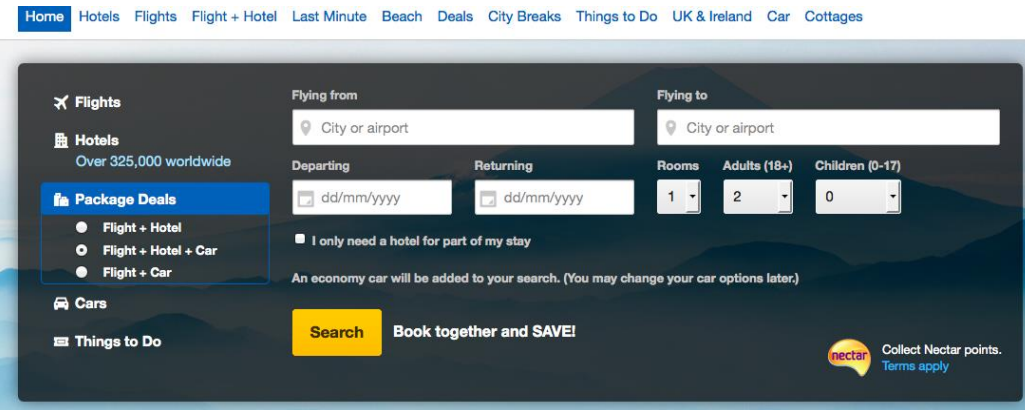


Web Applications

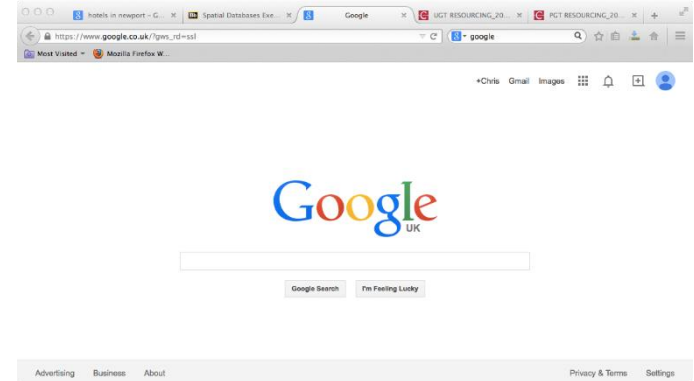
Forms & CGI with Python

Forms

Forms enable the user to enter data



The image shows a travel booking interface with a dark blue background. At the top, there is a navigation bar with links: Home, Hotels, Flights, Flight + Hotel, Last Minute, Beach, Deals, City Breaks, Things to Do, UK & Ireland, Car, and Cottages. The main content area is divided into sections. On the left, there is a sidebar with 'Flights' (selected), 'Hotels' (Over 325,000 worldwide), 'Package Deals' (Flight + Hotel, Flight + Hotel + Car, Flight + Car), 'Cars', and 'Things to Do'. The main search area has two columns. The left column has 'Flying from' and 'Flying to' fields, both with a location pin icon and the text 'City or airport'. Below these are 'Departing' and 'Returning' date pickers, both showing 'dd/mm/yyyy'. To the right of these are 'Rooms' (1), 'Adults (18+)' (2), and 'Children (0-17)' (0) dropdown menus. Below the date pickers is a checkbox 'I only need a hotel for part of my stay' and a note 'An economy car will be added to your search. (You may change your car options later.)'. At the bottom of the search area are a yellow 'Search' button and a 'Book together and SAVE!' button. On the right side of the search area, there is a 'nectar' logo and the text 'Collect Nectar points. Terms apply'.



The image shows the Google UK homepage. At the top, there is a navigation bar with links: Home, Maps, News, Shopping, Finance, and More. The main content area features the Google logo (Google UK) and a large search bar. Below the search bar are two buttons: 'Google Search' and 'I'm Feeling Lucky'. At the bottom, there is a footer with links: Advertising, Business, About, Privacy & Terms, and Settings.

- Main applications: provide user input for
- programs and databases located on a web server
 - generating a customized results web page
 - or to client-side scripts for local processing

Example applications

- **e-commerce**, to enter name, address, details of purchase and credit-card number etc
 - request brochures from a company
 - make a booking for holiday, cinema etc.
 - buy a book, cd, etc
 - obtain a map giving directions to a shop
- to support input to a database query
- to obtain feedback on a web site
 - + other type of polls or surveys.

Input types

- text
 - checkbox
 - radio (buttons)
 - password
 - submit
 - reset
 - hidden
 - file
 - button
 - image
 - select (options)
 - textarea
- +++ other HTML5 types
- See later

Tell us what you think

Your name:

And address:

How did you find out about us?:

☐ A friend told me

☐ In a dream

☐ Followed a link

In the general scheme of life, how would you rate us?:

head blowing

Feel free to write some comments here

Thanks for your interest

The method and action attributes

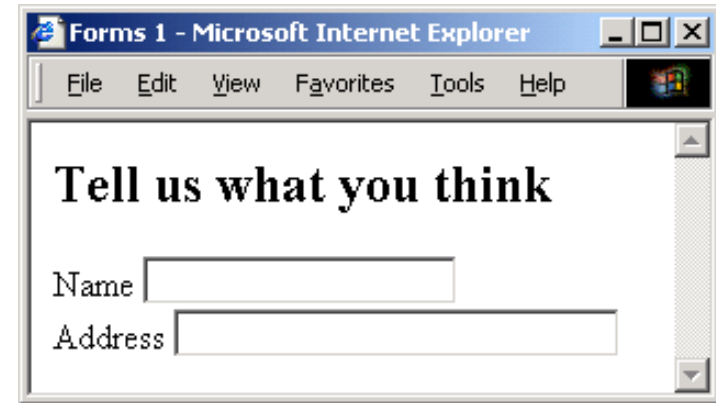
- The **method** attribute specifies the way that form data is sent to the server program
 - **GET** appends the data to the URL
 - **POST** sends the data in body of HTTP message
- The **action** attribute specifies a server program that processes the form data (often as a URL)

```
<body>
  <form method="POST" action="comments.py">
    <h2>Tell us what you think</h2>
    <!-- etc -->
  </form>
</body>
```

The `input` element: `type="text"`

type attribute specifies the type of user input

name attribute gives an identifier to the input data –
used by the server program

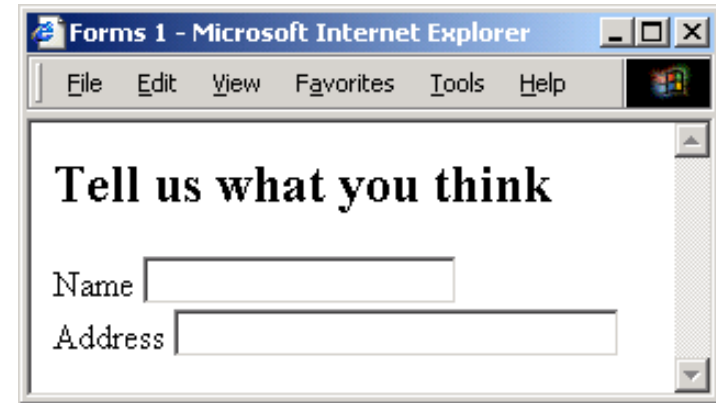


```
<form method="POST" action="comments.py">
  <h2>Tell us what you think</h2>
  Name <input type="text" name="AName" size="20" /> <br/>
  Address <input type="text" name="address" size="30" />
</form>
```

The `input` element: `type="text"`

size attribute specifies the length of a `text` input field in characters

value attribute specifies an initial value for the input data (optional)



```
<form method="POST" action="comments.py">
  <h2>Tell us what you think</h2>
  Name <input type="text" name="AName" size="20" /> <br/>
  Address <input type="text" name="address" size="30" />
</form>
```

<label> connects a prompt to the input

important for accessibility

for attribute provides **id** of associated control

Tell us what you think

Your name:

And address:


```
<form action="feedback.py" >
<h2>Tell us what you think</h2>
<label for="user">Your name: </label>
<input type="text" id="user" name="username"
                                             size="35"/>
<label for="address">And address:</label>
<input type="text" id="address" name="address"
                                             size="35"/>
```


The `input` element: `type="checkbox"`

name attribute is used to define a set of checkboxes

value attribute identifies the individual checkbox

If the **checked** attribute is set to “checked” the box is initially checked

How did you find out about us?:

- ☒ A friend told me
- ☐ In a dream
- ☐ Followed a link

```
<input type="checkbox" name="howheard"
  id="friend" value="friend" checked="checked">
  A friend told me <br />
<input type="checkbox" name="howheard" id="dream"
  value="dream"> In a dream <br />
<input type="checkbox" name="howheard" id="link"
  value="link"> Followed a link
```

For checkbox and radio, user can click on the label text to set the control

How did you find out about us?:

- ☐ A friend told me
- ☐ In a dream
- ☐ Followed a link

```
How did you find out about this web site?: <br>
<input type="checkbox" name="howheard"
        id="friend" value="friend">
<label for="friend"> A friend told me</label><br>
<input type="checkbox" name="howheard"
        id="dream" value="dream">
<label for="dream"> In a dream </label><br>
<input type="checkbox" name="howheard"
        id="link" value="link">
<label for="link"> Followed a link </label>
```

The `input` element: `type="radio"`

- Radio buttons are similar to checkboxes, but only one of a set can be selected
- To select a button by default, use the **checked** attribute (for one button only)

How did you find out about us?:

- ☒ A friend told me
- ☐ In a dream
- ☐ Followed a link

```
<input type="radio" name="howheard" id="friend"
value="friend" checked="checked">
                                A friend told me <br >
<input type="radio" name="howheard" id="dream"
value="dream"> In a dream <br>
<input type="radio" name="howheard" id="link"
value="link"> Followed a link
```

The `input` element: `type="button"`

name attribute uniquely identifies a button

value attribute gives a label to the button

Actions can be associated with buttons using JavaScript

Do you want to receive any further information:

A screenshot of a web form. It contains a text label "Do you want to receive any further information:" followed by two buttons. The first button is labeled "Yes" and the second button is labeled "No". Both buttons have a light gray background and a thin border.

```
<h3>Do you want to receive any further  
information: </h3>
```

```
<input type="button" name="yes" value='Yes' />
```

```
<input type="button" name="no" value='No' />
```

The `input` element: submit and reset

`type="submit"`

- clicking this button sends the form data to the program (URL) specified in the `action` attribute of the form



`type="reset"`

- clicking this button clears all data entered so far

```
Thank you <br>
<input type="submit" name="send" value="Send" >
<input type="reset" name="clear" value="Clear" >
```

The `input` element (cont.)

`type="password"`

- similar to `type="text"` except that the input is echoed with asterisks (so not visible)

`type="file"`

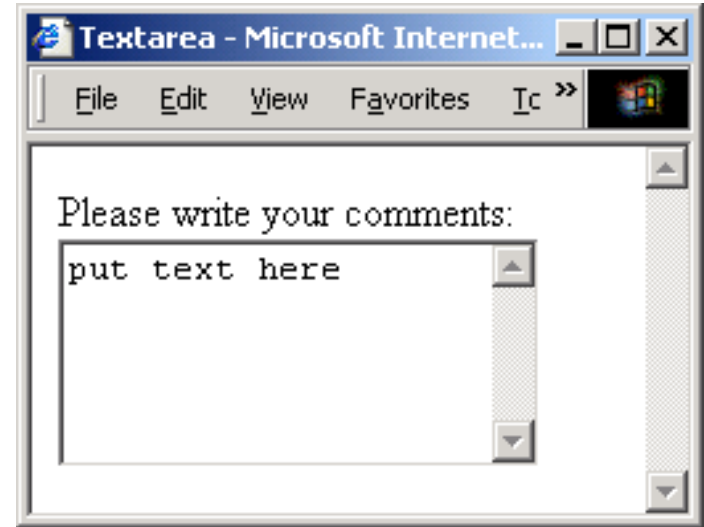
- provides a file dialogue box to specify a file that is sent to the server

`type="hidden"`

- similar to text input, but the `value` attribute is used to specify data that is to be sent to the server. Nothing appears on the screen.

The `textarea` element

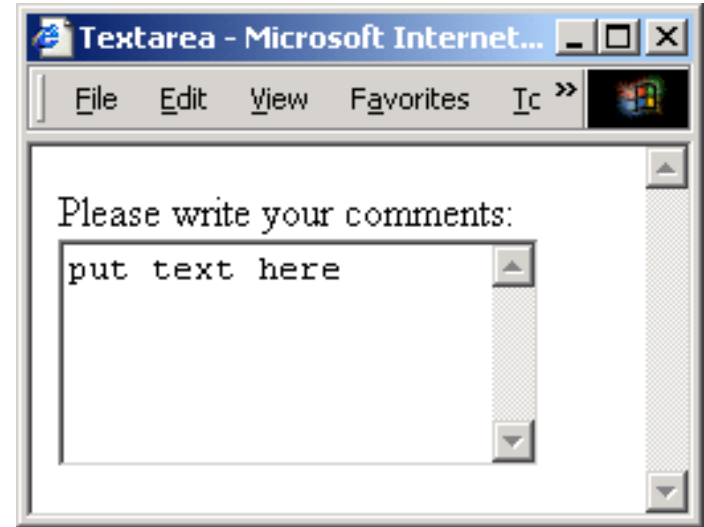
- Used for multi-line text input
- The size of the input area in characters is specified with the `cols` and `rows` attributes
- Any text placed inside the element appears in the input area (this can be deleted).



```
Please write your comments:<br>
<textarea name="comments" rows="5" cols="20">
  put text here
</textarea>
```

The `textarea` element

- Used for multi-line text input
- The size of the input area in characters is specified with the `cols` and `rows` attributes
- Use the `placeholder` attribute for text displayed as temporary suggestions

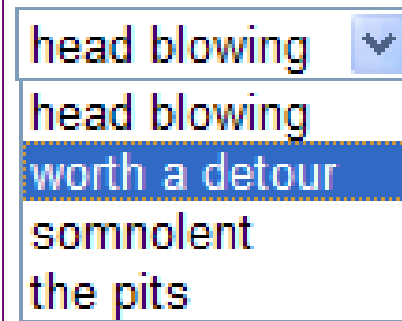


```
Please write your comments:<br>
<textarea name="comments" rows="5" cols="20"
placeholder="put text here">
</textarea>
```


The `select` element

- The `select` element provides a drop down menu of options
- An option can be selected by default using the `selected` attribute (otherwise the first in the list is initially selected)

In the general scheme of life, how would you rate us?:



In the general scheme of life, how would you rate us?:

```
<select name="quality">  
  <option value="headblowing">head blowing</option>  
  <option value="worthadetour">worth a detour</option>  
  <option value="somnolent">somnolent</option>  
  <option value="thepits">the pits</option>  
</select>
```

<button> element

Place text or/and image inside the element

Press to win the jackpot

Doesn't have to be inside a form element
– use **form** attribute to specify id of the associated form

Hi Dylan



```
<button form="theform" name="jackpot"
type="submit"> Press to win the jackpot
</button>
```

```
<button form="theform" name="dt"
type="submit">      <h3>Hi Dylan</h3>
 </button>
```

HTML5 additional <input> types and attributes

HTML5 provides support for various types of input – but may be handled differently by different browsers

e.g.:

date, **email**, **url**, **number**, **range**, **color**,
pattern (regular expression)...

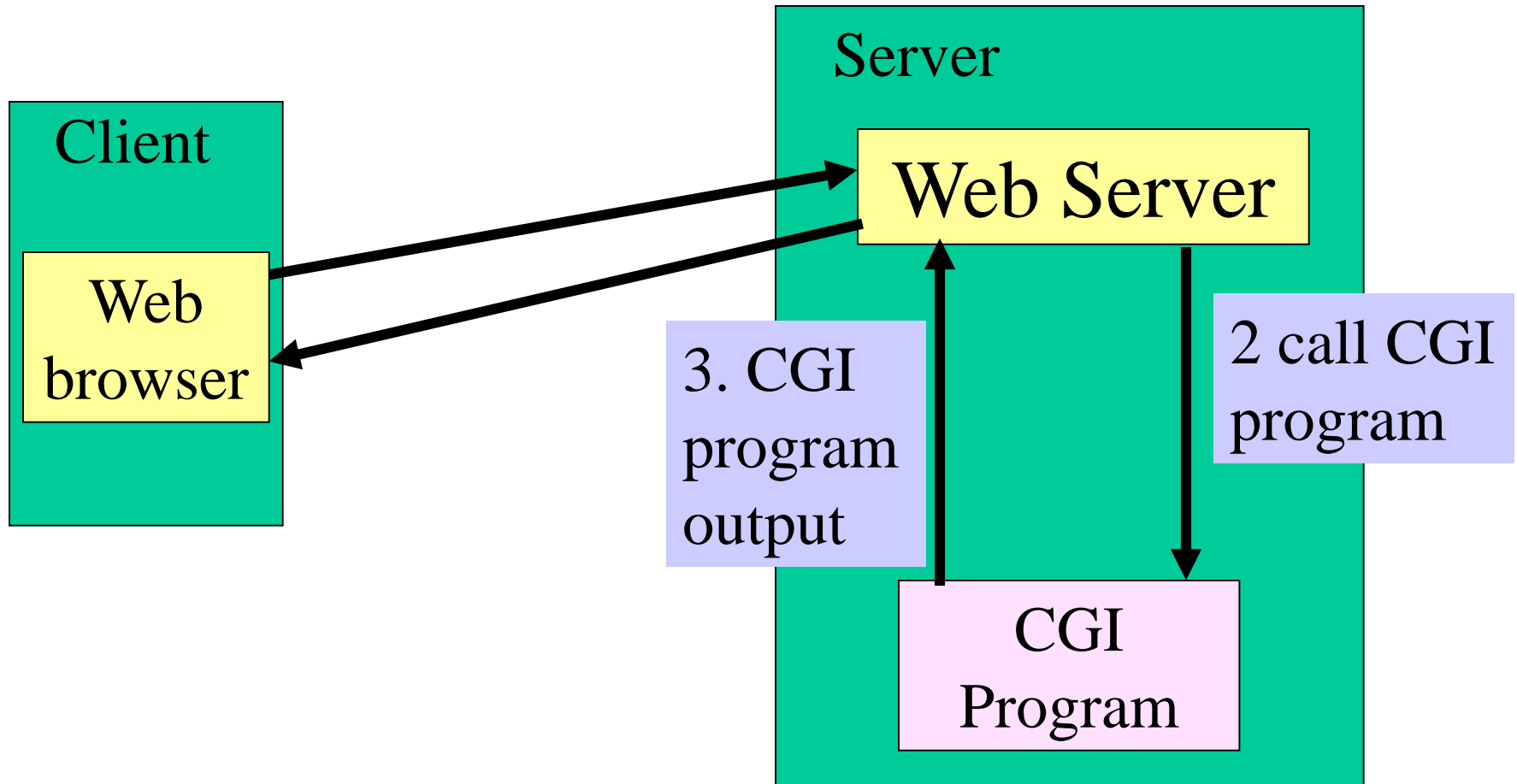
Attribute **required** results in automatic check that data are entered in appropriate format

Attribute **autocomplete** can be used to specify whether the browser should remember and suggest previously filled in data (set this to **"off"** for sensitive data like credit card numbers!)

CGI – Common Gateway Interface

A method of running programs on the server

CGI – Common Gateway Interface



CGI with Python

The Python program can access the values of all data items from the form

- with `cgi.FieldStorage()` function

```
form = cgi.FieldStorage()  
num1 = form.getvalue('theNumber')
```

Where **theNumber** is the name of a data item as specified with the **name** attribute, i.e. **name="theNumber"**

It returns a string – *so if you are expecting a number convert the string to appropriate type*

The Python program generates a new web page by writing HTML with the **print** statement

```
<!DOCTYPE html>
<html lang="en" >
<head>
  <meta charset="utf-8" >
  <title>Test CGI Python number squared</title>
</head>
<body>
  <form action="cgi-bin/NumSquared.py" >
    <label for="theNumber">Enter a number
      to be squared</label>
    <input type="text" id="theNumber"
      name="theNumber" /> <br />
    <input type="submit" value="submit" />
  </form>
</body>
</html>
```

Python CGI

NumSquared.py

```
#!/usr/bin/python3
import cgi, cgitb
form = cgi.FieldStorage()
num1 = form.getvalue('theNumber')
squared = float(num1)**2

print('Content-Type: text/html; charset=utf-8')
print('')
print('<!DOCTYPE html>')
print('<html>')
print('<head> <title> Python script to output the square  
of a number </title> </head>')
print('<body>')
print('<p>')
print('The square of %s is %g' % (num1, squared))
print('</p>')
print('</body>')
print('</html>')
```


Inside the <head>

Elements in the <head> section include

- **<title>**, **<base>**, **<link>**, **<meta>**, **<script>**, **<style>**

Example of <meta>:

```
<meta name="keywords"  
      content="desks, chairs, pens" />
```

Name attribute can have other values such as
“author”, “description”, date-last-modified...

MIME file types

MIME – Multipurpose Internet Mail Extension
= Internet Media Type

Examples : type/subtype

text e.g. `text/html`, `text/plain`,
`text/javascript`, `text/css`

image e.g. `image/gif`, `image/jpeg`, `image/png`

video e.g. `video/mpeg`, `video/mp4`, `video/ogg`

audio e.g. `audio/mpeg`, `audio/ogg`, `audio/mp4`

HTML Validation

Check validity of your document

<http://validator.w3.org/>

(be aware, however, that not *all* errors might be detected)

(also, distinguish *errors* from *warnings*)

More Information

book Rob Larsen “Beginning HTML and CSS”
(Chapter 6)

online tutorials at <http://w3schools.com>
(seach for “HTML forms”)

using Python for CGI:
<http://docs.python.org/3/library/cgi.html>

lots of detail about web technology and the
standards at <http://www.w3.org/>