

# **Comp 125 - Visual Information Processing**

---

Spring Semester 2019 - Week 11 - Monday

Dr Nick Hayward

# Final Demo and Presentation

---

## Dates

- Week 15 - Wednesday 24th & Friday 26th April 2019
- Final report due 3rd May 2019 by 1.25pm

# Final Demo and Presentation

---

- group project - maximum 3 persons per group
- develop an app concept and prototype
- working app (as close as possible...)
  - *must use technologies outlined during the course*
- show and explain code used to develop the app
- explain design decisions
  - *describe patterns used in design of app*
  - *layout choices...*
- show and explain implemented differences
  - *where and why did you update the app?*
  - *perceived benefits of the updates?*

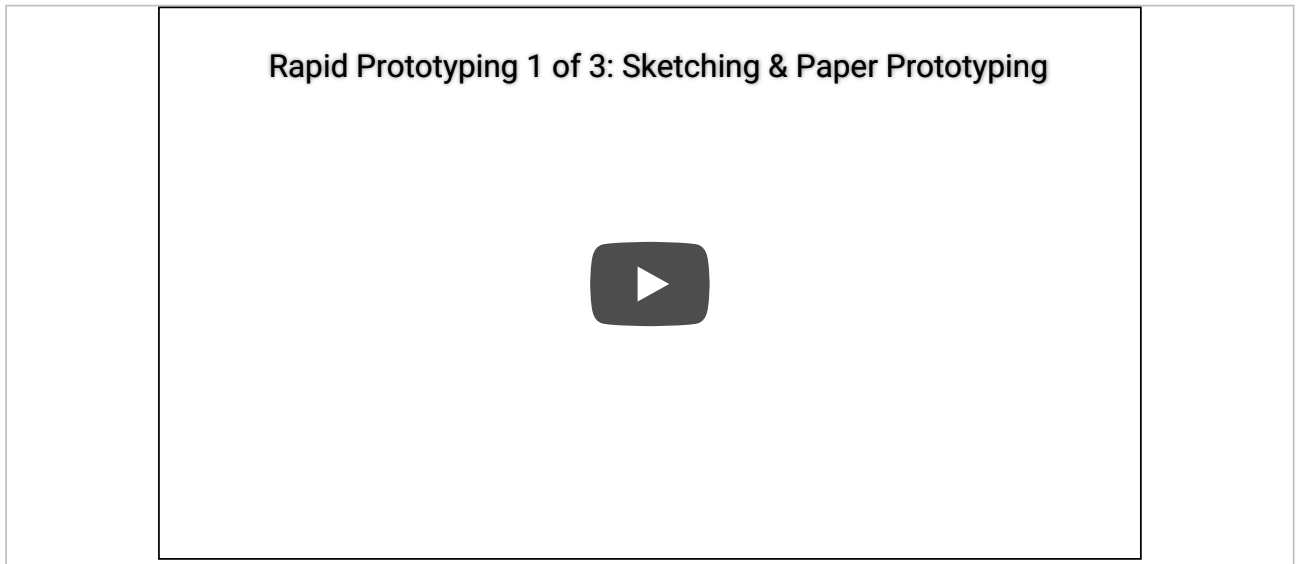
Further details on course website -

<https://csteach125.github.io/coursework/#assessment2>

## Video - Design

---

### Paper Prototyping



Rapid Prototyping I of 3: Paper Prototyping

Source: YouTube - Google

# HTML5 - semantic elements and structure - intro

---

- perceived issue or concern with HTML5 semantic elements
  - *how and when to add them to our document*
  - *where and when do we add them to our page?*
- non-semantic elements often considered simpler to apply
  - *generalised application and context for usage*

# HTML5 - semantic elements and structure

---

## header and nav

- `<header>`
  - *used to collect and contain introductory content*
  - *semantically appropriate for the head or top of a page*
  - *technically feasible and acceptable to include multiple `<header>` elements*
  - *e.g. `<header>` within main content, sidebar content, an article, a section...*
  
- `<nav>`
  - *short for navigation*
  - *stores and defines a set of links for internal or external navigation*
  - *not meant to define all page navigation links*
  - *often considered suitable for primary site links*
  - *additional links can be placed in*
  - *sidebar, footer, main content...*
  - *no need to consider a `<nav>` element for these links...*

# HTML5 - semantic elements and structure

---

## main

- this element tag defines our **main** content
- traditionally the central content area of our page or document
- HTML4 often used a `<div>` element
  - *plus a class or id to define central content*
  - e.g.

```
<!-- e.g. HTML4 main content -->
<div id="main">
  ...
</div>
```

- HTML5 semantically defines and marks content as `<main>`
- `<main>` should not include any page features such as
  - *nav links, headers etc, that are repeated across multiple pages*
- cannot add multiple `<main>` elements to a single page
- must not be structured as a child element to
  - *<article>, <aside>, <footer>, <header>, or <nav>*

# HTML5 - semantic elements and structure

---

## section, article, aside - part I

- `<section>`
  - *defines a section of a page or document*
  - *W3C Documentation defines as follows,*

*a section is a thematic grouping of content. The theme of each section should be identified, typically by including a heading as a child of the section element.*

- a site can be sub-divided into multiple `<section>` groupings
  - *e.g. as we might consider a chapter or section break in a book...*
- `<article>`
  - *suitable for organising and containing independent content*
  - *include multiple `<article>` elements within a page*
  - *use to establish logical, individual groups of content*
  - *again, newspaper analogy is useful to remember*
  - *e.g. a blog post, story, news report...might be a useful article*
  - *key to using this element is often whether content can be used in isolation*
- `<aside>`
  - *used to define some content aside from containing parent content*
  - *normally used to help define or relate material to surrounding content*
  - *effectively acts as supporting, contextual material*



# HTML5 - semantic elements and structure

---

## section, article, aside - part 2

- MDN Documentation suggests,

*if it makes sense to separately syndicate the content of a `<section>` element, use an `<article>` element instead*

and

*do not use the `<section>` element as a generic container; this is what `<div>` is for, especially when the sectioning is only for styling purposes. A rule of thumb is that a section should logically appear in the outline of a document.*

# HTML5 - semantic elements and structure

---

## figure, figcaption

- `<figure>` & `<figcaption>`
  - *as with print media, we can logically group image and caption*
  - *`<figure>` acts as parent for image grouping*
  - *child elements include*
  - *`<img>` and `<figcaption>`*

```
<figure>

<figcaption>Ptolemaic temple at Philae, Egypt</figcaption>
</figure>
```

- updated demo with figure grouping - Demo - Semantic structuring

# HTML5 - semantic elements and structure

---

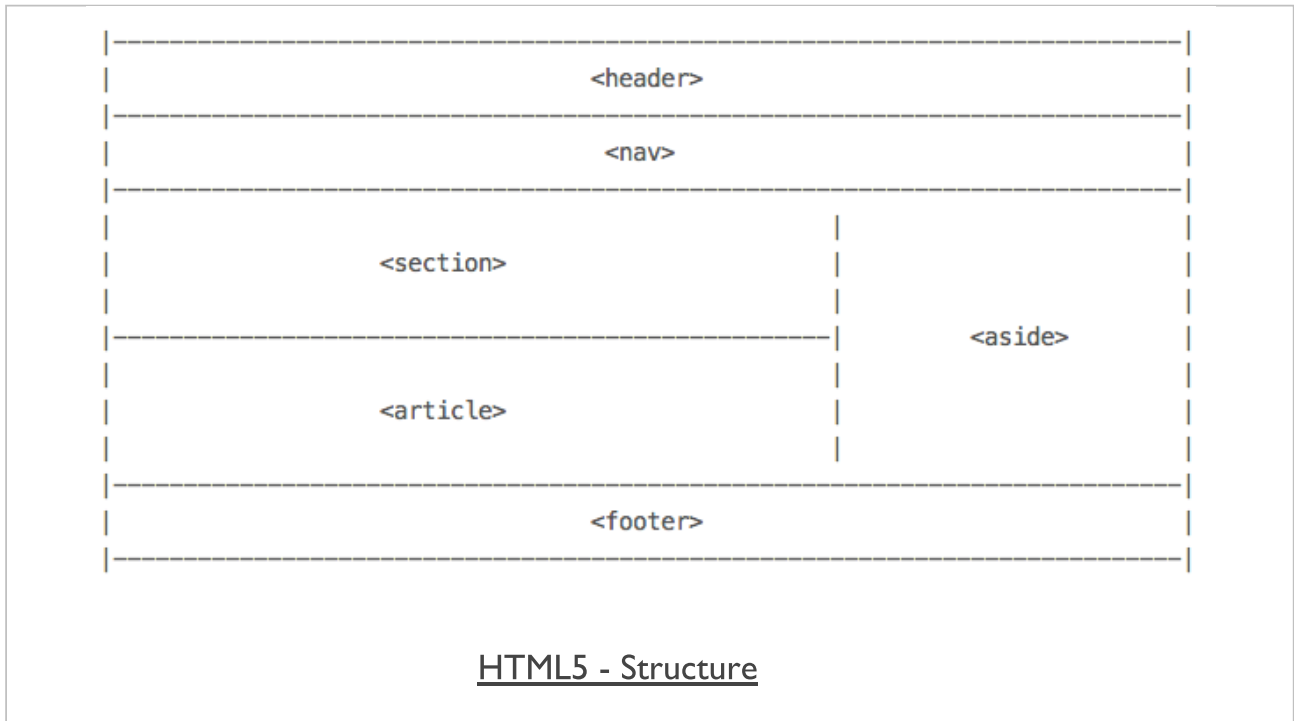
## footer

- `<footer>`
  - *usually contains information about its containing element*
- example 1 - in a footer for an article
  - *might use this element to define and record*
  - *author of the article*
  - *publication date*
  - *suitable tags or metadata*
  - *associated documents...*
- example 2 - a footer simply placed at the **foot** of a page
  - *record copyright information*
  - *contextual links*
  - *contact information*
  - *small logos...*
- example 2 considered standard usage for `<footer>`
  - *continues from HTML4 and earlier generic usage...*

# Image - HTML5 page structure - part I

---

## semantic elements



# Image - HTML5 page structure - part 2

---

## semantic elements

