SWINBURNE UNIVERSITY OF TECHNOLOGY

COS10005 Web Development

Module 3 – HTML Part 2

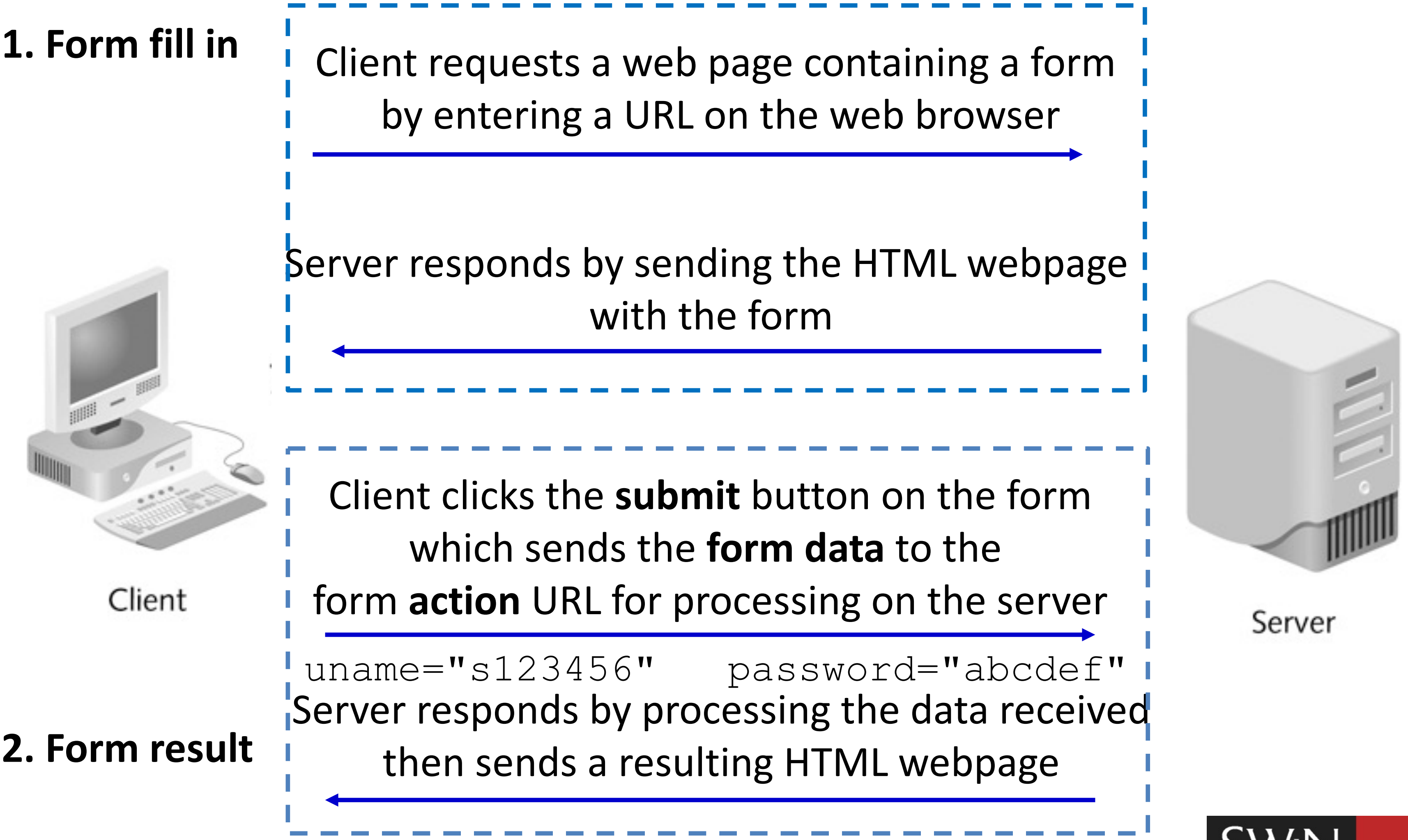
Contents

- HTML Elements -- continued
  - Forms
    - Form Element
    - Form Control Elements
- Website Development Process
  - Information Design
    - Folder Structure
    - Website Structure
- Website Application Environment
- Website Layout

Form Element

- `<form>` … `<form>` provides a mechanism to allow a user to enter information into a web page.
- Entered information can be submitted to a server, which takes actions upon receipt of the information.
- Possible actions include, but are not limited to:
  - Verifying the received information;
  - Retrieving data from database based on the received information;
  - Generating a web page and sending it back to the user;
  - Adding data to a database.

Form Element (continued)



1. Form fill in

Client requests a web page containing a form by entering a URL on the web browser

Server responds by sending the HTML webpage with the form

Client clicks the **submit** button on the form which sends the **form data** to the form **action** URL for processing on the server

```
uname="s123456"   password="abcdef"
```

Server responds by processing the data received then sends a resulting HTML webpage

2. Form result

Form Data

- Form data are submitted in the form of parameter name-value pairs

  parameterName = parameterValue

  – E.g., username = "s123456"

    password = "abcdef“

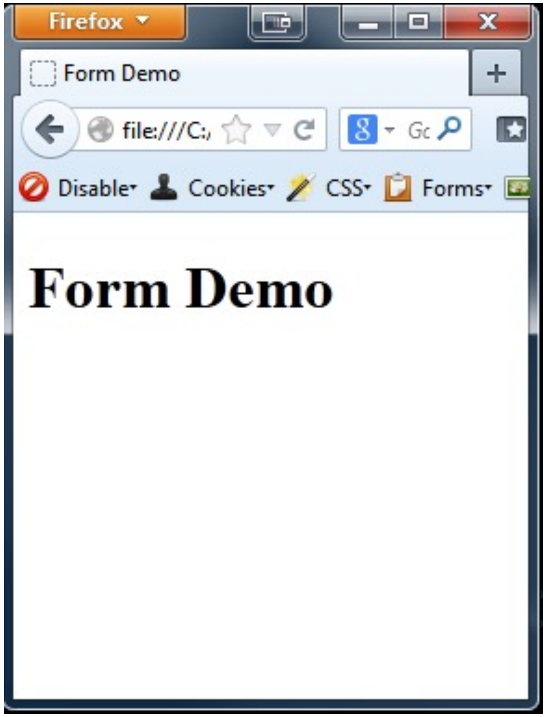
    gender = "female"

- Multiple such pairs can be sent in one submission to the server

`<form>` Attributes

```
<form
   id="survey"
   method="post" action="process.php">
   <!-- Form control elements here -->
</form>
```

Absolute path is used if processing is on a different server.



Form Demo

Usually the `<form>` element contains ***form control elements*** and form structuring elements.

The form will not be displayed or available, unless there are ***form control elements***.

`<form>` Attributes (continued)

- *action* - An URL referring to where the data is to be submitted for processing
- *method* - HTTP method used to submit the form – **get** or **post**
  - **get** is often used to read/retrieve data from a server to obtain something, e.g., search, or see a product (URL is visible in the browser)
  - **post** is often used to submit data for storage e.g. registration (URL is not visible in the browser)

Contents

- HTML Elements -- continued
  - Forms
    - Form Element
    - Form Control Elements
- Website Development Process
  - Information Design
    - Folder Structure
    - Website Structure
  - Website Application Environment
  - Website Layout

Form Control Elements

Common Form Control elements:

- `<input>` defines a ***form control element*** for the user to enter ***data***. Different ***input elements*** can be displayed ***based on*** the ***`type`*** *attribute*. Its possible values include: **`text, checkbox, radio, password, submit, reset, hidden, file, image, button`**
- `<select>` defines a ***selection of options/list*** and can have the following *attributes:* **`size, multiple, tabindex, disabled`**
- `<textarea>` defines a ***form control*** for the user to enter ***multi-line text input*** and can have the following *attributes:* **`rows, cols, readonly, tabindex, accesskey, disabled`**

Form Control Elements (Label)

- `<label>`…`</label>`

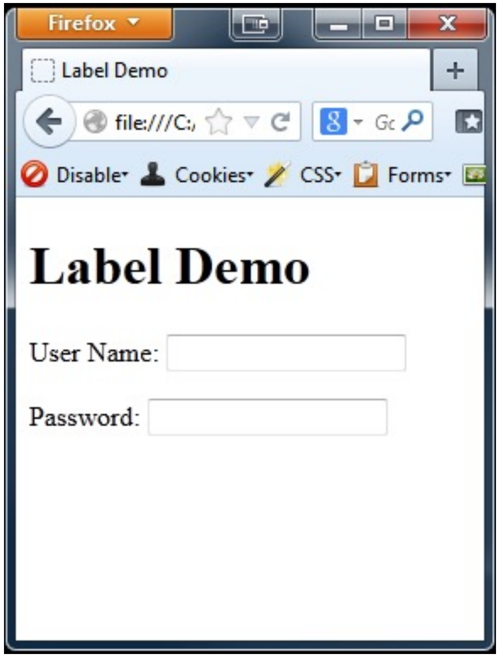
  associates a **label** with a **form control element.**

- The label element attributes can associate a **label** with a **form control element,** e.g., `for="element-id"`.
  - It allows users to clicking on a label to select the associated control element.

Form Control Elements (Label)

- Example

```
<form action="" method="post">
  <p><label for="tbUserName">User Name:</label>
     <input type="text" id="tbUserName" /></p>
  <p><label for="tbPassword">Password:</label>
     <input type="password" id="tbPassword" /></p>
</form>
```



Label Demo

User Name:

Password:

Form Control Elements (Input)

- `<input … >` ***Note: void element***
  - defines a form control element for users to enter data.
- It can have the following attributes:

  `type, name, value, id`
- The ***type*** attribute specifies the type of the input element, including:
  - text
  - checkbox
  - radio
  - password
  - submit
  - reset
  - hidden
  - file
  - image
  - button

Form Control Elements (Input)

```
<p><label>Name</label>
 <input type="text" name="fname" maxlength="20"
                         size="20">
</p>
<p><label>Age</label>
 <input type="text" name="age" maxlength="2"
                     size="2" >
</p>
```

DEMO!

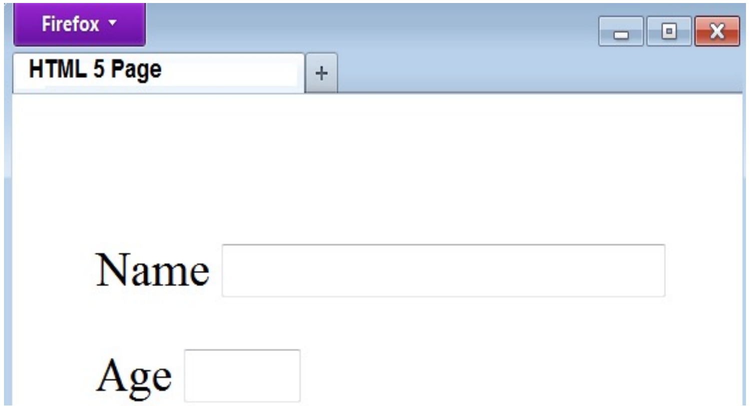
If **type** is not included, or is unidentified, type="text" is assumed.

type="text" is used for both text and numbers

name attribute is used to pass data for form processing

maxlength specify the maximum number of characters allowed

size sets the visible width of the text box



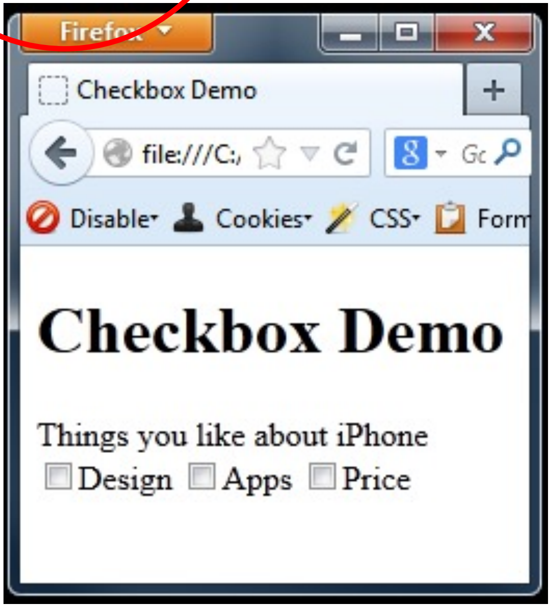
Name

Age

```
Data to send: fname=?   age=?
```

Form Control Elements (Checkbox)

```
<p>Things you like about iPhone <br />
  <input type="checkbox" name="cbDesign"
       value="design" >Design
  <input type="checkbox" name="cbApps"
       value="apps" >Apps
  <input type="checkbox" name="cbPrice"
       value="price" >Price
</p>
```



Checkbox Demo

Things you like about iPhone

□ Design □ Apps □ Price

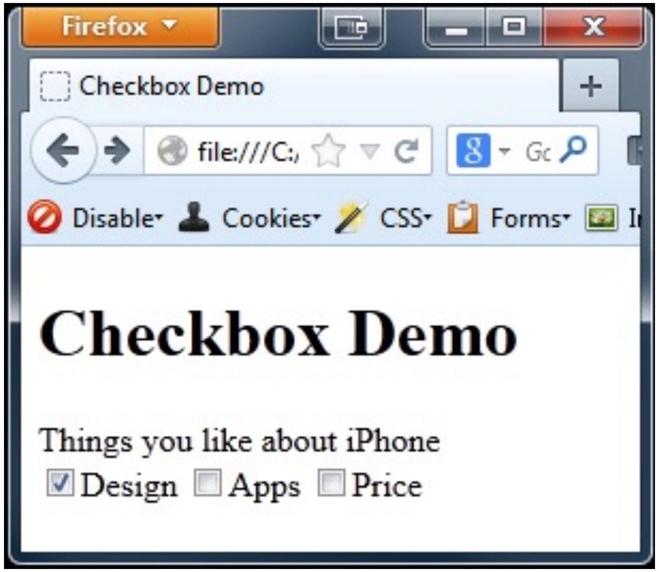
`<fieldset>` and `<legend>` elements are usually used to group the checklist

```
Data to send: cbDesign="design"
AND/OR cbApps="apps" AND/OR cbPrice="price"
```

Form Control Elements (Checkbox)

```
<p>Things you like about iPhone <br>
  <input type="checkbox" name="cbDesign"
       value="design" checked="checked" >Design
  <input type="checkbox" name="cbApps"
       value="apps" >Apps
  <input type="checkbox" name="cbPrice"
       value="price" >Price
</p>
```

The *checked* attribute is used to initialise a checked checkbox by default.



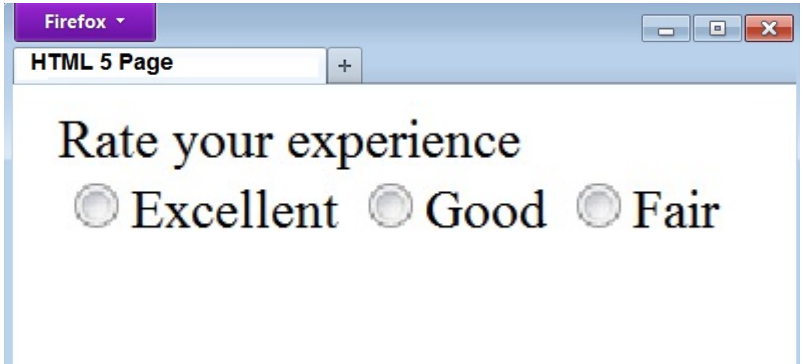
Checkbox Demo

Things you like about iPhone
☑ Design □ Apps □ Price

Form Control Elements (Radio Button)

```
<p>Rate your experience<br >
   <input type="radio" name="rbRating"
       value="Exel" >Excellent
   <input type="radio" name="rbRating"
       value="Good" >Good
   <input type="radio" name="rbRating"
       value="Fair" >Fair
</p>
```

Note that only one choice is allowed. ***Thus, the names of those radio buttons must be the same.***



Rate your experience

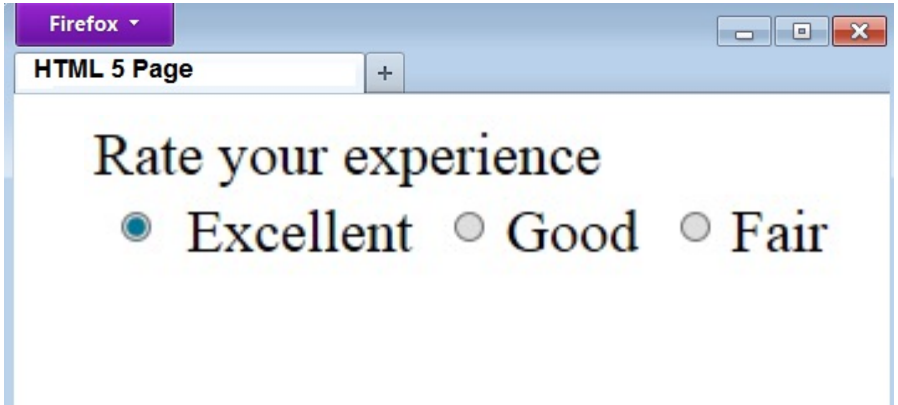
○ Excellent ○ Good ○ Fair

```
Data to send: rbRating="Exel" OR rbRating="Good"
OR rbRating="Fair"
```

Form Control Elements (Radio Button)

```
<p>Rate your experience<br >
   <input type="radio" name="rating"
       value="Exel" checked="checked" >
       Excellent
   <input type="radio" name="rating"
       value="Good" >Good
   <input type="radio" name="rating"
       value="fair" >Fair</p>
```

The ***checked*** attribute is used to check a radio by default when the web page is loaded.



Rate your experience

● Excellent ○ Good ○ Fair

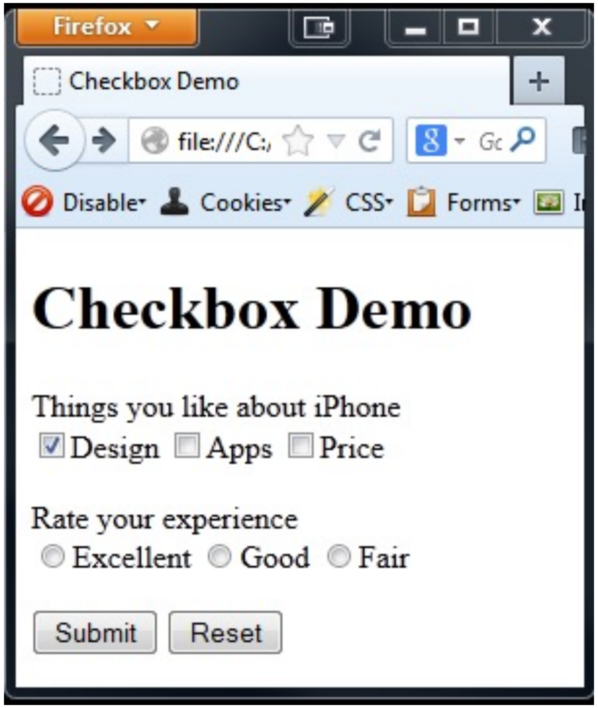
Form Control Elements (Submit Button)

```
<p>
 <input type="submit" value="Submit" >
 <input type="reset" value="Reset" >
</p>
```

Texts to be displayed on buttons

Make sure that your form has an input of type submit.

Note: reset means set all input form fields to their initial values.



Checkbox Demo

Things you like about iPhone
☑ Design □ Apps □ Price

Rate your experience
○ Excellent ○ Good ○ Fair

Form Control Elements (Select & Option)

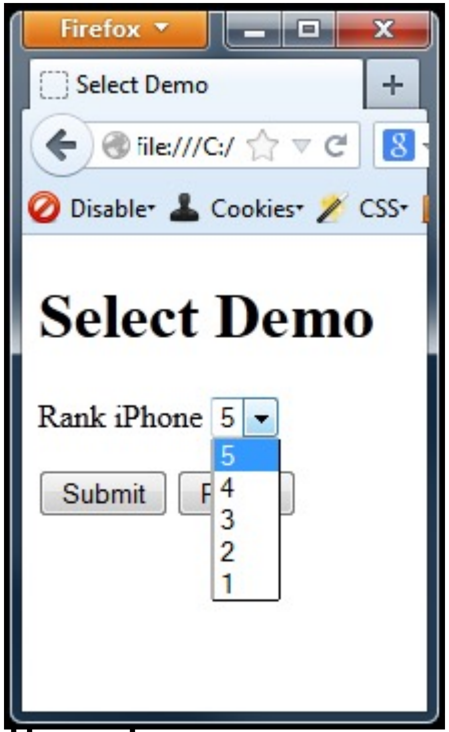
- `<select>`…`</select>` defines a form control for the selection of options from a selection list
- It can have the following attributes:

  multiple, *disabled*

- The ***selected*** attribute sets the default selected value
- Options are listed by using
  - <option> … </option>

Form Control Elements (Select & Option)

```
<p><label>Rank iPhone</label>
  <select name="rank">
    <option value="5" selected>5<
    <option value="4">4</option>
    <option value="3">3</option>
    <option value="2">2</option>
    <option value="1">1</option>
  </select>
</p>
```



Select Demo

Rank iPhone 5

5
4
3
2
1

The ***selected*** attribute is used to initialise a default option.

```
Data to send: rank="1" OR rank="2" OR rank="3" OR
rank="4" OR rank="5"
```

Forms – Watch out!

Checkboxes

```
<input type="checkbox" name="cbname1" value="value1" >
<input type="checkbox" name="cbname2" value="value2" >
```

Checkboxes in a group usually have ***different values for their name attributes, and different values for their value attributes.***

Radio Buttons

```
<input type="radio" name="rbname" value="value1" >
<input type="radio" name="rbname" value="value2" >
```

Radio buttons in a group are mutually exclusive – one checked, all others unchecked. They have ***the same values for their name attribute, but different values for their value attributes.***

Forms – Watch out!

- Errors in *Form Control elements* may lead to data *errors*

```
<select name="rank">
    <option value="5">5</option>
    <option value="4">4</option>
    <option value="3">3</option>
    <option value="2">2</option>
    <option value="1">1</option>
</select>
```

select** and **option *(dropdown box)*

*Only the select element has the **name** attribute.*

*The option elements in the same group usually have **different values for their value attributes**.*

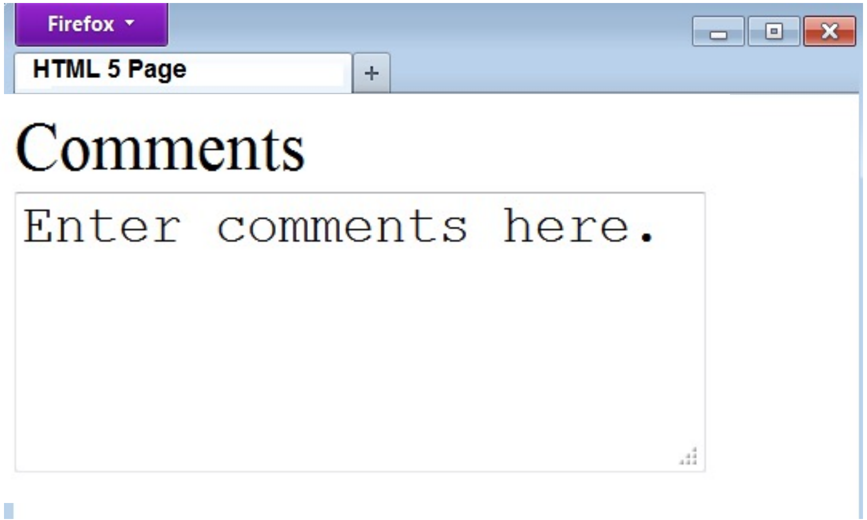
Form Control Elements (Text Area)

- `<textarea>`…`</textarea>` defines a form control for the user to enter multi-line text input
- It can have the following attributes: rows, cols, readonly, disabled
- Example

```
<textarea readonly="readonly">
<textarea disabled="disabled">
```

Form Control Elements (Text Area)

```
<p><label>Comments</label><br >
  <textarea name="comments" rows="4" cols="20">
Enter comments here.
  </textarea>
</p>
```



Comments

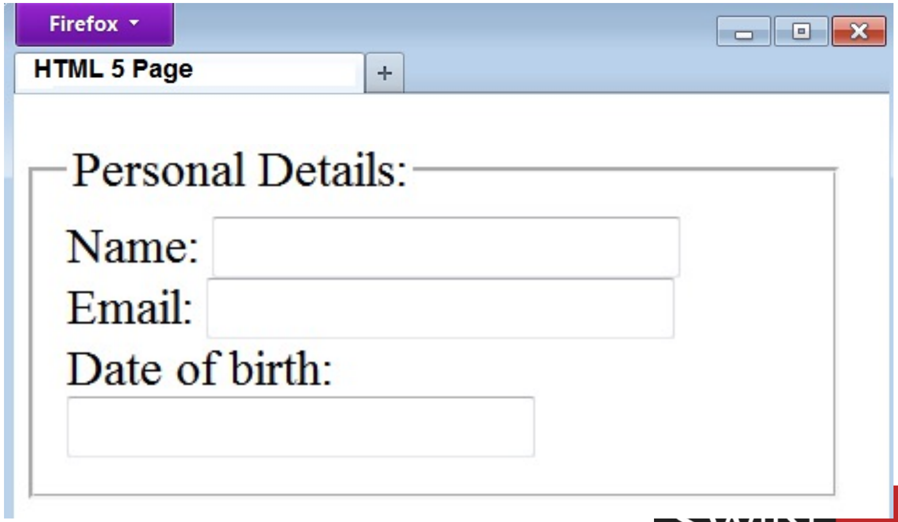
Enter comments here.

Form Elements (Fieldset & Legend)

- **`<fieldset>`**…**`</fieldset>`**
  - Used for ***grouping*** related form controls.
  - Enables authors to divide a form into smaller, more manageable parts, improving the usability of the form.
  - Draws a box around the related elements.
- **`<legend>`**…**`</legend>`**
  - Defines a ***caption*** for a `<fieldset>`
  - Must be at the start of a `<fieldset>` element, before any other elements.

Form Elements (Fieldset & Legend)

```
<fieldset>
  <legend>Personal Details:</legend>
  <label>Name:</label>
  <input type="text" name="fname" >
  <label>Email:</label>
  <input type="text" name="email" >
  <label>Date of birth:</label>
  <input type="text" name="dob" >
</fieldset>
```



Personal Details:

Name:

Email:

Date of birth:

Forms – How do they work?

- The `form` element ***must*** have an *action* attribute and value. It specifies where the form data will be submitted.
- A form ***must*** contain an `<input type="submit">`
  - When the submit button is clicked, or the ‘enter’ key is pressed, the form is ‘actioned’.
- Form control elements for data collecting ***must*** have *name* attributes.
  - These **names** are paired with user entered attribute **values** and then sent as **“name=value”** data pairs to the server.

HTML5 FORM ELEMENTS

SWINBURNE UNIVERSITY OF TECHNOLOGY

HTML5 Form Elements

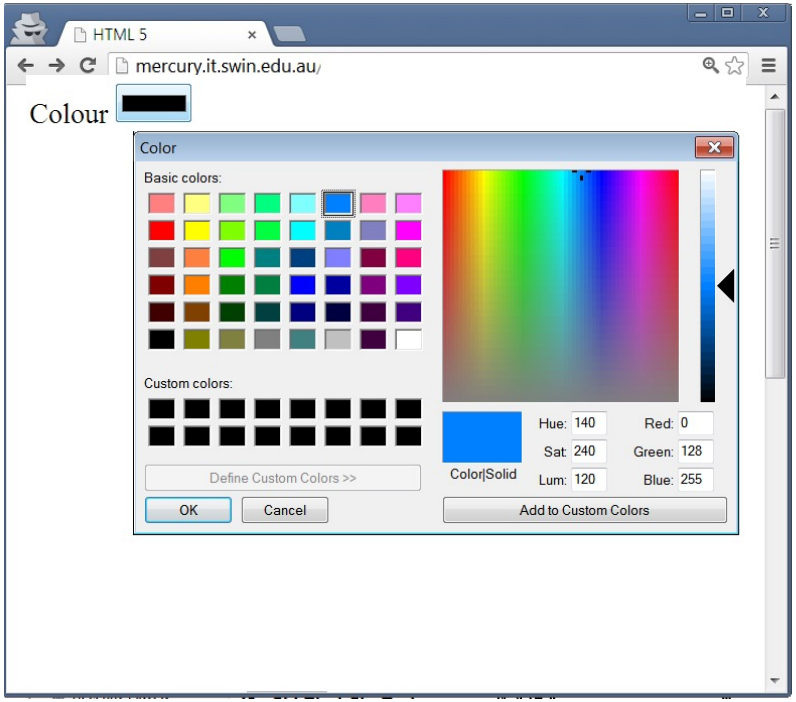
- HTML5 introduces new form <input ...> **type.** ***Note that these are not yet universally supported by all browsers.***
  - color
  - date
  - datetime
  - email
  - month
  - number
  - range
  - search
  - tel
  - time
  - url
  - week
- New attributes include: autofocus, placeholder, pattern, required

HTML5 Form Elements (Colour)

```
<p><label>Colour</label>
   <input type="color" name="favcolor"
      autofocus="autofocus" >
</p>
```

The *autofocus* attribute defines which input element should have the default cursor position.

There can only be one input element with *autofocus*. If there is more than 1 the **last** instance gets the focus.

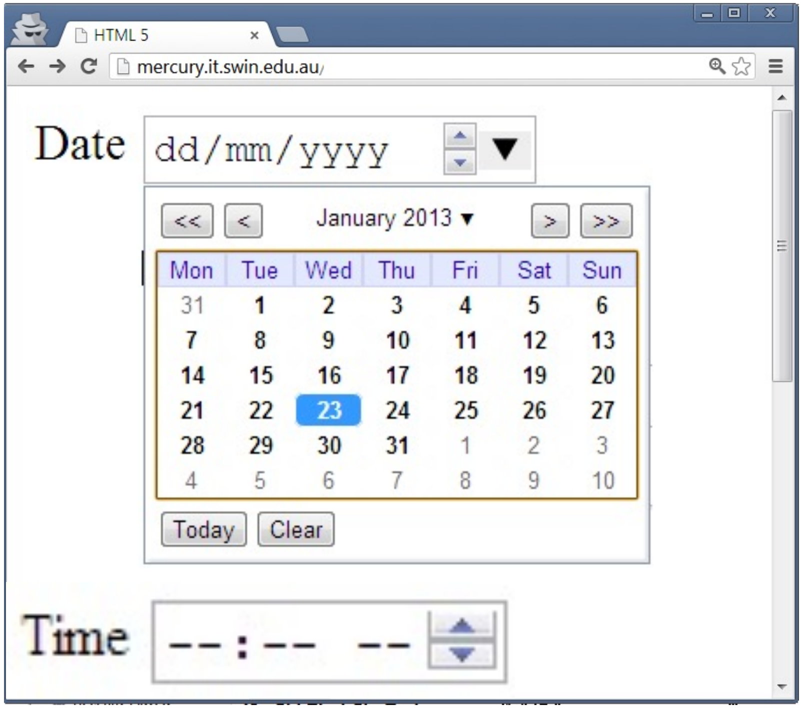


Hue: 140
Sat: 240
Lum: 120
Red: 0
Green: 128
Blue: 255

HTML5 Form Elements (Date)

```
<p><label>Date</label>
   <input type="date" name="date" >
</p>

<p><label>Time</label>
   <input type="time"
      name="time" >
</p>
```



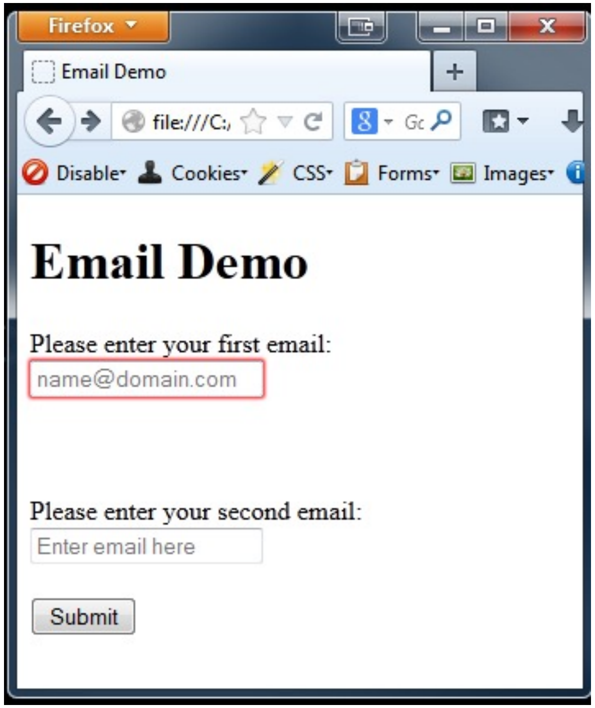
Mon	Tue	Wed	Thu	Fri	Sat	Sun
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10

HTML5 Form Elements (Email)

```
<p><label>Email</label>
   <input type="email" name="emailContact"
     placeholder="name@domain.com" required=
"required" >
</p>
```

The ***required*** attribute indicates that email field must be filled prior to submission.

The ***placeholder*** attribute specifies a hint that describes the expected value of an input field.



Email Demo

Please enter your first email:

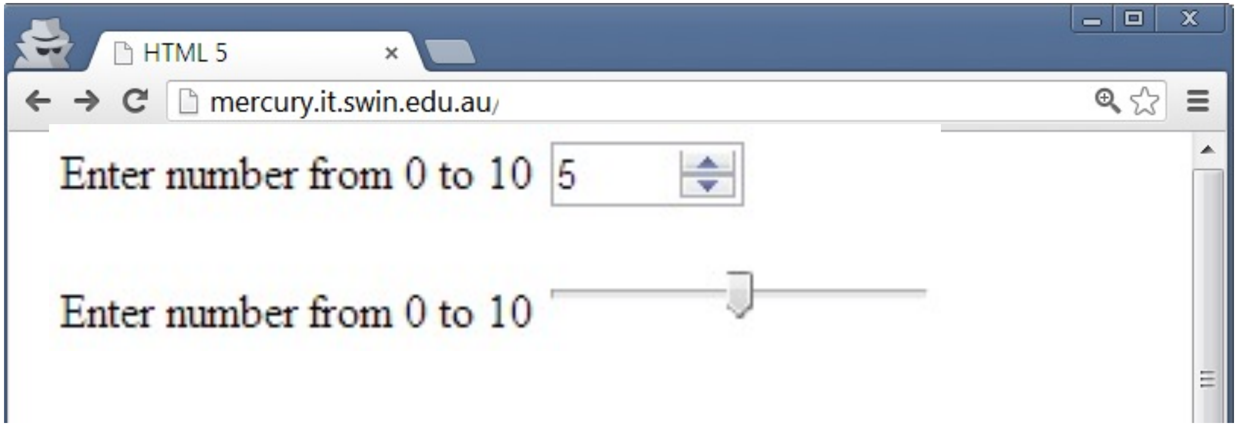
name@domain.com

Please enter your second email:

Enter email here

HTML5 Form Elements (Number)

```
<p><label>Enter number from 0 to 10</label>
   <input type="number" name="nbScore" min="0"
           max="10" step="1" value="5" >
</p>
<p><label>Enter number from 0 to 10</label>
   <input type="range" name="rgRating" min="0"
           max="10" value="5" >
</p>
```

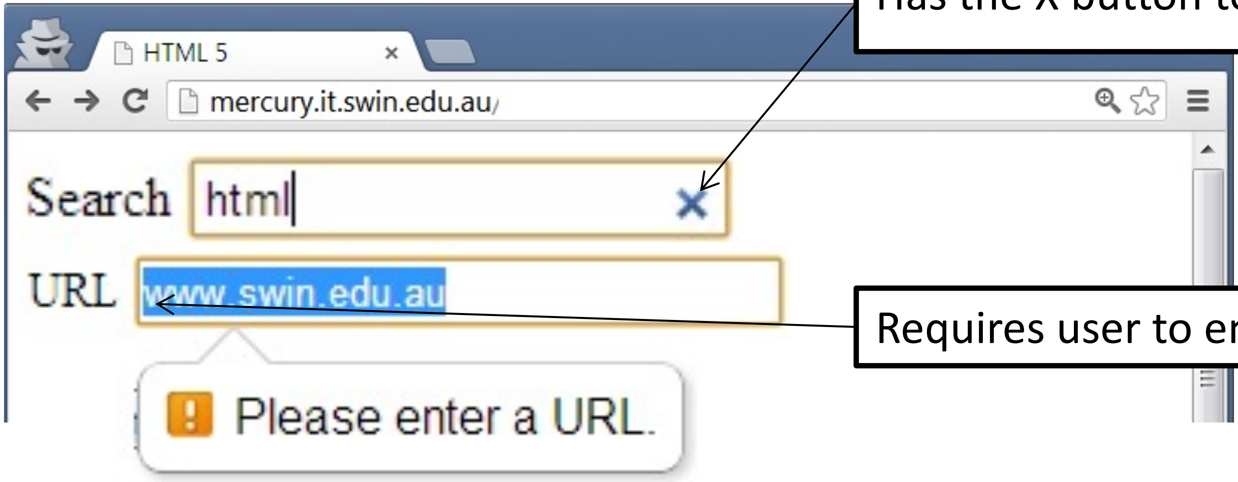


Enter number from 0 to 10 5

Enter number from 0 to 10

HTML5 Form Elements (Search and URL)

```
<p><label>Search
   <input type="search" name="scQuery"
    placeholder="search query" >
</label></p>
<p><label>URL</label>
   <input type="url" name="urlWebsite"
    placeholder="http://www.domainname.au" >
</p>
```



Search html

URL www.swin.edu.au

Please enter a URL.

Has the X button to clear

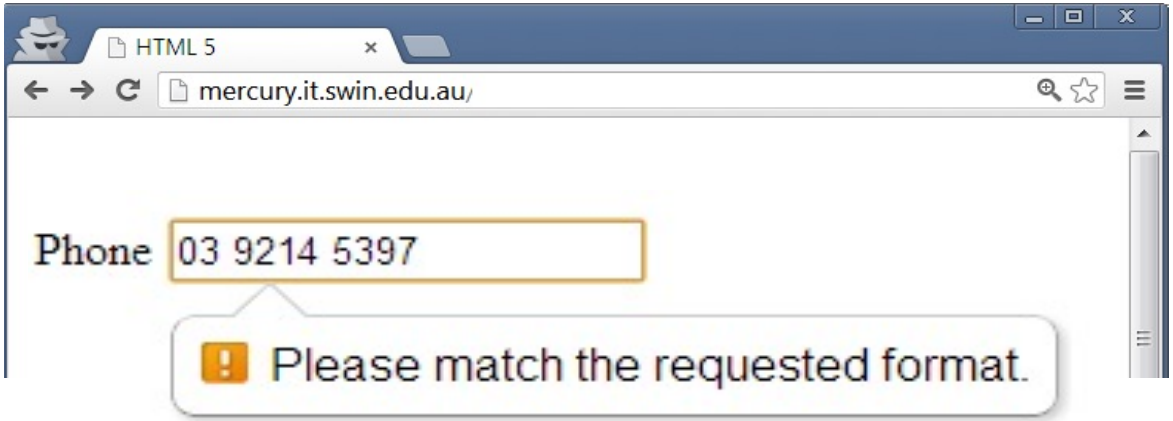
Requires user to encode http://

HTML5 Form Elements (Phone)

```
<p><label>Phone</label>
   <input type="tel" name="phone‶
      placeholder="(##) ####-####"
      pattern="\(\d{2}\) +\d{4}-\d{4}" >
</p>
```

The ***pattern*** attribute specifies a ***regular expression*** that the <input> element's value is checked against.

It works with the following input types: text, search, url, tel, email, and password.



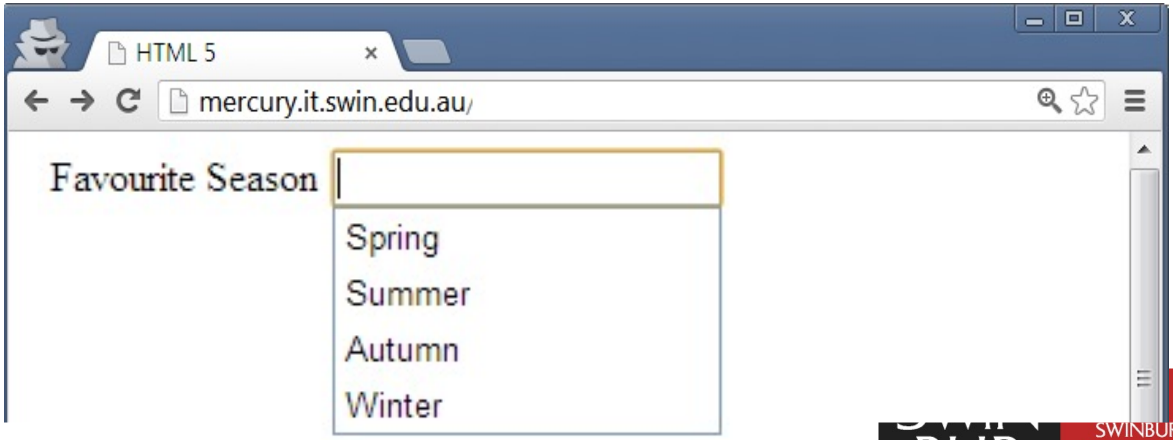
Phone 03 9214 5397

Please match the requested format.

HTML5 Form Elements (Data List)

```
<p><label>Favourite Season</label>
   <input list="dlSeasons"
      name="favseason" >
   <datalist id="dlSeasons">
     <option value = "Spring">
     <option value = "Summer">
     <option value = "Autumn">
     <option value = "Winter">
   </datalist>
</p>
```

Make sure the *list* attribute matches the *id* attribute of the list.



Favourite Season

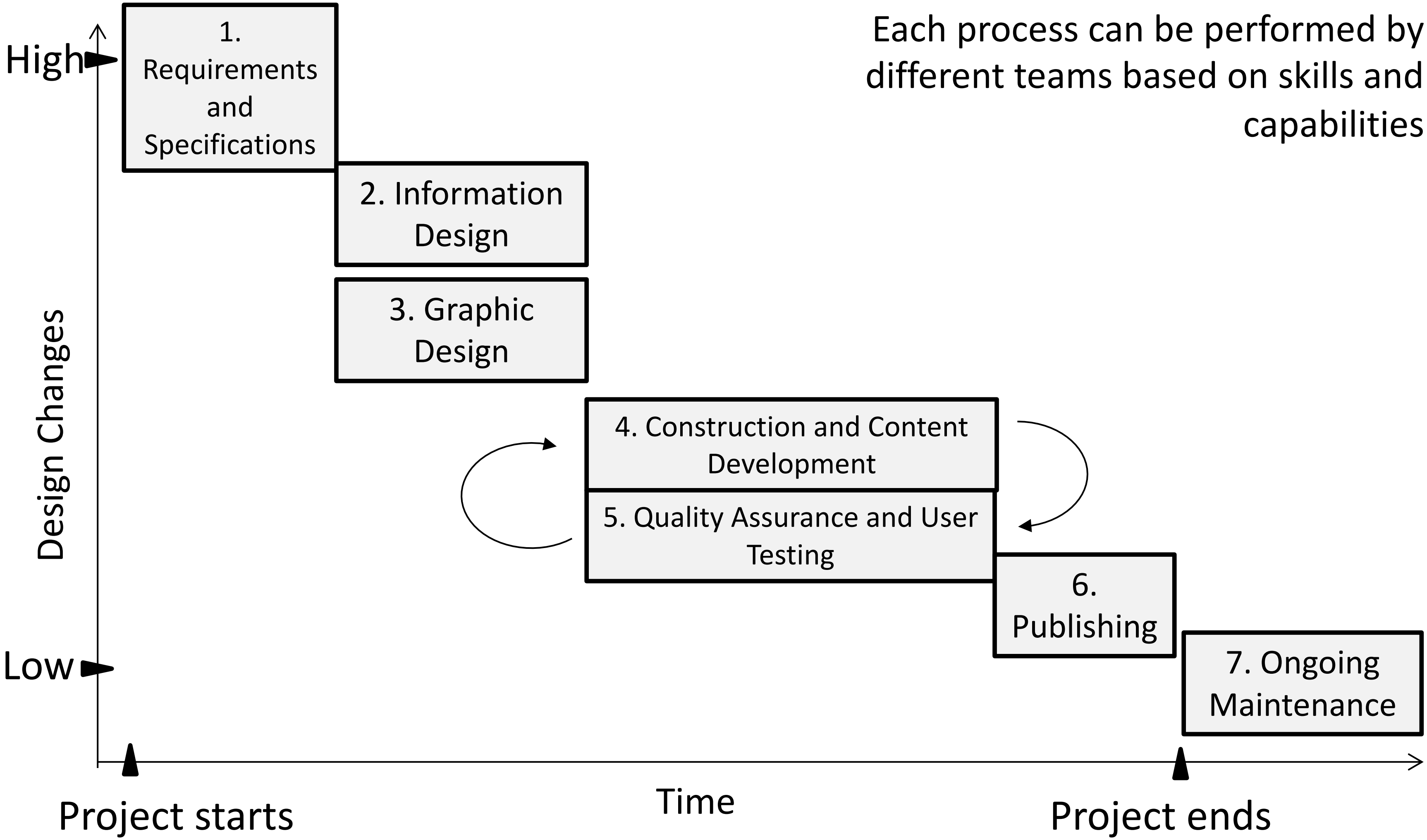
Spring
Summer
Autumn
Winter

Contents

- HTML Elements -- continued
  - Forms
    - Form Element
    - Form Control Elements
- Website Development Process
  - Project Life Cycle
  - Information Design
    - Folder Structure
    - Website Structure
  - Website Application Environment
  - Website Layout

Process: Project Life Cycle (continued)

Each process can be performed by different teams based on skills and capabilities



1. Requirements and Specifications

- Establish the client's needs
  - Gain visibility or attract customer
  - Provide a service or sell a product
  - Create a community or disseminate information
- Determine requirements
  - Search capabilities, menu navigation
  - Colour and branding
- Analyse and assess viability

Process: Design

- 2. Information Design
  - – Set up a **directory structure** and create conventions for filenames and URLs
  - – Select an appropriate **website structure** that is meaningful and support user navigation
- 3. Graphic Design
  - – Understand the **web design environment**
  - – Design **page mock ups** for discussion
  - – Capture refined mock ups as **wireframes** for developers

Process: Construction and Testing

- 4. Construction
  - Coding and validation starts
  - Templates are established
  - Contents are encoded
- 5. Testing
  - Cross browser compatibility and connectivity at different bandwidths
  - Valid links, forms and multimedia resources
  - Accessibility to all users and usability tests

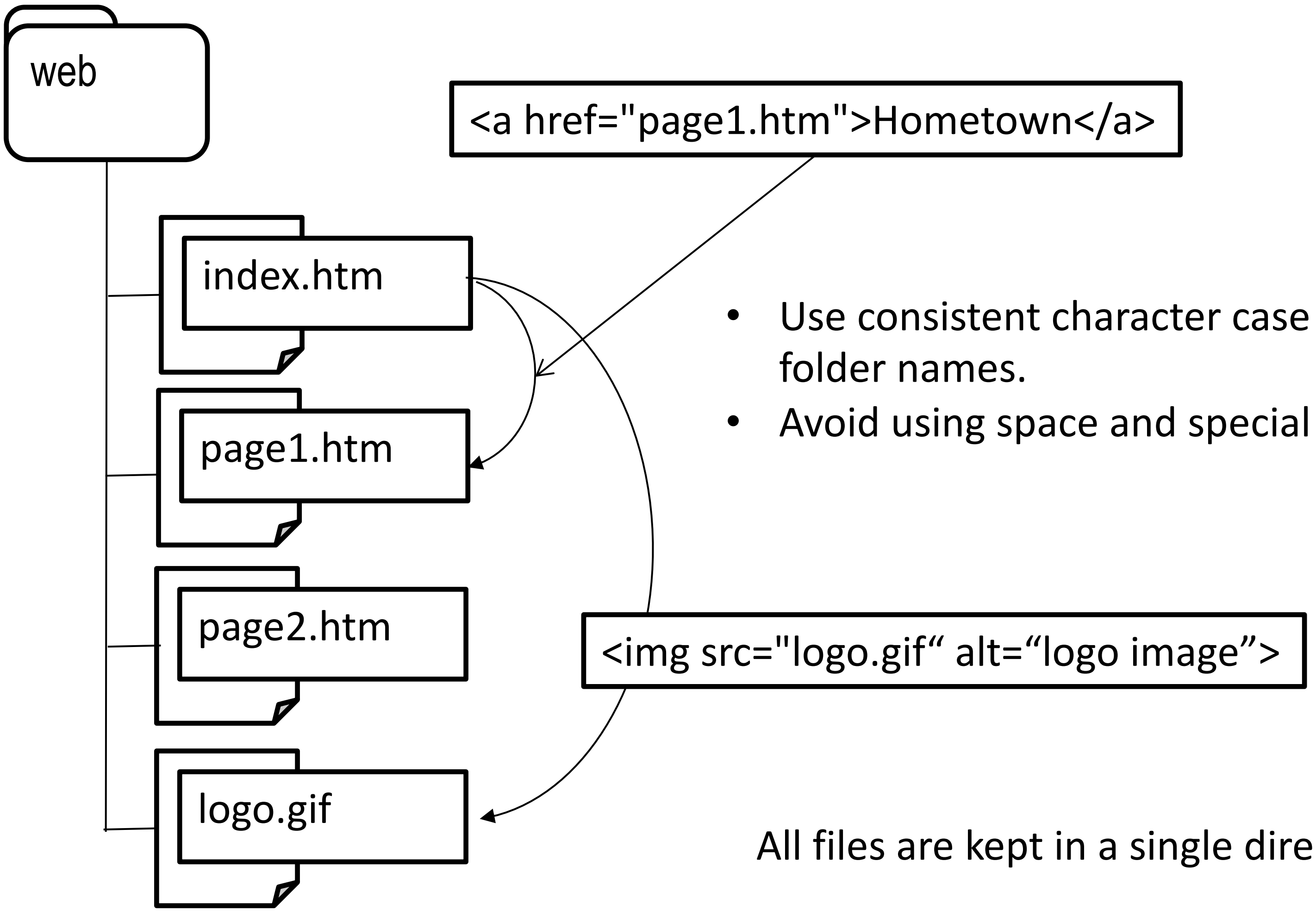
Process: Publishing and Maintenance

- 6. Publishing
  - Make the website known to the public
  - Registering with search engines
- 7. Maintenance
  - Ensure that the web **content** is updated
  - Restart project life cycle if new requirements are to be addressed

Contents

- HTML Elements -- continued
  - Forms
    - Form Element
    - Form Control Elements
- Website Development Process
  - Project Life Cycle
  - Information Design
    - Folder Structure
    - Website Structure
  - Website Application Environment
  - Website Layout

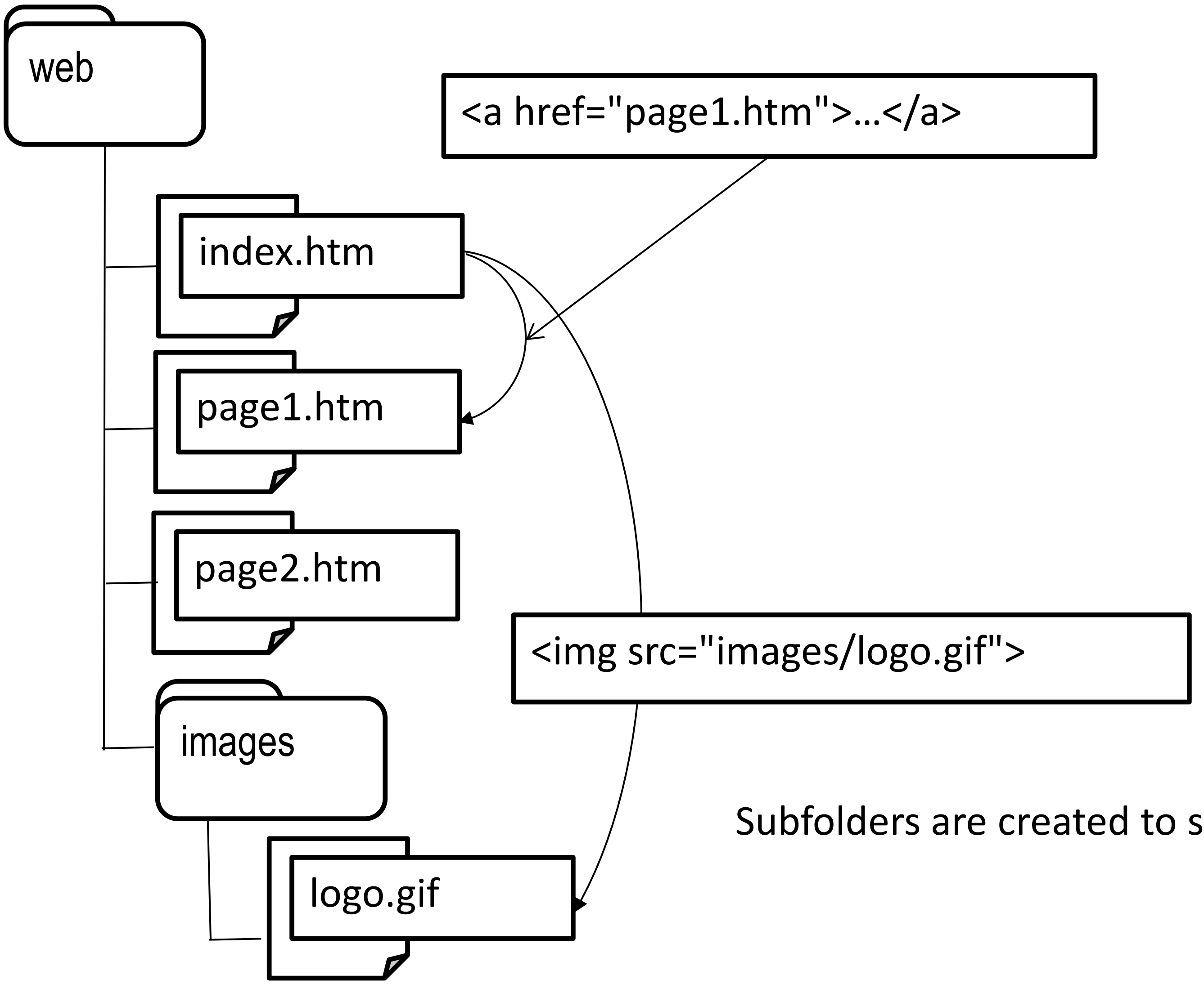
Folder Structure: Single Folder


web
index.htm
page1.htm
page2.htm
logo.gif
<a href="page1.htm">Hometown</a>
<img src="logo.gif“ alt=“logo image”>

- Use consistent character case for file and folder names.
- Avoid using space and special characters

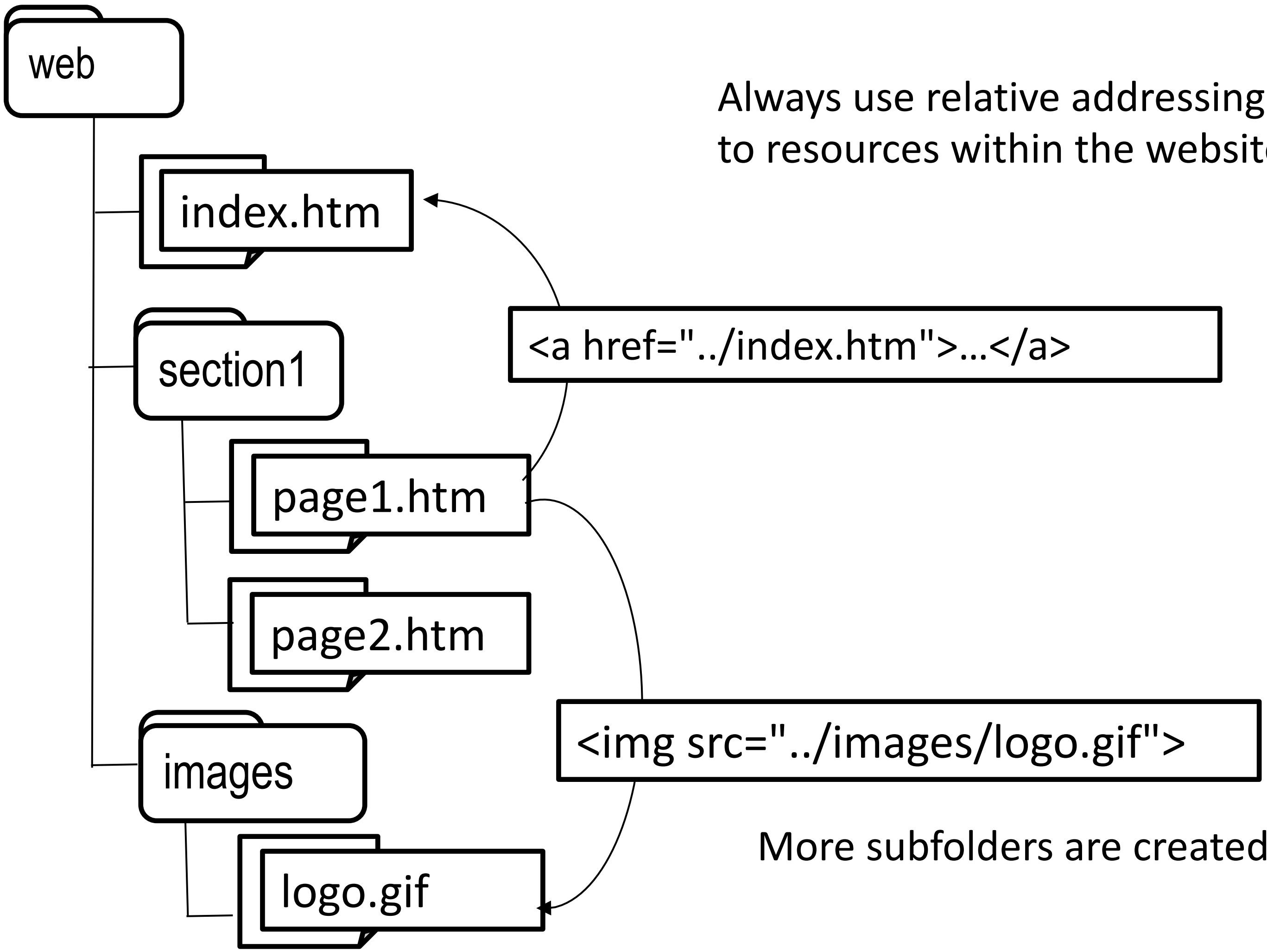
All files are kept in a single directory

Folder Structure: Hierarchical Folder



Subfolders are created to store image files.

Folder Structure: Hierarchical Folder



`<a href="../index.htm">...</a>`

`<img src="../images/logo.gif">`

Always use relative addressing when referring to resources within the website.

More subfolders are created.

Contents

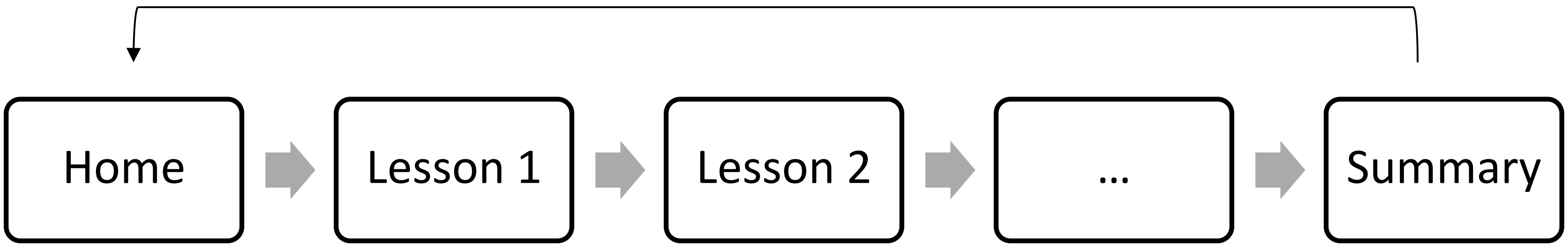
- HTML Elements -- continued
  - Forms
    - Form Element
    - Form Control Elements
- Website Development Process
  - Project Life Cycle
  - Information Design
    - Folder Structure
    - Website Structure
  - Website Application Environment
  - Website Layout

Website Organisation

- Organise the website based on the site's **content**
- Understand its effects on navigation
  - Folder structure, menu depth, navigation
  - Common information structure or website organisation
    - **Linear**
    - **Hierarchical**
    - **Network**

Structure: Linear

Some website uses linear structure in a small area, and will generally be using hierarchical organisation.



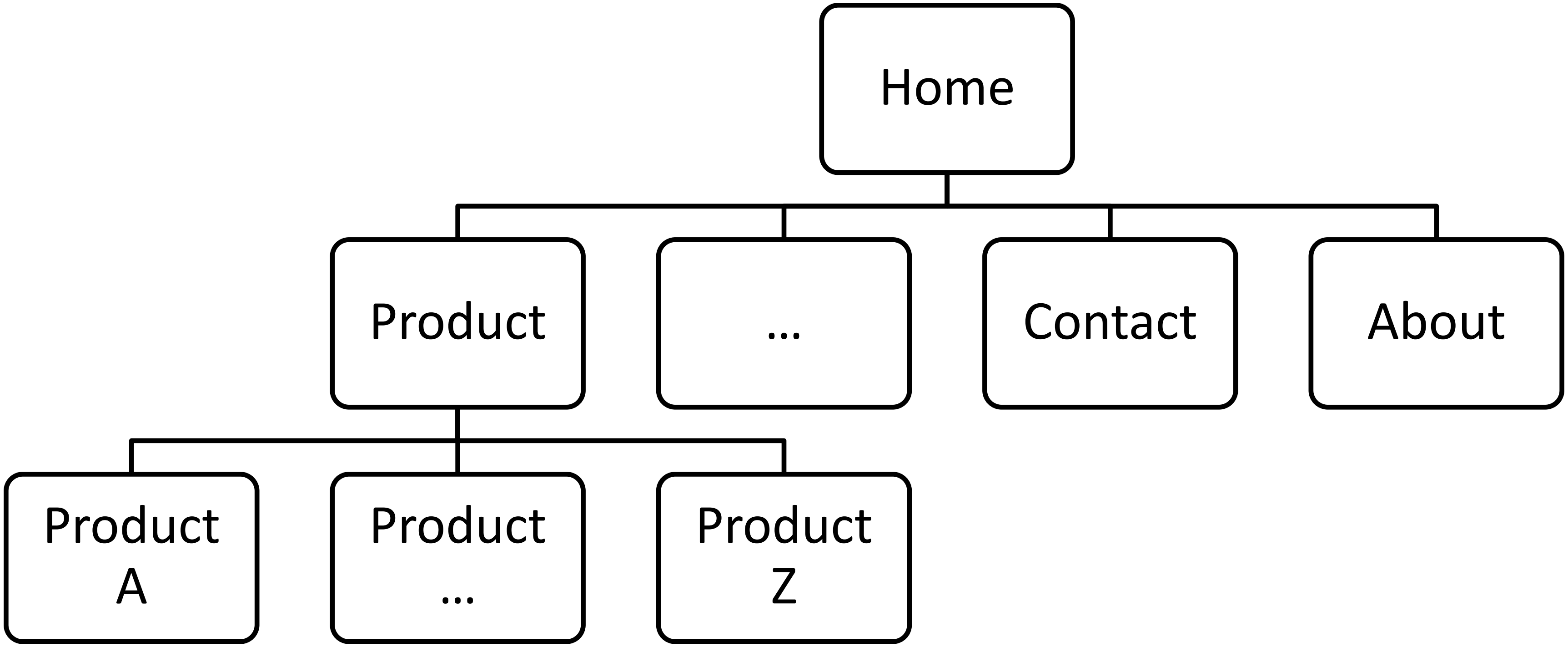
Site map for linear organisation

Structure: Linear (continued)

- Linear structure supports **forward** and **back** movement through a ***sequence*** of Web pages.
- This structure is suitable for describing **step-by-step** procedures, instructions or for dividing up text that is to be read sequentially, for example, online course and survey websites.
- Users will generally have no navigational difficulties however there should be an easy way to exit.

Structure: Hierarchical

Website generally use hierarchical organisation



Site map for hierarchical organisation

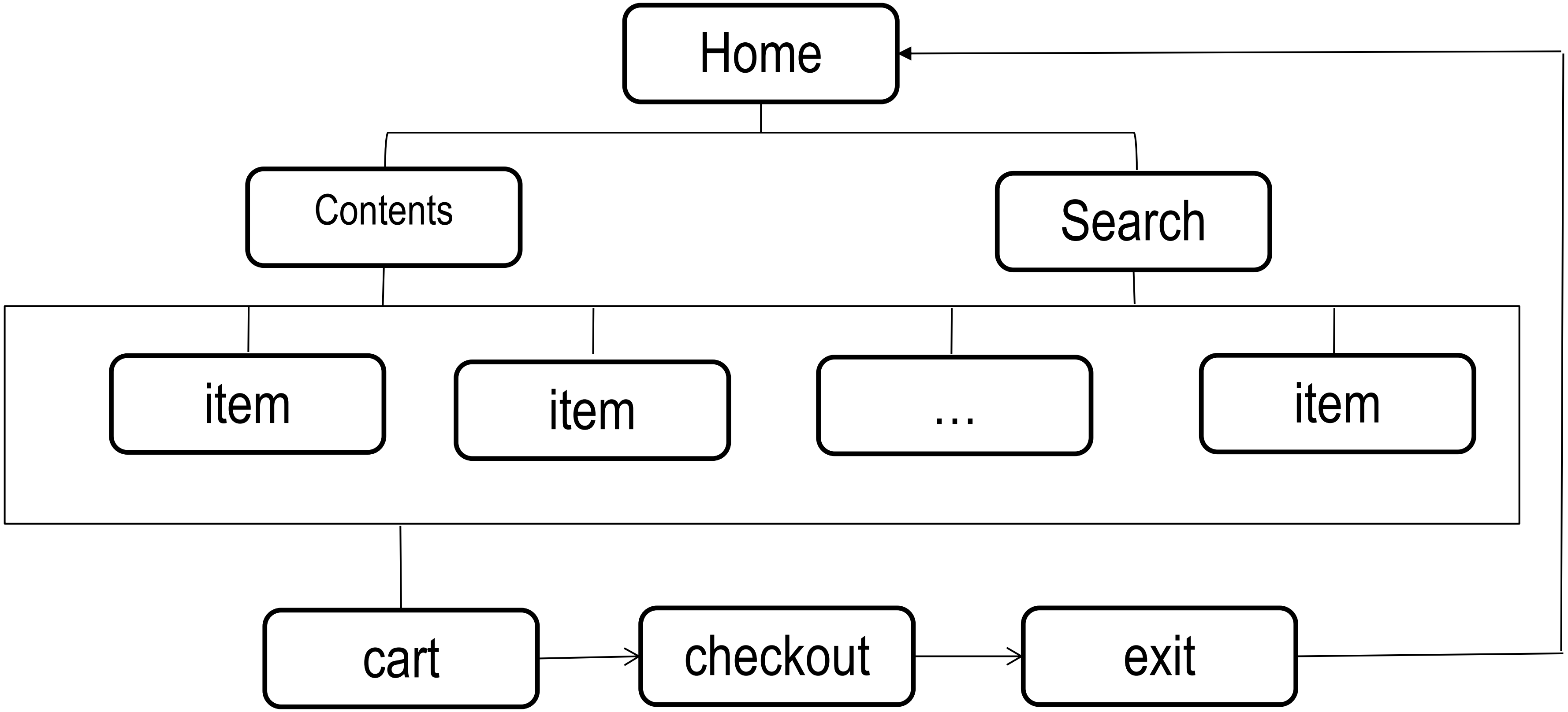
Structure: Hierarchical (continued)

- Hierarchical structure has an index page that contains links to other pages, which contain links to other pages
  - Users can navigate towards their desired information from top down.
- Usability studies suggest that **breadth** (or **"fanout"**) should be kept to less than **10** options, and depth less than **5** layers.
  - The **three click rule** is an unofficial web design rule which suggests that users should be able to find any information with no more than three mouse clicks. This is based on the belief that users become frustrated and often leave if they cannot find the information within the three clicks.
  - Usability studies considered this a **myth**.

http://uxmyths.com/post/654026581/myth-all-pages-should-be-accessible-in-3-clicks

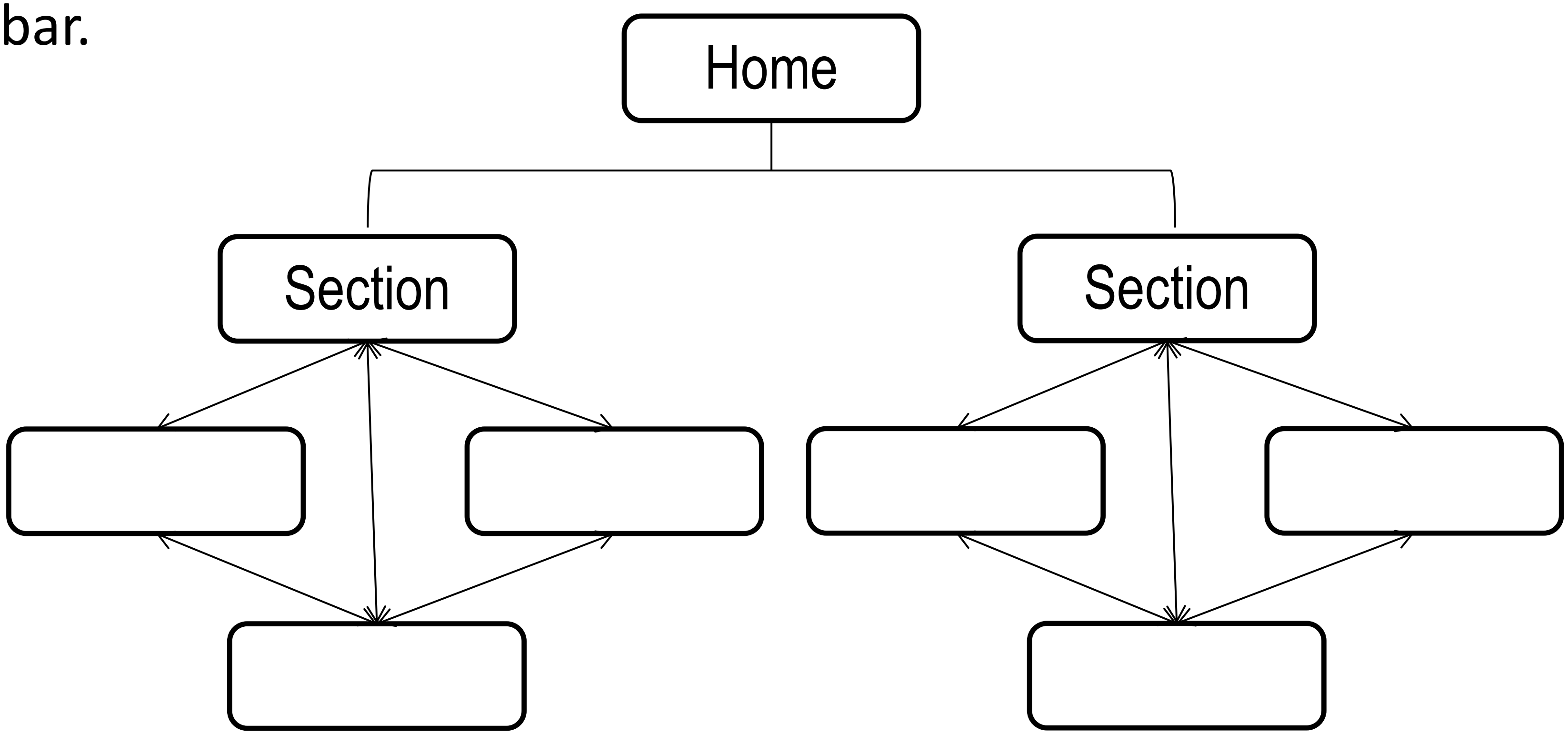
Structure: Network - Catalogue

Catalogue structure supports shopping cart system. Make sure all items include a clear navigation bar.



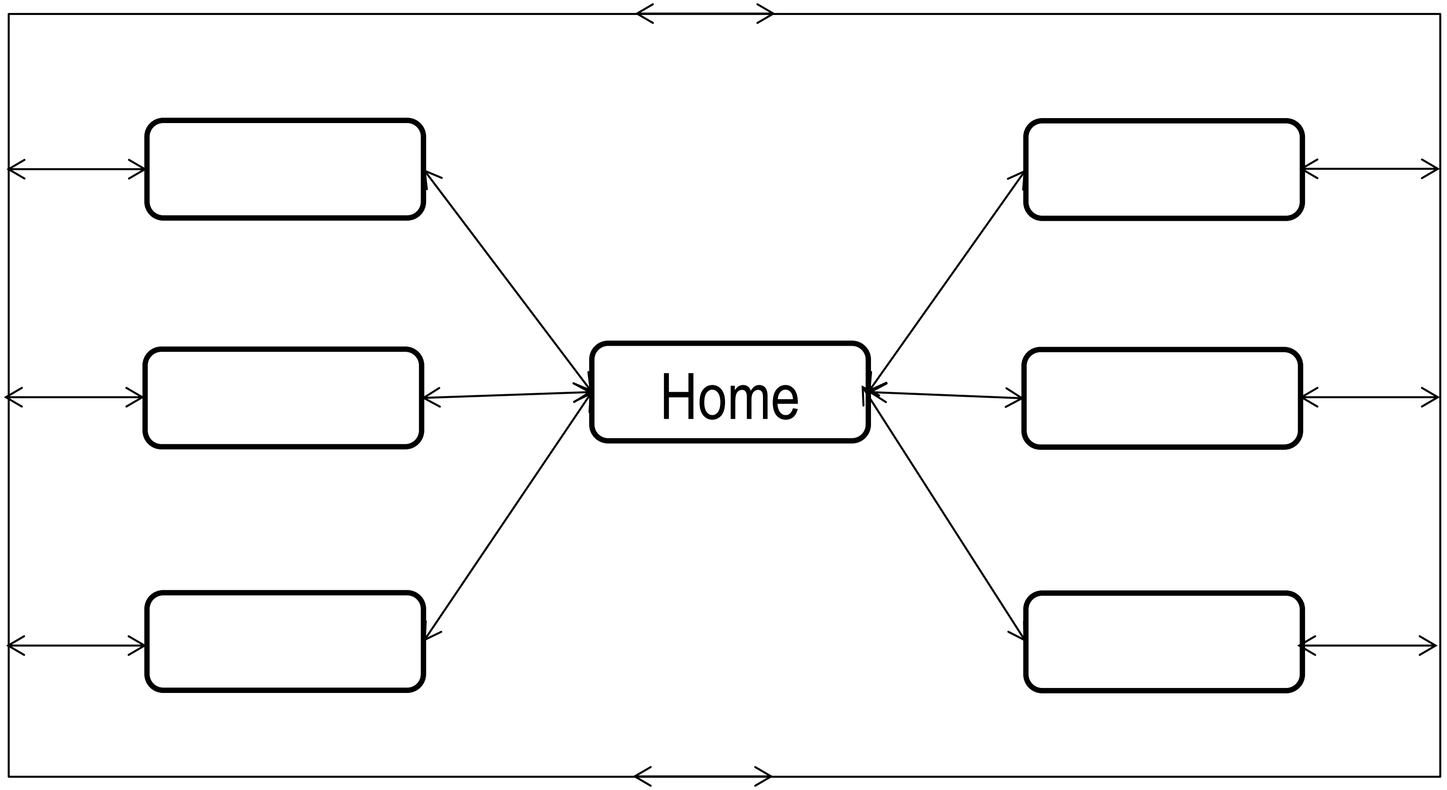
Structure: Network Cluster

Cluster structure encourages exploration within a section. Make sure all pages in each section include a clear navigation bar.



Structure: Network - Web

Web structure allows free navigation. Make sure each page includes a standardised navigation bar.



Structure: Network (continued)

- This structure can result in a user **easily become lost**.
- This type of structure can also cause a significant **maintenance** problems.

Contents

- HTML Elements -- continued
  - – Forms
    - Form Element
    - Form Control Elements
- Website Development Process
  - – Project Life Cycle
  - – Information Design
    - Folder Structure
    - Website Structure
  - – Website Application Environment
  - – Website Layout

Browser Compatibility

- Design must be **portable** and **accessible** by users who have
  - different browsers and device platforms
  - different level of physical abilities
- Guidelines for compatibility
  - follow W3C standards
  - validate your code
  - test your web site using different browsers (including old versions) on different device platforms

Speed and Resolution

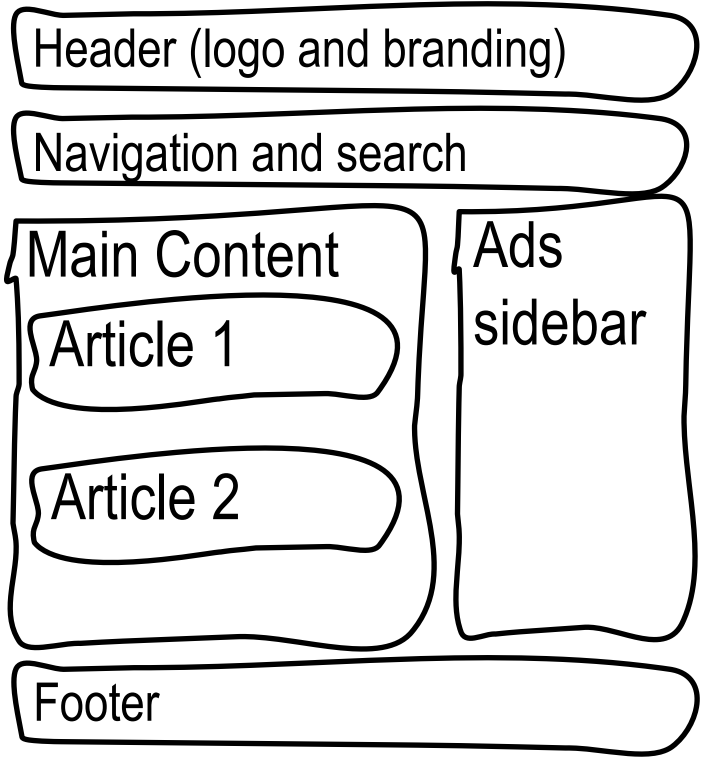
- Consider internet connection speed
  - On first visit, the entire contents of the HTML file, every referenced image, and CSS are downloaded
- Consider screen resolution
- Consider the choice of fonts

Contents

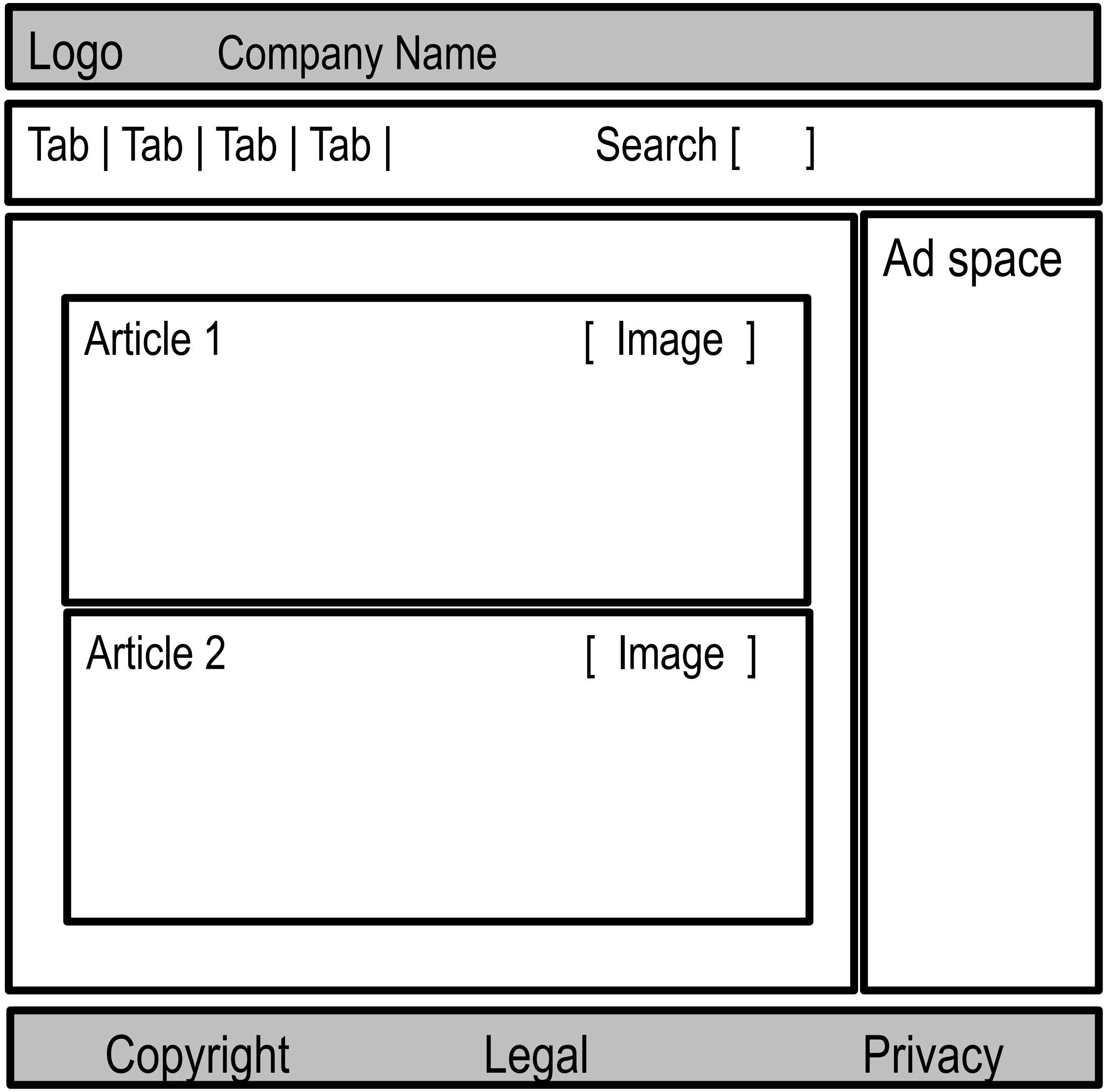
- HTML Elements -- continued
  - Forms
    - Form Element
    - Form Control Elements
- Website Development Process
  - Project Life Cycle
  - Information Design
    - Folder Structure
    - Website Structure
  - Website Application Environment
  - Website Layout

Website Layout: Page Mock Up

- It is a sketch of the desired design for discussion and critique
- Indicates the general layout of the website



Website Layout: Wireframe



- Wireframe shows a more complete version of the page design
- Contains a more detailed elements

References

- Web Style Guide

  http://webstyleguide.com/

  Web Style Guide

  http://webstyleguide.com/wsg3/index.html

  Web Style Guide Online

  http://webstyleguide.com/wsg3/1-process/index.html

  Website Development Process

NEXT LECTURE:

HTML PAGE STRUCTURE
CSS PRESENTATION