

Agile Software Development

Produced
by

Eamonn de Leastar (edeleastar@wit.ie)

Department of Computing, Maths & Physics
Waterford Institute of Technology

<http://www.wit.ie>

<http://elearning.wit.ie>



Waterford Institute of Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE



Semantic HTML + UI

HTML5

A subset of
Section 4 of the
HTML5 spec...

[4 The elements of HTML](#)

[4.1 The root element](#)

[4.1.1 The `html` element](#)

[4.2 Document metadata](#)

[4.2.1 The `head` element](#)

[4.2.2 The `title` element](#)

[4.2.3 The `base` element](#)

[4.2.4 The `link` element](#)

[4.2.5 The `meta` element](#)

[4.2.5.1 Standard metadata names](#)

[4.2.5.2 Other metadata names](#)

[4.2.5.3 Pragma directives](#)

[4.2.5.4 Other pragma directives](#)

[4.2.5.5 Specifying the document's character encoding](#)

[4.2.6 The `style` element](#)

[4.2.7 Styling](#)

[4.3 Sections](#)

[4.3.1 The `body` element](#)

[4.3.2 The `article` element](#)

[4.3.3 The `section` element](#)

[4.3.4 The `nav` element](#)

[4.3.5 The `aside` element](#)

[4.3.6 The `h1`, `h2`, `h3`, `h4`, `h5`, and `h6` elements](#)

[4.3.7 The `header` element](#)

[4.3.8 The `footer` element](#)

[4.3.9 The `address` element](#)

[4.3.10 Headings and sections](#)

[4.3.10.1 Creating an outline](#)

[4.3.10.2 Sample outlines](#)

[4.3.11 Usage summary](#)

[4.3.11.1 Article or section?](#)

[4.4 Grouping content](#)

[4.4.1 The `p` element](#)

[4.4.2 The `hr` element](#)

[4.4.3 The `pre` element](#)

[4.4.4 The `blockquote` element](#)

[4.4.5 The `ol` element](#)

[4.4.6 The `ul` element](#)

[4.4.7 The `li` element](#)

[4.4.8 The `dl` element](#)

[4.4.9 The `dt` element](#)

[4.4.10 The `dd` element](#)

[4.4.11 The `figure` element](#)

[4.4.12 The `figcaption` element](#)

[4.4.13 The `div` element](#)

[4.4.14 The `main` element](#)

[4.5 Text-level semantics](#)

[4.5.1 The `a` element](#)

[4.5.2 The `em` element](#)

[4.5.3 The `strong` element](#)

[4.5.4 The `small` element](#)

[4.5.5 The `s` element](#)

[4.5.6 The `cite` element](#)

[4.5.7 The `q` element](#)

[4.5.8 The `dfn` element](#)

[4.5.9 The `abbr` element](#)

[4.5.10 The `data` element](#)

[4.5.11 The `time` element](#)

[4.5.12 The `code` element](#)

[4.5.13 The `var` element](#)

[4.5.14 The `samp` element](#)

[4.5.15 The `kbd` element](#)

[4.5.16 The `sub` and `sup` elements](#)

[4.5.17 The `i` element](#)

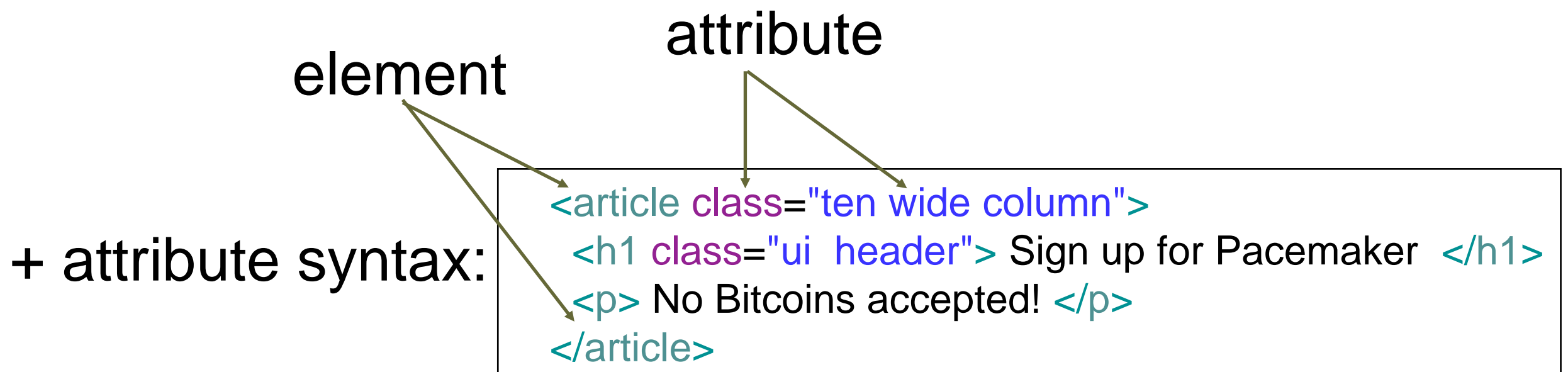


HTML5

A vocabulary and associated APIs for HTML and XHTML

W3C Recommendation 28 October 2014

<i>metadata</i>	• <code><html> <head></code>
<i>sections</i>	• <code><body><section><article><nav><aside><header><footer><h1><h2><h3></code>
<i>grouping</i>	• <code><p> <div></code>
<i>links</i>	• <code><a></code>
<i>embedding</i>	• <code></code>
<i>forms</i>	• <code><form></code>
<i>tabular data</i>	• <code><table></code>



The Need for Semantic Elements

- Give content on the page **meaning** and **structure**.
- Semantics portray the value of content on a page, and are not just its style.

- Semantic markup can be interpreted by tools for the visually impaired.
- Search Engines can use semantic markup to better categorise and classify content.

- Semantic markup can make site maintenance easier as new developers can quickly grasp the site structure.
- Updating / Restyling may be streamlined by using semantic elements.

The HTML5 Semantic Elements

<header>

<nav>

<section>

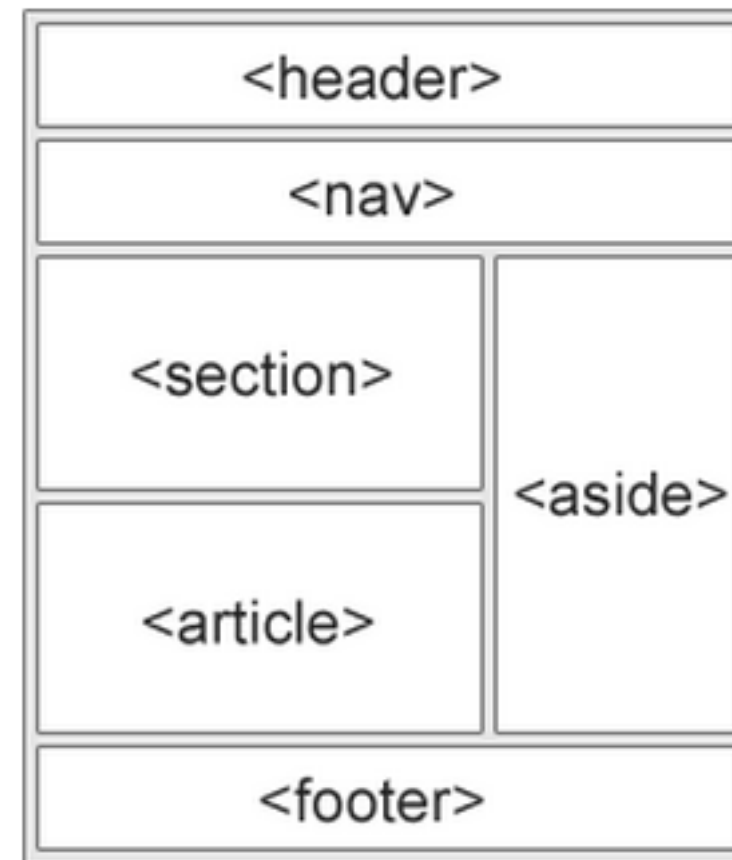
<article>

<aside>

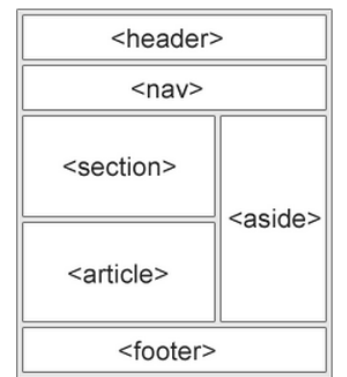
<figcaption>

<figure>

<footer>



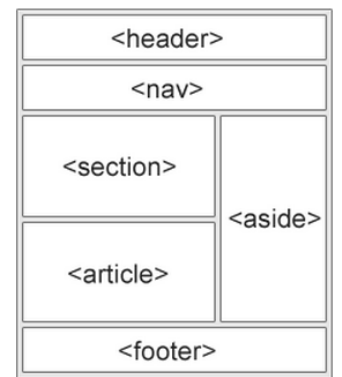
<nav>



```
<nav>
<h1>Navigation</h1>
<ul>
  <li><a href="articles.html">Index of all articles</a></li>
  <li><a href="today.html">Things sheeple need to wake up for today</a></li>
  <li><a href="successes.html">Sheeple we have managed to wake</a></li>
</ul>
</nav>
```

"The nav element represents a section of a page that links to other pages or to parts within the page: a section with navigation links."

<article>



```
<article itemscope itemtype="http://schema.org/BlogPosting">
  <header>
    <h1 itemprop="headline">The Very First Rule of Life</h1>
    <p><time itemprop="datePublished" datetime="2009-10-09">3 days ago</time></p>
    <link itemprop="url" href="?comments=0">
  </header>
  <p>If there's a microphone anywhere near you, assume it's hot and
    sending whatever you're saying to the world. Seriously.</p>
  <p>...</p>
  <footer>
    <a itemprop="discussionUrl" href="?comments=1">Show comments...</a>
  </footer>
</article>
```

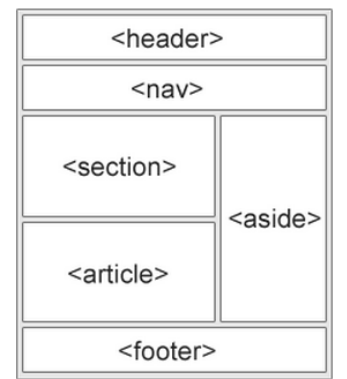
"The article element represents a complete, or self-contained, composition in a document, page, application, or site and that is, in principle, independently distributable or reusable, e.g. in syndication. This could be a forum post, a magazine or newspaper article, a blog entry, a user-submitted comment, an interactive widget or gadget, or any other independent item of content."

A brief note on MicroData and Items

- Sometimes, it is desirable to annotate HTML content with specific machine-readable labels.
- **Microdata** allows nested groups of name-value pairs to be added to documents, in parallel with the existing content.
- The microdata model consists of a group of name-value pairs. The groups are called **items**. Each name-value pair is a property.

itemscope	boolean attribute that creates a new item for the element.
itemtype	Gives the item a URL type (used in conjunction with itemscope).
itemprop	Used on one of the item's descendant's to add a property to the item.

<section>



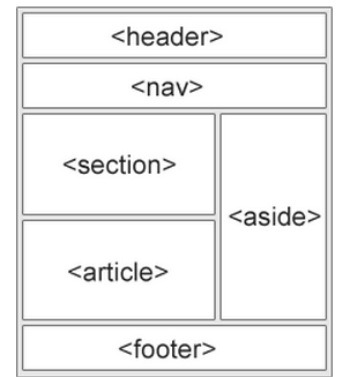
"The section element represents a generic section of a document or application."

A section, in this context, is a thematic grouping of content.

The theme of each section should be identified, typically by including a heading (h1-h6 element) as a child of the section element."

```
<article>
  <header>
    <h2>Apples</h2>
    <p>Tasty, delicious fruit!</p>
  </header>
  <p>The apple is the pomaceous fruit of the apple tree.</p>
  <section>
    <h3>Red Delicious</h3>
    <p>These bright red apples are the most common found in many
      supermarkets.</p>
  </section>
  <section>
    <h3>Granny Smith</h3>
    <p>These juicy, green apples make a great filling for apple
      pies.</p>
  </section>
</article>
```

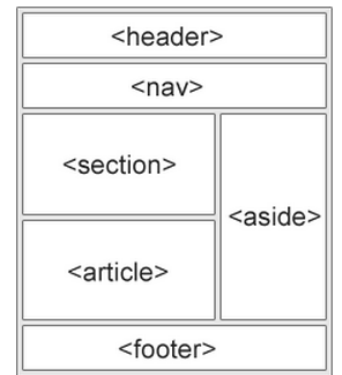
<aside>



"The aside element represents a section of a page that consists of content that is tangentially related to the content around the aside element, and which could be considered separate from that content. Such sections are often represented as sidebars in printed typography."

```
<aside>
<h1>Switzerland</h1>
<p>Switzerland, a land-locked country in the middle of
geographic Europe, has not joined the geopolitical
European Union, though it is a signatory to a
number of European treaties.</p>
</aside>
```

<header> & <footer>



"The header element represents introductory content for its nearest ancestor sectioning content or sectioning root element. A header typically contains a group of introductory or navigational aids."

"The footer element represents a footer for its nearest ancestor sectioning content or sectioning root element. A footer typically contains information about its section such as who wrote it, links to related documents, copyright data, and the like."

```
<header>
  <p>Welcome to...</p>
  <h1>Voidwars!</h1>
</header>
```

```
<footer>
  <a href="..">Back to index...</a>
</footer>
```

<figure> & <figcaption>

The <figcaption> tag defines a caption for a <figure> element.

The <figcaption> element can be placed as the first or last child of the <figure> element.

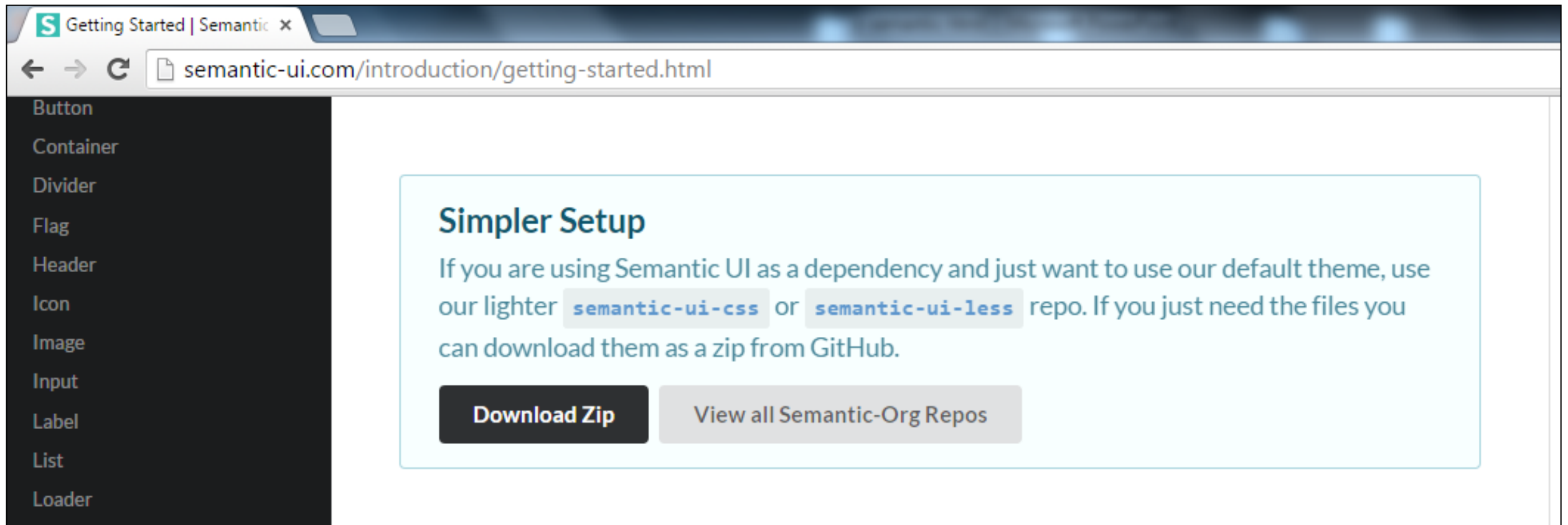
```
<figure>  
    
  <figcaption>Fig1. - A view of the pulpit rock in Norway.</figcaption>  
</figure>
```

Semantic UI

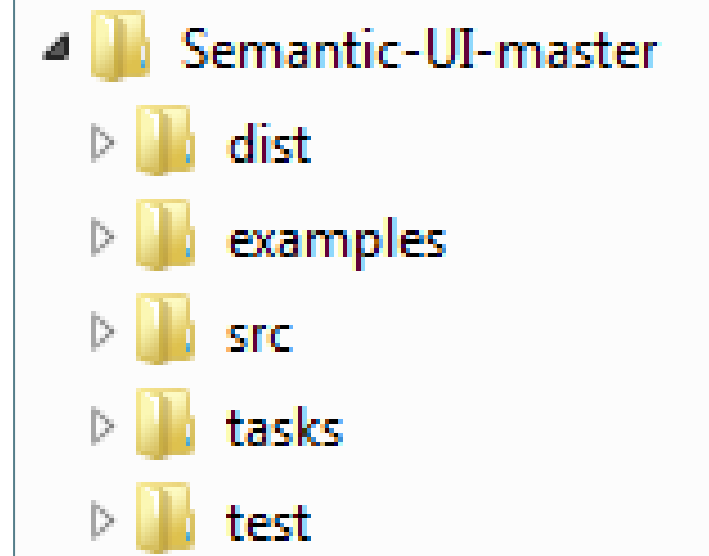


<http://semantic-ui.com/>

Installing Semantic UI as a project dependency

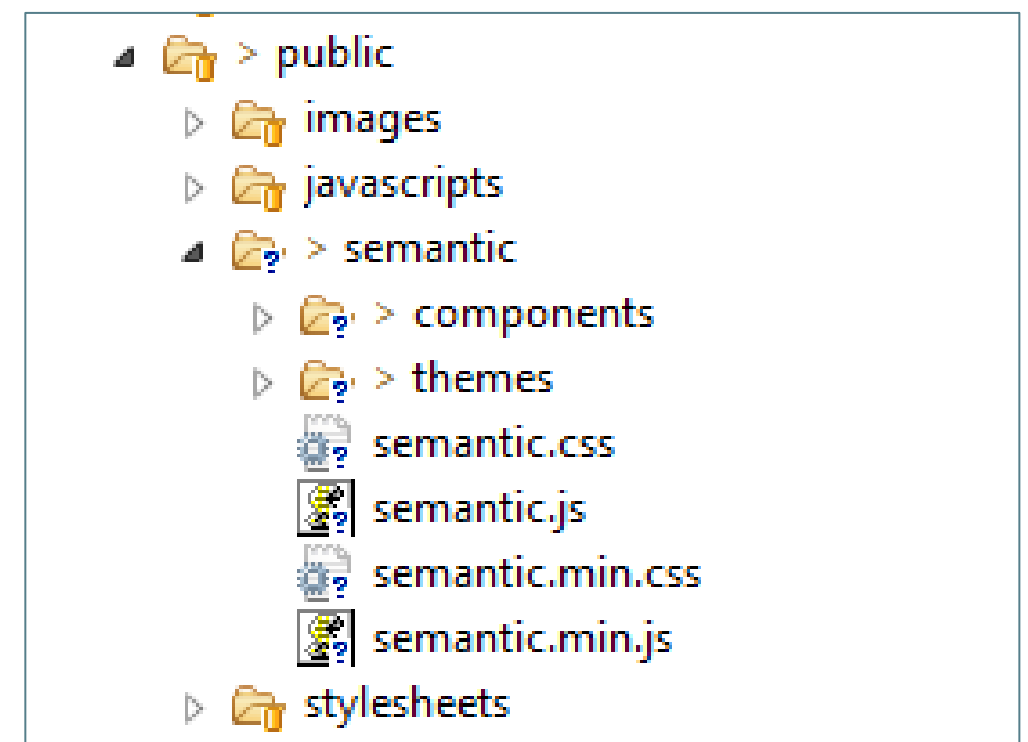
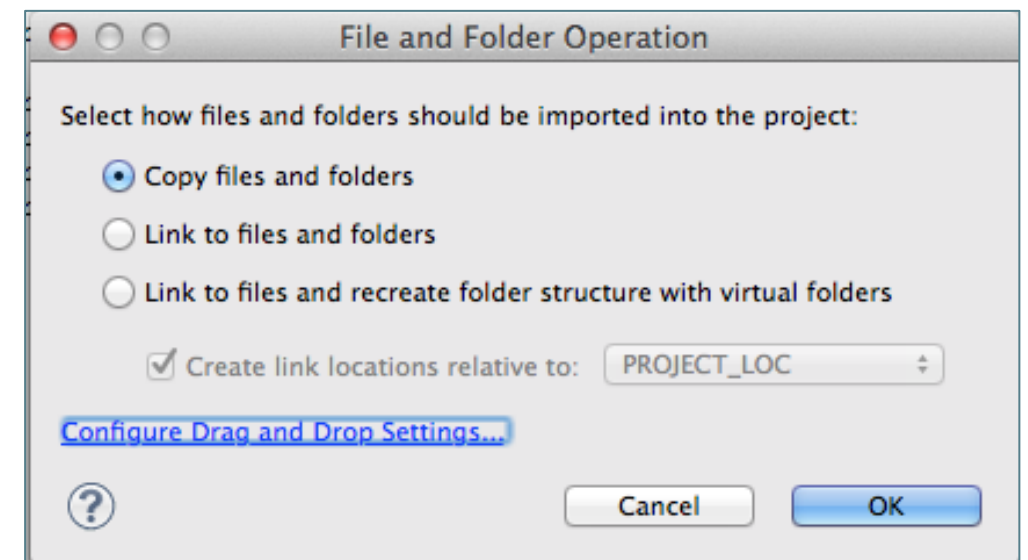


- Download and expand the zip file.
- You should have this file structure →



Installing Semantic UI as a project dependency

- Open your eclipse project and drag the **dist** folder and drop it into the **public** folder.
- When prompted, select the copy files and folders option:
- Once the folder is copied over rename it from **dist** to **semantic**. Your **public** folder structure should now look similar to this:



Terminology – Definitions and Components

- A **definition** is a set of CSS and Javascript which describe a component's essential qualities.
- A **component** is a general term used to refer to any user interface element packaged for distribution.
- Semantic UI classifies **components** into five different **definition types** depending on its qualities:
 - Element
 - Collection
 - Views
 - Modules
 - Behavior

Terminology – Global & Five Component Types

Globals	Globals are styles that are applied across a site. These include things like sitewide font, link and sizing defaults.
Element	UI elements are page elements with a single function. They can exist alone or in a plural form with elements sharing qualities e.g.: A group of buttons may use ui red buttons as a grouping with individual ui button children.
Collection	Collections are heterogeneous groups of components which are usually found together. They describe a list of "usual suspects" which appear in a certain context.
Views	A view is a convention for presenting specific types of content that is usually consistent across a website. These include things like comments , activity feeds , etc.
Modules	Modules are components that include both a definition of how they appear and how they behave. These include components like dropdowns, popups, etc.
Behaviors	Behaviors are standalone Javascript components that describe how page elements should act, but not how they should appear. Behaviors include things like form validation, state management, and API request routing.

Component Types Catalogue

Elements

Button
Container
Divider
Flag
Header
Icon
Image
Input
Label
List
Loader
Rail
Reveal
Segment
Step

Collections

Breadcrumb
Form
Grid
Menu
Message
Table

Views

Advertisement
Card
Comment
Feed
Item
Statistic

Modules

Accordion
Checkbox
Dimmer
Dropdown
Embed
Modal
Nag
Popup
Progress
Rating
Search
Shape
Sidebar
Sticky
Tab
Transition

Behaviors



API
Form Validation
Visibility

Terminology - Definition Sections



- All UI components of a single type are defined similarly

Types	Types are mutually exclusive versions of an element that modify the element's standard appearance.
Variations	Variations modify qualities of an element like size, or color. Variations are mutually inclusive, and can be used together to modify an element.
Content	Content are parts which only have meaning in the context of a component. Content use names which describe the type of expected content like header, description, menu, or item.
States	States are modifications to an element that help indicate a change in affordances . Common states include loading, disabled, hovered and active.
Behaviors	Behaviors are actions that a component can perform. Behaviors are represented by simple phrases like "set value" or "increment", that can be invoked in Javascript..

Components (Elements & Collections)

Elements	<p>Group An element can optionally define how attributes can be shared across a group</p> <hr/> <p> UI Button</p>
Collections	<p>Content A collection can define elements which might be found inside</p> <hr/> <p>States A collection may define states for content elements or itself</p> <hr/> <p>Variations A collection may define variations for content elements or itself</p> <hr/> <p> UI Form</p>

Components (Views, Modules & Behaviours)

Views	<p>Content A view may define elements which can exist inside of the view</p> <hr/> <p>States A view may define states for content elements or itself</p> <hr/> <p>Variations A view may define variations for a content elements or itself</p> <hr/> <p> UI Item</p>
Modules & Behaviors	<p>Behavior A module will define a set of behaviors which can be used as an API</p> <hr/> <p>Settings A settings object which can alter the default behavior when instantiating a module</p> <hr/> <p>Examples A list of examples to showcase the variations in behavior of a module</p> <hr/> <p> UI Dropdown</p>

Component Types Catalogue

Elements

Button
Container
Divider
Flag
Header
Icon
Image
Input
Label
List
Loader
Rail
Reveal
Segment
Step

Collections

Breadcrumb
Form
Grid
Menu
Message
Table

Views

Advertisement
Card
Comment
Feed
Item
Statistic

Modules

Accordion
Checkbox
Dimmer
Dropdown
Embed
Modal
Nag
Popup
Progress
Rating
Search
Shape
Sidebar
Sticky
Tab
Transition

Behaviors

API
Form Validation
Visibility

Elements

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante.

Segment

A segment is used to create a grouping of related content.



Image

An image is a graphic representation of something

 FOLLOW



Button

A button indicates a possible user action.



Divider

A divider visually segments content into separate groups

First **Second** Third Last

Step

Steps show the current activity in a series of steps.



Icon

An icon is a glyph used to represent another concept more simply

Section 2

The second section of the website

Header

Headers provide a short summary of content

DOG

HTML

NEW

Label

Labels give descriptions to sections of content.

Collections


Food / Fruit / Apples

Food > Fruit > Apples

Breadcrumb

A breadcrumb is a menu to show the location of the current section in relation to other sections.

Name

 Name

E-mail

E-mail

Form

A form is used to solicit a user response

1

2

3

Grid

A grid helps harmonize negative space in a layout

Friends

Messages

Profile

Friends

Messages

Profile

Menu

A menu organizes related links

We're sorry we can't process your idea just yet

Please enter your name

Message

Messages alert a user to something important.

Name

Status

John

Approved

John

Unconfirmed

Sally

Denied

Table

A table collects related data into rows of content

ui – Special Class

- **ui** is a special class name used to distinguish parts of components from components e.g.
 - a **list** will receive the class **ui** **list** because it has a corresponding definition, however a list item, will receive just the class **item**.
- The **ui** class name helps encapsulate CSS rules by making sure all 'parts of a component' are defined in context to a 'whole' component.
- It also helps make scanning unknown code simpler. If you see **ui** you know you are looking at a component.

List

A list groups related content

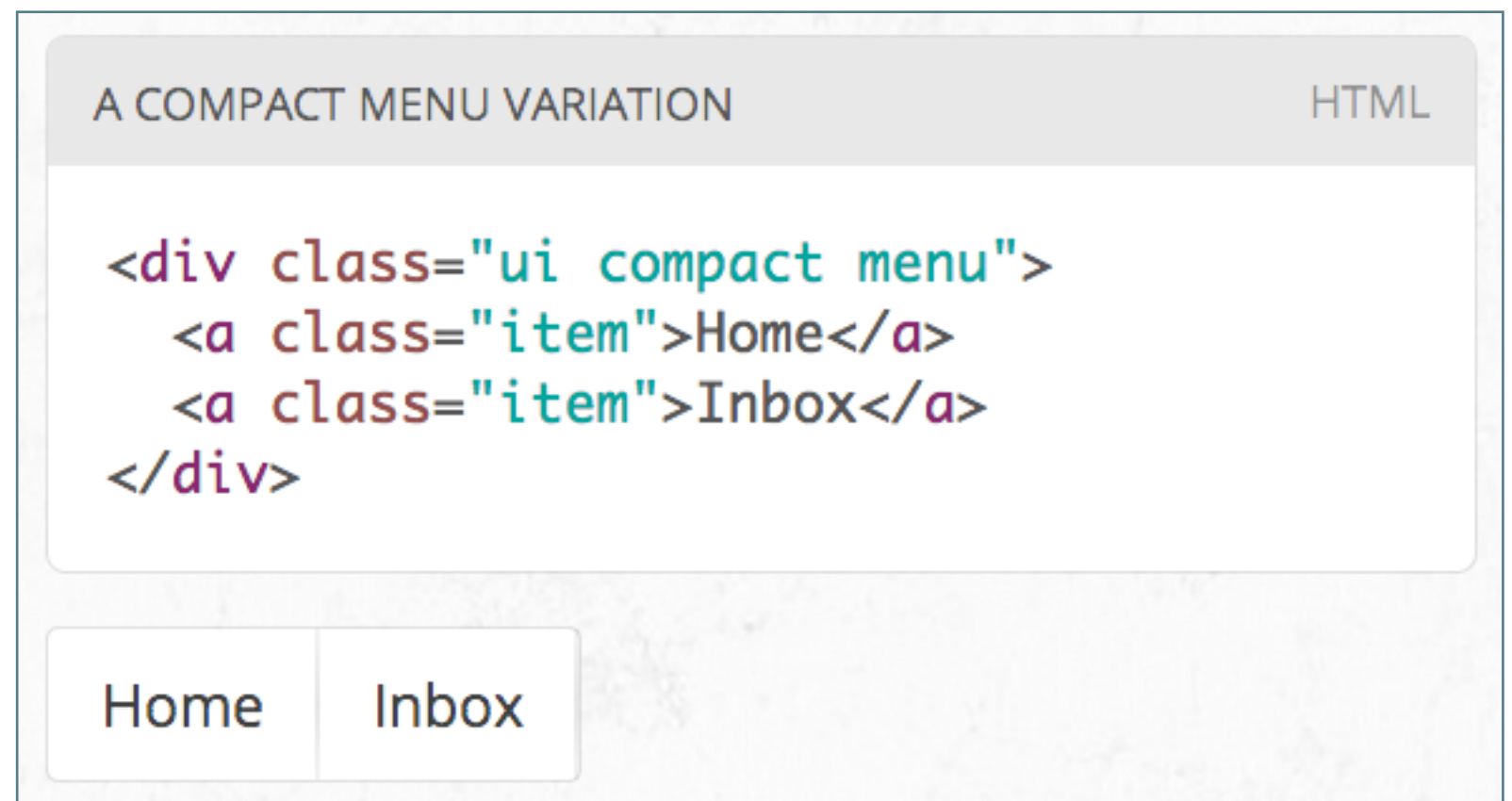
Example

Apples
Pears
Oranges

```
<div class="ui list">  
  <div class="item">Apples</div>  
  <div class="item">Pears</div>  
  <div class="item">Oranges</div>  
</div>
```

Changing an Element

- Class names in Semantic always use single english words.
- If a class name is an adjective it is either a type of element or variation of an element.
- CSS definitions always define adjectives in the context of a noun. In this way class names cannot pollute the namespace.



Combining an Element

- All UI definitions in semantic are stand-alone, and do not require other components to function.
- However, components can choose to have optional couplings with other components.
- For example you might want to include a badge inside a menu. A label inside of a menu will automatically function as a badge

USING A UI LABEL INSIDE A UI MENU

HTML

```
<div class="ui compact menu">  
  <a class="item">Home</a>  
  <a class="item">  
    Inbox  
    <div class="ui label">22</div>  
  </a>  
</div>
```

Home

Inbox 22


Types / Variations


- A ui definition in Semantic usually contains a list of mutually exclusive variations on an element design.
- A type is designated by an additional class name on a UI element


TYPES OF UI BUTTON

HTML

```
<div class="ui labeled icon button">
  Download <i class="download icon"></i>
</div>
<div class="ui icon button">
  <i class="download icon"></i>
</div>
<div class="ui button">
  Download
</div>
<div class="ui facebook button">
  <i class="facebook icon"></i>
  Facebook
</div>
```

 DOWNLOAD

 DOWNLOAD

 FACEBOOK




Types / Content

- Types may require different html structures to work correctly.
- For example, an icon menu might expect different content like icons glyphs instead of text to be formatted correctly

ICON MENU TYPE

HTML

```
<div class="ui icon menu">  
  <a class="item">  
    <i class="mail icon"></i>  
  </a>  
  <a class="item">  
    <i class="lab icon"></i>  
  </a>  
  <a class="item">  
    <i class="star icon"></i>  
  </a>  
</div>
```




Types / HTML Differences


- Types may also each require slightly different html.
- For example, a tiered menu needs html specified for a sub menu to display itself correctly

TIERED MENU TYPE

HTML

```
<div class="ui tiered menu">
  <div class="menu">
    <div class="active item">
      <i class="home icon"></i>
      Home
    </div>
    <a class="item">
      <i class="mail icon"></i>
      Mail
      <span class="ui label">22</span>
    </a>
  </div>
  <div class="sub menu">
    <div class="active item">Activity</div>
    <a class="item">Profile</a>
  </div>
</div>
```

 Home

 Mail 22

Activity Profile

Variations

- A variation alters the design of an element but is not mutually exclusive.
- Variations can be stacked together, or be used along with altering an element's type.
- For example, having wide menus that take up the full width of its parent may sometimes be overwhelming. You can use the compact variation of a menu to alter its format to only take up the necessary space.

```
<div class="ui compact tiered menu">  
  ...  
</div>
```

 Home

 Mail 22

Activity

Profile

Intersecting Variations

- The definition for the variation red contains css specifically for describing the intersection of both red and inverted.

```
<div class="ui red tiered menu">  
  ...  
</div>
```

 Home

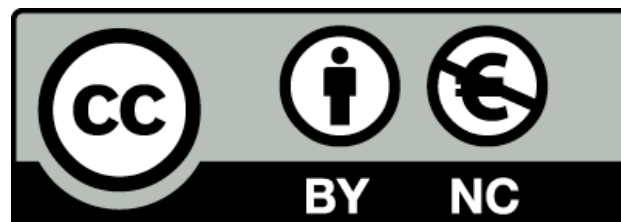


Mail

22

Activity

Profile



Except where otherwise noted, this content is licensed under a [Creative Commons Attribution-NonCommercial 3.0 License](http://creativecommons.org/licenses/by-nc/3.0/).

For more information, please see <http://creativecommons.org/licenses/by-nc/3.0/>

