# HTML, CSS, Responsive Design

# HTML 5

- What is HTML for ?

# Answer 1:  Document Layout

- `<h1>`

- `<p>`

- `<ul>`,`<ol>`,`<li>`

- `<table>` `<td>` `<tr>`

Answer 2:  Semantics

- <nav>

- <article>

- <header>,<footer>

- <figure> <caption>

- But wait

  - <h1>,<h2> etc. convey semantics

  - <tr>, <td>

# Why use HTML Semantic Tags?

- Accessibility

  - screen readers can change voice tone on a <strong> tag

- Design

  - meaning of page is always the same regardless of style

- Search Engine Optimization

  - text in a nav bar might be weighted differently

  - density of keywords is higher when more semantic tags are used

# CSS

- General syntax

  ```
  selector { properties;}
  ```

- Example:

  ```
  h2 {
    color: red;
  }
  ```
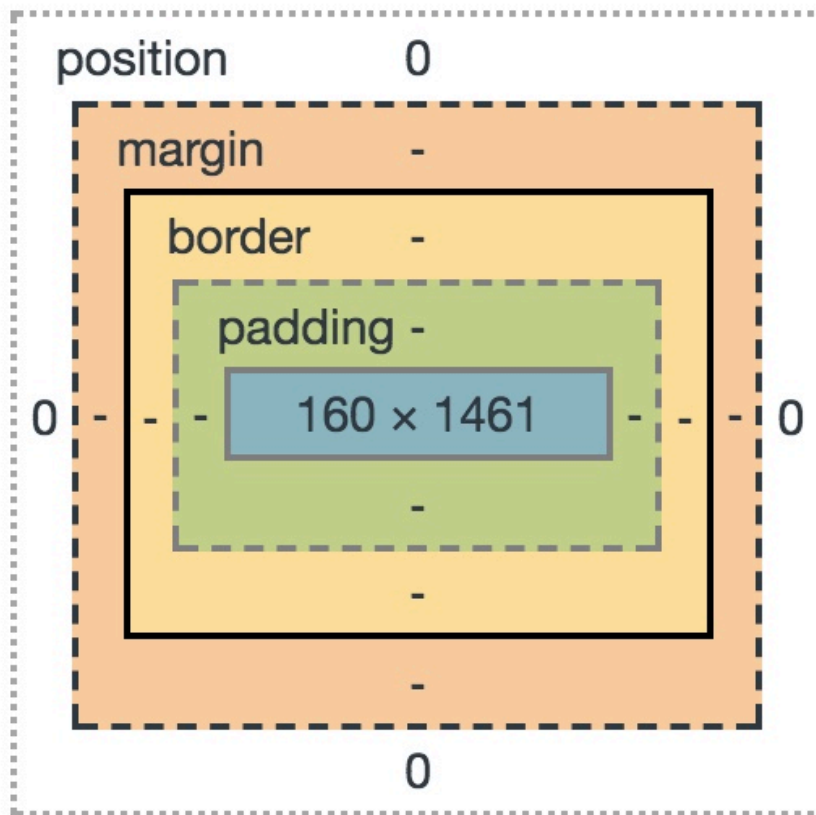
# Selectors

- **Type**: match an element
  - `p` matches `<p>`

- **Class**: match elements the have a particular class
  - `.inline` matches an element with the `class="inline"`

- **Id**: match the element with id.  The id should be unique in the document.
  - `#sidebar` matches element with `id=sidebar`

# More selectors

- `Universal`: matches all elements of a document
  - `*`

- **Attribute**: selects based on value of given attribute
  - `[attr]` `[attr=value]` `[attr~=value]` etc

# CSS Box Model



Used to specify the space around content

# CSS Frameworks

- Bootstrap
  - from Twitter now open source

- Grid

- Flexbox

- And many more

- Also languages that compile to CSS: SASS, LESS, SCSS

# Responsive Web Design

- Structure content for meaning, use style for how it looks.

- Same content, different devices
  - Don't want a completely different web page

# Responsive Web Design

- Flexible grid based layout
  - Instead of specifying widths in pixels, use percentages

  ```
  <meta name="viewport"
  content="width=device-width,
  initial-scale=1">
  ```

- Flexible media
  - E.g. max-width for images

- Media queries

# Media Queries

- Define CSS selectors that take effect when the media is a certain size.

```
@media screen and (width: 320px) {
    body { color: red; }
}
```

# Best Practices

- Design for smallest device first

- Keep content and style separate

- Choose breakpoints based on content rather than specific devices

- Optimize text for reading (70-80 characters per line)