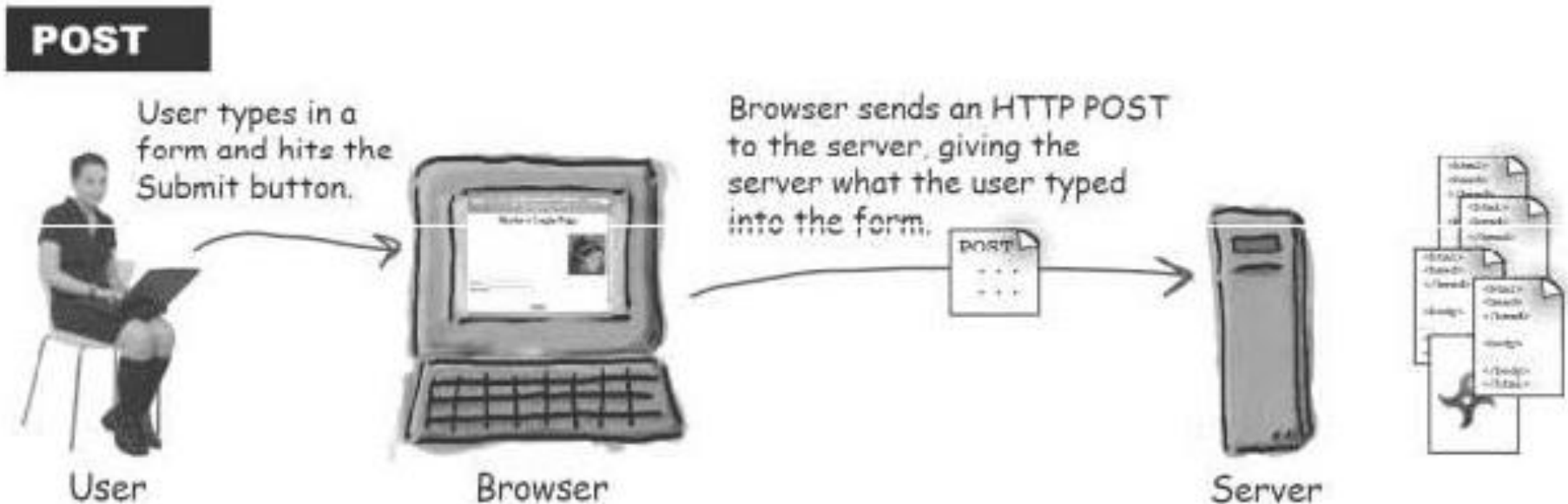
- El método GET es el más simple método HTTP
  - Su principal función es pedirle al servidor que obtenga un recurso y lo envíe de regreso
    - Este recurso puede ser una página HTML, un archivo JPEG, PDF, etc

## GET



# POST

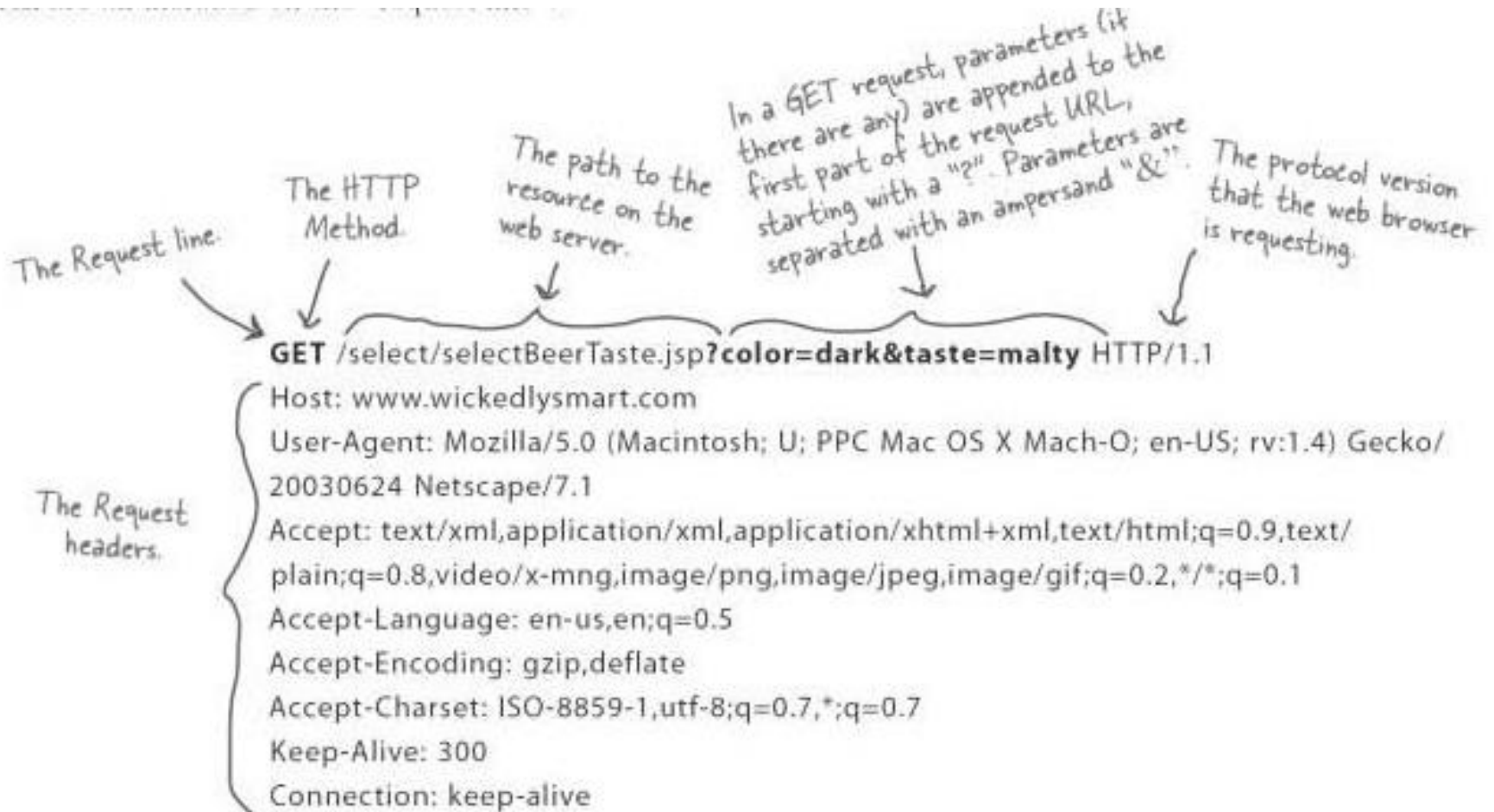
- El método POST es una solicitud más poderosa
  - Con éste método se puede solicitar algo y al mismo tiempo enviar datos de un formulario al servidor
    - Nota: Si los datos son limitados se puede utilizar GET. Ver siguiente lámina



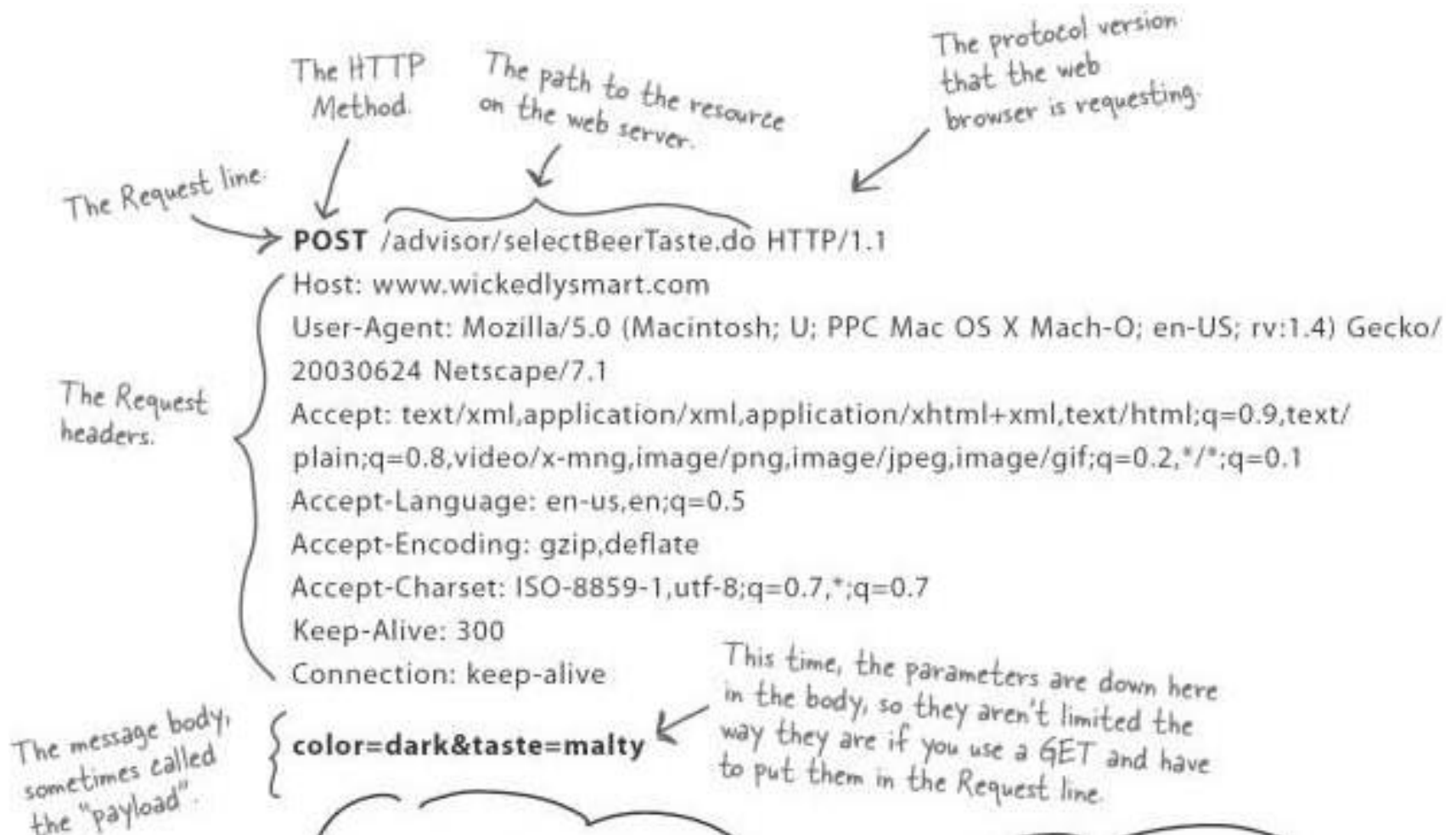
# Uso de POST vs GET

- La cantidad total de caracteres en un método GET es muy limitado (dependiendo del servidor)
- El dato enviado con el método GET es insertado al final del URL en la barra del navegador, de tal forma que los datos son mostrados
  - Esto no puede ser adecuado en ciertos casos, como el uso de password u otros datos sensibles

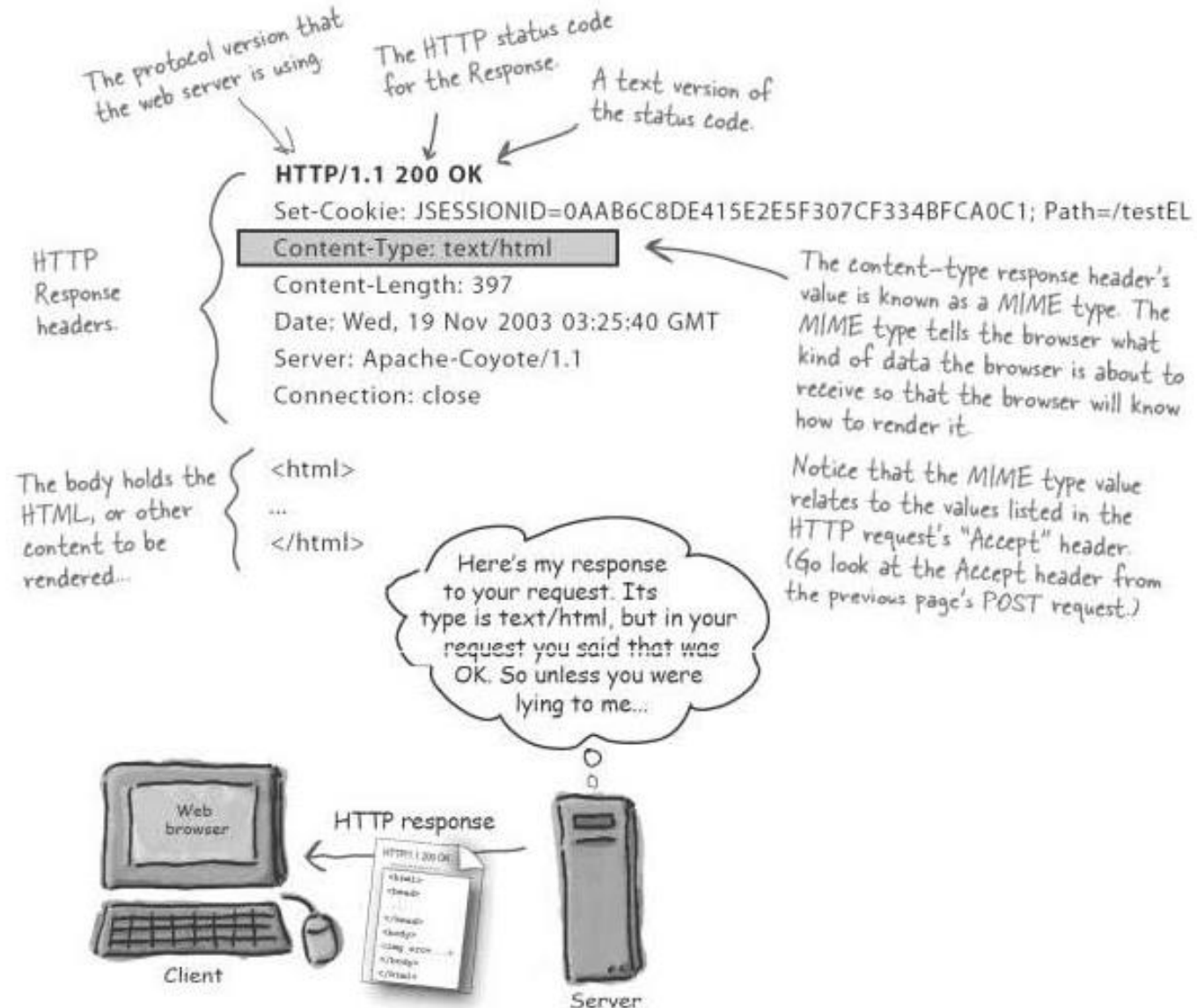
# Anatomía de una solicitud HTTP GET



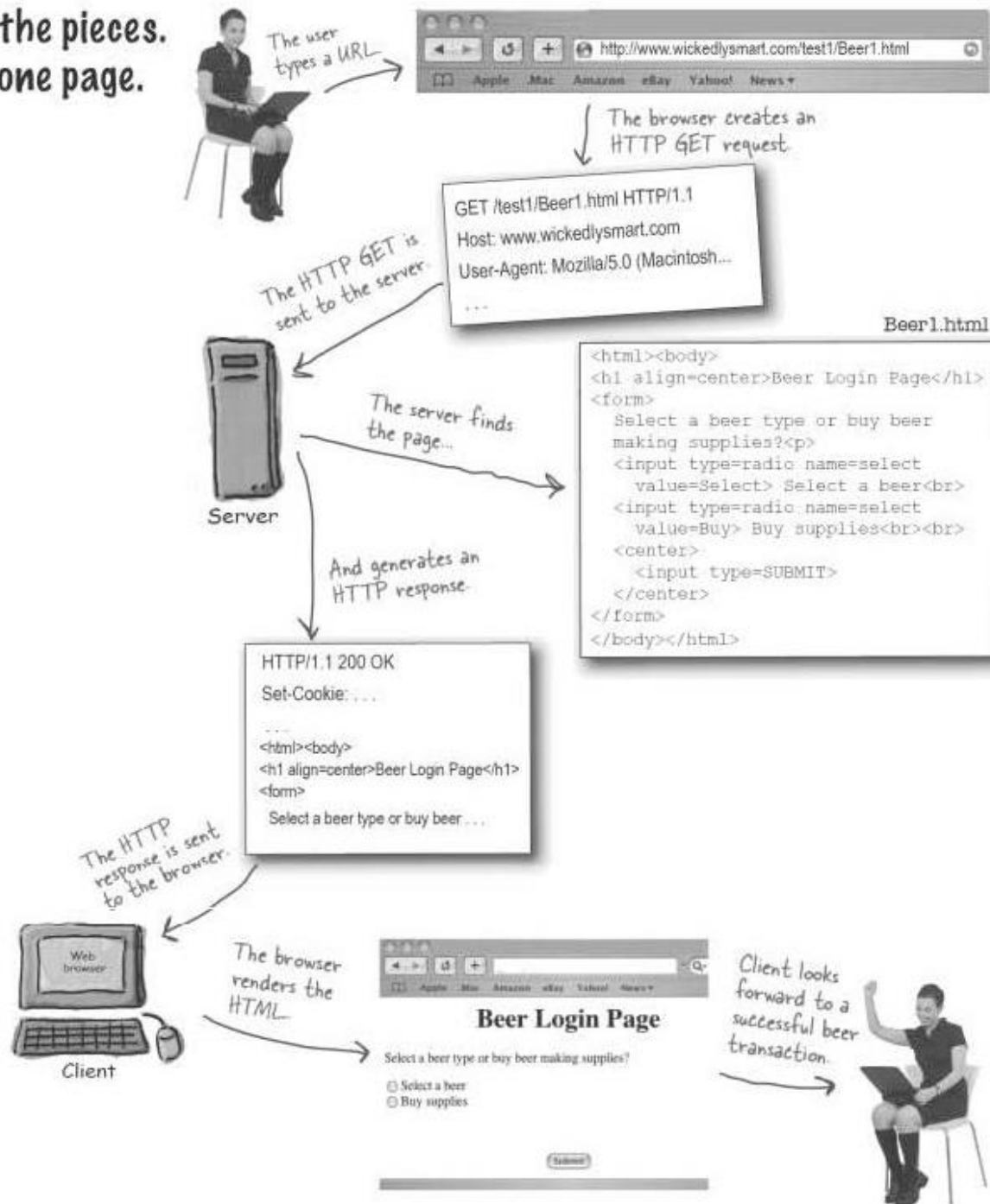
# Anatomía de una solicitud HTTP POST



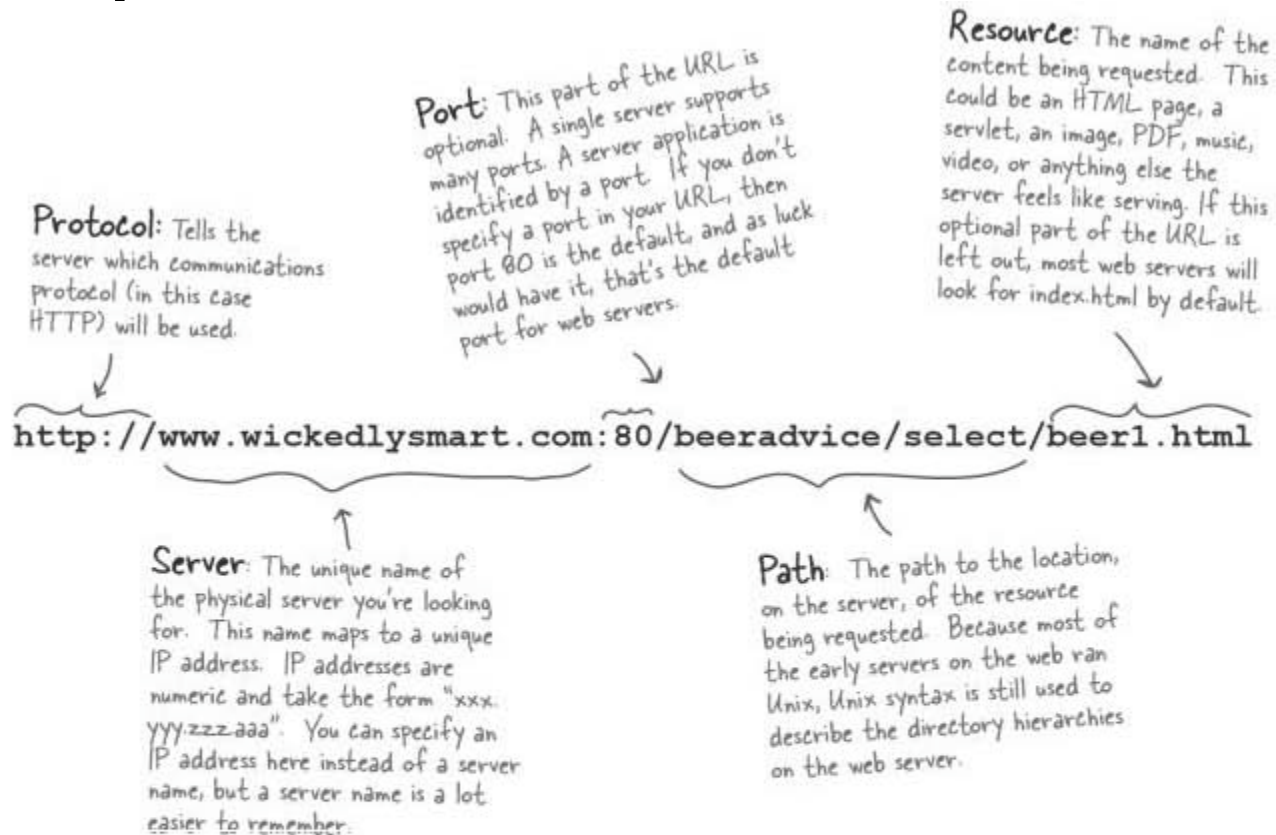
# Anatomía de una respuesta HTTP



All the pieces.  
On one page.



# URL (Uniform Resource Locator)



Cualquier recurso en la web tiene una dirección única, en el formato URL

Not shown:

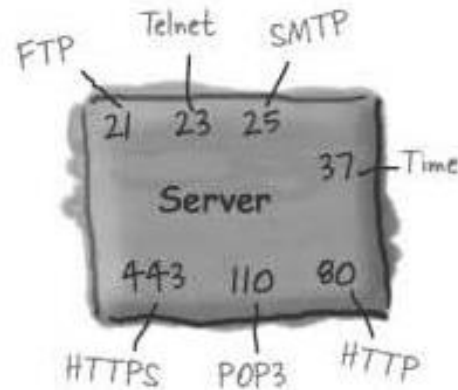
**Optional Query String:**  
Remember, if this was a GET request, the extra info (parameters) would be appended to the end of this URL, starting with a question mark "?", and with each parameter (name/value pair) separated by an ampersand "&".



# Puerto TCP

- Un puerto TCP es un número de 16 bits que identifica un programa de software en el hardware del servidor

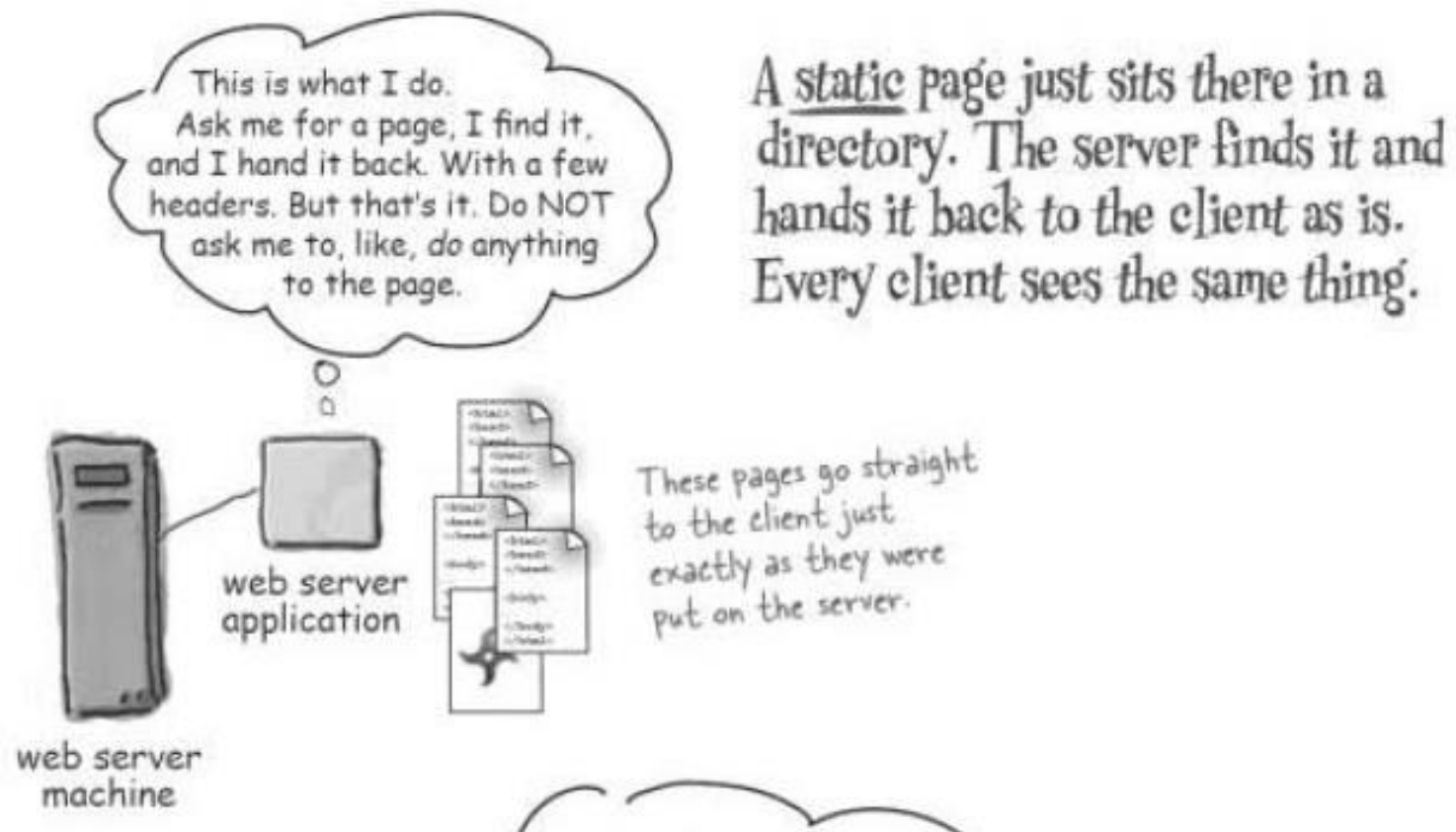
Well-known TCP port numbers  
for common server applications



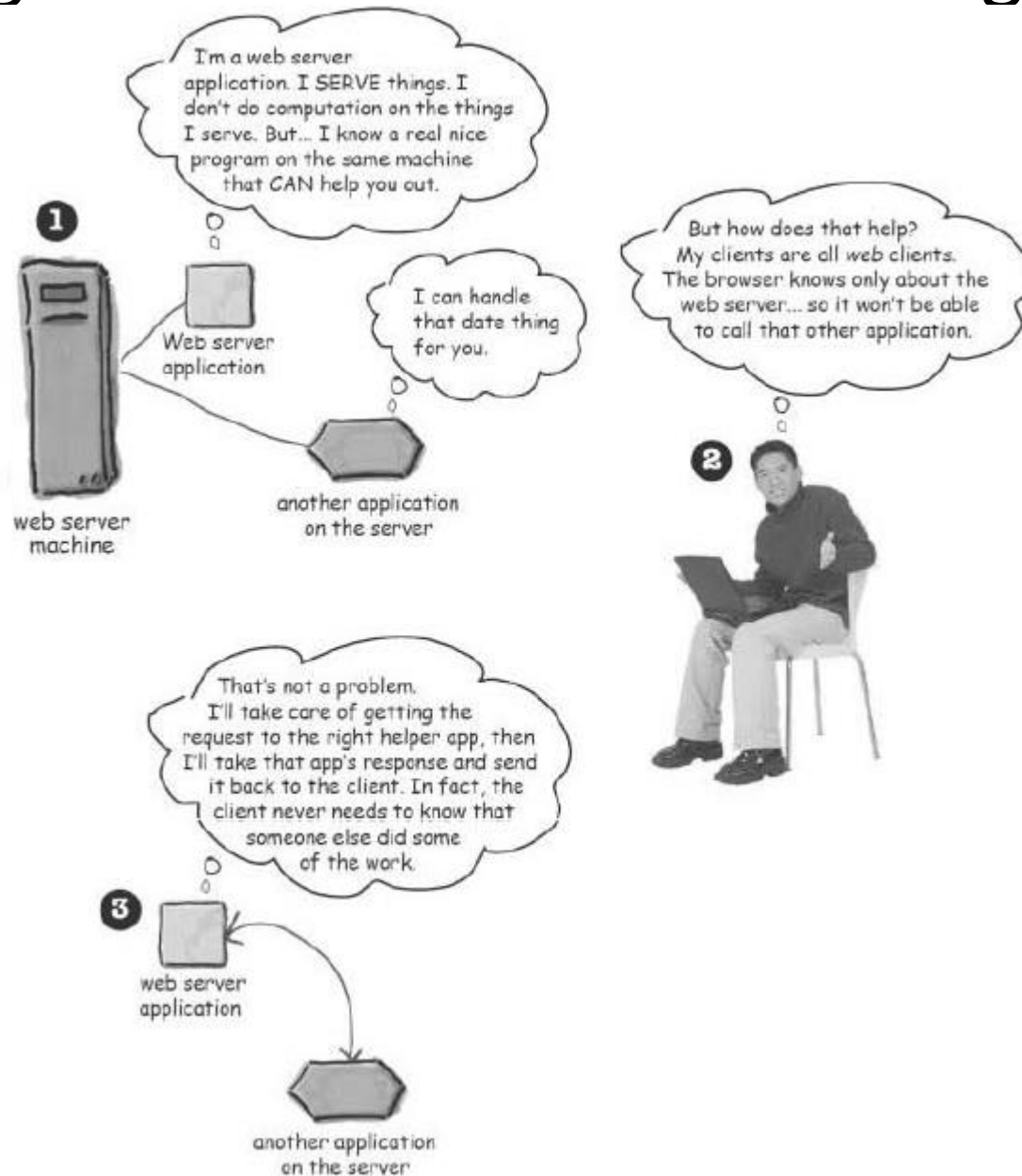
Using one server app per port, a server  
can have up to 65536 different server  
apps running.

The TCP port numbers  
from 0 to 1023 are  
reserved for well-known  
services (including the  
Big One we care about—  
port 80). Don't use these  
ports for your own  
custom server programs!

# Los servidores web adoran servir páginas web estáticas



# Pero algunas veces es necesario algo más...



# CGI

- CGI es el acrónimo de 'Common Gateway Interface'
- La mayoría de los programas CGI son escritos en scripts Perl, pero algunos otros lenguajes pueden trabajar, por ejemplo, C, Python y PHP
- En la siguiente lámina se muestra como un programa CGI trabaja para una página web dinámica

