Java Servlets

CSCI 201
Principles of Software Development

Jeffrey Miller, Ph.D.
jeffrey.miller@usc.edu

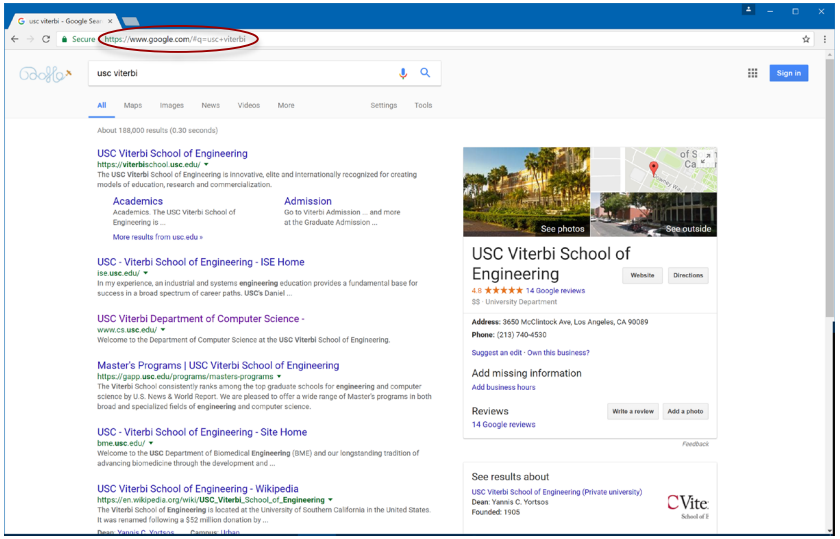
USC Viterbi
School of Engineering

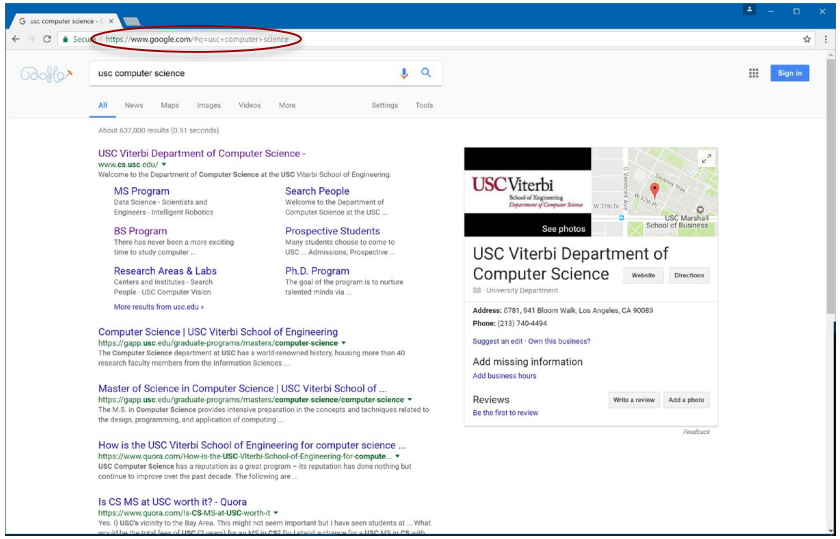
Outline

- Java Servlets
- Program

Dynamic Web Content

- Dynamic web content allows different data to be shown on the same web page
  - For example, the Google search result page is the same page that loads different data based on the search terms





Front-End vs Back-End

Browser

Request web page →

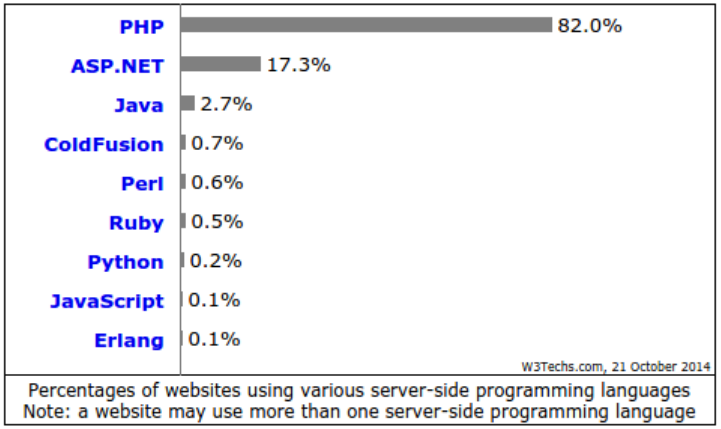
← Return HTML, CSS, JavaScript

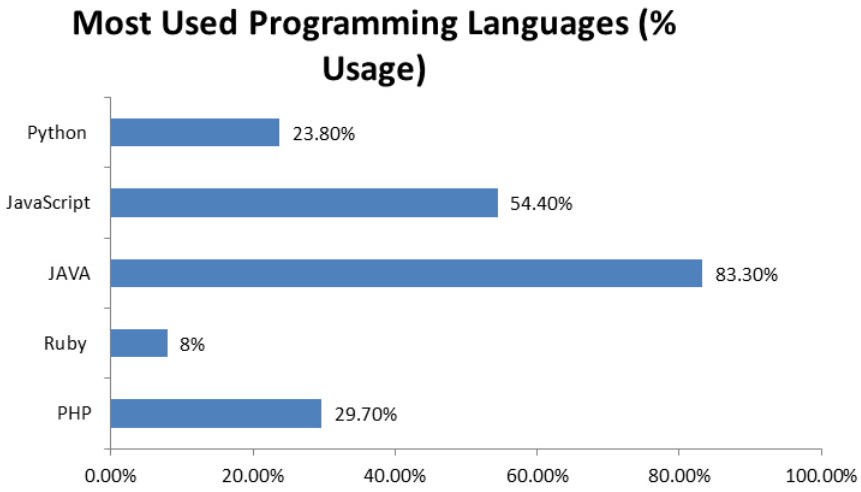
Web Server

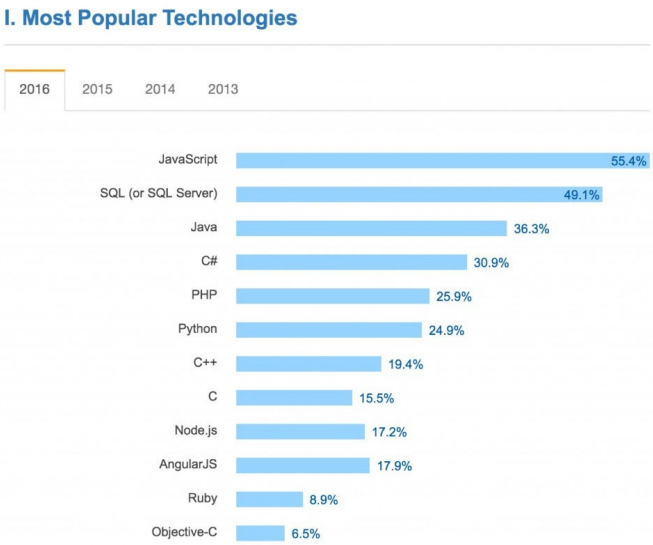
Web Languages

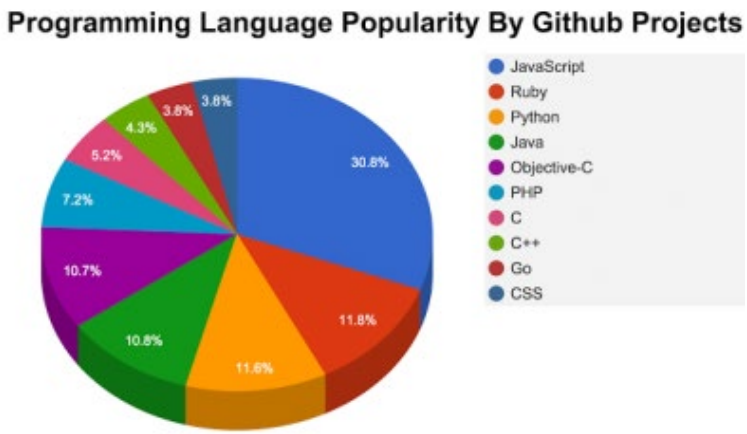
- Most dynamic content is based on a server program running and generating the front-end code in real-time
- There are many back-end web languages
  - Java – Servlets, JSPs, JavaBeans, Enterprise Java Beans (EJBs)
  - .NET – C#, VB
  - PHP
  - Ruby
  - Python
  - Server-side JavaScript – node.js
  - CGI – C, Perl
- There are three front-end web languages
  - HTML
  - CSS
  - JavaScript
  - (Browser plug-ins could be used on the front-end as well)

Which Back-End Language to Choose?









Which Back-End Language to Choose?

Web Developer Jobs, Employm

web developer | los angeles, ca

job title, keywords or company | city, state, or zip

New! Join Indeed Prime - Get offers from great tech companies

Page 1 of 512 jobs

Show: all jobs - 192 new jobs

Web Developer
NHN Global ★★★☆☆ 2 reviews
Los Angeles, CA 90010

Lead / Participate in web site or web application projects. We are seeking a Web Developer who will be responsible for full-stack coding for our eLAMBS team....

Desired Experience: Linux, Microsoft SQL Server, JSON, SQL, MySQL, PostgreSQL, Web Development, Spring, Mesos, Oracle, Java, Hibernate, Angular, Git, Docker, REST

Easily apply

Sponsored save job

Business Analyst Developer
Arcsona
Los Angeles, CA

Web:. Possess excellent English written and verbal communication skills....

Desired Experience: Java, JavaScript, Salesforce CRM, Oracle, CSS, Python, HTML5

Easily apply

Sponsored save job

Web Developer Jobs, Employm

Backend PHP Developer
LAShowroom.com
Los Angeles, CA 90079

Back End Developer*. We are looking to expand our development team with passionate back-end developers that love building scalable, maintainable, and optimized...

Desired Experience: Ubuntu, HTML5, MySQL, CentOS, NGINX, Software Development, PostgreSQL, CSS, OOP, PHP, SVN, JavaScript, SDLC, Ansible, Git, Docker, Redis, NoSQL, PCI, Unit Testing

Easily apply

Sponsored save job

Full Stack Java Web Developer
NHN Global ★★★☆☆ 2 reviews
Los Angeles, CA 90010

Solid understanding of web technologies:. 5+ years web development experience with Java Spring/Spring Boot. Design, implementation and maintenance of current...

Desired Experience: Spring, Linux, Tomcat, HTML5, Apache, Java, Hibernate, Maven, Angular, Git, CSS, Web Development

Easily apply

Sponsored save job

FileMaker Developer - new
California Medical Evaluators ★★★☆☆ 11 reviews
Los Angeles, CA 90025

Ideal candidate would have some experience with web fundamentals including:. URLs to any publicly accessible web applications you have worked on....

Desired Experience: JavaScript, Database Administration, XML, Go, JSON, HTML5, Excel, SSL, Accounting, FileMaker, CSS, PHP, AWS

Easily apply

Sponsored save job

Web Developer Jobs, Employm

Unreal Engine 4 Developer - new
Inhance Digital ★★★☆☆ 12 reviews
Los Angeles, CA

Unreal Engine 4 developer:. Are an Unreal Engine 4 developer. Experience with Unity, web front-end and JavaScript, knowledge of iOS or Android, Interactive...

Desired Experience: JavaScript, iOS, AI, Unreal Engine, Unity, C/C++

Easily apply

Sponsored save job

Wordpress/Woocommerce Developer - new
ganjarunner
Los Angeles, CA
$20 - $50 an hour

Licensed Cannabis Company is looking for a freelance web developer to help update our website, implement new features and optimize our user experience....

Desired Experience: JavaScript, Bootstrap, Front-End Development, Adobe Photoshop, Adobe CS, WordPress, CSS, HTML5, PHP

Easily apply

2 days ago save job more...

Front End Web Developer - new
Lunada Biomedical ★★★☆☆ 8 reviews
Los Angeles, CA 90045

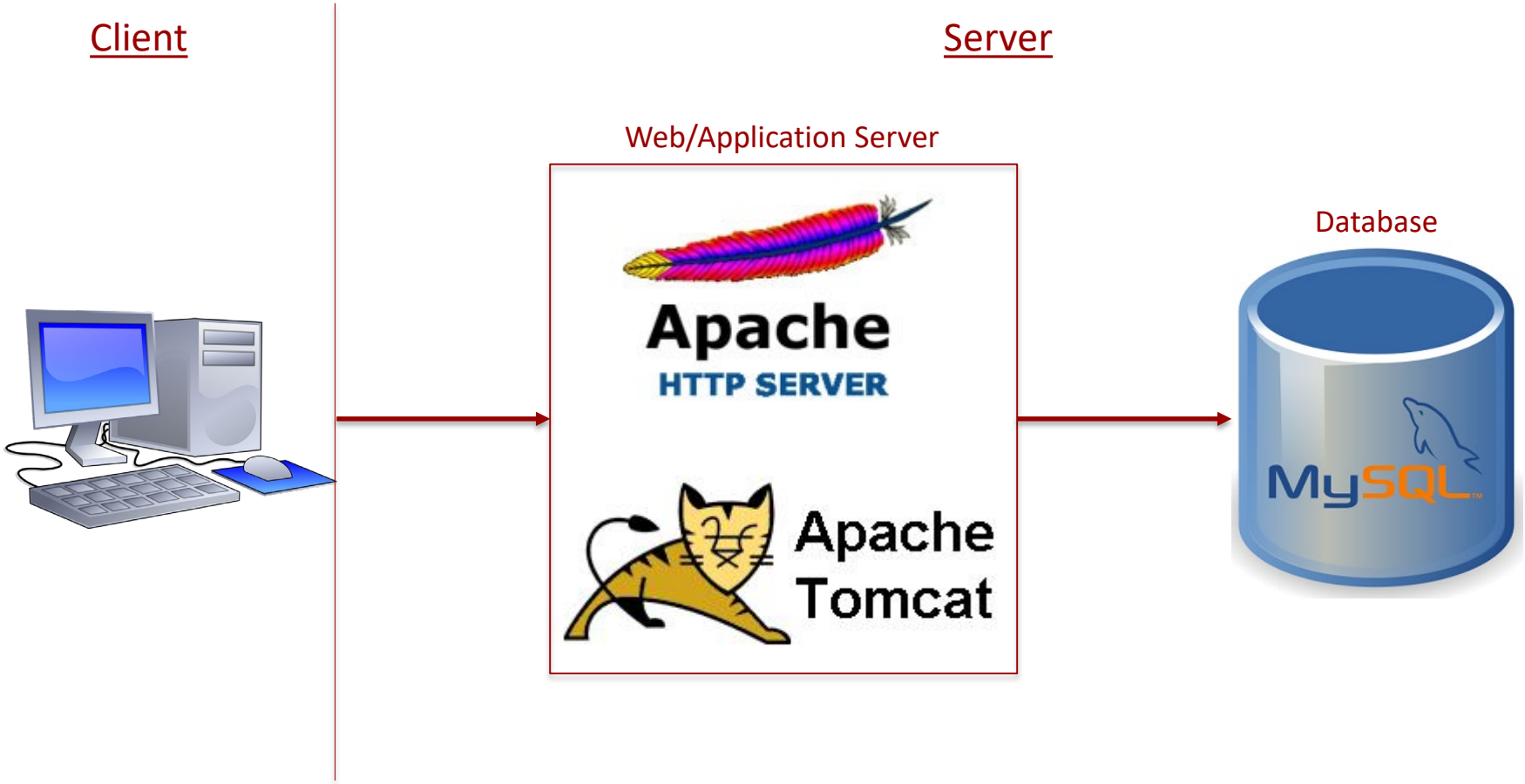
We are looking for an experienced Front End Web Developer to join our IT & Marketing team. Strong understanding of browser compatibility and web standards and...

Desired Experience: JavaScript, Google AdWords, SEO, Adobe CS, HTML5, MailChimp, Adobe Illustrator, Adobe Photoshop, Git, WordPress, CSS, SEO Tools, PHP

Easily apply

5 days ago save job more...

Servlet 3-Tier Architecture

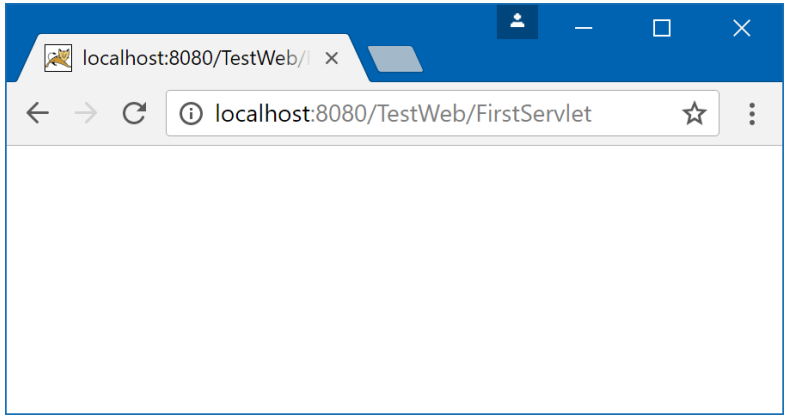


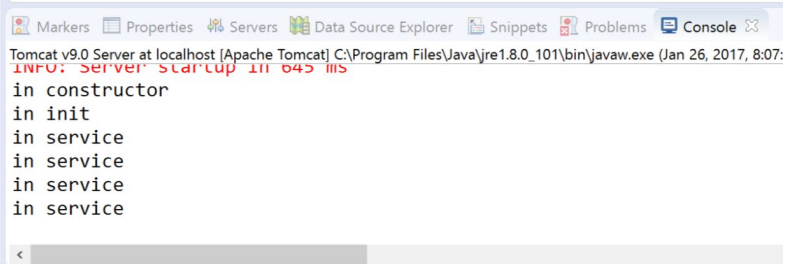
Java Servlets

- Servlets are Java classes that can serve dynamic web content
- Servlets get compiled and executed **on the server** and generate client-side code to send back out to the browser
  - Client-side code is HTML, CSS, and JavaScript
- To create a servlet, extend the `HttpServlet` class
  - `void doGet(HttpServletRequest req, HttpServletResponse resp)`
  - `void doPost(HttpServletRequest req, HttpServletResponse resp)`
  - `void service(HttpServletRequest req, HttpServletResponse resp)`
    - Dispatches to `doGet` or `doPost` if not overridden
  - `void init(ServetConfig config)`

My First Servlet

```
1  package csci201;
2  import java.io.IOException;
3  import javax.servlet.ServletConfig;
4  import javax.servlet.ServletException;
5  import javax.servlet.annotation.WebServlet;
6  import javax.servlet.http.*;
7
8  @WebServlet("/FirstServlet")
9  public class TestServlet extends HttpServlet {
10   private static final long serialVersionUID = 1L;
11   public TestServlet() {
12     super();
13     System.out.println("in constructor");
14   }
15   public void init(ServletConfig config) throws ServletException {
16     System.out.println("in init");
17   }
18   protected void service(HttpServletRequest request, HttpServletResponse response)
19     throws ServletException, IOException {
20     System.out.println("in service");
21   }
22 }
```





Annotations

- Annotations allow us to specify configuration parameters that are used by the application server in our .java files
- Before annotations (or even now if you want), you would have to modify the web.xml file of the application server

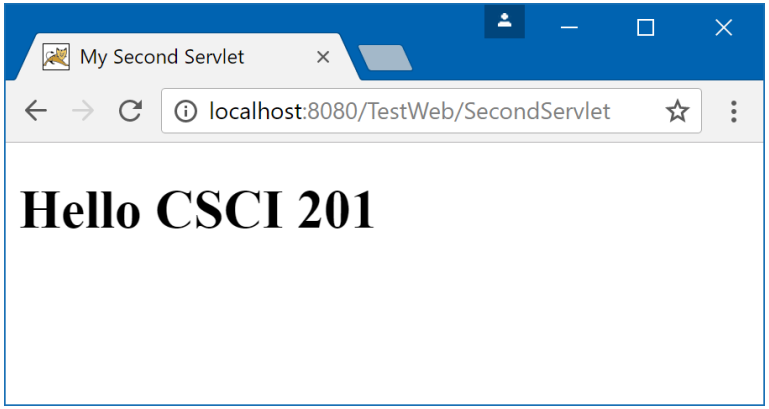
```
<servlet>
  <servlet-name>mytest</servlet-name>
  <init-param>
    <param-name>n1</param-name>
    <param-value>v1</param-value>
  </init-param>
  <init-param>
    <param-name>n2</param-name>
    <param-value>v2</param-value>
  </init-param>
</servlet>
<servlet-mapping>
  <servlet-name>mytest</servlet-name>
  <url-mapping>/myurl</url-mapping>
</servlet-mapping>
```

- Instead, we can write the following line immediately above the class declaration of our servlet

```
@WebServlet(name="mytest",
            urlPatterns={"/myurl"},
            initParams={
             @InitParam(name="n1",
                        value="v1"),
             @InitParam(name="n2",
                        value="v2")
           })
```

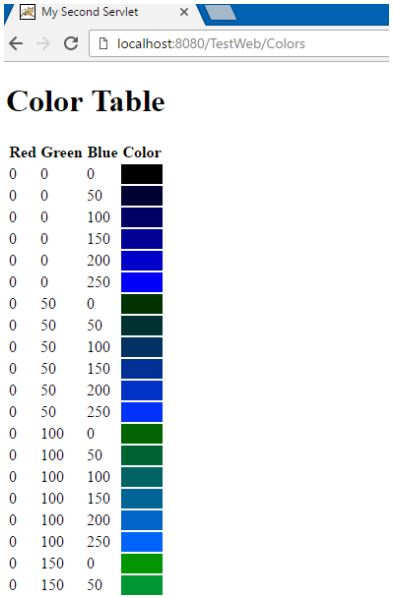
My Second Servlet

```
1  package csci201;
2  import java.io.IOException;
3  import java.io.PrintWriter;
4  import javax.servlet.ServletException;
5  import javax.servlet.annotation.WebServlet;
6  import javax.servlet.http.*;
7
8  @WebServlet("/SecondServlet")
9  public class TestServlet extends HttpServlet {
10   private static final long serialVersionUID = 1L;
11   protected void service(HttpServletRequest request, HttpServletResponse response)
12     throws ServletException, IOException {
13     response.setContentType("text/html");
14     PrintWriter out = response.getWriter();
15     out.println("<!DOCTYPE html>");
16     out.println("<html>");
17     out.println("<head>");
18     out.println("<title>My Second Servlet</title>");
19     out.println("</head>");
20     out.println("<body>");
21     out.println("<h1>Hello CSCI 201</h1>");
22     out.println("</body>");
23     out.println("</html>");
24   }
25 }
```



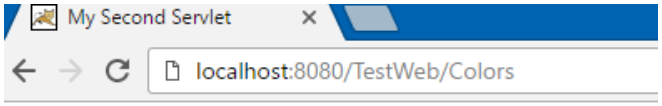
Color Servlet

```
1  // omitted package and imports for space
2  @WebServlet("/Colors")
3  public class TestServlet extends HttpServlet {
4    private static final long serialVersionUID = 1L;
5    private String getColor(int r, int g, int b) {
6      String color = "";
7      color += makeHex(r);
8      color += makeHex(g);
9      color += makeHex(b);
10     return color;
11   }
12   private String makeHex(int color) {
13     String hexString = Integer.toHexString(color);
14     if (hexString.length() == 1) {
15       hexString = "0" + hexString;
16     }
17     return hexString;
18   }
19   protected void service(HttpServletRequest request,
20                          HttpServletResponse response)
21     throws ServletException, IOException {
22     response.setContentType("text/html");
23     PrintWriter out = response.getWriter();
24     out.println("<!DOCTYPE html>");
25     out.println("<html>");
26     out.println("<head>");
27     out.println("<title>My Second Servlet</title>");
28     out.println("</head>");
29     out.println("<body>");
30     out.println("<h1>Color Table</h1>");
31     out.println("<table>");
32     out.println("<tr><th>Red</th>”);
33     out.println(“<th>Green</th>”);
34     out.println(“<th>Blue</th>”);
35     out.println(“<th>Color</th></tr>");
36     for (int red=0; red < 255; red+=50) {
37       for (int green=0; green < 255; green+=50) {
38         for (int blue=0; blue < 255; blue+=50) {
39           out.println("<tr>");
40           out.print("<td>" + red + "</td>");
41           out.print("<td>" + green + "</td>");
42           out.print("<td>" + blue + "</td>");
43           String color = getColor(red, green, blue);
44           out.print("<td style=\"background-color:#" + color + ";\">”);
45           out.print(“ </td>");
46           out.println("</tr>");
47         }
48       }
49     }
50     out.println("</table>");
51     out.println("</body>");
52     out.println("</html>");
53   }
54 }
```



Color Servlet Generated HTML

```
1   <!DOCTYPE html>
2   <html>
3   <head>
4   <title>My Second Servlet</title>
5   </head>
6   <body>
7   <h1>Color Table</h1>
8   <table>
9   <tr><th>Red</th><th>Green</th><th>Blue</th><th>Color</th></tr>
10  <tr>
11  <td>0</td>
12  <td>0</td>
13  <td>0</td>
14  <td style="background-color:#000000;"> </td>
15  </tr>
16  <tr>
17  <td>0</td>
18  <td>0</td>
19  <td>50</td>
20  <td style="background-color:#000032;"> </td>
21  </tr>
22  <tr>
23  <td>0</td>
24  <td>0</td>
25  <td>100</td>
26  <td style="background-color:#000064;"> </td>
27  </tr>
...
1306 </table>
1307 </body>
1308 </html>
```

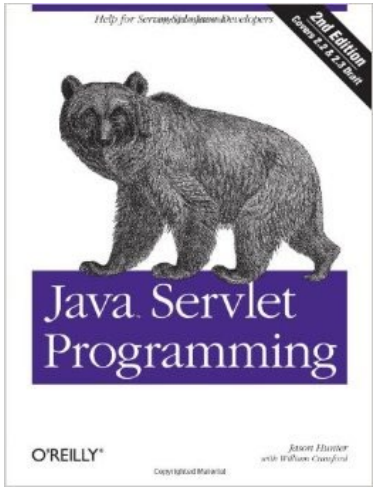


Color Table

Red	Green	Blue	Color
0	0	0	
0	0	50	
0	0	100	
0	0	150	
0	0	200	
0	0	250	
0	50	0	
0	50	50	
0	50	100	
0	50	150	
0	50	200	
0	50	250	
0	100	0	
0	100	50	
0	100	100	
0	100	150	
0	100	200	
0	100	250	
0	150	0	
0	150	50	

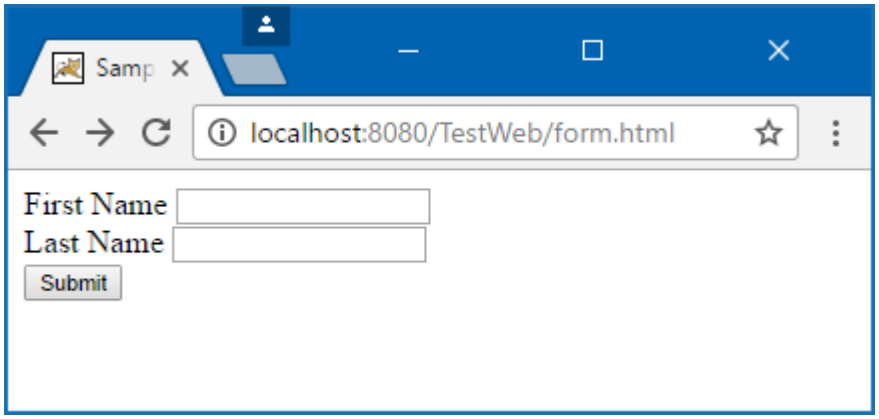
Servlets and HTML Forms

- Servlets can be used to process the data submitted from an HTML form through the `HttpServletRequest` variable in the `doGet`, `doPost`, or `service` method
- Here are a few of the more commonly used methods
  - `Cookie[] getCookies()`
  - `String getQueryString()`
  - `HttpSession getSession()`
  - `String getParameter(String)`
  - `Enumeration<String> getParameterNames()`
  - `String[] getParameterValues(String)`
    - Returns all of the values associated with a specific parameter name



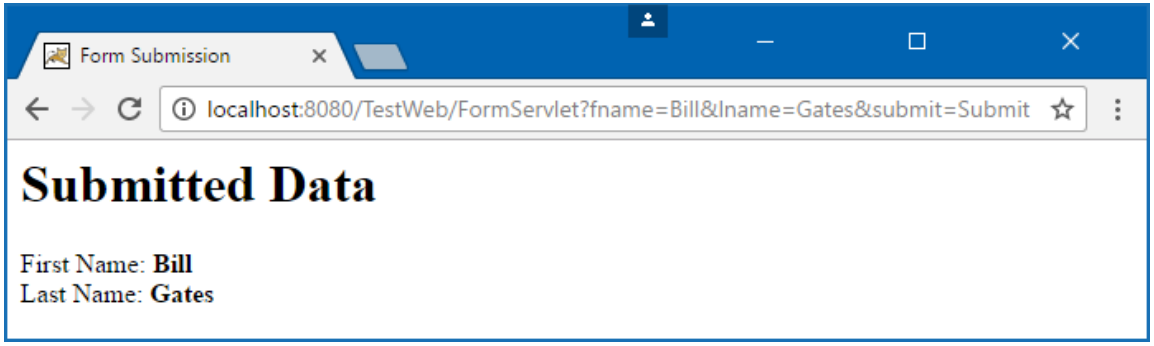
Servlet Form Example

```
<!DOCTYPE html>
<html>
  <head>
    <title>Sample Form</title>
  </head>
  <body>
    <form name="myform" method="GET" action="FormServlet">
      First Name <input type="text" name="fname" /><br />
      Last Name <input type="text" name="lname" /><br />
      <input type="submit" name="submit" value="Submit" />
    </form>
  </body>
</html>
```



Servlet Form Example

```
1  // package and imports omitted for space
2  @WebServlet("/FormServlet")
3  public class FormServlet extends HttpServlet {
4    private static final long serialVersionUID = 1L;
5    protected void service(HttpServletRequest request, HttpServletResponse response)
6      throws ServletException, IOException {
7      PrintWriter out = response.getWriter();
8      String fname = request.getParameter("fname");
9      String lname = request.getParameter("lname");
10     System.out.println("fname = " + fname);
11     System.out.println("lname = " + lname);
12
13     response.setContentType("text/html");
14     out.println("<html>");
15     out.println("<head><title>Form Submission</title></head>");
16     out.println("<body>");
17     out.println("<h1>Submitted Data</h1>");
18     out.println("First Name:<strong> " + fname + "</strong><br />");
19     out.println("Last Name:<strong> " + lname + "</strong>");
20     out.println("</body>");
21     out.println("</html>");
22   }
23 }
```



Forwarding from a Servlet

- Because servlets have a lot of overhead when generating client-side code, forwarding to a different page is often used
  - The request and response objects can be forwarded to the page too
  - The servlet can do some processing of the data, possibly even modify or amend it, then forward to another page
  - This is separating the display and business logic (view and controller in MVC)

```
// omitted package and import statements for space
@WebServlet("/FormServlet")
public class FormServlet extends HttpServlet {
  private static final long serialVersionUID = 1L;
  protected void service(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    String username = request.getParameter("username");
    String next = "/invalidUsername.jsp";
    if (username != null && username.equals("csci201")) {
      next = "/validUsername.jsp";
    }
    RequestDispatcher dispatch = getServletContext().getRequestDispatcher(next);
    dispatch.forward(request,response);
  }
}
```

Java Servlet Frameworks

- There is a popular Java Servlet framework currently called Spring Boot (https://projects.spring.io/spring-boot/)
  - Spring Boot tries to remove all configuration and allow a programmer to focus solely on writing Java code

Spring Boot makes it easy to create stand-alone, production-grade Spring based Applications that you can "just run". We take an opinionated view of the Spring platform and third-party libraries so you can get started with minimum fuss. Most Spring Boot applications need very little Spring configuration.

Features

- Create stand-alone Spring applications
- Embed Tomcat, Jetty or Undertow directly (no need to deploy WAR files)
- Provide opinionated 'starter' POMs to simplify your Maven configuration
- Automatically configure Spring whenever possible
- Provide production-ready features such as metrics, health checks and externalized configuration
- Absolutely **no code generation** and **no requirement for XML** configuration

More Servlets

- For more information on Servlets
  - Go to https://docs.oracle.com/javaee/7/tutorial/servlets.htm
  - Go through one of the many servlet tutorials online

Outline

- Java Servlets
- Program

Program

- Create the following form and process the submitted data with a servlet to display the page on the right with a servlet.