

COS10011/60004 Creating Web Applications

Lecture 2

HTML 2



Unit of Study Outline



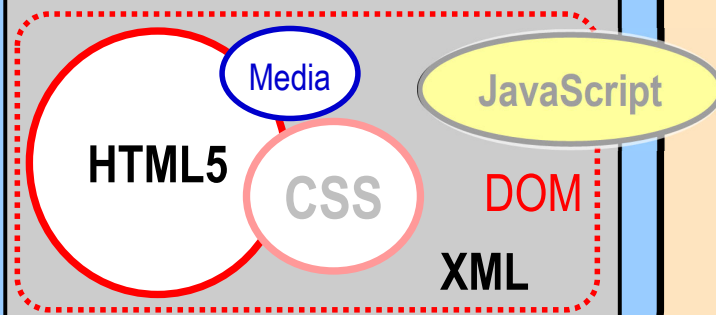
Internet Technologies: TCP/IP, URLs, URIs, DNS, MIME, SSL

Web Technologies: HTTP, HTTPS, Web Architectural Principles

Client Side Technologies:

Web Applications, Markup Languages

Web Documents



Server Side Technologies:

PHP, SSI, ...

Server-Side Data:
MySQL

*Standards
Quality Assurance
Accessibility
Usability
Security*

Last Lecture



■ HTML Documents

- ☐ HTML and XML elements
- ☐ HTML Head (meta information) and body (content)

■ HTML Body elements (page content)

- ☐ Headings and Paragraph
- ☐ Phrase tags and Special Characters
- ☐ Lists and Table
- ☐ Image and Anchor
- ☐ Form, Form Attributes and Form Elements

■ HTML Structure



3

This Lecture - overview



■ HTML page content (continued)

- ☐ ...
- ☐ Image, Figure
- ☐ Hyperlinks (anchor)
- ☐ Form, Form Attributes and Form Elements

■ Validating HTML5 form data with regular expressions

■ HTML Structure

- ☐ Div and Span
- ☐ Navigation, Article, Section, Header, Footer, Aside

Example



4

Image element



- `` is an 'empty' or 'void' inline element that defines an image in an HTML page.

- ☐ Must have the two required attributes: **src** and **alt**

``

- The three well supported image formats for the web are:
 - ☐ PNG (Portable Networks Graphics) *.png
 - ☐ JPEG (Joint Photographic Experts Group) *.jpg or *.jpeg
 - ☐ GIF (Graphics Interchange Format) *.gif
- Images can be edited / created using software such as:
 - ☐ GIMP www.gimp.org
 - ☐ Inkscape www.inkscape.org
 - ☐ Adobe Photoshop www.photoshop.com
 - ☐ IrfanView www.irfanview.com

*Note: The Extras section on Canvas has some notes on **Image Maps**
Not examined but might be nice in your assignments.*

Image (continued)



- **src** attribute - indicates the location and filename of the image
 - ☐ A **relative** or **absolute (URL)** path can be used
- **alt** attribute - for clients that do not support image display (or if the user has "turned off" images).
- **height** and **width** attributes – used to set and reserve the dimensions of the image on the web page.
(This speeds page loading)

***Do not** use height and width attributes to force a large image to "scale down" in appearance.
Resize the image first in an image editing program to the final on-page size and put the correct height and width size in the attribute values.*

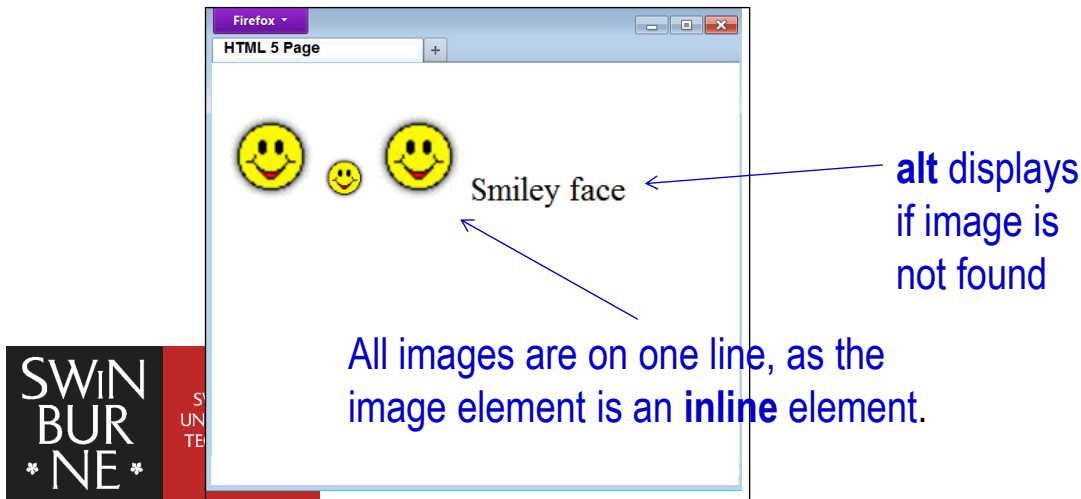
Image (continued)



```
<p>




</p>
```



Figure



- **<figure>... </figure>** encloses a self-contained content such as illustrations, diagrams, photos, code listings
- Its position is independent of the main flow, and if removed it should not affect the flow of the document.
- The **<figcaption>** element is used to add a caption for the **<figure>** element.

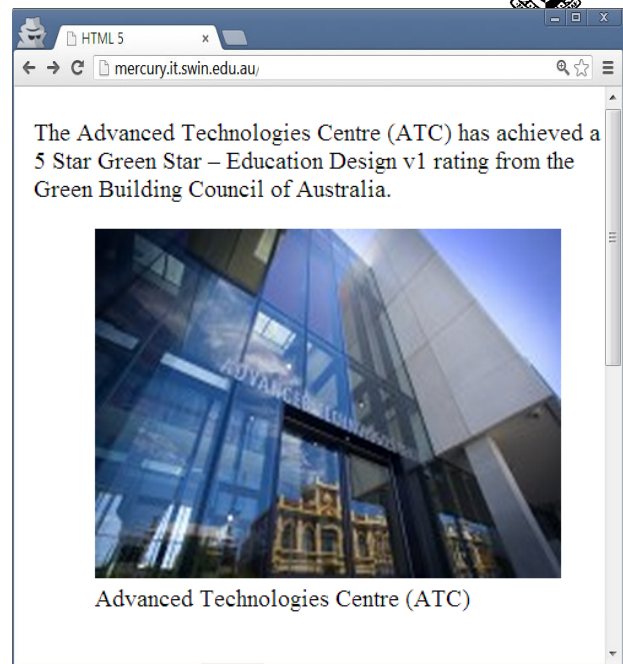
Figure (continued)



```
<p>The Advanced Technologies
Centre (ATC) has achieved a 5
Star Green Star – Education
Design v1 rating from the Green
Building Council of
Australia.</p>
```

```
<figure>

<figcaption>
Advanced Technologies Centre
(ATC)
</figcaption>
</figure>
```



9

This Lecture - overview



■ HTML Content

- ☐ ...
- ☐ Image
- ☐ **Hyperlinks (anchor)**
- ☐ Form, Form Attributes and Form Elements

■ Validating HTML5 form data with regular expressions

■ HTML Structure

- ☐ Div and Span
- ☐ Navigation, Article, Section, Header, Footer, Main
- ☐ Aside, Details, Figure

Example



10



Hyperlinks - Anchor element

- `<a> ... ` is an inline element that defines a hyperlink, used to link from one page to another.

```
<a href="chemistry.html">Chemistry</a>
```

- The **href** attribute - indicates the resource's location for the hyperlink
 - ☐ Resource can be a
 - ☐ location within a web page
 - ☐ web page within the same site
 - ☐ web page on another site
 - ☐ file
 - ☐ email address
- In HTML5, the `<a>` element can only be a hyperlink.



An image can be used as a hyperlink, by nesting an `` element as the anchor's element content



Note: Specifying Resource Locations

- **Relative path** is best used in specifying resources, such as files, images and web pages within website
 - ☐ It allows a whole folder of web pages to be moved from the "testing" environment to the "live" environment, and still retain their relative paths

e.g. "index.html", "images/logo.png"

avoid referencing your home directory `"//images/logo.png"`
- **Absolute path** is used if the resource is not part of the website
 - e.g. "http://www.swinburne.edu.au/update.htm" ✓
 - Do not** use local directory `"file://c:./update/htm"` ✗

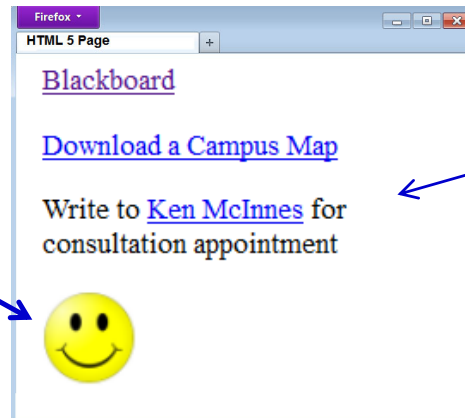


Anchor (continued)



```
<p><a href="http://ilearn.swin.edu.au"> Blackboard</a></p>
<p><a href="http://www.swinburne.edu.au/campuses/hawthorn/
documents/hawthorn.pdf" >Download a Campus Map</a></p>
<p>Write to <a href="mailto:kmcinnes@swin.edu.au" >Ken McInnes</a>
for consultation appointment</p>
<p><a href="http://ilearn.swin.edu.au">
</a></p>
```

Image as
hyperlink



Paragraph tags are used in the example to display a multiple line example. List elements could also have been used.



Paths – Relative



Note: The mercury server is configured to make the folder **www/htdocs** your root directory

You are here.
What is the *relative* link to **bio.htm** in the directory **lee** ?



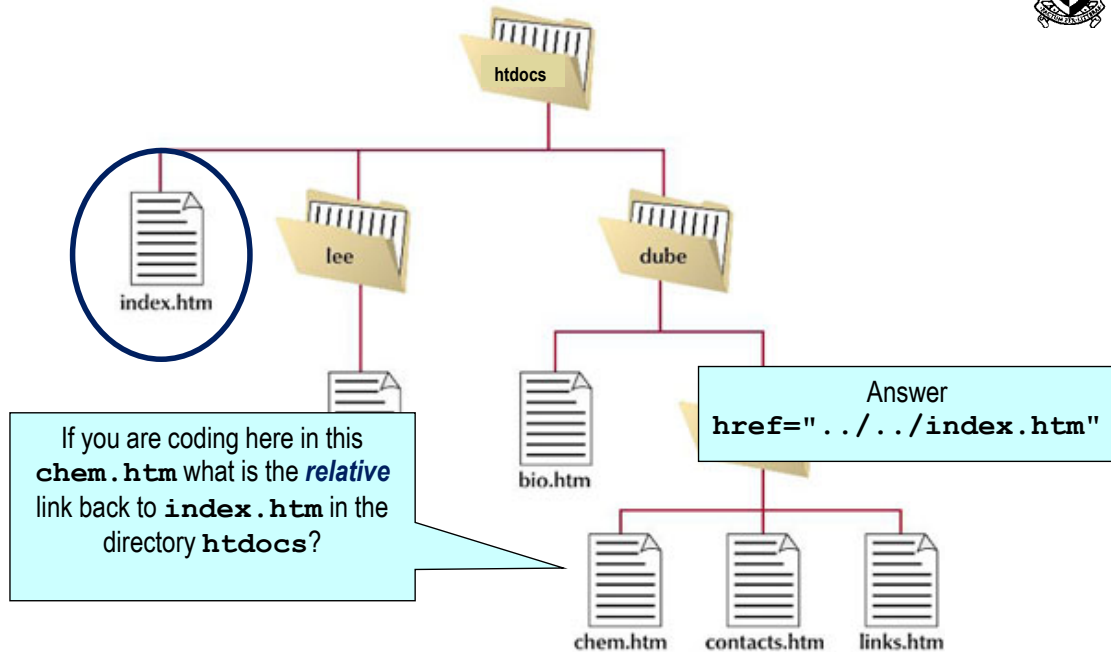
Answer
href="lee/bio.htm"

If you are coding here in this **bio.htm** what is the *relative* link back to **index.htm** in the directory **htdocs**?

Answer
href="../../index.htm"



Paths – Relative



Linking to Sections



- How do we link to sections within a page?

MR. DUBÉ'S CHEMISTRY CLASSES
AT
ROBERT SERVICE HIGH SCHOOL

Classes • Grading • Appointments • Safety

Welcome to the Chemistry Classes Website.
I hope you will use these pages to learn more about my class and my expectations.
Mr. Dubé

Chemistry Classes

Conceptual Chemistry
An introductory course requiring basic math but no algebra

Chemistry I
An introductory course requiring solid algebra

Applied Chemistry
An introductory course requiring solid algebra and an interest in using critical thinking to solve real-world, chemistry-related problems

Advanced Placement Chemistry
An advanced course requiring a grade of B or better in Chemistry I designed for students who want to prepare for the AP Chemistry exam (which can count towards college credits)

clicking a link in the list jumps the user to the correct heading in the page

Grading

Homework: Homework will be given daily and is due at the beginning of the period the day after it was assigned. Homework is worth 5 to 10 points. A periodic quiz consisting of 1 or 2 homework problems from the past week may be given in place of



Anchor element

- How to link to a resource location **within** a web page
- Identify a resource location with an **id** attribute in the page

```
<h1 id="chapter01">Chapter One</h1>
```

- Refer to the resource location within an anchor using an **#** symbol

```
<a href="home.htm#chapter01">Chapter 1</a>
```

- By default, links in all browsers are displayed as follows :
 - An unvisited link is underlined and **blue**
 - A visited link is underlined and **purple**
 - An active link is underlined and **red**



HTML: Common Link Protocols

Protocol	Purpose
file	Access document stored on a local computer system
ftp	Access files on a server using the file transfer protocol
http	Web document transfer protocol
mailto	Use local user configuration to create new email message
news	Usenet newsgroup service
wais	Wide Area Information Server database
tel	User local user configuration to activate the phone and dial the number <u>RFC3996</u>
sms	User local user configuration to create / send sms <u>RFC5724</u>

Include full international prefix.

```
<a href="tel:+61-3-9214-8000">+61-3-9214-8000</a>
```

```
<a href="sms:+61-3-9214-8000?body=Remember%20the%20milk">Send reminder!</a>
```



Other **<a>** attributes



- Apart from **href** **<a>** has number of other useful attributes

```
<a href="..." title="..." target="..." ... >...</a>
```

- **title** a core attribute, shows information about the link as a "tool tip".
- **target** specifies **where to display** the information retrieved from the URL.

Value	Description
_blank	Opens the linked document in a new window or tab
_self	Opens the linked document in the same frame as it was clicked (this is default)
_parent	Opens the linked document in the parent frame
_top	Opens the linked document in the full body of the window
framename	Opens the linked document in a named frame

19

Link **title** attribute



- An example of a hyperlink **title** attribute value being displayed as a tool-tip by the browser.

```
<a href="home.htm"
  title="Return to the Chemistry Home Page">
  Home Page</a>
```



20

This Lecture - overview



■ HTML Content

- ☐ ...
- ☐ Image
- ☐ Hyperlinks (anchor)
- ☐ **Form, Form Attributes and Form Elements**

■ Validating HTML5 form data with regular expressions

■ HTML Structure

- ☐ Div and Span
- ☐ Navigation, Article, Section, Header, Footer, Main
- ☐ Aside, Details, Figure

Example

21



HTML Content

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 in turn can receive the data, process the data and generate a response.
- Possible responses may include:
 - ☐ display information on a web page;
 - ☐ adding data to a database; or
 - ☐ sending an email message.

Firefox

HTML 5 Page

Personal Details:

Name:

Email:

Date of birth:



Form (continued)



1. Form filled



Client

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

Server responds by sending the webpage
of the form as HTML



Server

2. Form result

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

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



23

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 (continued)



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

- **id** - unique identifier of the form
- **method** -HTTP method used to submit the form – **get** or **post**
 - **get** is often used to submit data 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)
- **action** - URL referring to where the data is to be submitted for processing
 - Absolute path is used if processing is from a different site
- **Usually** the <form> element contains all **form control elements** and all other form structuring elements.
- Nothing will be displayed or actioned, unless there are **form control elements**.



HTML Forms



Form control elements: *Note: input is an empty or void element*

- **<input ../>** defines a **form control** for the user 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`

- **<select>** defines a **form control** for the **selection of options** from a **selection 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`

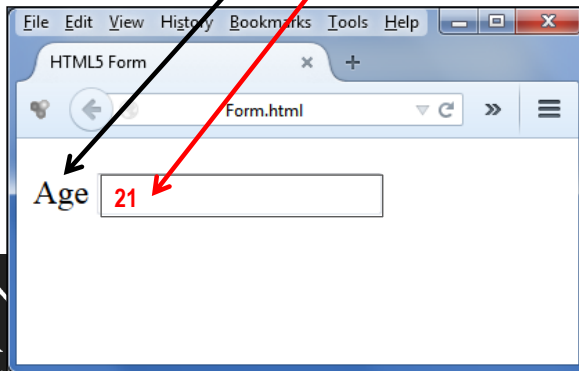


Form Elements: Label



- `<label>...</label>` element associates a **label** with a **form control**.

```
<form id="survey" method="post" action="process.php">
  <p><label for="age">Age</label>
  <input type="text" name="years" id="age"/></p>
  <!-- other form controls -->
</form>
```



Note: Each **form control** element has a **name** attribute. Data is passed in **name = value** pairs e.g. **years=21**

When label text is 'clicked' or 'touched' the form control is focused. This provides a better / larger target on touch screens.

27

Form Elements: Label



Technique #1 Preferred Technique! `for` is explicitly connected to `id`

```
<form id="survey" method="post" action="process.php">
  <p><label for="age">Age</label>
  <input type="text" name="age" id="age" /></p>
</form>
```

Technique #2

```
<form id="survey" method="post" action="process.php">
  <p><label>Age
    <input type="text" name="age" />
  </label></p>
</form>
```

This technique is very common but the **label** is **not explicitly** connected to an **identifier**

28



Form Elements: Input Text

```
<p><label>Name
  <input type="text" name="name" maxlength="40"
        size="20"/>

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

</label></p>
```

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

29



Form Elements: Input Checkbox

```
<p>Things you like about the seminar
  <br />
  <label><input type="checkbox" name="things[]"
    value="loc" />Location</label>
  <label><input type="checkbox" name="things[]"
    value="fud" />Food</label>
  <label><input type="checkbox" name="things[]"
    value="spk" />Speaker</label>

</p>
```

An array called "things"

As multiple checks are allowed, the **name** (sent to the server) must either

- Be different
- or the same but terminate with []

30



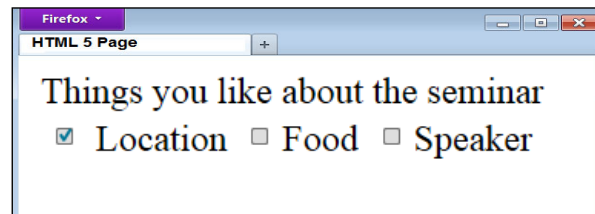


Form Elements: Input Checkbox

```
<p>Things you like about the seminar
<br />
<label><input type="checkbox" name="things[]"
      value="loc" checked="checked" />
      Location</label>
<label><input type="checkbox" name="things[]"
      value="fud" />Food</label>
<label><input type="checkbox" name="things[]"
      value="spk" />Speaker</label>
</p>
```

checked is used to initialise checkbox with a default check

checked="checked" is used for XHTML compliant code



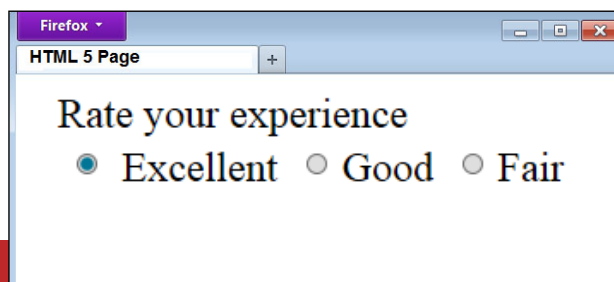
31



Form Elements: Input Radio Button

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

Best to group radio inputs with a fieldset and legend - see later.



Note: Only **one** choice is allowed, the **name** must be the same

32

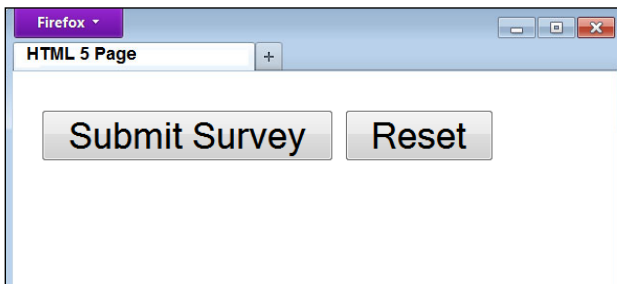
Form Elements: Input Submit and Reset



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

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

Note: Reset means set all input form fields to its initial value, and not clear its value. It only has the effect of clearing, if the initial values are blank or empty

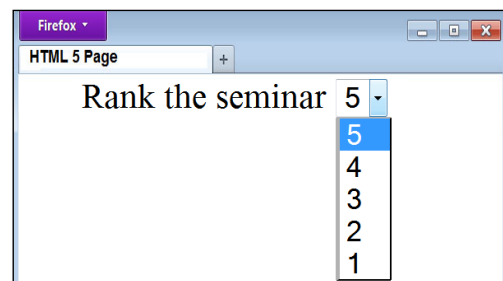


33

Form Elements: Select & Option



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



- **selected="selected"** is used for XHTML compliant code

34

Form Elements: Select & Option (Attributes)



select can have the following attributes: size, multiple, tabindex, disabled

Attribute	Value	Description
<u>autofocus</u>	autofocus	Specifies that the drop-down list should automatically get focus when the page loads
<u>disabled</u>	disabled	Specifies that a drop-down list should be disabled
<u>form</u>	<i>form_id</i>	Defines one or more forms the select field belongs to
<u>multiple</u>	multiple	Specifies that multiple options can be selected at once
<u>name</u>	<i>name</i>	Defines a name for the drop-down list
<u>required</u>	required	Specifies that the user is required to select a value before submitting the form
<u>size</u>	<i>number</i>	Defines the number of visible options in a drop-down list

35



Form Elements: Text Area



- `<textarea>...</textarea>` defines a form control for the user to enter multi-line text input
- Text Areas can have the following attributes: rows, cols, readonly, tabindex, accesskey, disabled

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

Note: all characters in the text area's element content are displayed verbose. So do not add blank spaces



36



Forms - Common errors – Watch out!



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



input type = "radio"

one, from a limited number of choices

mutually exclusive – one checked, all others unchecked

same 'name'

different 'value' for elements in the group

input type = "checkbox"

one or more, from a limited number of choices

usually have different 'name' or use an array[]

select and option (dropdown box)

one (or more) options, from a limited number of choices

'name' only for select (can have a 'multiple' attribute)

different 'value' for 'option' elements (not 'name')

HTML Forms: Fieldset & Legend



Other form elements:

- **<fieldset>** element is used for **grouping** related form controls, so authors can divide a form into smaller, more manageable parts, improving the usability of the form.
- **<legend>** element defines a **caption for a fieldset** and must be at the start of a fieldset, before any other elements. A legend can include an **accesskey** attribute.

Form Elements: Fieldset & Legend



<fieldset>

<legend>Personal Details:</legend>

<label>Name:

<input type="text" name="pname" /></label>

<label>Email:

<input type="text" name="pemail" /></label>

<label>Date of birth:

<input type="text" name="pdob" /></label>

</fieldset>

Firefox
HTML 5 Page

Personal Details:

Name:

Email:

Date of birth:

39



HTML5 FORM ELEMENTS

Example



40



HTML5 Form Elements

- HTML5 has new form input **types**
Note that these are not yet universally supported by all browsers

<http://html5test.com/>

- Examples generated using Chrome.

- | | |
|-----------------------------------|---------------------------------|
| <input type="checkbox"/> color | <input type="checkbox"/> range |
| <input type="checkbox"/> date | <input type="checkbox"/> search |
| <input type="checkbox"/> datetime | <input type="checkbox"/> tel |
| <input type="checkbox"/> email | <input type="checkbox"/> time |
| <input type="checkbox"/> month | <input type="checkbox"/> url |
| <input type="checkbox"/> number | <input type="checkbox"/> week |

Note: If the browser does not understand the type, it will default to type="text"

- Other **new attributes** include:
autofocus, placeholder, pattern, required



41

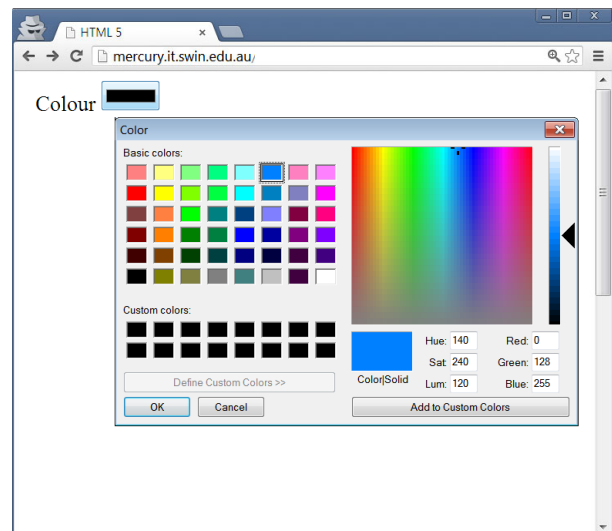


HTML5 Form Elements: Colour

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

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

There can only be one field with autofocus. If there is more than 1 the first instance gets the focus.



42

HTML5 Form Elements: Date



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

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

The screenshot shows a web browser window with a URL bar containing 'mercury.it.swin.edu.au/'. Below the browser window, there is a form with two sections: 'Date' and 'Time'. The 'Date' section has a text input field with the placeholder 'dd/mm/yyyy' and a calendar dropdown menu. The calendar is open, showing January 2013, with the 23rd highlighted. Below the calendar are 'Today' and 'Clear' buttons. The 'Time' section has a text input field with the placeholder '-- : -- --' and a time selection dropdown menu. The number '43' is displayed to the right of the time picker.



HTML5 Form Elements: Email



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

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

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

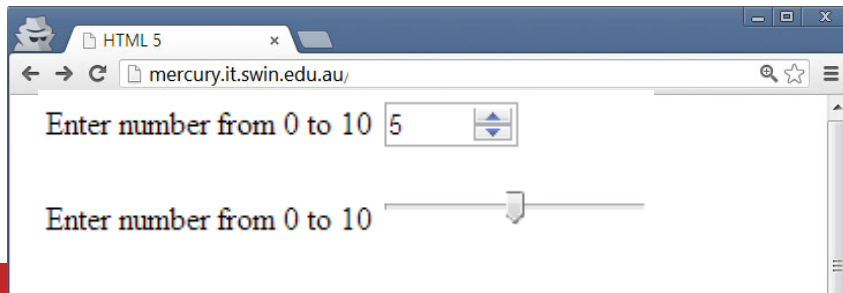
The screenshot shows a web browser window with a URL bar containing 'mercury.it.swin.edu.au/'. Below the browser window, there is a form with two email input fields. The first field has the placeholder text 'name@domain.com'. The second field contains the text 'cchua.swin.edu.au'. A tooltip message with an exclamation mark icon and the text 'Please enter an email address.' is displayed below the second field.



HTML5 Form Elements: Number



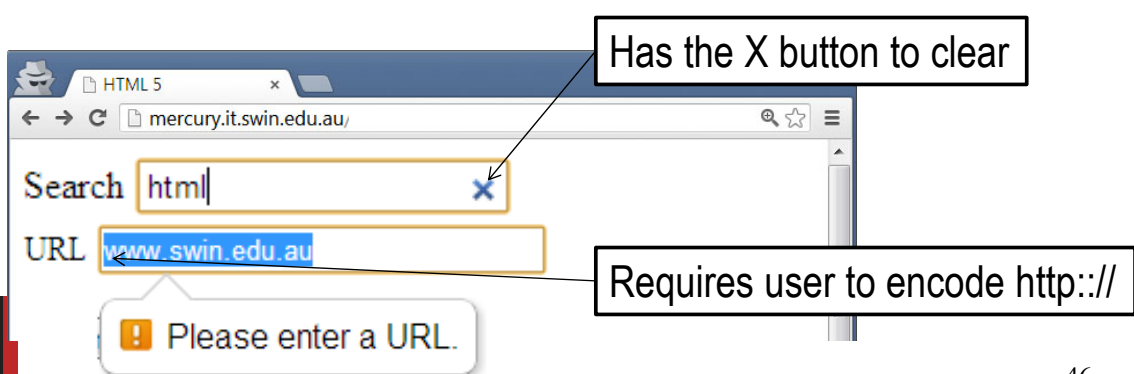
```
<p><label>Enter number from 0 to 10
  <input type="number" name="score" min="0"
    max="10" step="1" value="5" />
</label></p>
<p><label>Enter number from 0 to 10
  <input type="range" name="rating" min="0"
    max="10" value="5" />
</label></p>
```



HTML5 Form Elements: Search



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



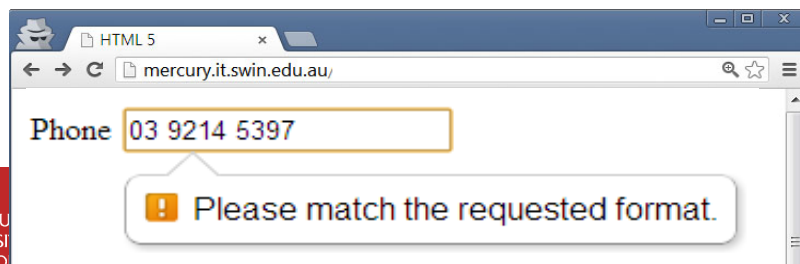


HTML5 Form Elements: Phone

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

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

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



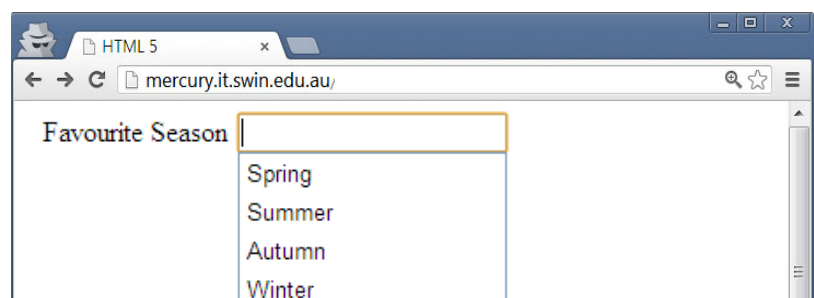
47



HTML5 Form Elements: Data List

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

Make sure the id matches the list



48



This Lecture - overview



■ HTML Content

- ...
- Form, Form Attributes and Form Elements

■ *Validating HTML5 form data with regular expressions*

■ HTML Structure

- Div and Span
- Navigation, Article, Section, Header, Footer, Main
- Aside, Details, Figure

Example

49



HTML Forms Input Data Control and Checking



■ Examples

```
<form id="regForm" method="post" action="....php">
```

```
...
```

```
<p><label for="catname">Cat's Name</label>
```

```
<input type="text" name="catname" id="catname"
maxlength="20"
```

Restricts the # characters

```
size="10"
```

Size of text box

```
required="required"/>
```

A value must be entered

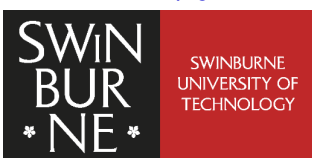
```
</p>
```

```
...
```

```
<input type="email" name="email" id="email" required="required"/>
```

```
</form>
```

HTML5 input control



Using patterns in HTML



- The pattern attribute uses a 'regular expression' to define the characters that can be entered into a field

```
<input type="text" name="catname" id="catname" maxlength="20"
      pattern="^[a-zA-Z ]+$"
      required="required"/>
```

Alpha characters or space only

```
<input type="text" name="dob" id="dob" maxlength="10" size="10"
```

Placeholders provide prompt to the user

```
placeholder="dd/mm/yyyy"
```

```
pattern="\d{1,2}\/\d{1,2}\/\d{4}"
required="required"/>
```

dd/mm/yyyy
???? no range

```
/(?=d)(?:31(?!(?:0?[2469])11))(?:30|29(?!(?:0?2)|29(?=.0?2.(?:1[6--9]|2--9)d)?(?:0[48]|[2468][048]|13579)[26]))(?:16|[2468][048]|3579)[26])00))(?:x20|$))(?:2[0--8]|1d|0?[1--9])([--/])(?:1[012]0?[1--9]1(?:1[6--9]|2--9)d)?2dd(?(?:x20d)x20|$))/?
```

Regular expressions not necessarily the best solution to every check!!

<http://html5pattern.com/>

What are Regular Expressions?



■ Regular Expressions:

- are strings that describe the 'pattern' or 'rules' for strings

e.g. `^[a-zA-Z]+$`

- are strings that follow a set of syntax rules
- can be used as a concise and consistent way to 'test' for matching patterns.

■ Regular expressions can be great for checking form values!

- A simple regular expression can be the equivalent of many lines of code.

Where are they Used?



EVERYWHERE!

- **Regular Expressions** have a history in unix environments, became popular in many text editors, and were supported in text processing (programming) languages like Perl and Tcl.

... but the catch was

there were some variations in syntax ☹

mostly now the same basic

syntax ☺

- **Regular Expressions** are used in many text search routines
- **Regular Expressions** are very easy to use ☺
.... but *not* so easy to define the 'pattern' ☹

```
/(?=d)(?: (?:(?:31(?! (?:0?[2469])11))|(?: 30|29)(?!0?2)|29(?=.0?2.(?: (?:1[6--9]| [2--9]d)?(?:0[48]||[2468]||[048]||[13579]
[26])|(?: (?:16|[2468]||[048]||[3579] [26])00)))(?:x20|$))|(?:2[0--8]|1d|0? [1--9])|([--/])(?:1[012]|0?[1--9])1(?: 1[6--9]||[2--
9]d)?dd(?: (?:x20d)x20| $))?)
```

* INE *

RegExp - Basic Syntax



/pattern/modifiers

■ Pattern Basics

^	Start of string
\$	End of string
.	Match any single character
(a b)	a or b
(...)	Group section
[abc]	match any character in the set
[^abc]	not match in the set
[a-z]	match the range
\d	match a single digit from 0 to 9

■ Pattern Quantifiers

a?	0 or 1 of a
a*	0 or more of a
a+	one or more instance of a
a{3}	exactly 3 a's = aaa
a{3,}	3 or more a's
a{3,6}	between 3 to 6 a's
!(pattern)	"not" pattern

[\ ^ \$. | ? * + () are the 11 meta-characters, or special characters, used in the syntax.
If you want to include these in a RegExp literal you need to escape them with \ eg. \ (

RegExp - Basic Examples



JavaScript	matches "Isn't JavaScript great?"
[^] JavaScript	matches "JavaScript rules!", not "What is JavaScript?"
JavaScript\$	matches "I love JavaScript", not "JavaScript is great!"
[^] JavaScript\$	matches "JavaScript", and nothing else

bana?na	matches "banana" and "banna", but not "banaana".
bana+na	matches "banana" and "banaana", but not "banna".
bana*na	matches "banna", "banana", and "banaaana", but not "bnana"
[^] [a-zA-z]+\$	matches any string of one or more letters and nothing else.

<http://www.sitepoint.com/article/expressions-javascript>



x?	0 or 1 of x
x*	0 or more of x
x+	one or more instance of x

RegExp - Basic Syntax



■ Groups & Ranges

- . Any character (except \n)
- (a|b) a or b
- (...) group
- (?:...) passive group
- [abc] set ("range") a, b or c
- [^abc] not a, b or c
- [a-g] set range a to g
- [3-6] set range of digits 3,4,5 and 6
- \n "nth" group or subpattern

■ Pattern Modifiers

- /i case insensitive
- /x allow comments and white space in pattern
- ...



RegExp - Some Sample Patterns



`[A-Za-z0-9-]+` = Letters, numbers and hyphens
`\d{1,2}\/\d{1,2}\/\d{4}` = date as 19/9/2006 \ = escape character
`^[^s]+(?:\.(jpg|gif|png))\.\d{2}` = jpg, gif or png filename
`^[1-9]{1}$|^[1-2]{1}[0-9]{1}$|^30$` = any number from 1 to 30
`#?([A-Fa-f0-9]){3}((([A-Fa-f0-9]){3})?)` = valid hex colour code
`{6,}` = password six or more characters
`(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,15}` = password string
(at least 1 uppercase letter, 1 lowercase letter and 1 digit)
`^.+@.+\. {2,3}$` = email address
`\<(/?[^>]+)>/` = HTML tags



Remember: Same pattern may be used for checking form values, **both** on client-side and server-side.

Lecture - overview



■ HTML Content

- ☐ Headings and Paragraph
- ☐ Phrase tags and Special Characters
- ☐ Lists and Table
- ☐ Image and Anchor
- ☐ Image Maps
- ☐ Form, Form Attributes and Form Elements

■ HTML Structure

Example

- ☐ **Div and Span**
- ☐ Navigation, Article, Section, Header, Footer, Main
- ☐ Aside, Details, Figure



Division



- `<div>...</div>` is a generic logical **block level container** used to divide content, e.g. section
- It has **no default** meaning or rendering behaviour, as it is a logical “division”.
- It plays a role in providing an arbitrary block container where a style can be applied to use CSS.
- **Do not** use a `<div>` when you should be using a logical element like `<p>` or `<nav>`.

Span



- `...` is a generic **inline level container** used to group other inline elements, such as text.
- It has **no default** meaning or presentation qualities.
- Similar to the `<div>` element, the `` tag has a role in providing an arbitrary container but for inline elements.
- **Do not** use a `` when you should be using a logical element like `` or ``.

Division (continued)



div id = "header"

There is a better way to do this!

div id = "banner"

Presented without CSS

div id="news"

div id="info"

div id="podcast"

div id = "footer"



- Always remember that **HTML** is only about **content** and **structure**, where these will appear on the screen will be specified through **CSS**

61

Division and Span



- With `<div>` and ``, the defined structure remain semantically *neutral* in terms of meaning to the browser.
- If a browser is to have a shortcut key to the page navigation, which `<div>` will it jump to?
Users may use different names, such as
 - ☐ `<div id="nav"> ... </div>`
 - ☐ `<div class="menu"> ... </div>`
 - ☐ `<div class="navigation"> ... </div>`

62

Lecture - overview



■ HTML Content

- ☐ Headings and Paragraph
- ☐ Phrase tags and Special Characters
- ☐ Lists and Table
- ☐ Image and Anchor
- ☐ Image Maps
- ☐ Form, Form Attributes and Form Elements

■ HTML Structure

- ☐ Div and Span
- ☐ *Navigation, Article, Section, Header, Footer, Aside, Figure*



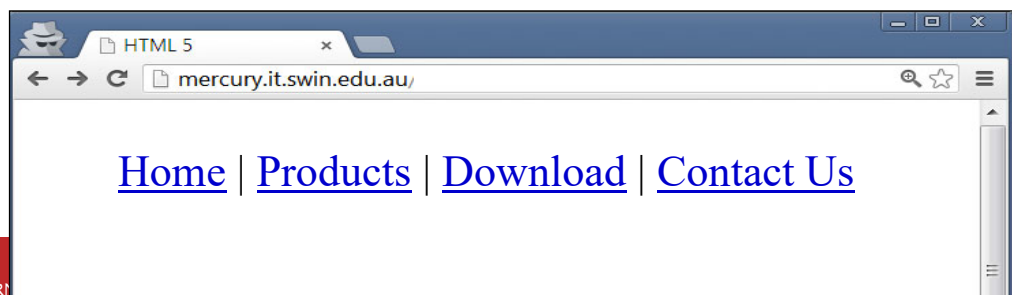
63

HTML Structure

Navigation



- `<nav>...</nav>` specifies a section of navigation links.
- It is intended only for major block of navigation links.
 - ☐ Not all links of a document must be in a `<nav>` element
- Browsers, such as screen readers for disabled users, can use this element to determine whether to omit the initial rendering of this content.



64



Navigation (continued)

<nav>

```
<a href="index.php">Home</a> |
<a href="product.php">Products</a> |
<a href="download.php">Download</a> |
<a href="contact.php">Contact Us</a>
</nav>
```

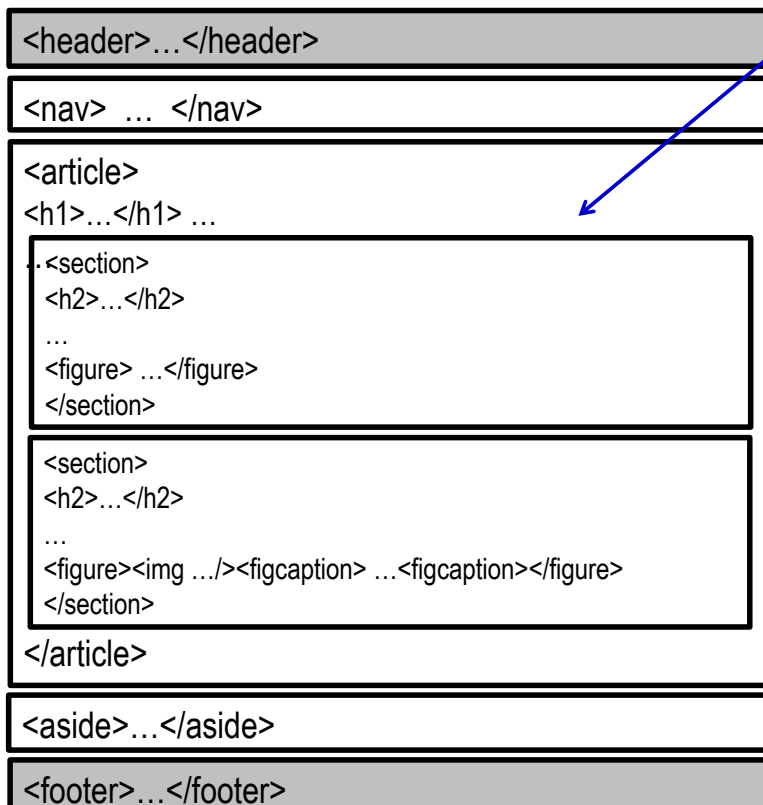


An unordered list is often used inside <nav>

65



Structure: Putting it all together



Presented without
CSS

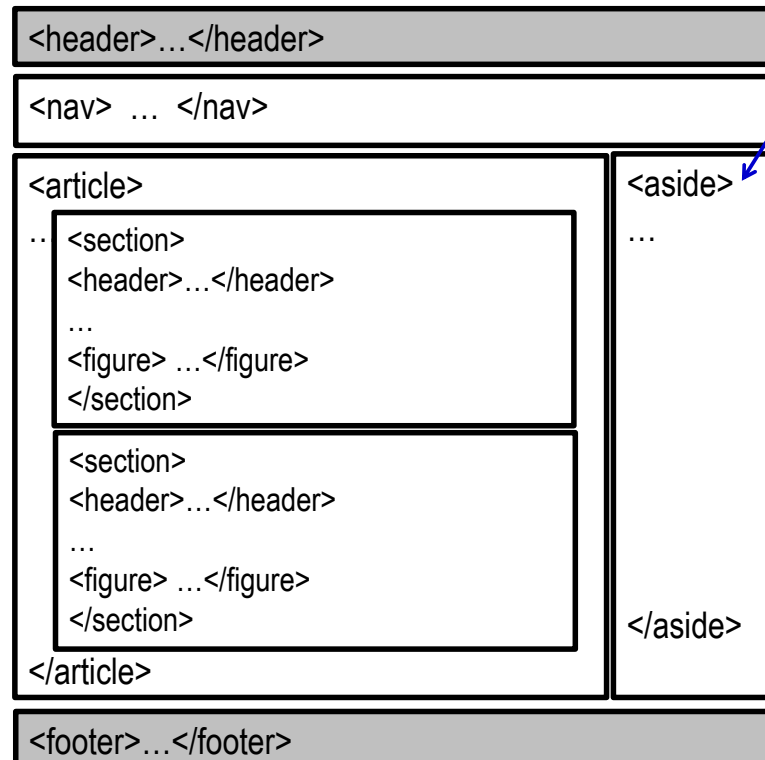
- Always remember that **HTML** is only about **content** and **structure**, where these will appear on the screen will be specified through **CSS**

66

Structure: Putting it all together



Presented with CSS



67

Next

- Design for Usability
- CSS – Presentation and Layout



68