



Keywords/Questions:

Notes:

Refactoring code rules

Rules for Refactoring code

1) Put all your HTML elements together

2) For your HTML elements(aka body, h1, h3,... tags) place only broad styling:

-like color, font-family, ...

3) If there is a repetition of the same property for different elements for example

let it be font family we can write it as

```
h1, h2, h3 {
  font-family: "Montserrat";
  font-weight: 900;
}
```

4) When using specific styling use classes to style each common occurrence of a

particular type of element. So this is what we mean about modularity, so it allows

you to drill down on a specific section easily and figure out what the problems are or

change the style of one particular section without affecting the rest of the web site.

5) When writing your code, when you see opportunities for refactoring you would do

it right there and then rather than waiting for a larger refactoring session

6) Combining selectors to specify settings

```
#title .container-fluid{
  padding: 3% 15% 7%;
}
.container-fluid{
  padding: 7% 15%;
}
```

-This targets the container-fluid div under the title section.

Advantage of using this is that the padding from the 2

won't add up. The more specific one will override it. If we

just use an id, it would result in addition of padding

Summary:

Refactoring Code rules:-Put HTML elements in style sheet together & use them for broad styling like color and font-family

-We can use multiple selectors in the same line i.e. **h1, h2 {font-family:"Montserrat"; }**

-Use specific classes for common occurrences

-Refactor code as you write it

-Combining selectors helps avoid addition of properties like padding

```
#title .container-fluid{
  padding: 3% 15% 7%;
}
.container-fluid{
  padding: 7% 15%;
}
```