INFO 151 Web Systems and Services

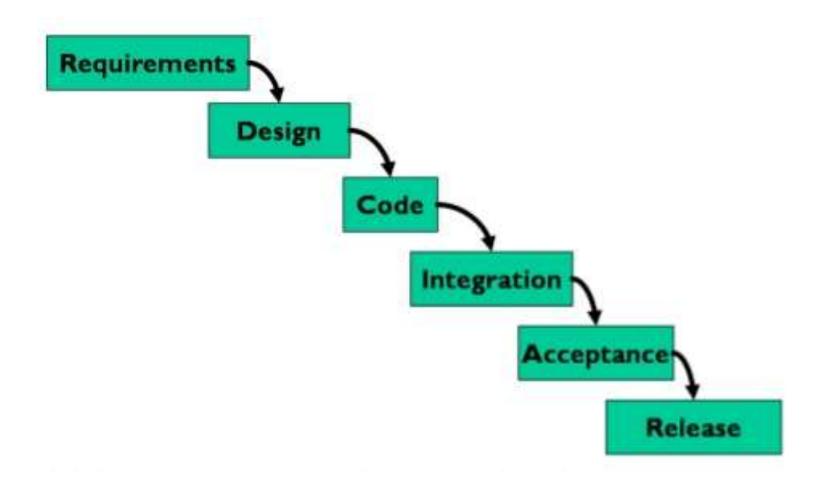
Principles of Design

Dr Philip Moore
Dr Zhili Zhao

Software Design Methodologies

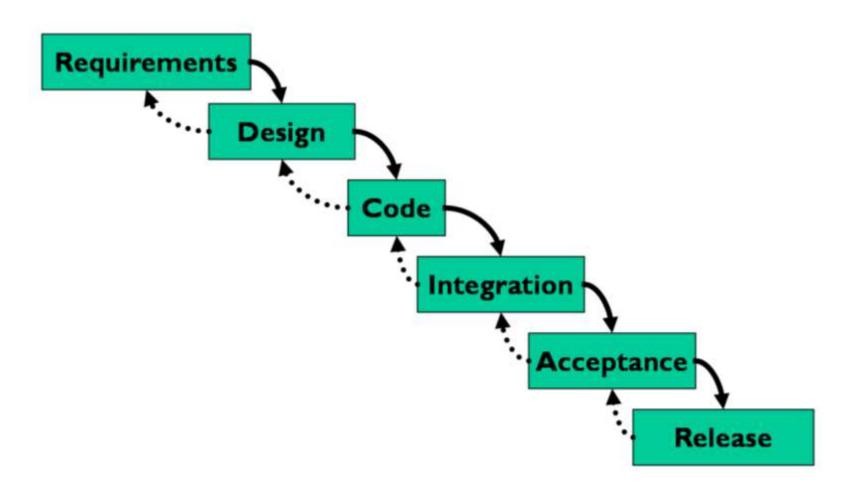
Waterfall Software Design

- The traditional waterfall Software Development Model
- A development of this model is shown in the following slide



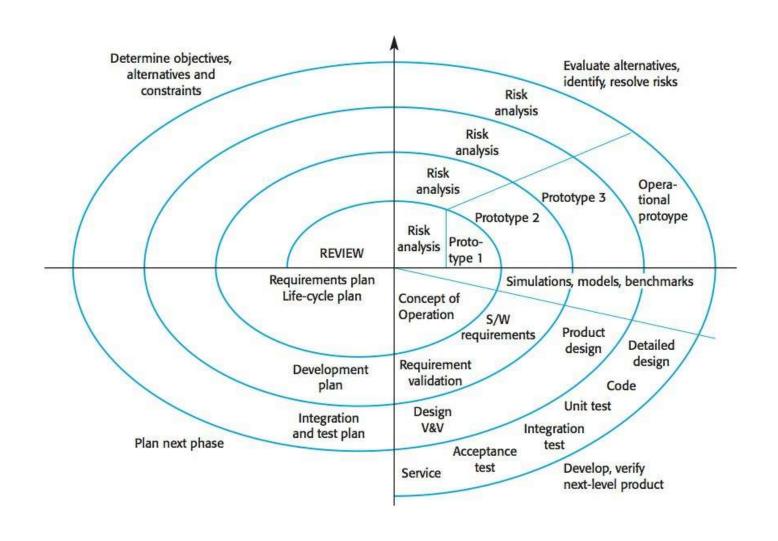
Improved Waterfall Software Design

- The waterfall
 Software
 Development
 Model with
 feedback
- A development of the traditional approach
 - A semi-iterative approach



Web Systems Software Design

- The Spiral Software Development Model
- This model tries to address the issues in the waterfall model(s)
- This method applies an iterative approach to software design
- The spiral model also tries to address the 30/70 software lifecycle



Computer Program Design

Program Design

- The design process for a computer program is:
 - Identify and document the requirements specification
 - This will set out the required input (data) and the required output (results)
 - Create a block layout showing the program algorithm structure
 - This will set out the sequence of operations (sequential / selection / iteration)
 - Write *pseudo program code* describing the algorithm
 - Plot the path of the data through the program
 - Design a test plan to:
 - Test and verify if the data values throughout the program are correct (to test if the result is consistent with the required output (logic testing)
 - Testing may use both black-box and white-box testing

Program Design

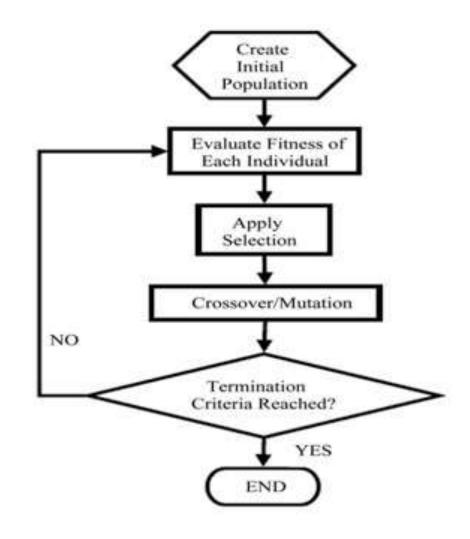
- In the design process for a computer program a number of terms are in general use:
 - Algorithm
 - Pseudo (program) code
 - Black-box testing
 - White-box testing
- You will need to understand and apply these terms and the related processes when designing 'real-world' programs
- The following slides provide a brief description of the terms and processes

Algorithm

- In computer science:
 - An *algorithm* is a finite sequence of well-defined, computer-implementable instructions, typically designed to solve a class of problems or to perform a computation
 - Algorithms are always unambiguous and are used as specifications for performing calculations, data processing, automated reasoning, and other tasks
- Algorithms are widely used in informatics:
 - Informatics addresses information systems, information technologies, and data science

Algorithm Example

- The figure shows:
 - A typical algorithm for computing a Genetic Algorithm (GA)
- In the figure we can see:
 - The steps to be implemented to run a GA
 - Each step must be completed before toe program moves on to the next step
 - The program flow and termination criteria are clearly seen



Pseudo Code

Pseudo code:

- Is a term used in programming and algorithm based design and implementation
- It is a methodology that allows the programmer to represent the implementation of an algorithm
- The process is a step-by-step written outline of your code that you can gradually transcribe into the programming language

Programmers:

Use it to plan out the function of an algorithm before starting actual programming

Pseudo Code Example

- How do we know when to stop?
 - Let the use tell us how many
- Print them in order?
 - IF the first number is smaller
 - Print the first number
 - Print the second number
 - IF the first number is larger
 - Print the second number
 - Print the first number

- The example is part of a larger pseudo code
- From the brief example we can see:
 - The tasks to be undertaken
 - The sequence of tasks
 - The constraints
 - The computing processes
 - The actual operations to be undertaken

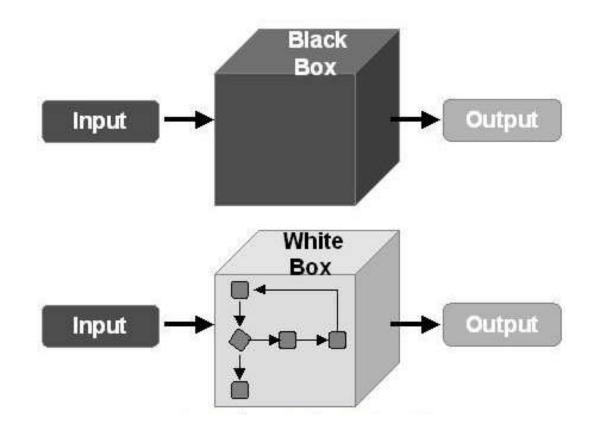
Testing

Black Box Testing:

 Is a method of testing software where the internal structure and design of the program is NOT known to the tester

White Box Testing:

 Is a method of testing software where the internal structure and design IS known to the tester

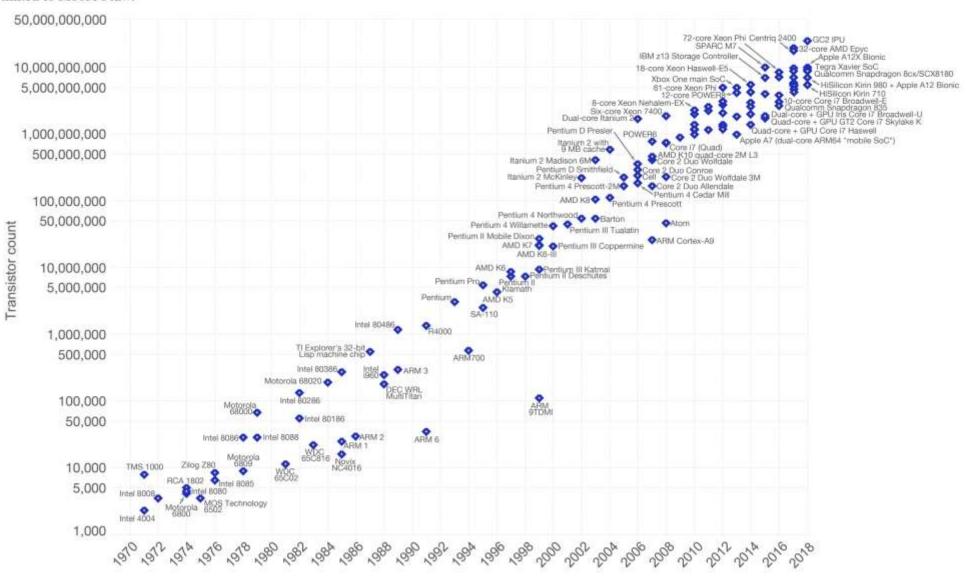


Program Code Optimisation

Moore's Law – The number of transistors on integrated circuit chips (1971-2018)



Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are linked to Moore's law.



Web Design Principles

Web Design Principles (Donald Norman)

Visibility

 Users need to know what all the options are and know straight away how to access them. In the case of websites, this is an easy win.

Feedback

• Every action needs a reaction. There needs to be some indication, like a sound, a moving dial, a spinning rainbow wheel, that the user's action caused something.

Affordance

 Affordance is the relationship between what something looks like and how it's used.

Six Web Design Principles (Donald Norman)

Mapping

 Mapping is the relationship between control and effect. The idea is that with good design, the controls to something will closely resemble what they affect.

Constraints

• Constraints are the limits to an interaction or an interface. Some are really obvious and physical, for example the screen size on a phone.

Consistency

- The same action has to cause the same reaction, every time.
- For more information on these web design principles with some examples see:
 - https://enginess.io/insights/6-principles-design-la-donald-norman

Donald Norman – Design Principles

- Provide rich, complex, and natural signals
- Be predictable
- Provide a good conceptual model
- Make the output understandable
- Provide continual awareness without annoyance
- Exploit natural mapping and make interaction understandable and effective

Coffee Pot (Users System View)

- This is an example of a users view a system
 - In this case a coffee pot
- The user can clearly see:
 - The purpose
 - The function
- In this example the *user* and the *designer* share a common view of the system



Coffee Pot (Designers System View)

- This is a famous example of a coffee pot where the *user* and the *designer* do not share a common view of the system
- In this example:
 - The purpose is not clear
 - The function is not clear
- In this example:
 - Not only is the design faulty (but)
 - The designer has failed to address the Donald Norman design principles



A System Design Conceptual Model

designers
view of
the system

A designer's system view must match the user's system view to create a system with good usability

users view of the system

System

Web Systems Design

- Web-based systems must address a two important factors:
 - Usability (and)
 - Security
- Usability:
 - Relates to how easy users can access a web-site and use the functions
 - Usability applies to both 'stateless' (static) and 'stateful' (dynamic) web sites
- Security:
 - While we must consider security for both 'stateless' and 'stateful' web sites
 - Security is very important for 'stateful' web sites
 - For example: security is essential in an on-line banking web-site / application
- There is a 'trade-off' between usability and security

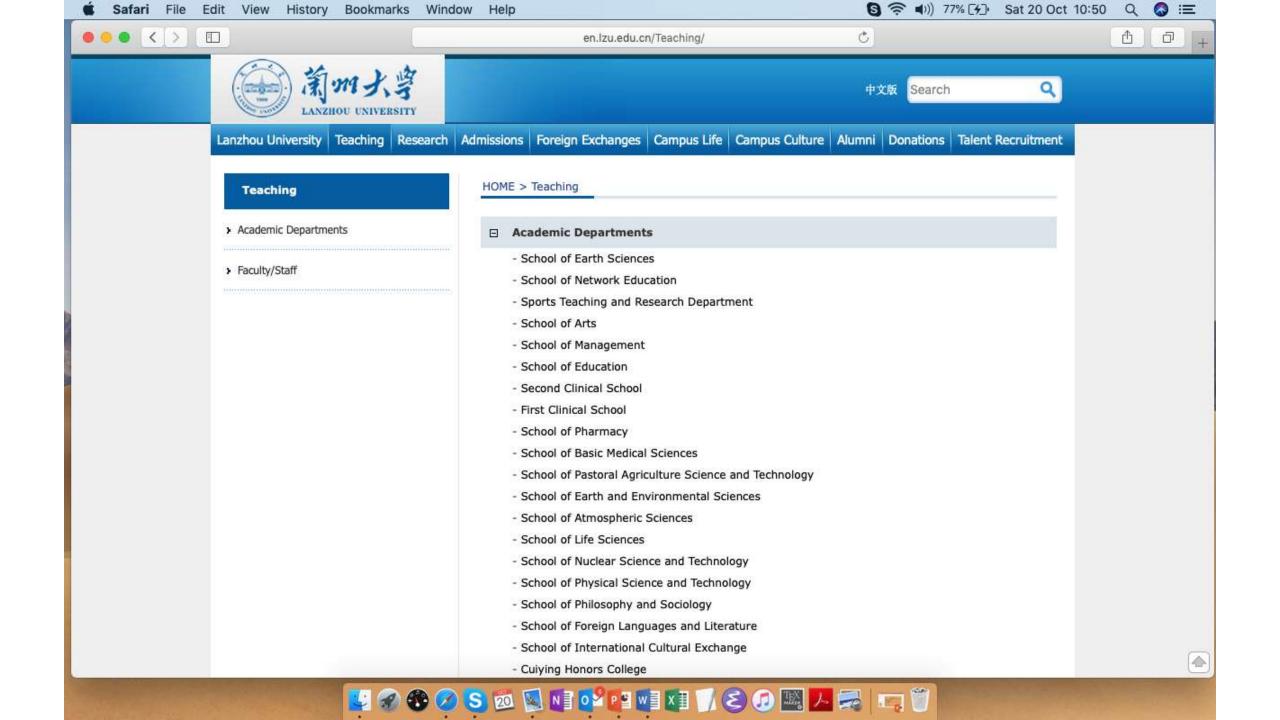
Feedback

Feedback

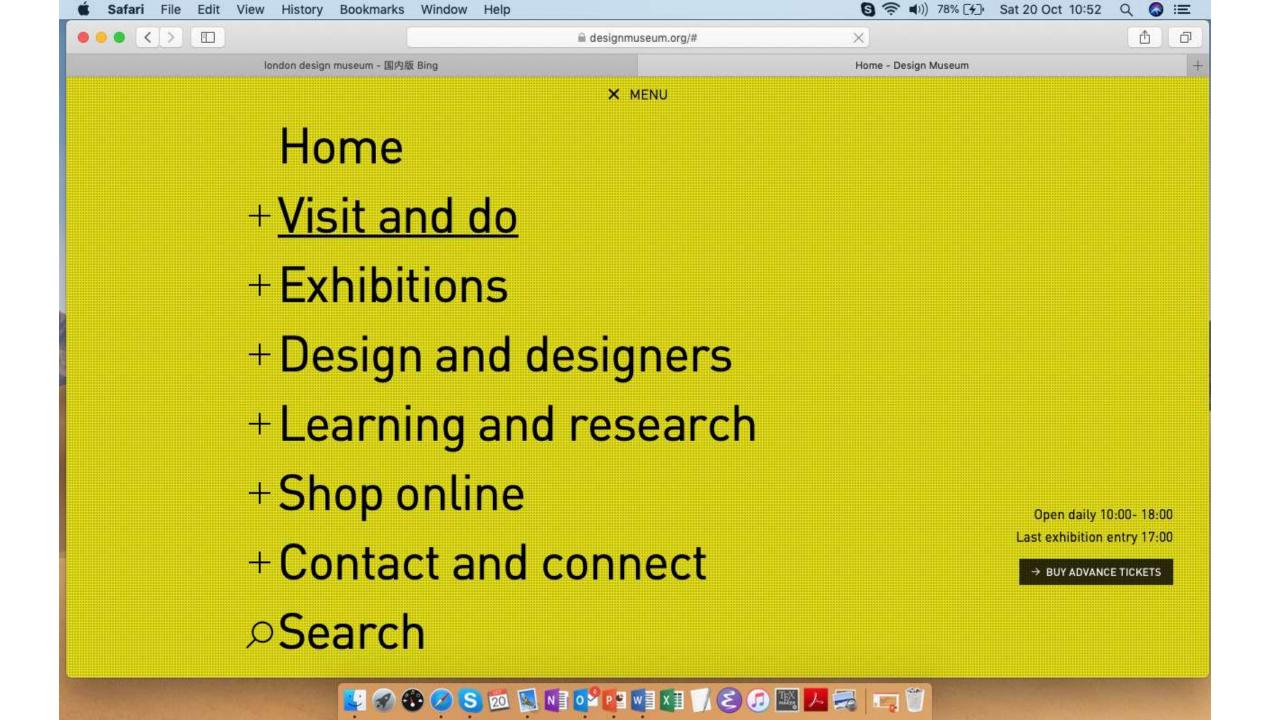
- When designing and developing a web-page (or web-site) we have in previous sessions introduced
 - Usability and security considerations and the 'trade-off'
- A further important consideration is 'feedback' to users
- We shall see in a later tutorial a CSS float example where a navigation bar was created with links to other web-pages
 - In this example when the cursor *hovers* over the home link (*mousover*) the colour changed from red to black
 - This is an example of feedback which informs the user of an action
- When developing web-pages and web-sites user feedback is important

Web-Site Design Web Site Building



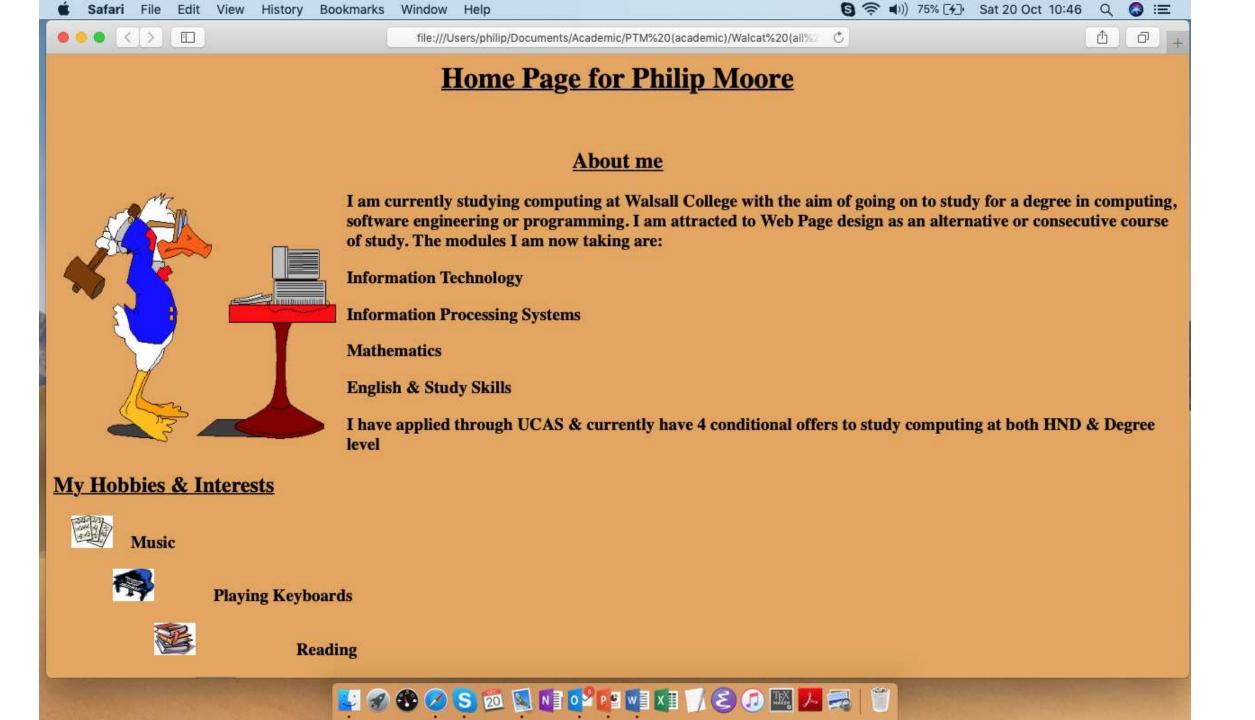


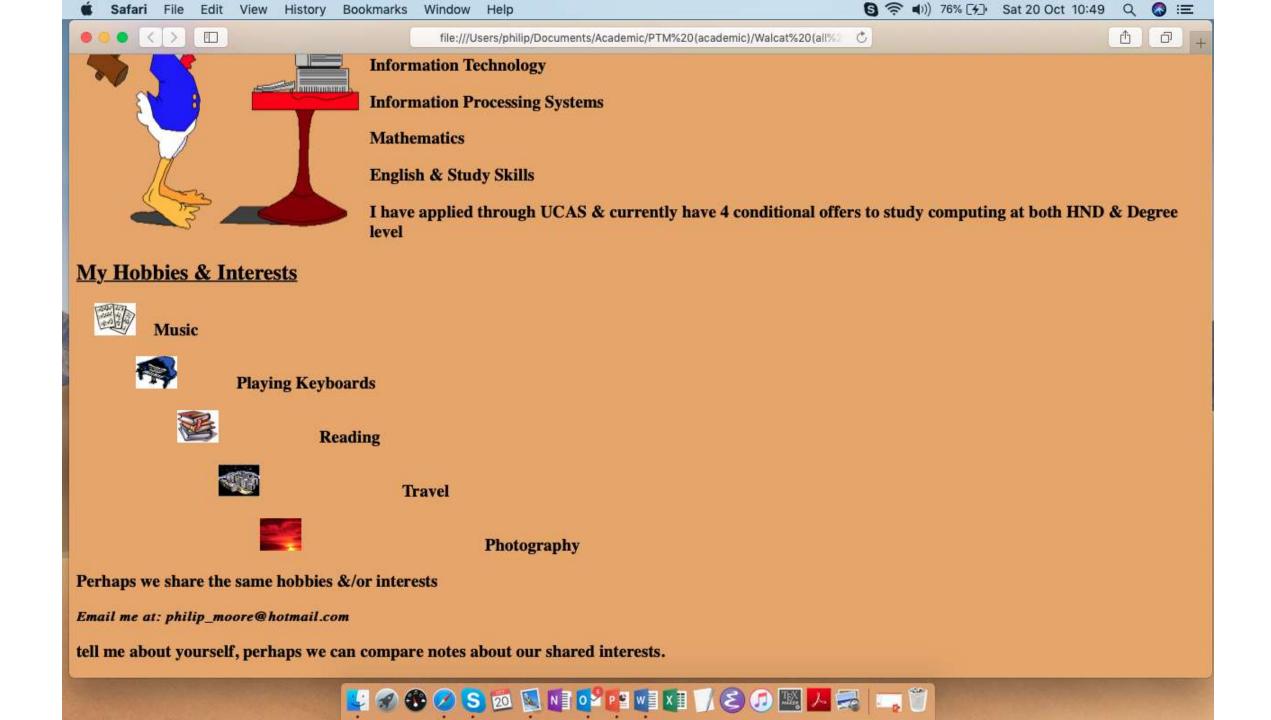


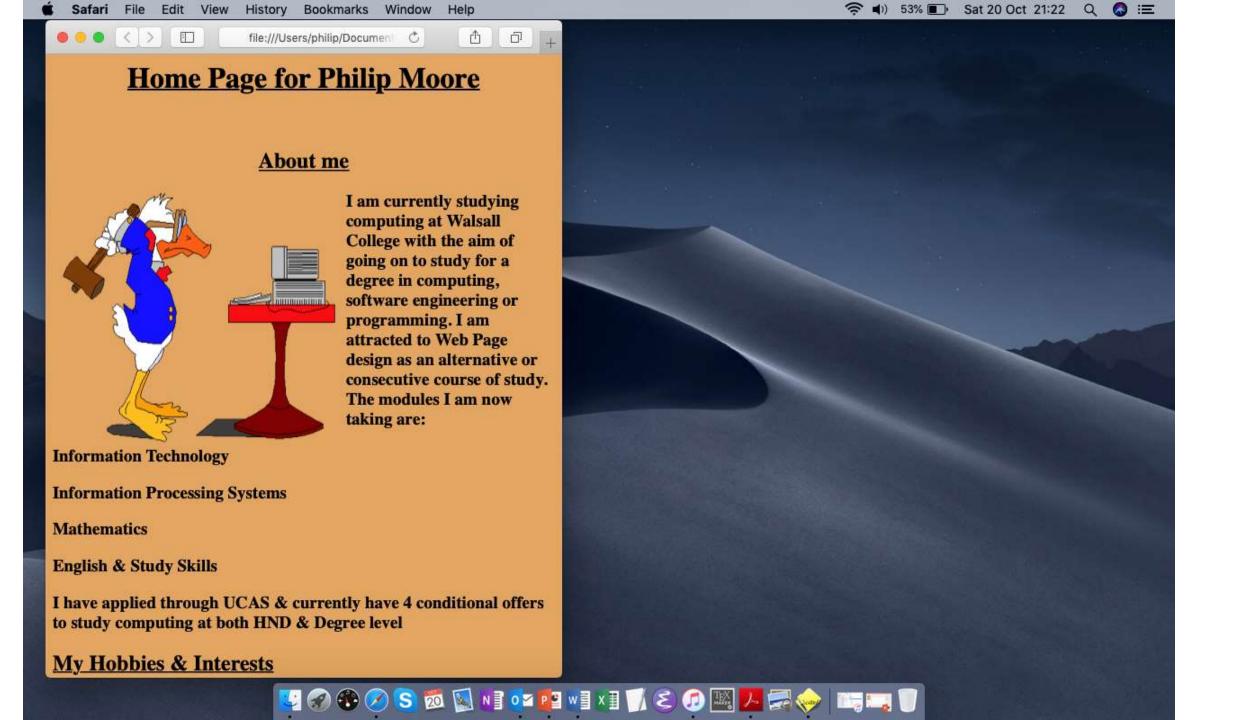


Web Systems Design

- From the previous slides we can see that web site design:
 - Is driven by the requirements of the web site owner
 - There is no one-way to design a web site
 - While there are correct design layouts (good design)
 - There are also incorrect design layouts (bad design)
- The design principles set out by Donald Norman must be applied to all web site design
- The following slides show what not to do!







What to do and what not to do!

- The previous (three) slides show my first attempt at building a web site (in 2001).
- There are parts that are acceptable (not good)
 - The introduction and graphic are acceptable (but can be improved)
- There are parts that are clearly bad design
 - The navigation is very poor (and must be changed)
 - The contact email and related text is bad design (and must be changed)
- The navigation must follow good design principles
- The contact email and related would be better in a separate linked web page with greater detail as required
- The web page is too long revisions as noted will partly correct this problem

Web-Page Layout Semantic Elements Web page Design and Build

The History of HTML

- HTML 2.0
 - November 24 1995
- HTML 3.2
 - January 14 1997
- HTML 4.0
 - December 18 1997
- HTML 4.01
 - December 24 1999
- HTML 5
 - October 28 2014
- HTML 5.1
 - November 1, 2016
- HTML 5.2
 - December 14, 2017



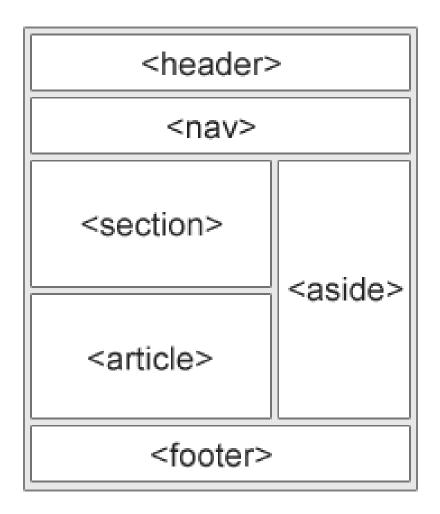
HTML4 Semantic Elements

- What are Semantic Elements?
- A semantic element describes the meaning and purpose to both the web browser and to web designers and developers
- In HTML 4 there were *non-semantic elements* such as:
 - <div>
>
 - These elements define and specify nothing about the content within the tag
- In HTML 4 there were limited semantic elements such as:
 - <form>
 - These elements define and specify the content within the tag

HTML5 Semantic Elements

- Many web sites contain HTML code such as:
 - <div id="nav"> <div class="header"> <div id="footer">
 - To indicate navigation, header, and footer
- HTML5 introduced new semantic elements to define different parts of a web page
- There are 13 *semantic elements* in HTML5
 - <article> <aside> <details> <figcaption> <figure> <footer> <header>
 - <main> <mark> <nav> <section> <summary> <time>
- Details of semantic elements may be found at:
 - The course text book (see pages 38 to 50)
 - The W3schools web site at:
 - https://www.quanzhanketang.com/html/html5_semantic_elements.html

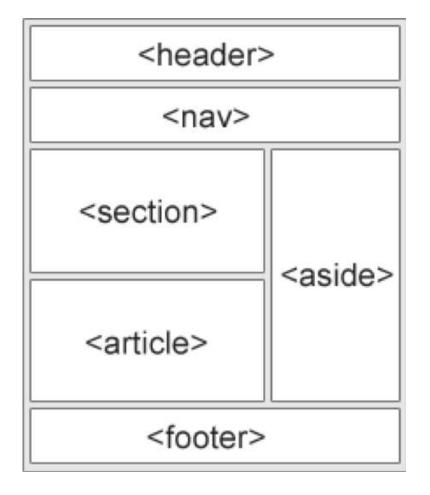
HTML5 Semantic Elements



- The figure shows an example of a web page structure defined using *semantic elements*
- The layout is only an example
- The actual layout of a web page will be defined by the web designer
- In general the location of the <header>
 <nav> <footer> elements are as shown in the figure
 - However: as shown in the previous examples the layout may change to suit the web site requirements

HTML Layout Elements

- <header>
 - Defines a header for a document or a section
- <nav>
 - Defines a container for navigation links
- <section>
 - Defines a section in a document
- <article>
 - Defines an independent self-contained article
- <aside>
 - Defines content aside from the content (like a sidebar)
- <footer>
 - Defines a footer for a document or a section.
- <details>
 - Defines additional details
- <summary>
 - Defines a heading for the <details> element



bringing it all together

- Knowing how to create HTML files and run web-pages / web-sites does not address the need for good web-site design
- Bringing it all together is the process of design where
 - The requirements are investigated and identified
 - The appropriate features are selected
 - The features selected are combined into a web-page / web-site that is
 - Attractive / easy to use / effective in presenting the information / provides user interactive feedback
- It is the task of the designer to
 - Meet the user requirements specification
 - Provide a multi-page web-site with clear navigation that users will want to visit

Usability and Security

- Web-based systems must address a two important factors:
 - Usability (and)
 - Security
- Usability:
 - Relates to how easy users can access a web-site and use the functions
 - Usability applies to both *stateless* (static) and *stateful* (dynamic) web sites
- Security:
 - While we must consider security for both stateless and stateful web sites
 - Security is very important for *stateful* web sites
 - For example: security is essential in an on-line banking web-site / mobile application
- There is a *trade-off* between *usability* and *security*

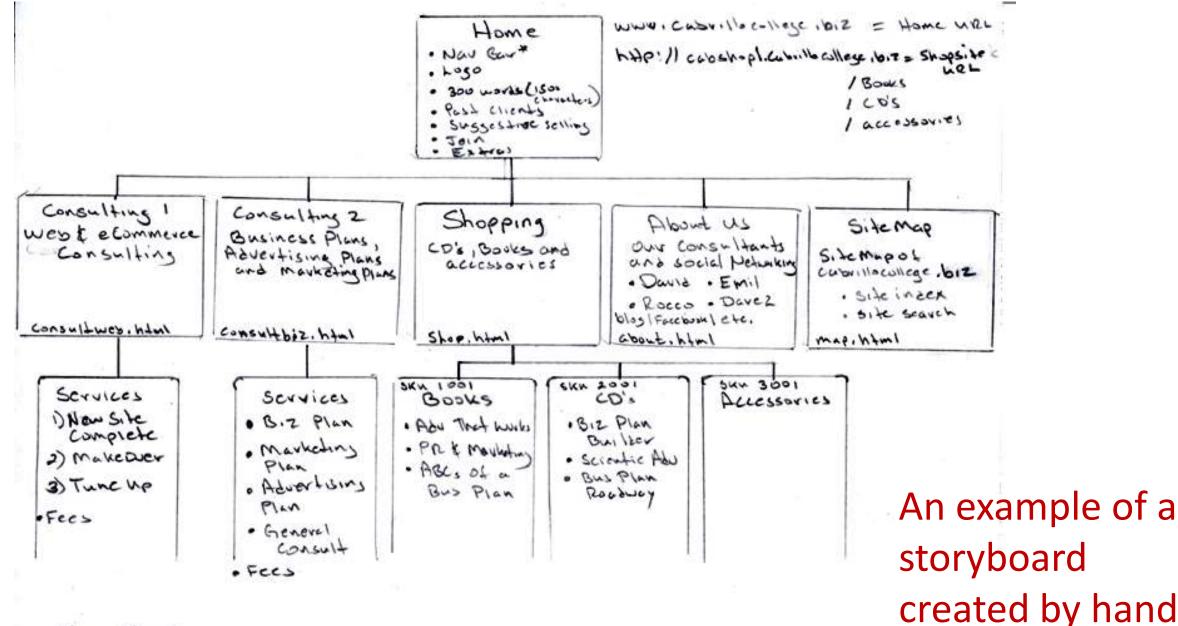
The Design Process

To Design a Web Site

- Carry out the following steps:
 - Identify and document the requirements and uses for the web site
 - Design your web site on paper (termed the LoFi design process)
 - Create a 'storyboard'
 - A storyboard is used in software development to identify the specifications for a specific web site
 - During the specification phase, screens that the web site pages will display are drawn, either on paper or using other specialized modelling software, (to illustrate the important steps of the user experience)
 - The storyboard is then modified to address the specific needs of the web site

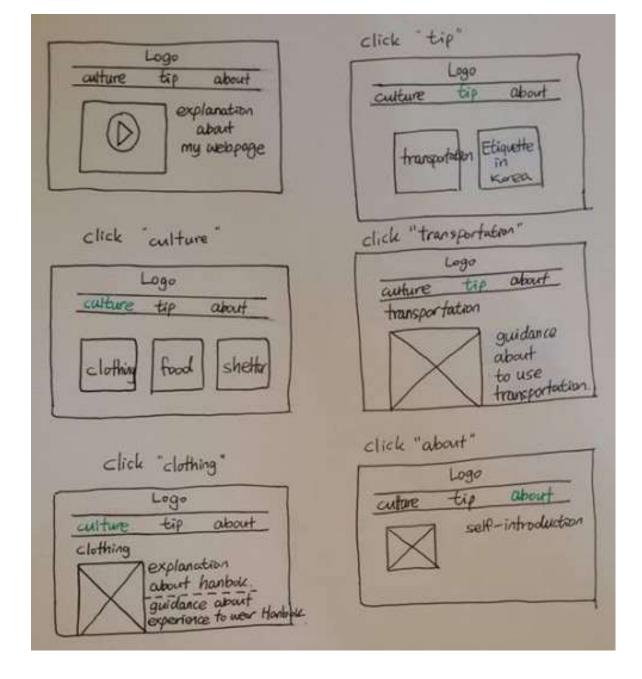
Create a Storyboard

- A storyboard shows how the web site (and the related links) will look and function
- The design process where the design is modelled using a storyboarding and the related testing (of the design) with potential users is termed the LoFi design process)
- It is cheaper to make changes to a storyboard than a built web site
 - The following slides show three typical examples of storyboarding
 - The following slides give typical examples of storyboards created using the manuscript (drawn by hand) and using specialized software
- 1 hour spent on design will save 10 hours spent in building a web site

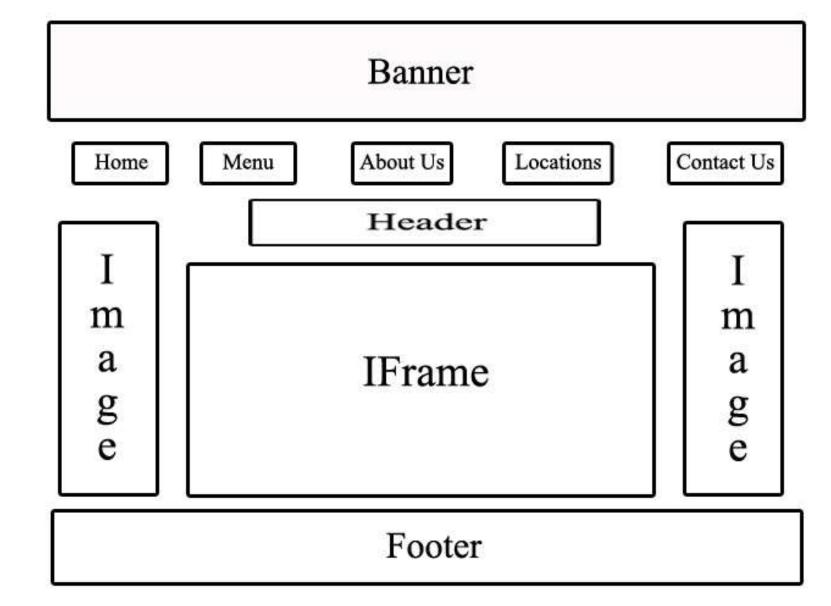


Home | west eCommerce | Business Plans | Shappins | About us | Site Map

 An example of a storyboard created by hand



- An example of a storyboard created using specialised software
- Note: the relationship to the HTML 5 semantic elements



- An example of a storyboard created using specialised software
- Note: the relationship to the HTML 5 semantic elements

Banner

Links to other website sub pages goes in here!

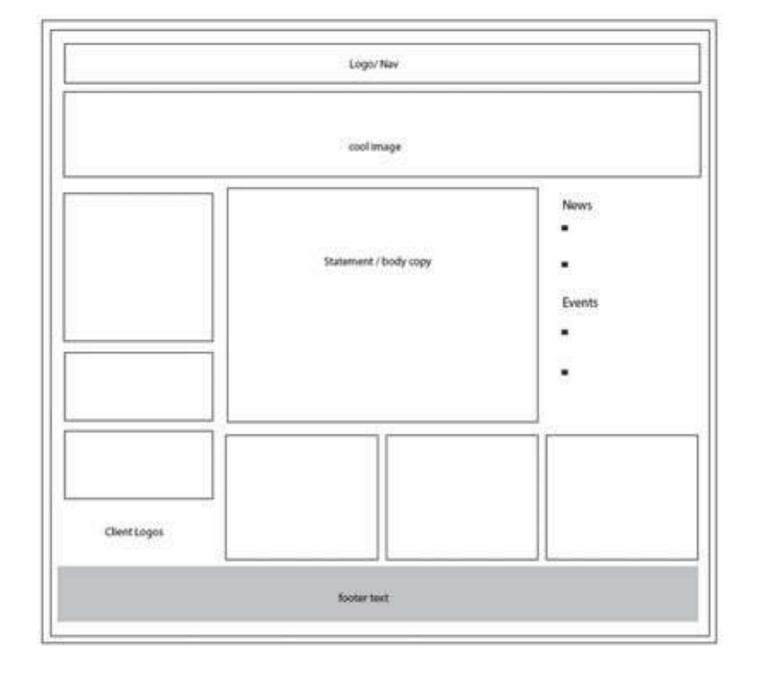
My Picture here!!

Links to other peoples or companies websites herel

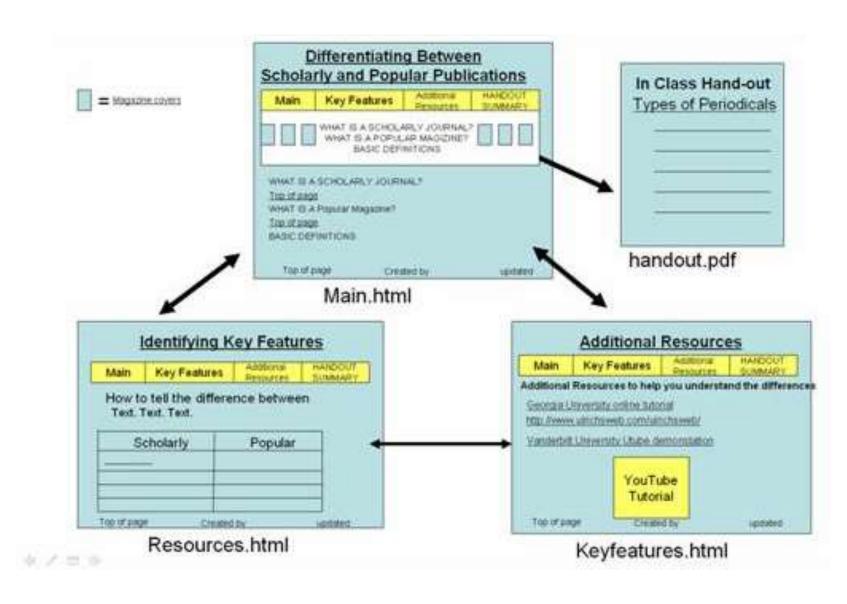
Information in a paragraph goes here!

Sub-page links here, too!

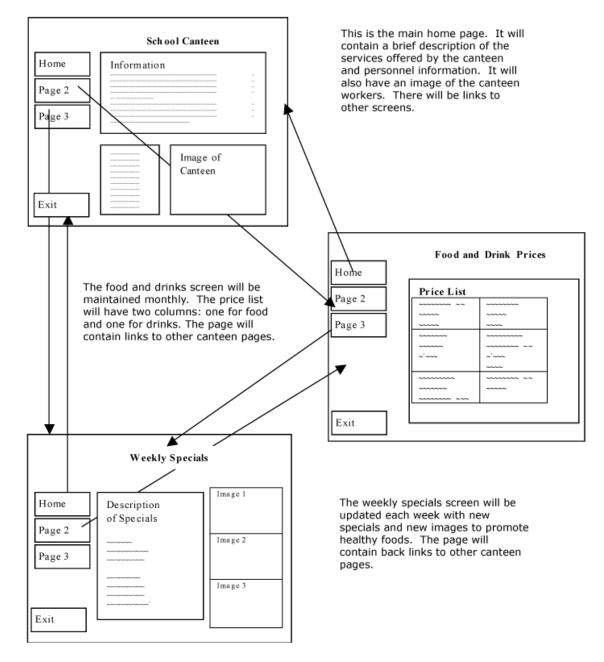
- An example of a storyboard created using specialised software
- Note: the relationship to the HTML 5 semantic elements



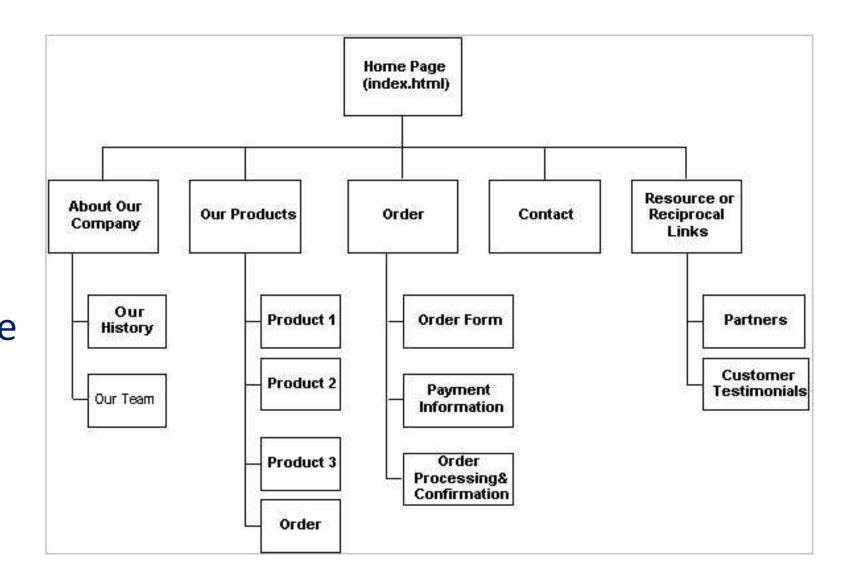
- An example of a storyboard created using specialised software
- Note: this storyboard uses specific web site design software

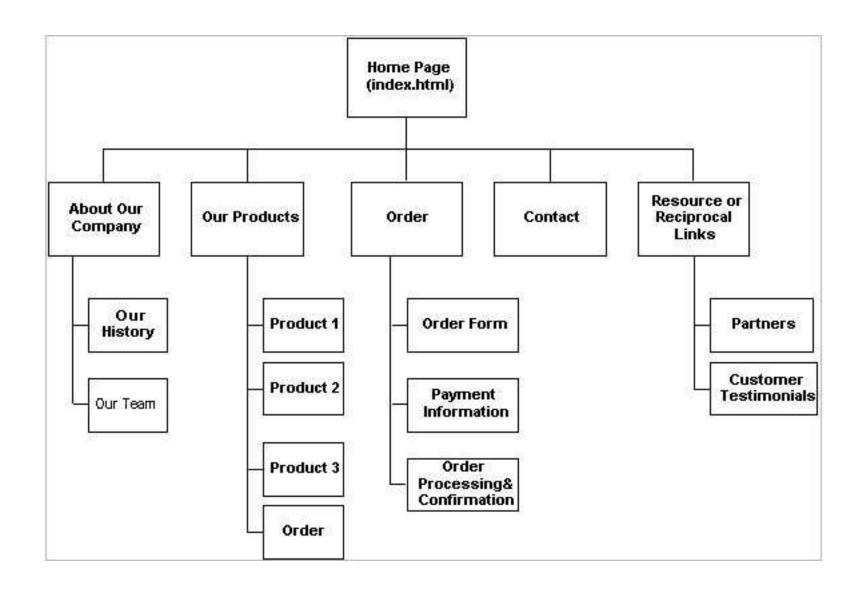


- An example of a storyboard created using specialised software
- Note:
 - The identification of the navigation links
 - The notes used in the creation of the web site model



- An example of a storyboard created using specialised software
- Note: the use of the tree structure to create a model of the web site





- In designing your own personal web site:
 - Prepare a storyboard showing:
 - The home web page (and)
 - The linked HTML web pages (showing information about you)
 - The site map
 - On the storyboard show:
 - The home page (*index*) file
 - Associated related files
 - The links to any related files (e.g., a separate file holding your images)

Modelling

- We have introduced:
 - Conceptual models
 - Storyboarding
- In is important to note that:
 - In computer design the creation of models at the design stage is a frequent and regular task
 - Modelling can be used in the design of many components and elements including: web sites / databases / networks / information systems / computer programs / etc

The Build Process

Web Site Building (1)

- When you are happy with your storyboard design
 - Create the home web page (*index.html*) structure (the layout) including the navigation links to the related web pages
- The structure for all the web pages (the layout) can be defined in the home web page (index.html)
- The index.html web page may be used as a template and copied (using a different page specific name - *.html) and used throughout the web site
 - This ensures consistency, saves time, and reduces errors

Web Site Building (2)

- When the home web page (index.html) structure and layout (including the navigation links) is complete:
 - Make renamed copies for each of the related web pages
- All the html web pages and related files plus the images must be in the same project folder
- You can then proceed to complete the HTML content in the *index.html* file and the related html web pages

Web Site Testing

- As the build continues periodically test the work in progress
- On completion of the build test and validate the final web site
- Remember to place a link in the footer of each web page to
 - A link (URL) to return to the top of the web page
- Also remember to:
 - Create uniform navigation across all web pages to include links (URL's)
 to all web pages in the web site

Review

- The previous slides have set out a skeleton structure and principles for the design and building of web sites
- The steps are:
 - 1. Identify and document the requirements and uses for the web site
 - 2. Design your web site on paper (or specialized modelling software) using a storyboard
 - 3. Create the home web page (index.html) setting out the structure (layout) with uniform navigation links
 - 4. Copy the index.html file and rename it (*.html)for each of the related web pages
 - 5. Complete the content (the HTML) for each of the web pages with return links.
 - 6. Test and validate the web site including the navigation and links across multiple platforms and devices

Create a Web Page

- In this course we have created web pages using the NetBeans IDE where we have written the HTML, JavaScript, and PHP and run the web pages using NetBeans
- As I have noted HTML, JavaScript, and PHP can be written without using an integrated development environment you can use a simple text editor such as Notepad (or) an application such as emacs using the following steps:
 - Design your web page on paper (LoFi design)
 - Write the HTML, JavaScript, and PHP
 - Save the file using the correct format (e.g., index.html)
 - Click on the *index* file (it will run in a web browser see the following example)

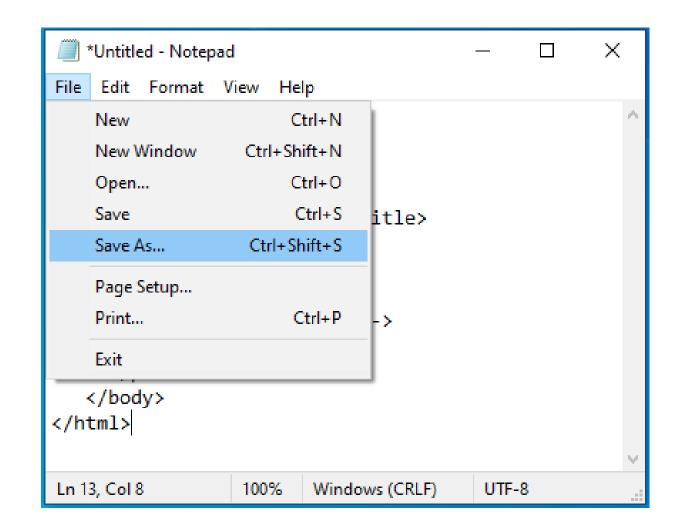
Write the HTML

- In this example the web page has been written using the Microsoft Notepad application in Windows 10
- Note: do not use a word processor such as Word
- This is the basic template for all HTML files

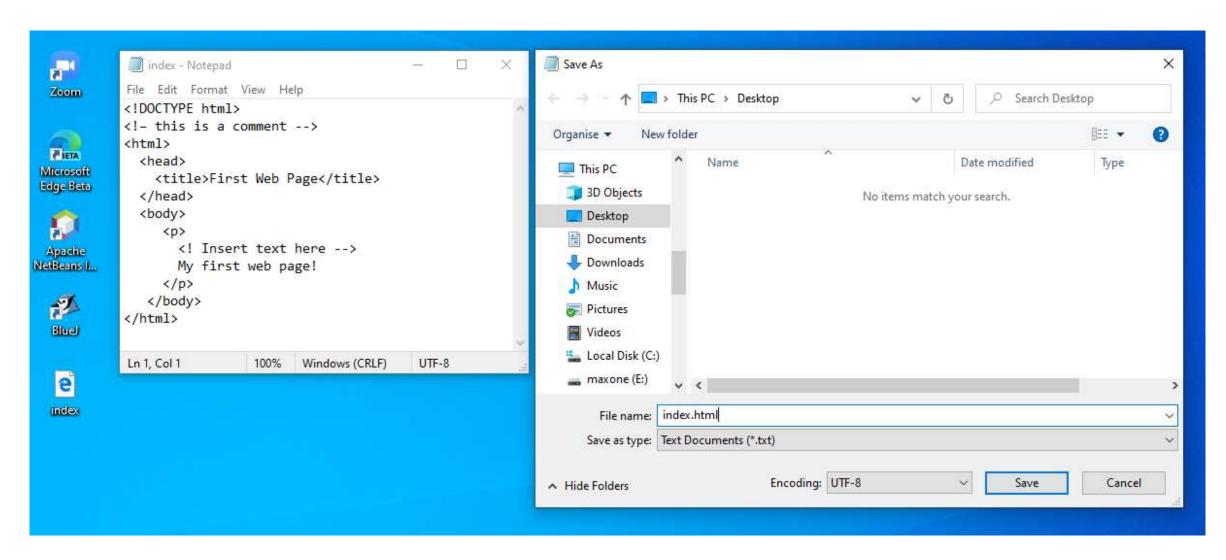
```
X
   *Untitled - Notepad
   Edit Format View Help
<!DOCTYPE html>
<!- this is a comment -->
<html>
  (head)
    <title>First Web Page</title>
  </head>
  <body>
     >
       <! Insert text here -->
       My first web page!
     </body>
</html>
                  100%
Ln 13, Col 8
                         Windows (CRLF)
                                         UTF-8
```

Save the File

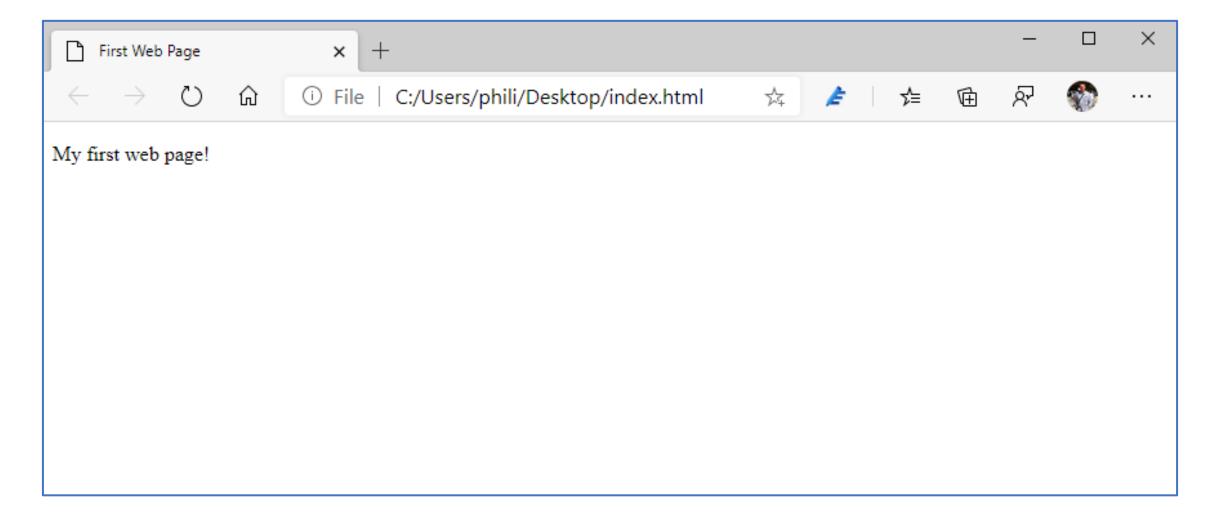
- Shown is the saving of the Notepad text file
- Use Ctrl + s (capitals are not needed)
- The following slides show the actual save and the output in a web browser



Create the index.html File



The index.html in MS Edge



Software Design and Web Page Building

- We have introduced and demonstrated:
 - The principles of software design and development for computer programs and web sites including an overview of feedback, testing, and validation
 - Design models and the use of simple LoFi design using paper and pen
 - The design, creation, and running of a HTML file
 - The output in a web browser
- Design is important:
 - Time spent on design will save many wasted hours!
 - 1 hour spent on design will save 10 hours spent in building