CHAPTER 6 –
CSS PT2: SELECTORS

TOPICS TO BE COVERED

- **CSS Selectors**

  Modify specific elements in HTML

More info on this topic can be found in e-books provided in LMS.
Presentation slides made by Fifah S., M. Khalid.

INHERITANCE

```
<body>
    <div>
      <p>Text</p>
    </div>
    <h1>Text</h1>
</body>
```

- Top-level element parent is <body> tag.
- Both and <h1> are the children of <body>.
- <p> is the child of?

- **If we apply style to the parent, everything within the tag will inherit the style.**

INHERITANCE

```
<body>
    <div id="header">
      <h1>Contact Us</h1>
    </div>
    <div>
      <p>
        Hello there!
      </p>
    </div>
    <span>Hi, I am a span</span>
    <span class="deck">I am a deck</span>
    <span class="deck">I am a deck too</span>
    <a href="#">Click me!</a></a>
</body>
```

- Notice that the parent is <body> tag.
- When we apply the styling on the parent element, as such, `body { color: red;}` all elements will inherit the same styling as the body tag.

Contact Us

Hello there!

Hi, I am a span I am a deck I am a deck too Click me!

INHERITANCE

```html
<body>
    <div id="header">
      <h1>Contact Us</h1>
    </div>
    <div>
      <p>
        Hello there!
      </p>
    </div>
    <span>Hi, I am a span</span>
    <span class="deck">I am a deck</span>
    <span class="deck">I am a deck too</span>
    <a href="#">Click me!</a></a>
</body>
```

- By adding specific styling onto tag, would override the inheritance styling.

```css
body { color: red; }
div{ color: blue; }
```

- As a result, text within the tags will be colored blue.

Contact Us

Hello there!

Hi, I am a span I am a deck I am a deck too Click me!

INHERITANCE

```html
<body>
    <div id="header">
      <h1>Contact Us</h1>
    </div>
    <div>
      <p>
        Hello there!
      </p>
    </div>
    <span>Hi, I am a span</span>
    <span class="deck">I am a deck</span>
    <span class="deck">I am a deck too</span>
    <a href="#">Click me!</a></a>
</body>
```

- As we add styling onto <p> tag, we automatically override the style. As such:

```css
body { color: red; }
div{ color: blue; }
p { color: green; }
```

- By doing that, all <p> tags text will be colored green.

Contact Us

Hello there!

Hi, I am a span I am a deck I am a deck too Click me!

INHERITANCE

- Notice that all anchor tag <a> styling are the same throughout the examples, whether or not we have applied different styles.

Contact Us

Hello there!

Hi, I am a span I am a deck I am a deck too Click me!

Contact Us

Hello there!

Hi, I am a span I am a deck I am a deck too Click me!

Contact Us

Hello there!

Hi, I am a span I am a deck I am a deck too Click me!

- All anchor tags are in blue with underline decoration. This is due to the **default browser style.**

CONFLICTS AND CASCADE

```html
<body>
  <p class="para"> I am </p>
  <p class="para"> a </p>
  <p class="para"> paragraph</p>
</body>
```

```css
p { color: blue; }
p { color: red;  }
```

- CSS runs / reads from the **TOP to bottom.**
- Bottommost rules always win.

If you wanted to override styling, as per the figure, you can either:

1. style it using internal CSS.
2. style it using inline CSS.
3. Important Declaration.

IMPORTANT DECLARATION

```
<body>
  <p class="para"> I am </p>
  <p class="para"> a </p>
  <p class="para"> paragraph</p>
</body>
```

```
p { color: blue !important; }
p { color: red;  }
```

- A tool to use within CSS conflicts to make a property important.
- No other styles can override the styling which has important declaration.
- If we want the paragraph to be blue, add !important statement at the end.
- **USE IT WITH CAUTION!**

DESCENDANT SELECTORS

```
<body>
    <div>
      <p>Text</p>
    </div>
    <h1>Text</h1>
</body>
```

- All are descendants of <body> tag.
- <p> is the descendant of tag.

DESCENDANT SELECTORS

```
<body>
    <div id="main-content">
      <p>Hello there!</p>
      <p>Yes, <strong>YOU!</strong> Hi.</p>
      <div id="sub-content">
        <p>More content.</p>
      </div>
    </div>
</body>
```

- More content. is a descendant of .
- If we wanted to select specific tag (e.g. all under , we do:

```
#main-content p{ color: purple;}
```

Stating the CSS to find main-content div (parent).

Signify that we want to target the specific element / id / class within the div.

DESCENDANT SELECTORS

```
<body>
    <div id="main-content">
      <p>Hello there!</p>
      <p>Yes, <strong>YOU!</strong> Hi.</p>
      <div id="sub-content">
        <p>More content.</p>
      </div>
    </div>
</body>
```

» If we want to target specifically More Content , we do:

```
#main-content #sub-content p{ color: purple;}
```

parent | child | grand-child

OR simply,

```
#sub-content p{ color: purple;}
```

Activity! Try it out

Now we apply what we have learnt to ***menu.html*** as per follows:

1. Add some color to the “new item!” elements next to certain menu items name. Any color you like.

2. BUT!! I don’t want “Very Spicy!” in the same color with new item elements. Use descendent selector to only change #1 color.

3. Now make all the text in **header** teal, considering the ID. Named the ID as info.

SPECIFICITY SELECTOR

» **Example – these styling are point to the same p tags.**

```html
<body>
    <div id="main">
      <p>Hello there!</p>
      <p class="test">I am a para</p>
    </div>
    <p>None contained <strong>p</strong>tag</p>
</body>
```

```css
#main p{
    color: purple;
}
p{ color: red; }
```

#main p styling wins due to specificity, and selector with most points wins.

» To make it more clearer, we make a point system for all the selector.

Styling with	Points
ID (#)	100
Class (.)	10
Element	1

SPECIFICITY SELECTOR

```html
<body>
    <div id="main">
      <p>Hello there!</p>
      <p class="test">I am a para</p>
    </div>
    <p>None contained <strong>p</strong>tag</p>
</body>
```

```css
#main p{ color: purple;}
p{ color: red; }
.test{ color: green;}
```

- Now, we add on styling to paragraph in test class. Does it work?
- Nope. This is because class styling has only 10 points, in compare to #main (ID) which has 100 pts. (Refer to table in prev slide)

#main p styling wins due to specificity.

- To make it more clearer, we make a point system for all the selector.

SPECIFICITY SELECTOR

```html
<body>
    <div id="main">
      <p>Hello there!</p>
      <p class="test">I am a para</p>
    </div>
    <p>None contained <strong>p</strong>tag</p>
</body>
```

```css
#main p{ color: purple;}
.test{ color: green;}
p{ color: red; }
strong { color: blue;}
```

- Initially strong text is in red due to inheriting the styling from p tag by default.
- By adding strong styling, we override the rule explicitly.

Hello there!

I am a para

None contained **p** tag

Activity! Try it out

4. Make the paragraph inside the header ***italic*** in a way that doesn't affect the other paragraphs on the page. Use specificity selector. Make sure to select ONLY paragraphs contained within the info section of the document.

5. Use a class selector to change all the prices on the menu into:
   - Font to Georgia or any serif font.
   - Italic
   - Color them gray.

MULTIPLE SELECTORS

» **Example – these codes have repetitive styles.**

```
h1 { font-family: “Georgia”;  }
.menu {  font-family: “Georgia”;  }
```

Instead, you can write it as:

```
.menu h1 { font-family: “Georgia”;  } OR  h1, .menu { font-family: “Georgia”;  }
```

Activity! Try it out

6. Similarly, change the appearance of the text in the header of class *label* to make them stand out.
   - Bold and small-caps

7. Finally, make the warning at the bottom as obvious by applying group selector.
   - Font size as x-small with red color.

CHILD SELECTORS

```
<body>
    <div>
        <p>Text</p>
    </div>
    <h1>Text</h1>
</body>
```

Both are child of the <body>

Not a direct child of <body>, but a child of

CHILD SELECTORS

» If we want to style the <p> tags, we do:

```
#main-content > p{color: blue;}
```

States that you want to **select direct children** of main-content, specifically the <p> tags

```
<body>
    <div id="main-content">
      <p>Hello there!</p>
      <p>Yes, <strong>YOU!</strong> Hi.</p>
      <div id="sub-content">
        <p>More content.</p>
      </div>
    </div>
</body>
```

Are direct child of i.e. 1-level deep

The greater than symbol (>) is known as Child Combinator.

ADJACENT SELECTORS

- A selector that selects element which comes directly after another element.

```html
<body>
    <div id="articles">
      <h2>Article #1</h2>
      <p>Published by: SK</p>
      <p>bla bla bla</p>
      <p>bla bla bla</p>
      <p>bla bla bla</p>
      <h2>Article #2</h2>
      <p>Published by: MF</p>
      <p>bla bla bla</p>
      <p>bla bla bla</p>
      <p>bla bla bla</p>
    </div>
</body>
```

- Imagine you have 100+ <p> tags.
- The target is to style the highlighted <p> tags without affecting the other <p> tags.
- To target those, we do:

```css
#articles h2 + p { background: yellow;}
```

The plus (+) symbol is called adjacent combiner.

- i.e. it will collect the **h2 descendants**, and the p tag that comes ***directly*** after h2.

Activity! Try it out

8. Now it's done, lets change a few from the body into these properties: font-family into *Georgia* or any *serif* fonts.

   Add **line-height** properties to open up the text lines and easier to read to 1.75em.

9. Redesign the header section. Delete the teal color settings and make:
   - **h1** as purple.
   - **Paragraph** in the header as gray.

Activity! Try it out

10. To imitate a fancy print menu, lets align a few key elements on the page (h1, h2 and #info) using **text-align** property. Make it center.

11. Make “Appetizer” and “Main courses”,
    - All uppercase letters
    - Extra letter spacing
    - Colour purple

12. Tweak the paragraphs right after h2 using **text-align** as center and italic.

ATTRIBUTE SELECTORS

» An attribute is a characteristic of a page element, such as ID, class, alt, color, font size etc.

Element start tag

Attributes

Textual Contents

Attributes

```
<h1 id="main_header">
    Welcome to Web Programming. <img src="head.jpg" width ="300" height = "250" />
</h1>
```

Element end tag

Graphical Contents

ATTRIBUTE SELECTORS

```html
<body>
  <div id="header">
    <h1>Contact Us!</h1>
  </div>
  <h2 id="content-header">Content</h2>
  <div id="main">
    <p>Hello There!</p>
    <p>Hi!</p>
  </div>
  <div>
    <p>This is the 3rd div.</p>
  </div>
  <span>Hi, I am a span</span>
  <span class="deck">I am a deck</span>
  <span class="deck">I am a deck too</span>
</body>
```

1. Targeting ** with class deck** associated.

```css
span[class]{ color: green; }
```

- As a result, only span with attribute class will be styled.

Contact Us!

Content

Hello There!

Hi!

This is the 3rd div.

Hi, I am a span I am a deck I am a deck too

ATTRIBUTE SELECTORS

```html
<body>
  <div id="header">
    <h1>Contact Us!</h1>
  </div>
  <h2 id="content-header">Content</h2>
  <div id="main">
    <p>Hello There!</p>
    <p>Hi!</p>
  </div>
  <div>
    <p>This is the 3rd div.</p>
  </div>
  <span>Hi, I am a span</span>
  <span class="deck">I am a deck</span>
  <span class="deck">I am a deck too</span>
</body>
```

2. Styling ** with attribute** ID.

```css
div[id]{background: grey; }
```

- As a result, every element which has id attribute will be styled.

Contact Us!

Content

Hello There!

Hi!

This is the 3rd div.

Hi, I am a span I am a deck I am a deck too

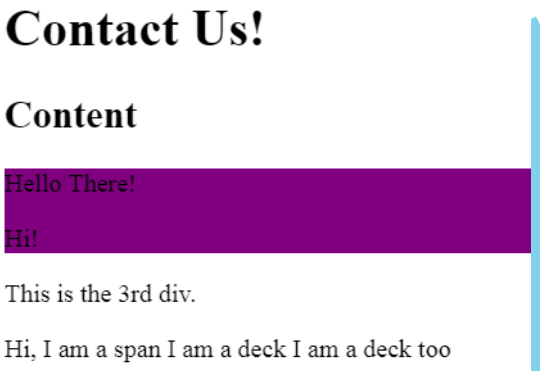
ATTRIBUTE SELECTORS

```html
<body>
  <div id="header">
    <h1>Contact Us!</h1>
  </div>
  <h2 id="content-header">Content</h2>
  <div id="main">
    <p>Hello There!</p>
    <p>Hi!</p>
  </div>
  <div>
    <p>This is the 3rd div.</p>
  </div>
  <span>Hi, I am a span</span>
  <span class="deck">I am a deck</span>
  <span class="deck">I am a deck too</span>
</body>
```

3. Styling ** with specific ID attribute.**

```css
div[id="main"]{ background-color: purple; }
```

- As a result, element which has id attribute of main will be styled.



ATTRIBUTE SELECTORS

```html
<body>
  <div id="header">
    <h1>Contact Us!</h1>
  </div>
  <h2 id="content-header">Content</h2>
  <div id="main">
    <p>Hello There!</p>
    <p>Hi!</p>
  </div>
  <div>
    <p>This is the 3rd div.</p>
  </div>
  <span>Hi, I am a span</span>
  <span class="deck red">I am a deck</span>
  <span class="deck">I am a deck too</span>
</body>
```

4. **Pattern Matching**
   - To target both span, we use *tilde* ~.

```css
span[class~="deck"]{background: pink;}
```

As long as it has “deck” class, even though the span has multiple classes, it will be styled.

Contact Us!

Content

Hello There!

Hi!

This is the 3rd div.

Hi, I am a span I am a deck I am a deck too

ATTRIBUTE SELECTORS

```html
<body>
  <div id="header">
    <h1>Contact Us!</h1>
  </div>
  <h2 id="content-header">Content</h2>
  <div id="main">
    <p>Hello There!</p>
    <p>Hi!</p>
  </div>
  <div>
    <p>This is the 3rd div.</p>
  </div>
  <a href="some.pdf">pdf file</a>
  <a href="http://www.google.com">Googlee</a>
</body>
```

5. **Target Selection**
   a. Ends with …

```css
a[href$="pdf"]{color: green;}
```

Grabs all <a> links that have href attribute associated that **ends with pdf.**

Contact Us!

Content

Hello There!

Hi!

This is the 3rd div.

pdf file Googlee

ATTRIBUTE SELECTORS

```html
<body>
  <div id="header">
    <h1>Contact Us!</h1>
  </div>
  <h2 id="content-header">Content</h2>
  <div id="main">
    <p>Hello There!</p>
    <p>Hi!</p>
  </div>
  <div>
    <p>This is the 3rd div.</p>
  </div>
  <a href="some.pdf">pdf file</a>
  <a href="http://www.google.com">Googlee</a>
</body>
```

5. **Target Selection**
   b. Starts with..

```css
a[href^="http"]{color: yellow;}
```

Grabs all <a> links that have href attribute associated that **starts with http**

Contact Us!

Content

Hello There!

Hi!

This is the 3rd div.

pdf file Googlee

COMMENTS

- Using comments for:
  - Notes. Describing what the codes all about.
  - To split / segment CSS file into logical areas.

- It's a way of adding text to CSS file without the browser displaying the text.

/*Comments here */

```
/* 2 - RADIO BOX STYLES  */
input[type="radio"]{ /*hide the default radio*/
  opacity:0;
  margin: 0;
  width:0;
}

label[for="male"], label[for="female"]{
  margin-bottom: 10px;
  display: inline-block;
  padding-left: 26px;
  background: url(img/checks.png) no-repeat;
  background-position:  0 -32px;
  line-height: 24px;
  cursor: pointer;
}

input:checked + label[for="male"], input:checked + label[for="female"]{
  background-position: 0 0;
  color: #ca1010;
}

/* 3 - CHECKBOX STYLES  */
input[type="checkbox"]{ /*hide the default checkbox*/
  opacity: 0;
  margin: 0;
  width: 0;
}
```

Activity! Try it out

13. Add “sienna” color to all **dt** elements.

14. To make it more appealing, add drop shadow to h1 headings.

DONE! SAVE THE MENU.HTML

PSEUDO-CLASSES

- Special keywords that go after selectors.

- TWO groups of pseudo-class:

Dynamic	Structural
Allow us to style an element in relation to **user actions.**	Allow us to style elements based on **advanced structural techniques**, which are not possible using CSS selectors.
Example: 1. Whether the link is being hovered over. 2. Whether a button is being pressed. 3. Whether a tick box has been clicked.	Example: 1. The 5th tag in the list. 2. A parent tag that has no children.

PSEUDO-CLASSES : DYNAMIC

- Also known as behavioral pseudo-class.

- Example:

```html
<body>
  <a href="#">I am a link</a>
  <a href="http://www.google.com" title="Google">I am a link too</a>
</body>
```

```css
a:hover{ color: red;}
a:active{ color: purple;}
a:visited{ color: orange;}
```

When the link is hovered

When the button is clicked down

Link which has been visited before

PSEUDO-CLASSES : STRUCTURAL

» Understand this concept:

```
<body>
  <ul>
    <li>item</li>
    <li>item</li>
    <li>item</li>
    <li>item</li>
  </ul>
</body>
```

First child

Children of <ul> tag

PSEUDO-CLASSES : STRUCTURAL

```html
<ul>
  <li>ITEM LIST</li>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
  <li>Item 4</li>
  <li>Item 5</li>
  <li>HALF WAY THERE</li>
  <li>Item 6</li>
  <li>Item 7</li>
  <li>Item 8</li>
  <li>Item 9</li>
  <li>Item 10</li>
</ul>
```

» To style each <li> tags differently, we use the **nth child** selectors.

Example 1:

Style the ITEM LIST (i.e. the first <li>) in bold.

```css
li:nth-child(1){font-weight: bold;}
```

Passed an argument, specify which child we want to style.

PSEUDO-CLASSES : STRUCTURAL

```html
<ul>
  <li>ITEM LIST</li>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
  <li>Item 4</li>
  <li>Item 5</li>
  <li>HALF WAY THERE</li>
  <li>Item 6</li>
  <li>Item 7</li>
  <li>Item 8</li>
  <li>Item 9</li>
  <li>Item 10</li>
</ul>
```

» To style each <li> tags differently, we use the **nth child** selectors.

Example 2:

Style 2 children in bold.

```css
li:nth-child(1), li:nth-child(7){ font-weight: bold;}
```

Passed an argument, specify which child we want to style.

PSEUDO-CLASSES : STRUCTURAL

```html
<ul>
  <li>ITEM LIST</li>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
  <li>Item 4</li>
  <li>Item 5</li>
  <li>HALF WAY THERE</li>
  <li>Item 6</li>
  <li>Item 7</li>
  <li>Item 8</li>
  <li>Item 9</li>
  <li>Item 10</li>
</ul>
```

» To style each <li> tags differently, we use the **nth child** selectors.

Example 3:

Style even/odd children.

```css
li:nth-child(even){color: pink;}
li:nth-child(odd){color: magenta;}
```