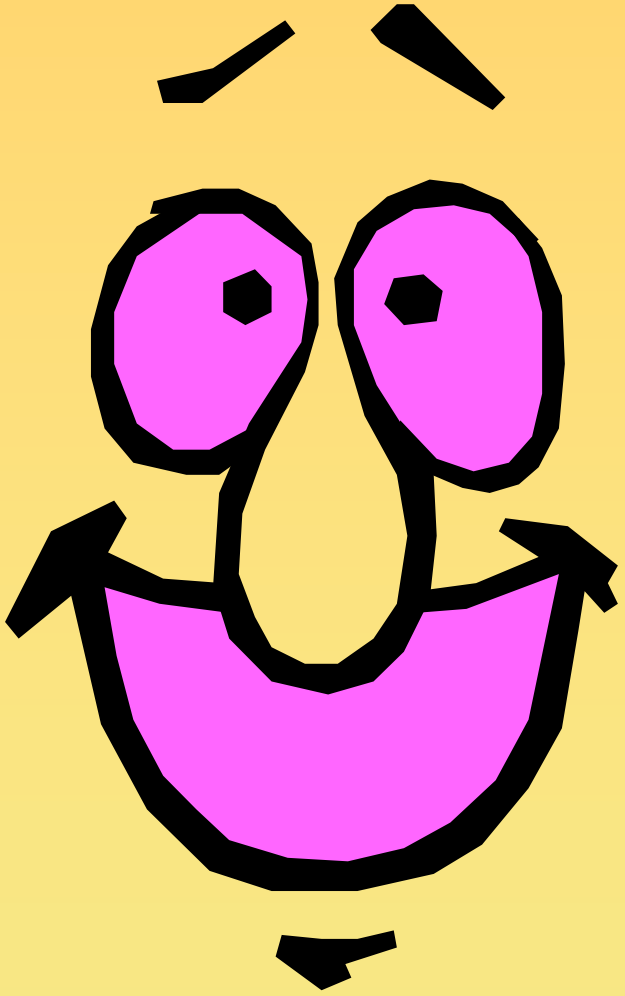
Welcome ☺



Dynamic Content

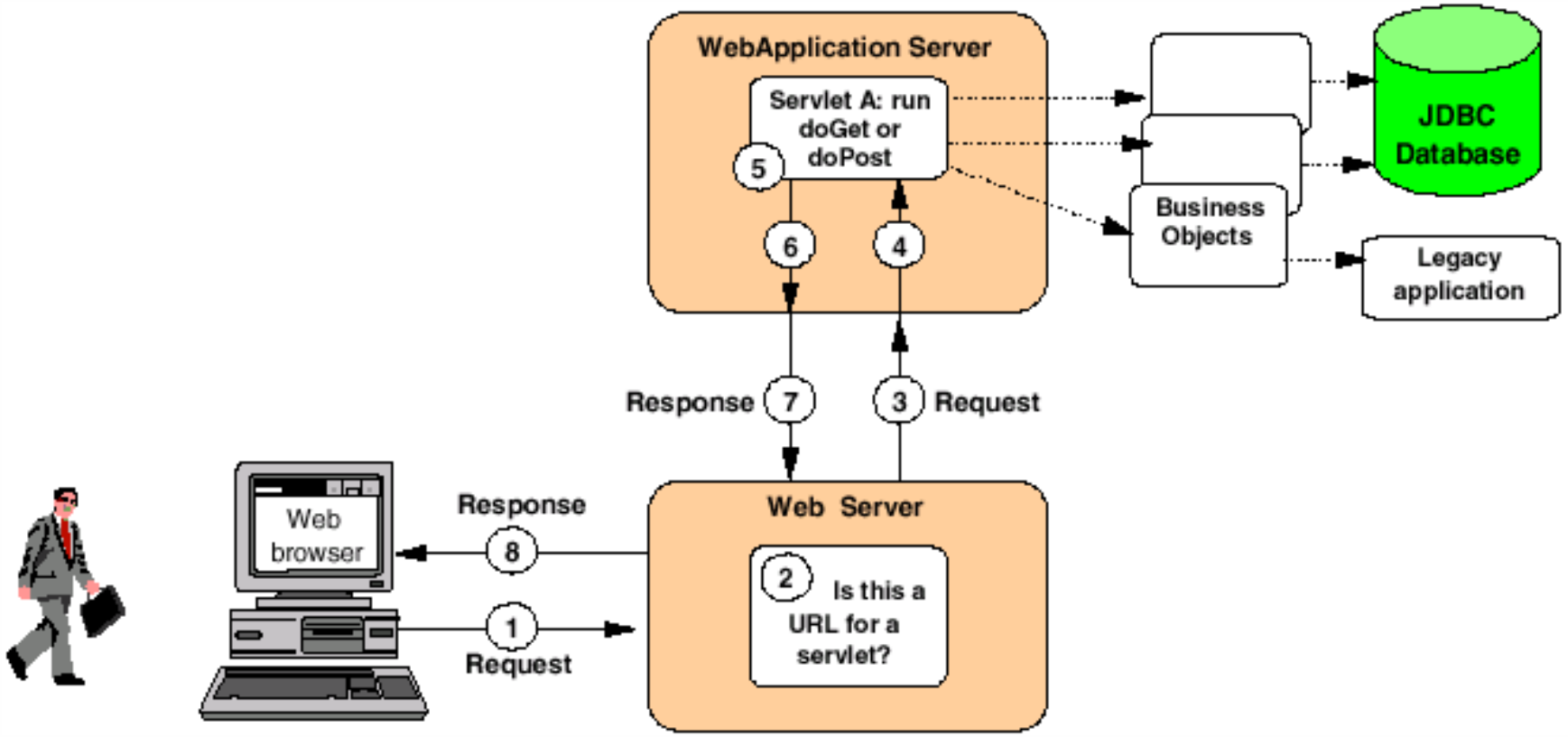
Outline

- Internet and World Wide Web
  - Web Browsers and Web Servers
  - Protocols and Standards
- Hyper Text Markup Language
  - Static Content
  - Dynamic Content
    - Dynamic Web Pages
    - CGI Programming
    - Form elements in HTML
    - Processing Data from a Form

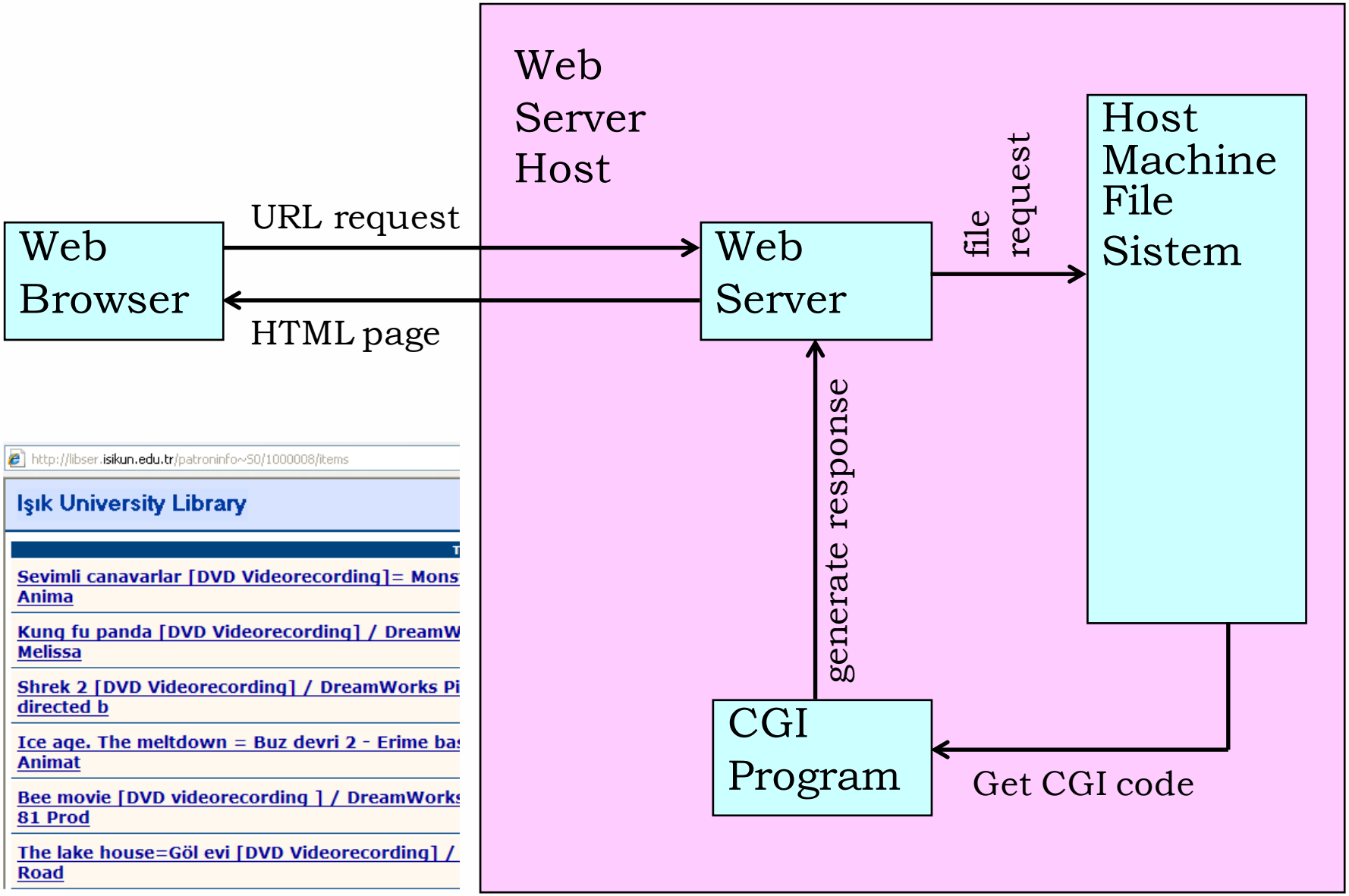
Dynamic Web Content

- Sometimes the content of a page may change from time to time
- For example, if a web page contains following information, its content will be (dynamically) changed each time it is displayed
  - Current Time
  - Currency rate
  - Shopping Chart content
  - ...

Running a (Dynamic) Web Content



Running a (Dynamic) Web Content



Steps

1. Read data
    a. Explicit data sent by the client user/program
        - User data entered in an HTML form
        - Data generated by an application such as an applet or a custom HTTP client program
    b. Implicit HTTP request data sent by browser
2. Generate the results
3. Send data to the client
    a. Explicit data to the client
    b. Implicit HTTP response data

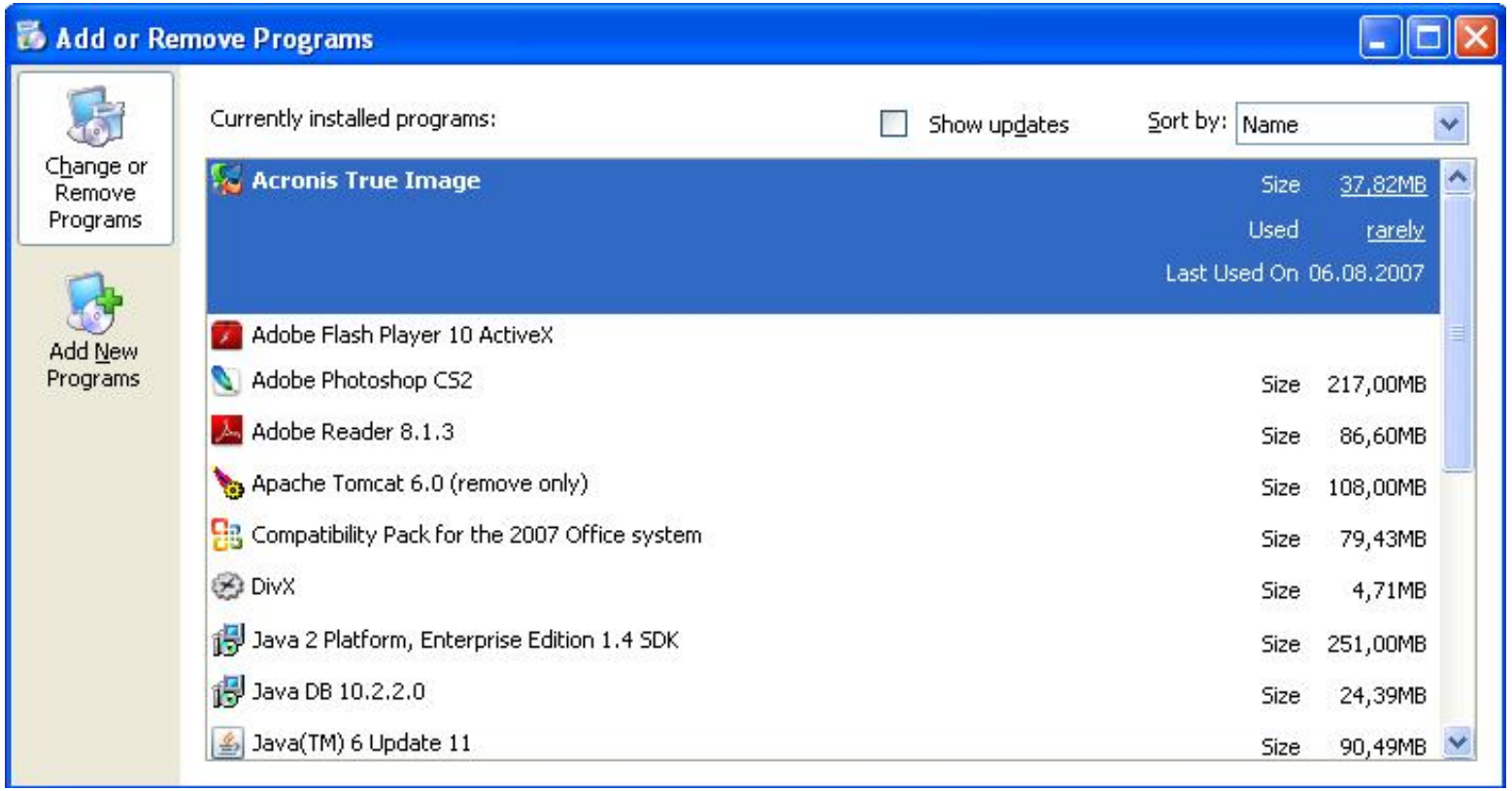
Web Server functionality

- Client may request data from the database, or the web server may require talking to the database
- DBMS doesn't speak HTTP or return results in HTML, so the Web browser can't talk directly to the database
- Even if it could, for security reasons, we would not want it
- The same argument applies to most other applications
- Web server talks to the application, receive the data, and embed the results inside a document

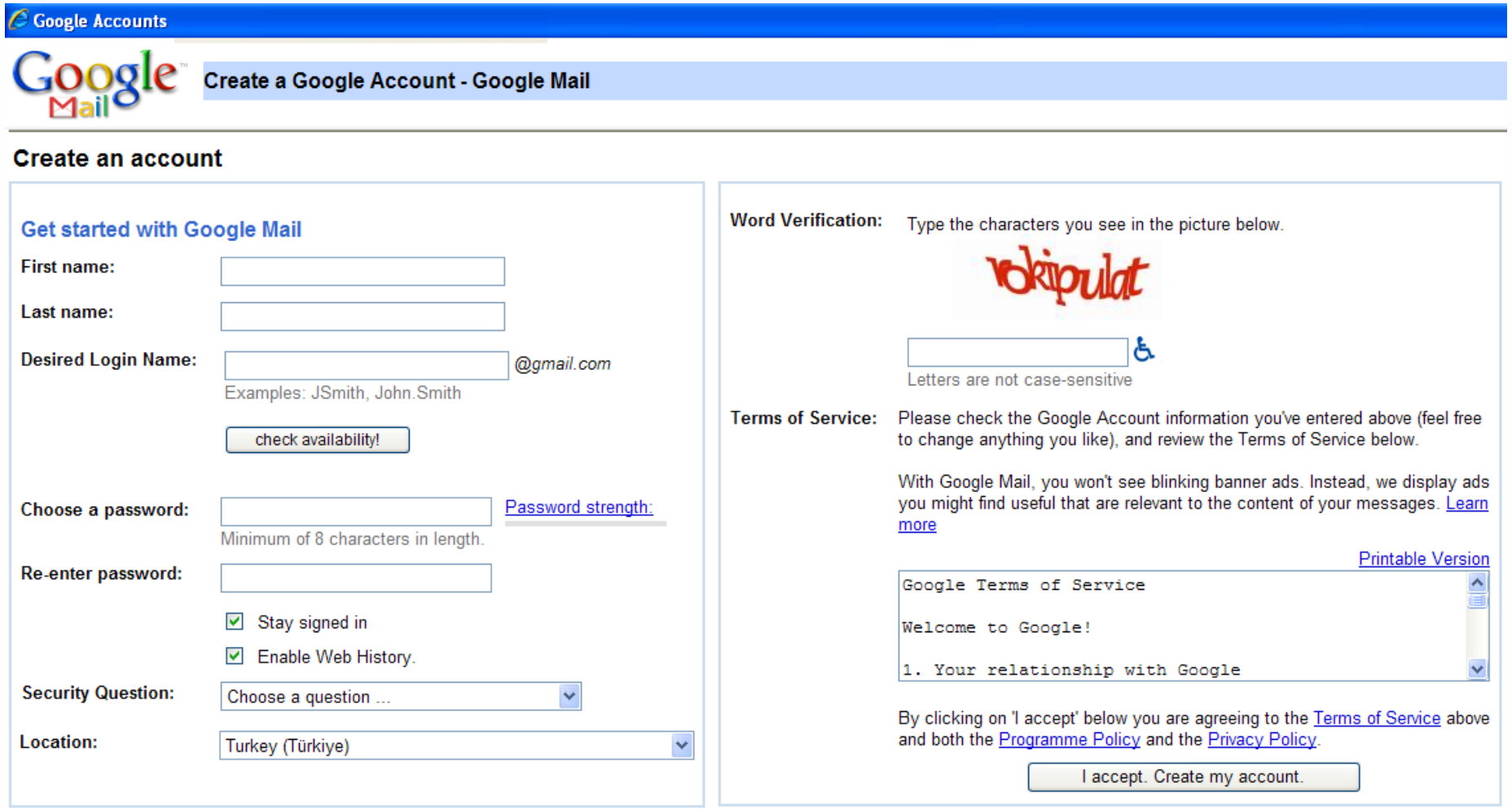
Why to build pages dynamically?

- Content of the page may depend upon user submitted data
  - a static page never satisfies the request, or
  - too many options may exist
- Information represented by the HTML may be dynamic (may change over time) by nature
- The HTML may be customizable on a per user basis (depend on user)

Why to build pages dynamically?



Why to build pages dynamically?



Outline

- Internet and World Wide Web
  - Web Browsers and Web Servers
  - Protocols and Standards
- Hyper Text Markup Language
  - Static Content
  - Dynamic Content
  - Forms
    - CGI Programming
    - Form elements in HTML
    - Processing Data from a Form

Form

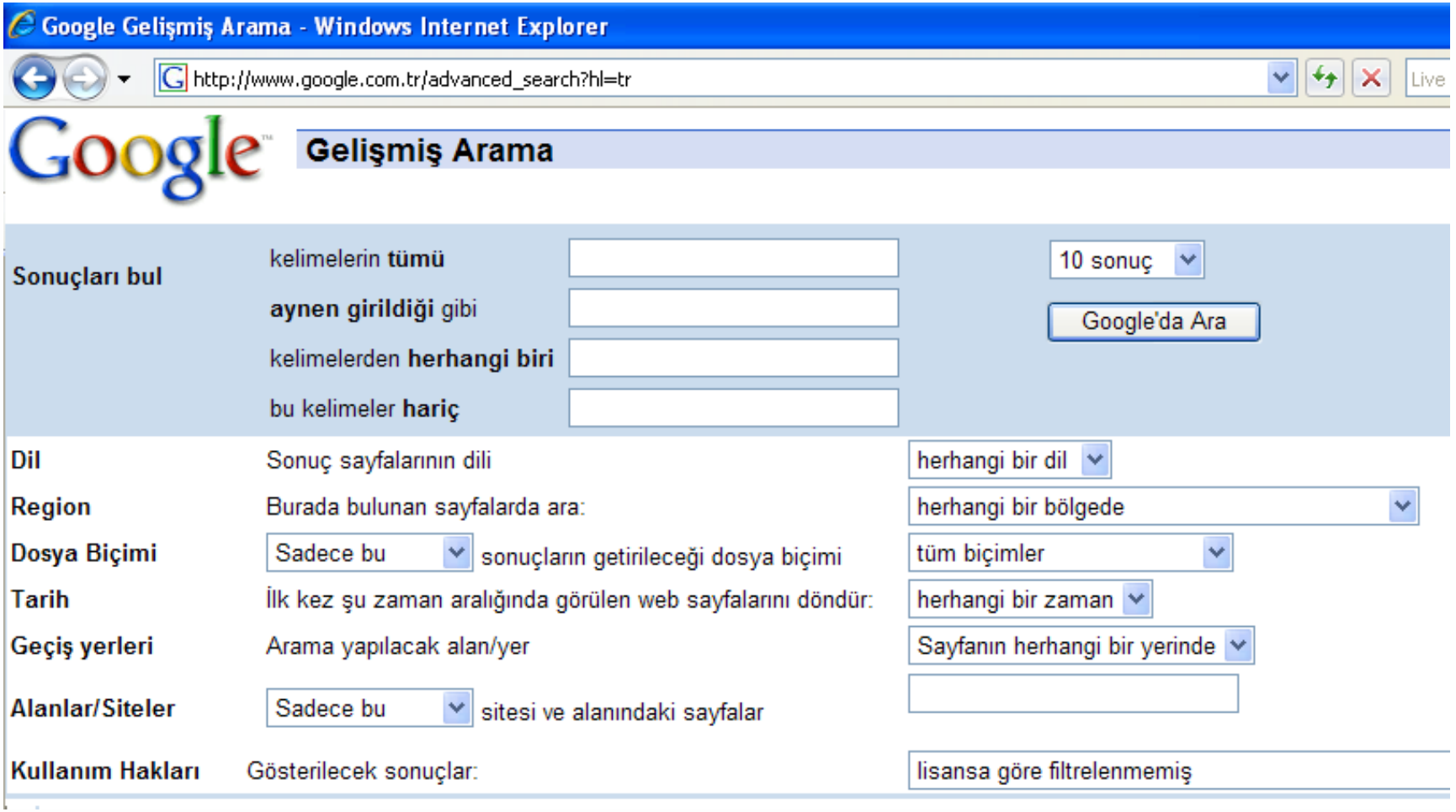
- A form on a web page allows a user to enter data which will be sent by the web browser to the web server (user name)
- Forms can also be used to
  - submit data to save on a server (e.g., ordering a product), or:
  - retrieve data (e.g., searching on a search engine)
    - which also includes submitting data to the server

Form

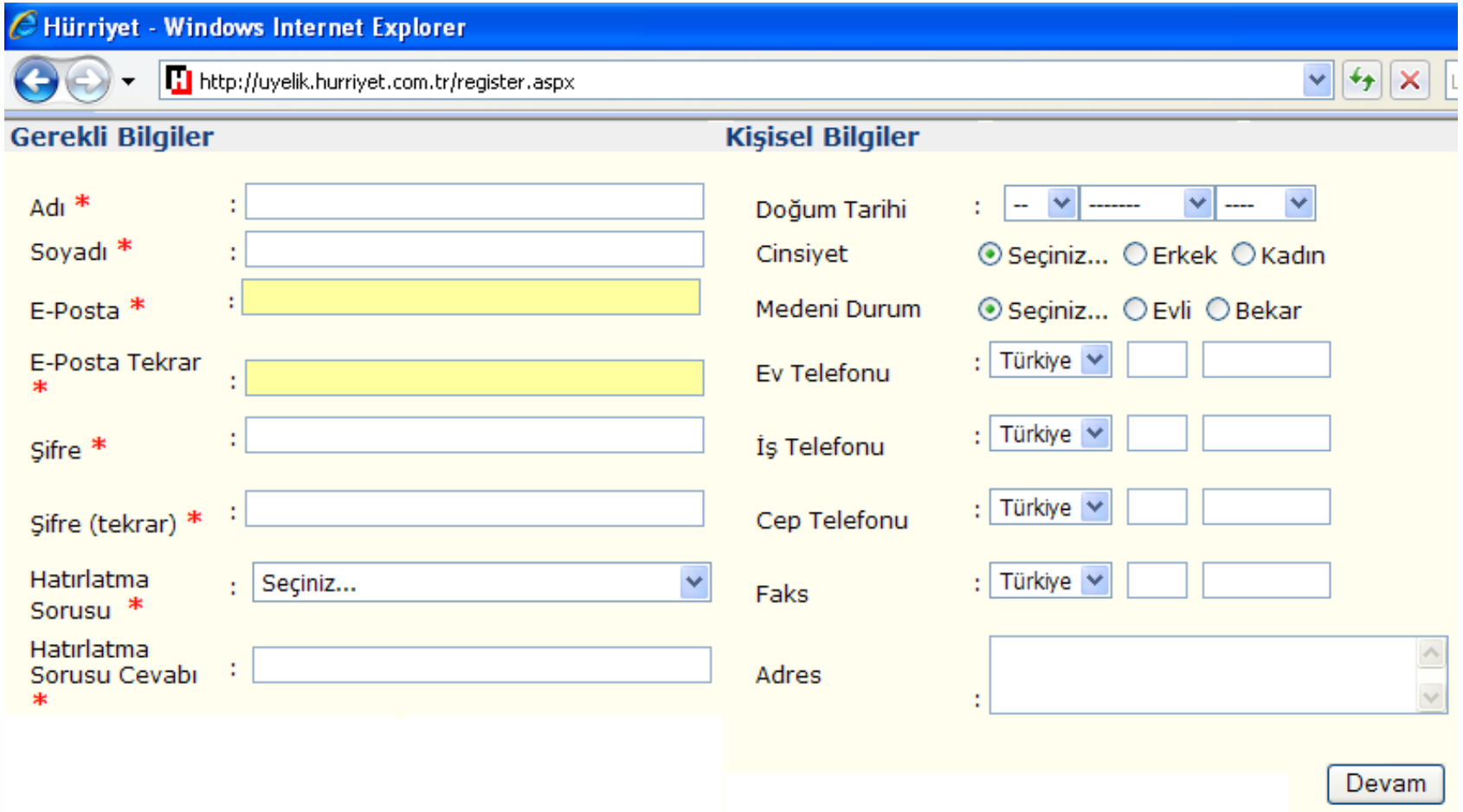
- When the data is entered, the user should press a button with type=submit to enable processing another (jsp) file on the web server, by reading the values that the user entered

Form Examples

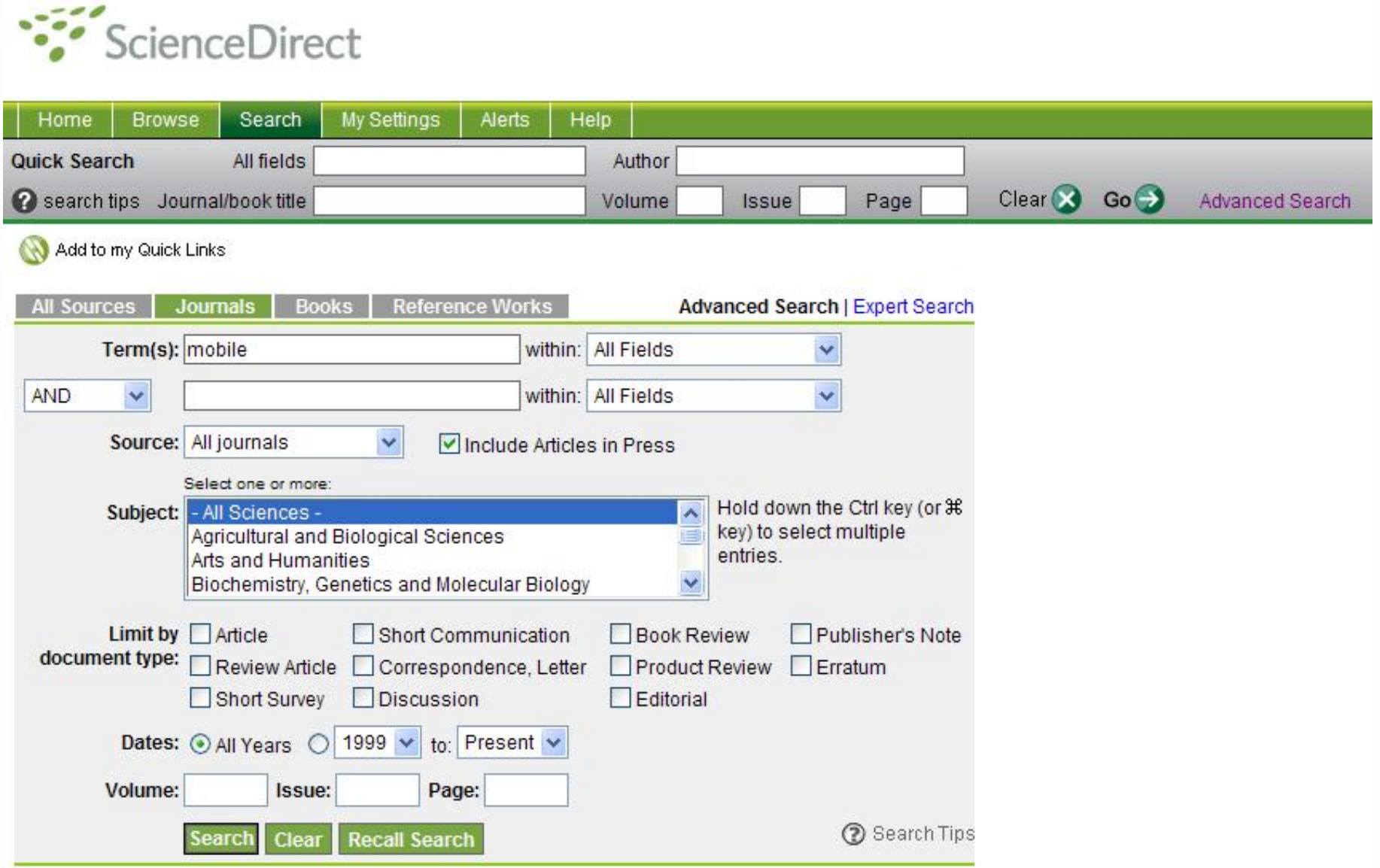
Form Examples



Form Examples



Form Examples



GET and POST methods

- Two most common parameter passing methods by the Web browser to the CGI program via Web server are GET and POST methods
- When issuing a CGI requests directly from a URL, GET method is used
- This URL is known as a Query String. It contains:
  - Name of the CGI program
  - Parameters
  - Values of the parameters

http://www.vedatcoskun.com/cgi-bin/
getResult.cgi?userid=vedat+coskun&pass=seal

GET and POST methods

- The form explicitly specifies which method will be used
- When GET method is used, data created in the form is appended to the Query String
- When POST method is used, a file containing the data is sent as a seperate file
- POST method is more secure

Outline

- Internet and World Wide Web
  - Web Browsers and Web Servers
  - Protocols and Standards
- Hyper Text Markup Language
  - Static Content
  - Dynamic Content
    - Forms
    - CGI Programming
    - Form elements in HTML
    - Processing Data from a Form

Dynamic Web Content

- Common Gateway Interface (CGI) is the standard to generate dynamic web content
- CGI provides a standard on the communication among:
  1. Web browsers
  2. Web servers
  3. Executable programs running on the Web Server (computer)

CGI is execution

- Web server responds to requests of the web browsers by returning the appropriate output:
  1. if the request identifies a (html) file stored on disk, it returns that file
  2. if the request identifies an executable program and arguments, it runs the program and returns its output
- CGI defines a standard way of doing the second

Processing User Request

- The task of Web Servers is to respond to requests of client web browsers
- Each time a request is received, the server analyzes what the request asks for, and returns the appropriate output:
  1. Static processing: if the request identifies a file stored on disk, return the contents of that file
  2. Dynamic processing: if the request identifies an executable command and possibly arguments, run the command and return its output

What CGI program outputs?

- The result of running a Common Gateway Interface (CGI) program is a static html file
- Hence, what a Web server returns to a web browser which requires running a CGI program is still a static web page

CGI protocol

- CGI is a standard protocol for interfacing the client and server on the internet
- CGI defines how information about the server and the request is passed to the command in the form of arguments and environment variables, and how the command can pass back extra information about the output (such as the type)

Outline

- Internet and World Wide Web
  - Web Browsers and Web Servers
  - Protocols and Standards
- Hyper Text Markup Language
  - Static Content
  - Dynamic Content
    - Forms
    - CGI Programming
    - Form elements in HTML
    - Processing Data from a Form

Form Elements

- `<form>` creates a form
- `<select name="xyz">…</select>` creates a selection list, from which the user can select a single option. May be rendered as a dropdown list.
- `<option value="x">` creates an item in a select list.
- `<label for="id">…</label>` creates a label for a form input (e.g. radio button). Clicking on the label fires a click on the matching input.

Form Elements

1. `<input type="text">` one-line text
2. `<input type="password">` masked text input
3. `<textarea>…</textarea>` multi-line text area
4. `<input type="checkbox">` checkbox
5. `<input type="radio">` radio button
6. `<input type="file">` file select
7. `<input type="hidden">` unvisible component
8. `<input type="button">` button
9. `<button>` button
10. `<input type="image“>` button with an image
11. `<input type="reset">` reset button
12. `<input type="submit">` submit button

1. <input type="text">

```
<INPUT
   type="text"
   name="firstname"
   size=10
   maxlength=20
>
```

- Creates a one-line text input
- size attribute specifies the default width of the input in character-widths
- maxlength sets the maximum number of characters the user can enter (which may be greater than size)

2. <input type="password">

```
<INPUT
   type="password"
   name="mypass"
  size=10
  maxlength=20
>
```

- Creates a one-line text input
- size and maxlength attributes behave same as in "text"

3. <input type="textarea">

```
<INPUT
   type="textarea"
   name="comment"
  rows=10
  cols=20
>
```

- Creates a multiple line text input
- rows and cols attributes defines size of the area

4. <input type="checkbox">

```
<INPUT
   type="checkbox"
   name="check1"
>Option 1
```

```
<INPUT
   type="checkbox"
   name="check2"
   CHECKED
>Option 1
```

5. <input type="radio">

```
<INPUT
   type="radio"
   name="radio1"
>Option 1
```

```
<INPUT
   type="radio"
   name="radio2"
   CHECKED
>Option 1
```

11. <input type="reset">

```
<INPUT
   type="reset"
   name="reset1"
>
```

- **Creates a button which resets all input areas with default values when clicked**

11. <input type="submit">

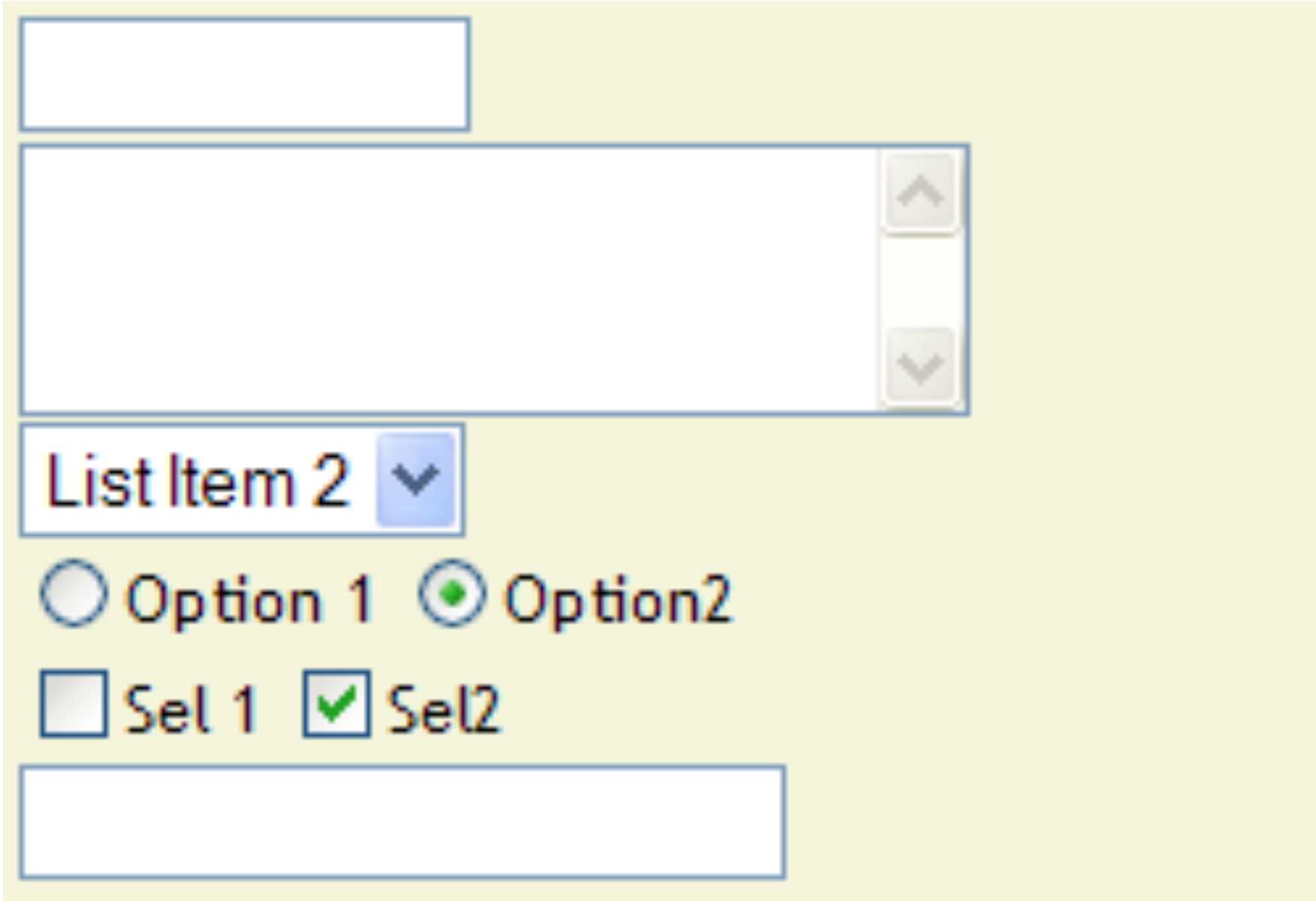
```
<INPUT
   type="submit"
   name="submit1"
>
```

- Creates a button which tells the web browser to take action on the form
- When it is pressed, web browser sends the inserted data to the web server

A Form Example

```
<FORM METHOD=post ACTION="example.jsp">
  <INPUT type="text" size="10">
  <TEXTAREA rows=3 cols=20></TEXTAREA>

  <SELECT>
    <OPTION >List Item 1
    <OPTION SELECTED>List Item 2
  </SELECT>

  <INPUT type="radio" name="o1">Option 1
  <INPUT type="radio" name="o1" CHECKED>Option2

  <INPUT type="checkbox" name="c1">Sel 1
  <INPUT type="checkbox" name="c1" CHECKED>Sel2

  <INPUT type="Submit">
  <INPUT type="Reset">
</FORM>
```

A Form Example



Outline

- Internet and World Wide Web
  - Web Browsers and Web Servers
  - Protocols and Standards
- Hyper Text Markup Language
  - Static Content
  - Dynamic Content
    - Forms
    - CGI Programming
    - Form elements in HTML
    - Processing Data from a Form

Combining Forms with Programs

- Forms can be combined with various scripting languages to allow developers to create dynamic web sites
- This includes both client-side and server-side languages

Client-Side Programs

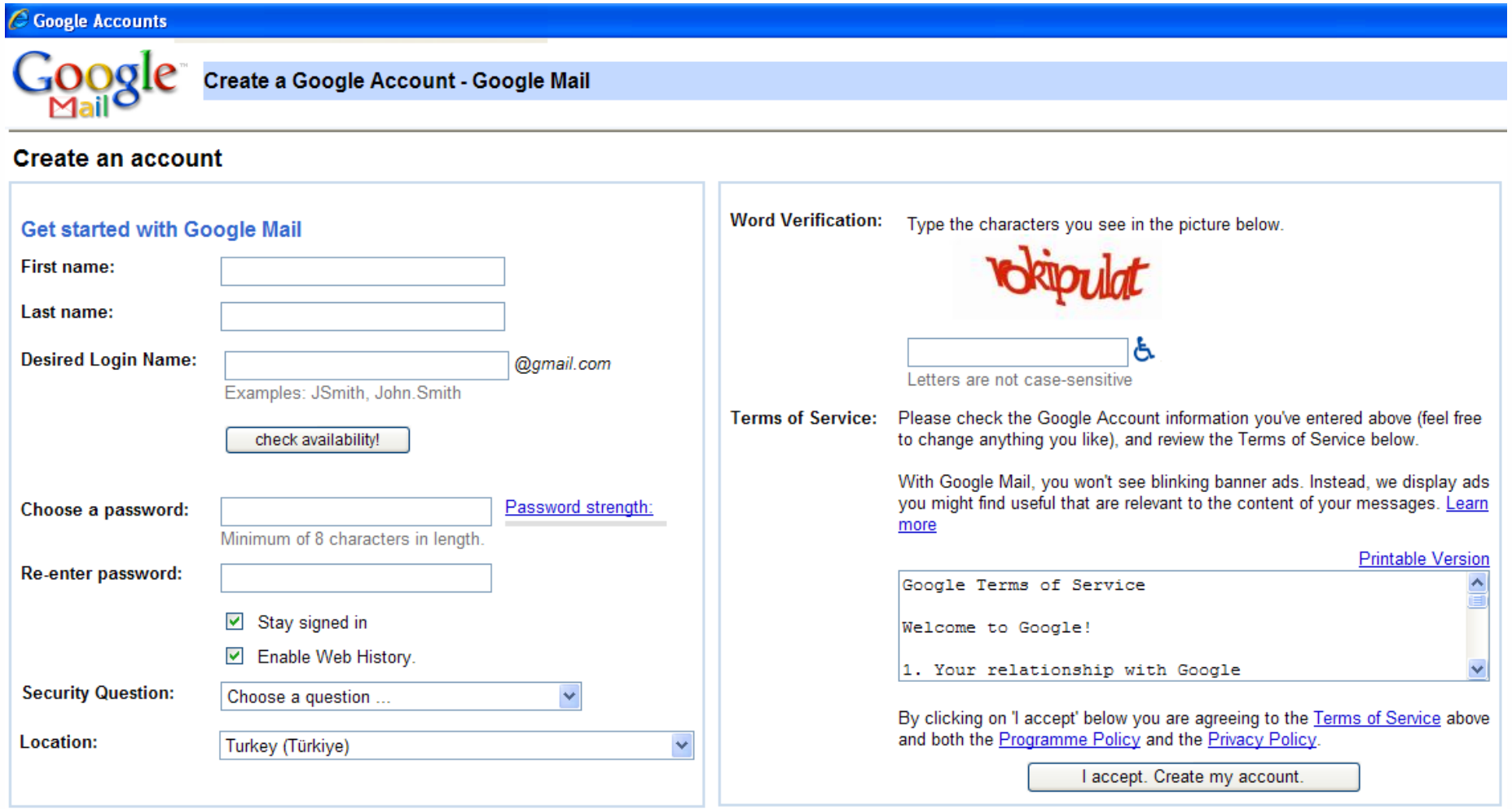
- The disadvantage of client-side programming is implementing same functionality with different codes for different web browser
- Another disadvantage is the possibility of users banning execution of client programs on the web client
- Javascript is the de facto standard client-side scripting language for web

Practical usage of Client-Side Programs

Client Side Programs (Javascript) are used for:

1. **Browser Detection: When same server side programs are to be viewed by different browsers, different behaviours might be required. Javascript can detect the client browser type, and behave accordingly**
2. **Browser Control: Various client-side actions such as opening a new window, triggering some time-dependent actions and sending a new request to server might be only done by client-side programs**
3. **Form Validation: When user enters an input to a form, it might be better to validate it, and send only the appropriate inputs to the server**

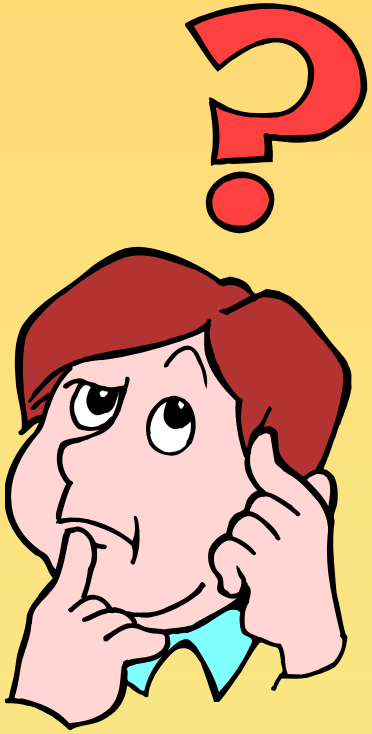
Why to build pages dynamically?

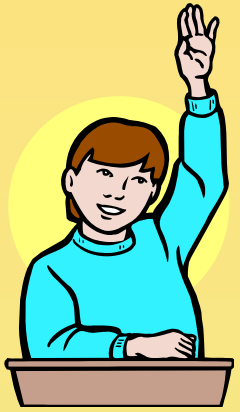


Server-Side Programs

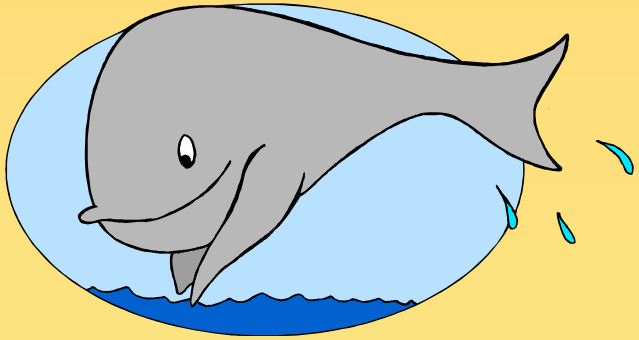
- Scripting languages are common:
  - PHP
  - Perl
  - ASP
  - Servlet, JSP, JSF
- Compiling languages are rarely used:
  - C
  - C++
  - Java

Questions?





Thank you. ☺



Teşekkürler