Web Development

BSc in Applied Computing

Produced by

Eamonn de Leastar (edeleastar@wit.ie)

Department of Computing, Maths & Physics
Waterford Institute of Technology

http://www.wit.ie

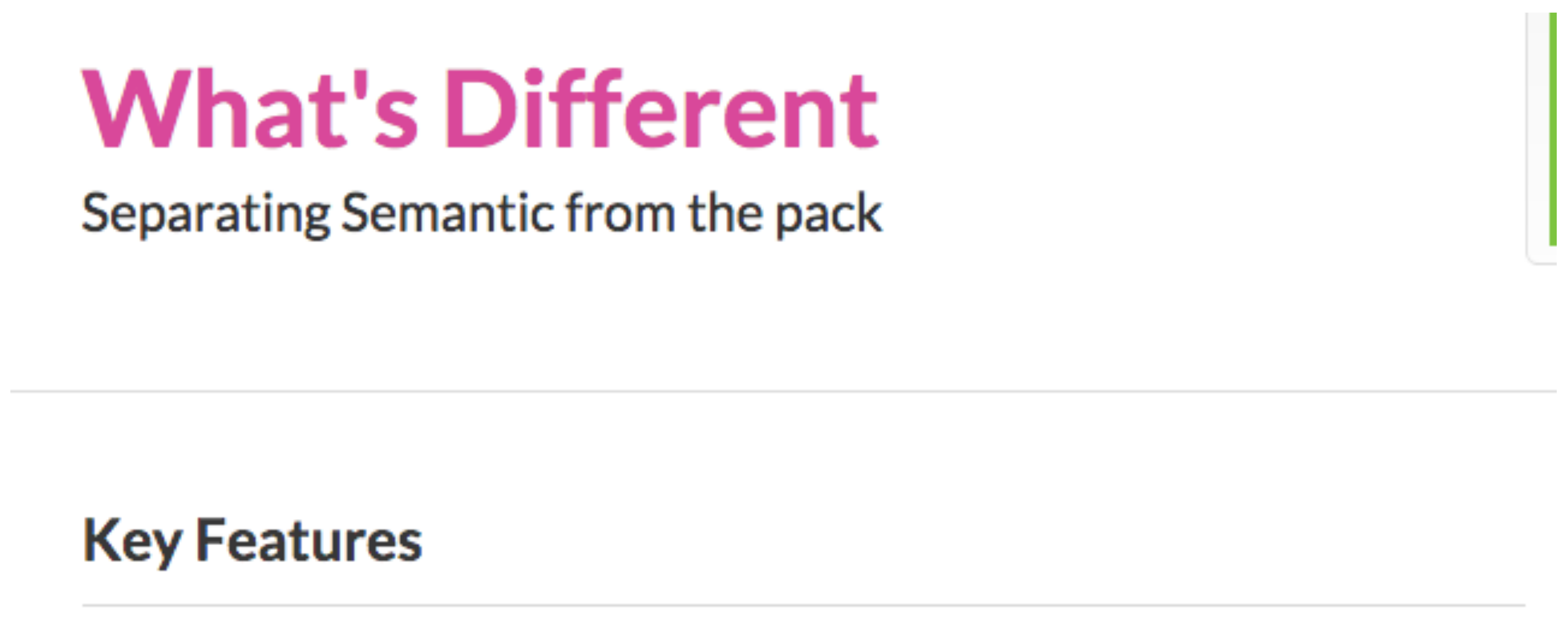
http://elearning.wit.ie

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

eLearning support unit

Semantic UI Usage

Key Features


What's Different

Separating Semantic from the pack

Key Features

http://learnsemantic.com/preface/whats-different.html

Build Responsive Layouts Easier

- Designed Completely with EM
- Every component is defined using em and rem so that components can be resized simply on the fly.

Easy to Learn

- Descriptive not Prescriptive
- Writing front end code shouldn't require learning the naming or programming conventions of a particular developer.
- Instead of using short-hand, or codifying naming conventions, Semantic uses simple, common language for parts of interface elements, and familiar patterns found in natural languages for describing elements.

Tag Agnostic

- Use whatever html tags you please.
- Interface definitions in Semantic are tag ambivalent.
- That means you can use div, article, section, span without affecting the display of the element.
- Special tags like a, table, td still carry special meaning in certain circumstances however.

Concise & Expressive

- Don't repeat yourself

- In English it's much easier to say "There are three tall men" than "There is a tall man, a tall man and a tall man".

- Semantic elements use principles of plurality to express similarities across groups to avoid repetitive declarations.

High-Level Theming

- All UI components share site-wide defaults which let you quickly change the look and feel of components.
- High level variables make sure you aren't specifying one to one matches with CSS properties.

Componentized UI

- Using Semantic doesn't mean adopting an entire framework, or rewriting your code base

- Semantic components are written in a singular style, but are not part of mandated overarching library. Only like a couple components? No problem, use only what you need.

- UI components in Semantic also define optional and required couplings with other components where their usage intersect. That means for example, a popup can check for the existence of CSS animation component before using the fallback javascript animations.

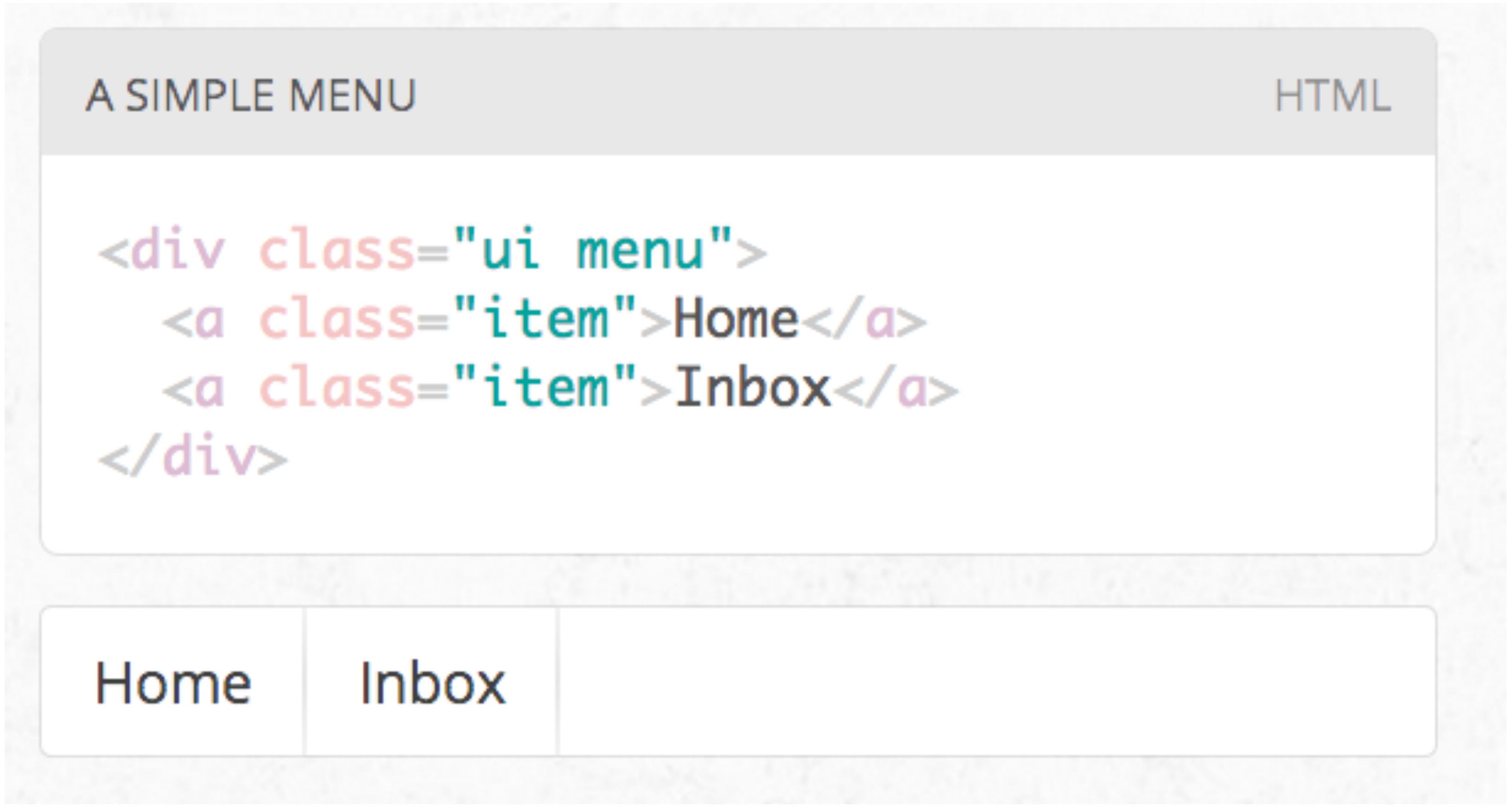
Develop Once, Redesign Infinitely

- Creating a site in Semantic means you never have to rewrite your codebase from scratch.
- Redesigning means retooling your UI toolkit, adjusting UI definitions, not creating entirely new HTML layouts.

Using Semantic UI

“UI” Class

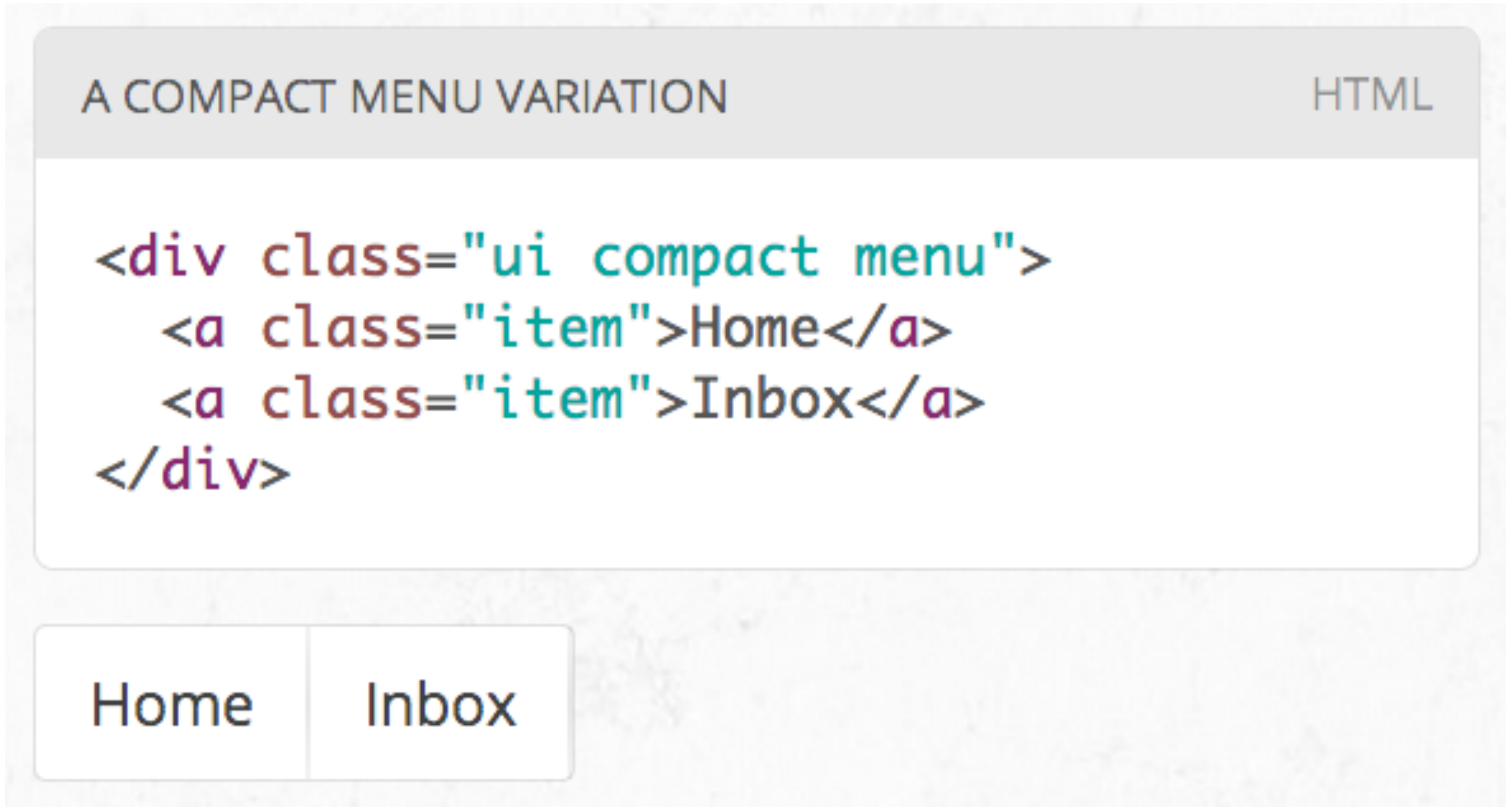
- UI definitions in Semantic are given the class name ui.
- This is to help tell the difference between ui elements and parts of the definition of an element.
- This means any element with the class name UI has a corresponding UI definition.



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

Class Names

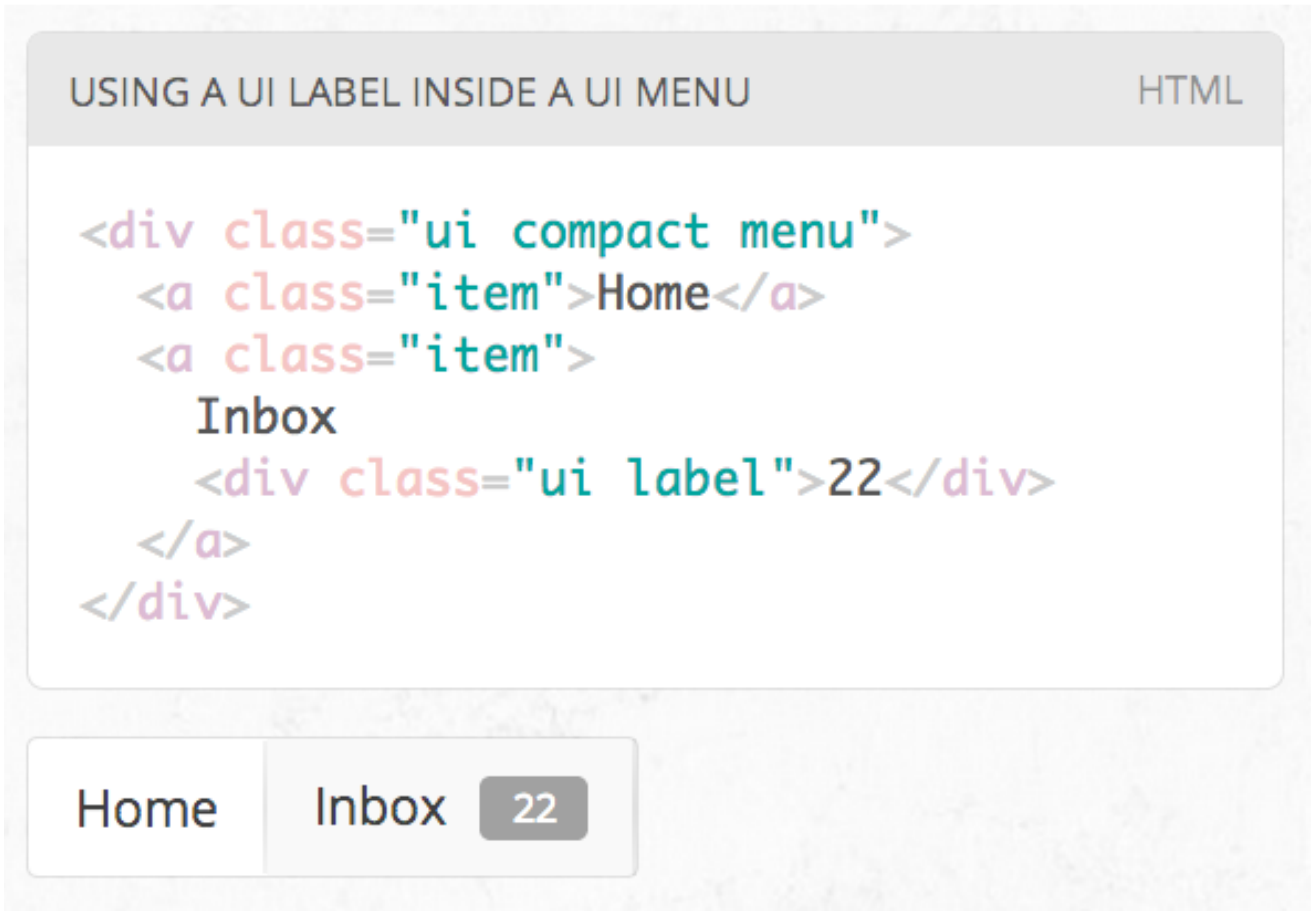
- Class names in Semantic always use single english words.
- If a class name is an adjective it is either a type of element or variation of an element.
- CSS definitions always define adjectives in the context of a noun. In this way class names cannot pollute the namespace.


A COMPACT MENU VARIATION HTML

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

Combining Classes

- 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

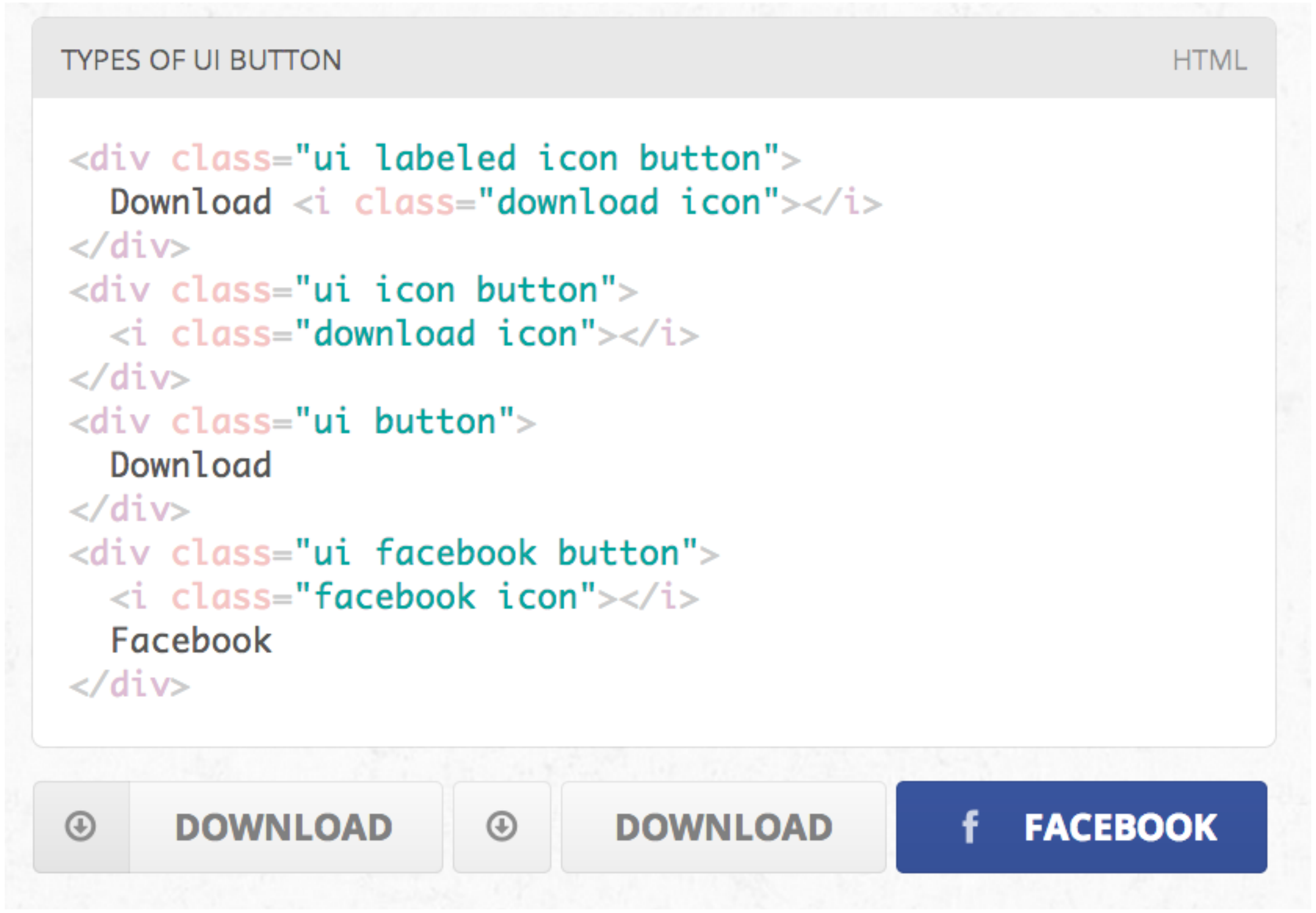

USING A UI LABEL INSIDE A UI MENU HTML

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

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

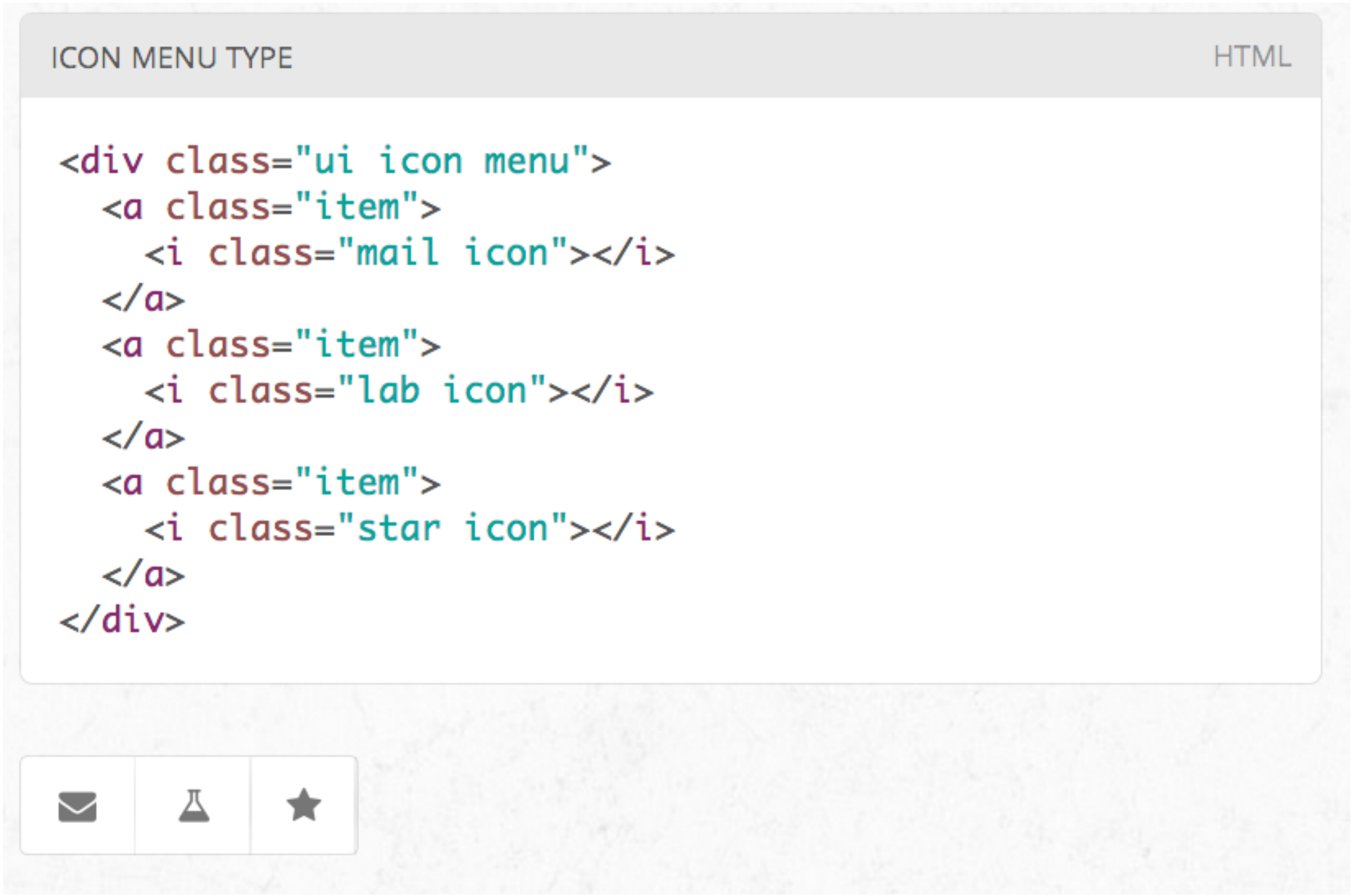


TYPES OF UI BUTTON HTML

```
<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>
```

Content/Structure

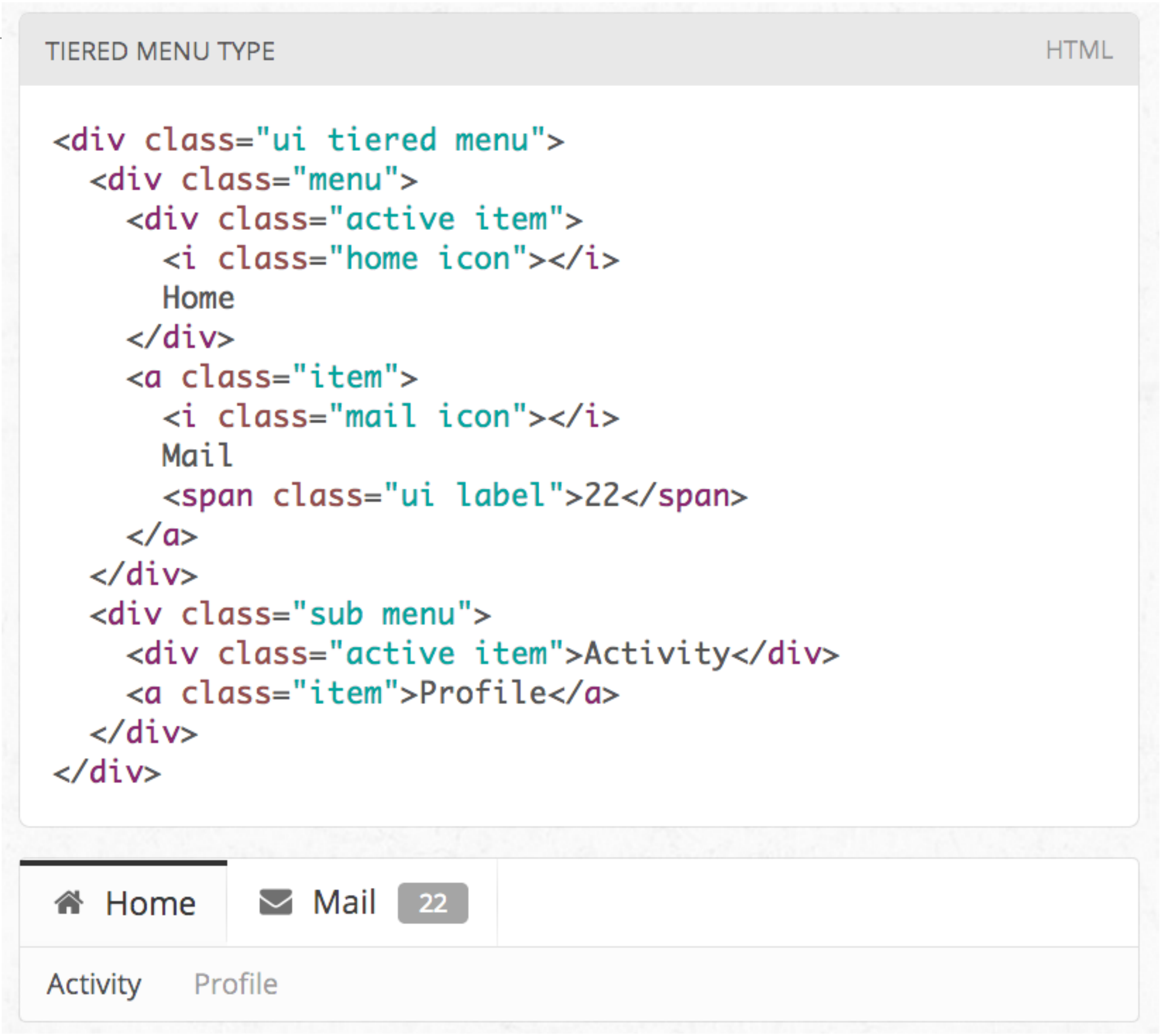
- Types may require different html structures to work correctly.
- For example, an icon menu might expect different content like icons glyphs instead of text to be formatted correctly


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>
```

HTML Variations

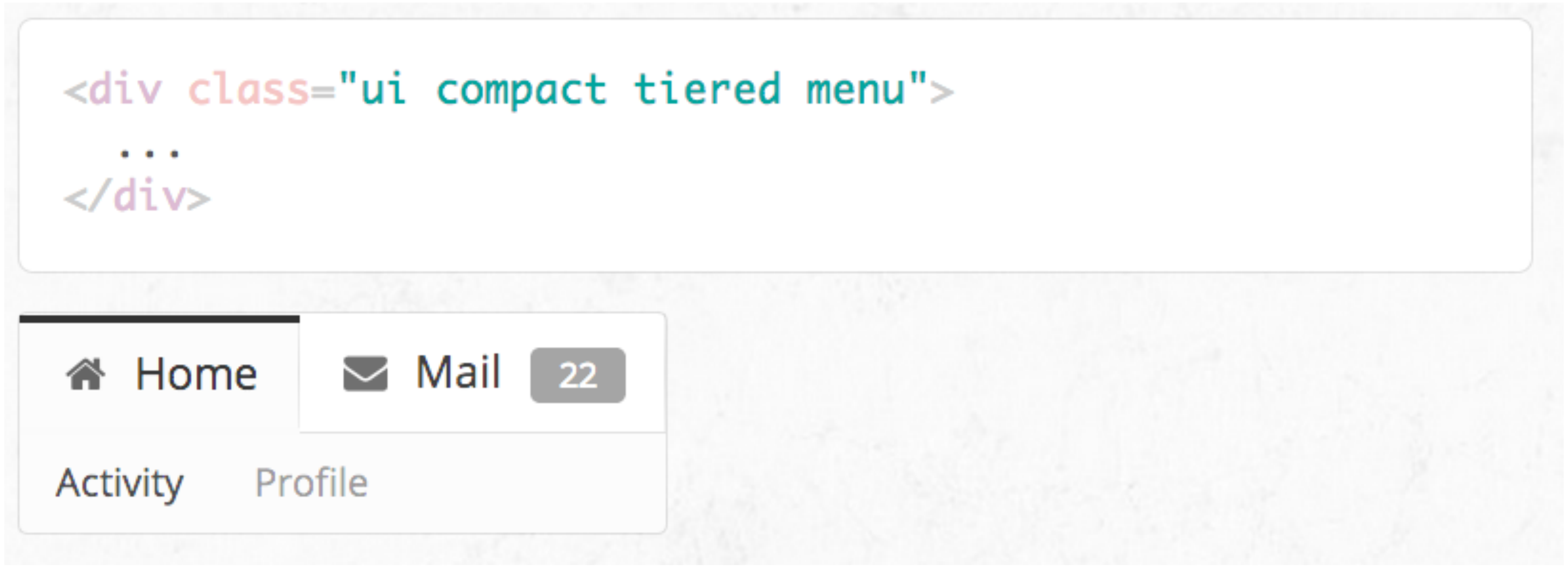
- Types may also each require slightly different html.
- For example, a tiered menu needs html specified for a sub menu to display itself correctly



```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>
```

More Variations

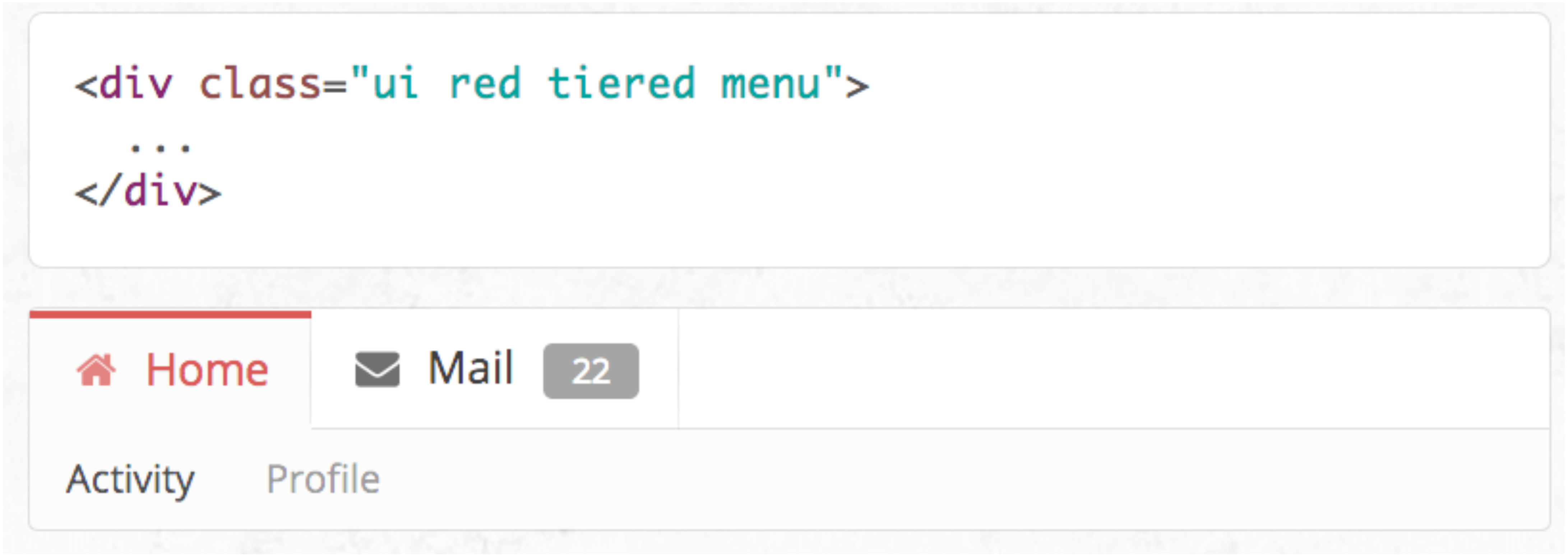
- A variation alters the design of an element but is not mutually exclusive.
- Variations can be stacked together, or be used along with altering an element's type.
- For example, having wide menus that take up the full width of its parent may sometimes be overwhelming. You can use the compact variation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



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

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>
```

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

eLearning support unit