Cascading Style Sheets (CSS)

Separating Presentation Details

- Separating content and its presentation details is important
- Content is consumed using various media
  - Computer monitor
  - Braille terminal
  - Text to speech system
  - Automated systems
- Focus on content during development
- Easily improve/change the presentation later
- Saves a lot of effort

CSS Properties

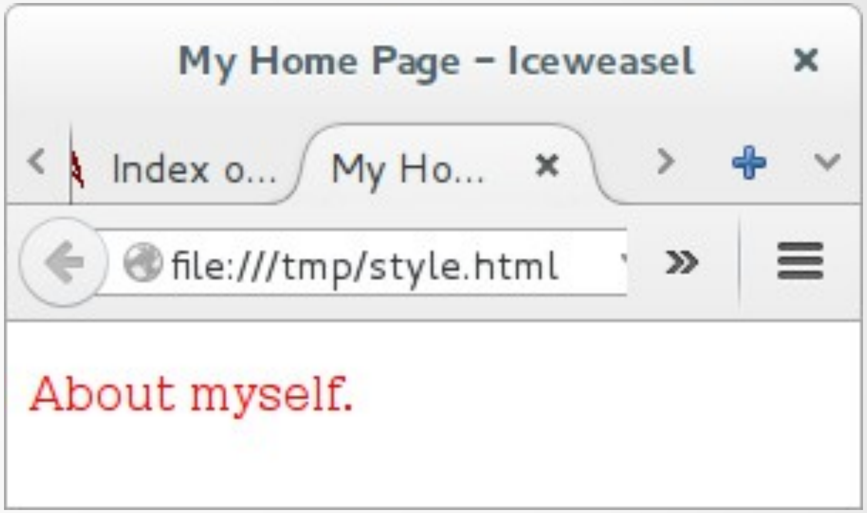
- Color
- Background and Borders
- Basic Box
- Flexible Box
- Text
- Text Decoration
- Fonts
- Writing Modes
- Table
- Lists and Counters
- Animation
- Transform
- Transition
- Basic User Interface
- Multi-column
- Paged Media
- Generated Content
- Filter Effects
- Image/Replaced Content
- Masking
- Speech

In-line CSS (Bad)

You Write:

```
<!DOCTYPE html>
<html>
  <head>
    <title> My Home Page
    </title>
  </head>
  <body style="color: red;">
    <p>About myself.</p>
  </body>
</html>
```

You See:



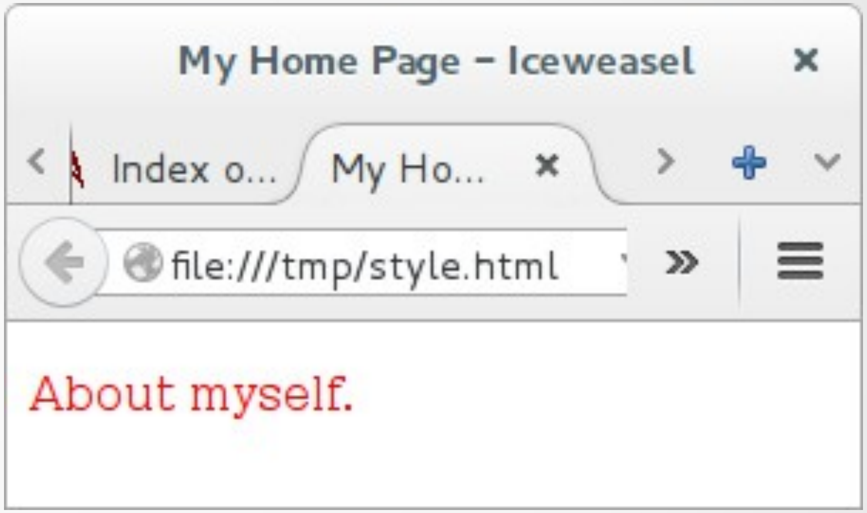
About myself.

Internal CSS (Better)

You Write:

```
<head>
  <title> My Home </title>
  <style>
    body {
      color: red;
    }
  </style>
</head>
```

You See:



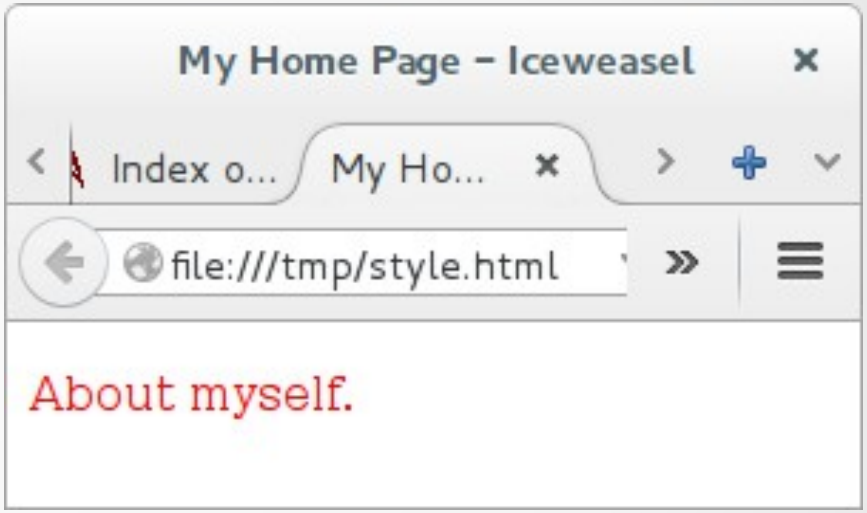
About myself.

External CSS (Good)

In HTML You Write:

```
<head>
  <title> My Home </title>
  <link
     rel="stylesheet"
     href="style.css">
</head>
```

You See:

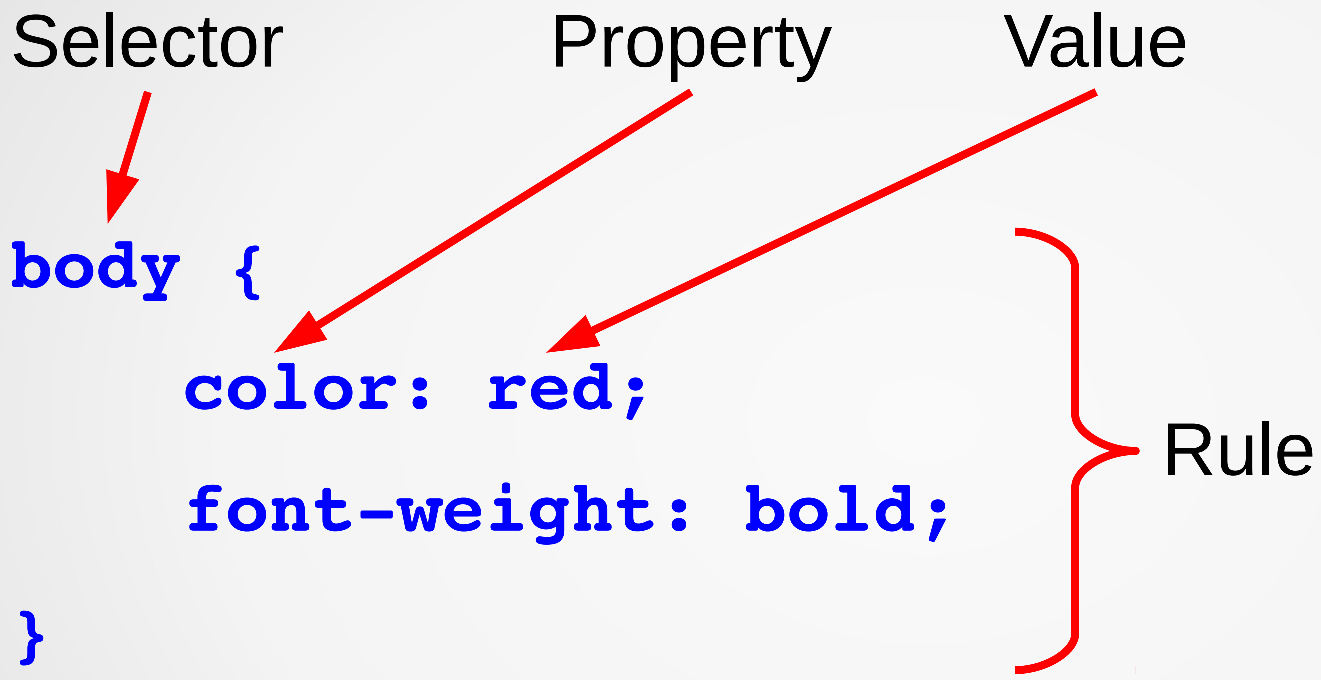


About myself.

In *style.css* You Write:

```
body {
  color: red;
}
```

CSS Syntax



Selector Property Value

```
body {
    color: red;
    font-weight: bold;
}
h1 {
    font-size: 20px;
}
```

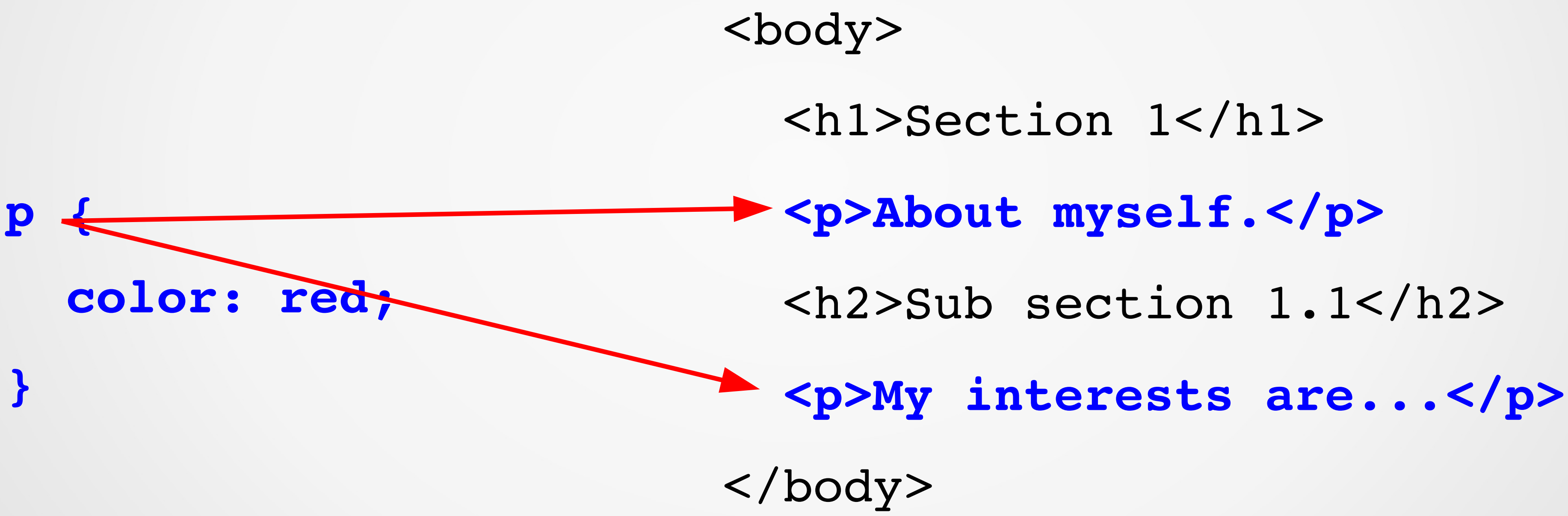
Rule

CSS Comments

```
/* This is a comment */
body {
    color: red;
    font-weight: bold;
}
h1 {
    font-size: 20px;
}
```

CSS Selectors – Tag Name

- Select all elements in a page matching a HTML tag name

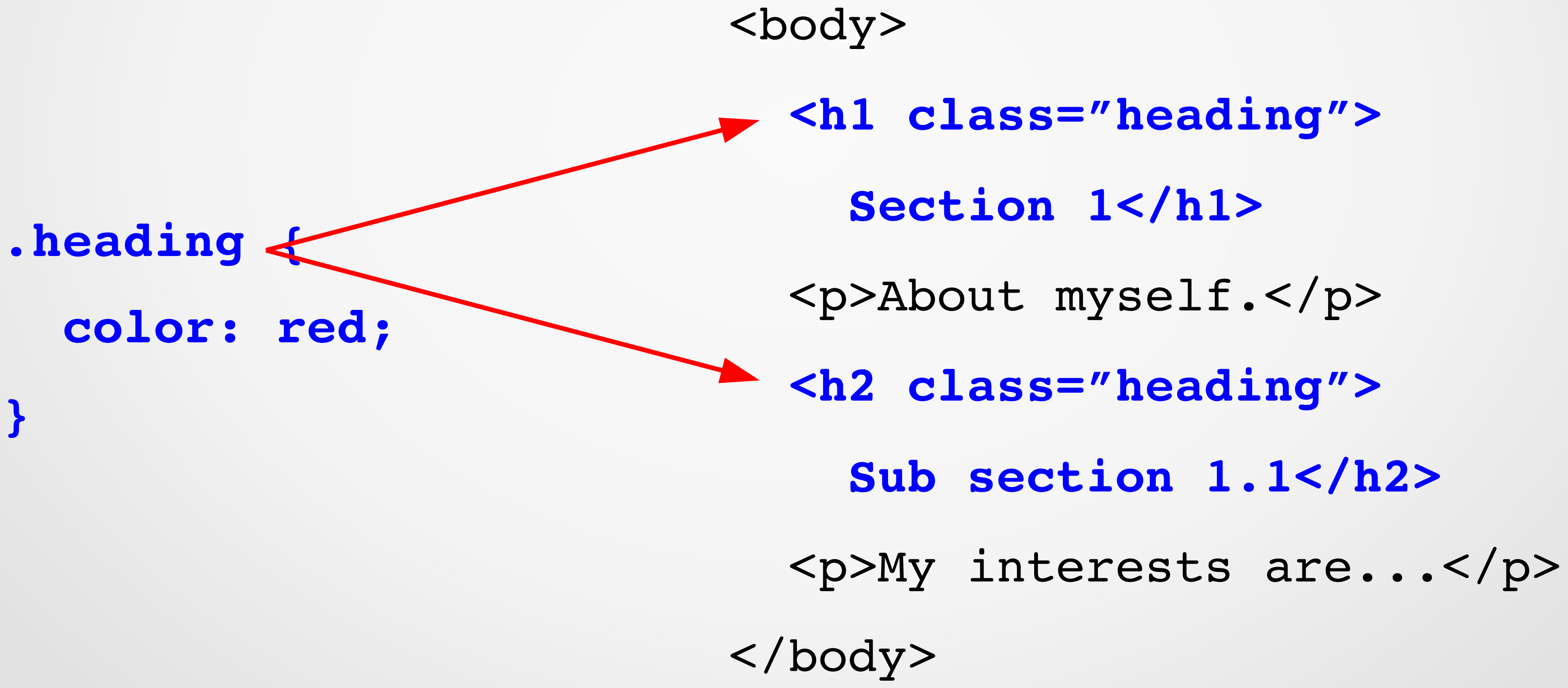


```
p {
  color: red;
}
```

```
<body>
  <h1>Section 1</h1>
  <p>About myself.</p>
  <h2>Sub section 1.1</h2>
  <p>My interests are...</p>
</body>
```

CSS Selectors – Class Name

- Select all elements in a page have a “class” attribute
- An element can have multiple classes

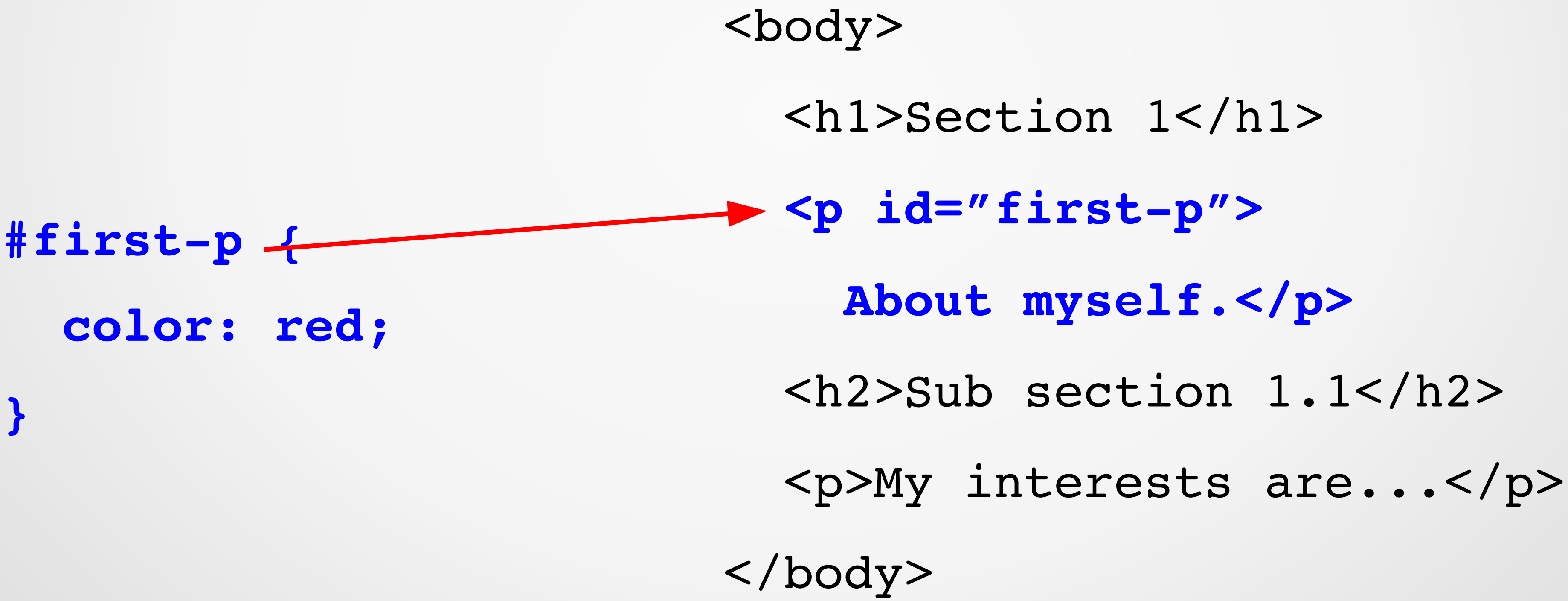


```
.heading {
  color: red;
}
```

```
<body>
  <h1 class=”heading”>
    Section 1</h1>
  <p>About myself.</p>
  <h2 class=”heading”>
    Sub section 1.1</h2>
  <p>My interests are...</p>
</body>
```

CSS Selectors – ID

- Select all elements in a page a ID attribute
- ID must be unique in the entire document



```
#first-p {
  color: red;
}
```

```
<body>
  <h1>Section 1</h1>
  <p id="first-p">
    About myself.</p>
  <h2>Sub section 1.1</h2>
  <p>My interests are...</p>
</body>
```

Grouping Selectors

```
h1 { color: red; }
h2 { color: red; }
```

is same as:

```
h1, h2 { color: red; }
```

Grouping Selectors

```
p { color: red; }
p { background: black; }
```

is same as:

```
p { color: red; background: black; }
```

Combining Selectors

```
p.even { color: red; }
```

means all elements with “p” tag **and “**even” class, and similarly

```
p#first { color: red; }
```

means all elements with “p” tag **and “**first” as ID attribute

Other Selectors

- [P]
  - Select elements containing attribute P
- [P=Q]
  - Select elements containing attribute P and value Q
- :hover :link :visited
  - Select elements with a particular state
- :before :after
  - Select pseudo elements before/after an element
- And more...

Pseudo Selectors

defines a special state of an element or a phantom state that can be targeted with CSS.

Can be psuedo-class or a pseudo-element

Pseudoclass : style the selected elements ***only*** when they are in certain state

Ex: only when element is hovered over by the mouse pointer, or a checkbox when it is disabled or checked, or an input is required,

Pseudoelement: A CSS pseudo-element is used to style specified parts of an element.

- Style the first letter, or line, of an element, Insert content before, or after, the content of an element

Some pseudoclasses:

- `:default`
- `:fullscreen`
- `:focus`
- `:hover`
- `:invalid`
- `:lang()`
- `:link`
- `:not()`
- `:optional`
- `:out-of-range`
- `:read-only`
- `:required`
- `:target`
- `:valid`
- `:visited`
- ……

Some pseudoelements

- `::after`
- `::before`
- `::first-letter`
- `::first-line`
- `::selection`
- ….

Cascading..

Cascading preperences

sources of style definition for a given document element are in this order of precedence:

1. The STYLE attribute on an individual element tag
2. The STYLE element that defines a specific style sheet containing style declarations or a LINK element that links to a separate document containing the STYLE element. In a Web page, the STYLE element is placed between the TITLE statement and the BODY statement.
3. An imported style sheet, using the CSS @import notation to automatically import and merge an external style sheet with the current style sheet
4. Style attributes specified by the viewer to the browser
5. The default style sheet assumed by the browser

Cascading

- Priority among various selectors
  - Tag (low)
  - Class
  - ID
  - Style attribute (high)
- See specificity algorithm for full details

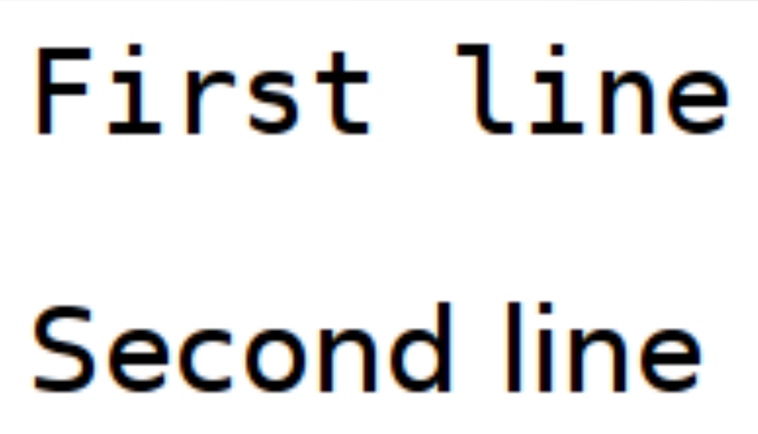
Font Family

You Write:

```
#first {
   font-family: monospace;
}


#second {
   font-family: sans-serif;
}
```

You See:



First line

Second line

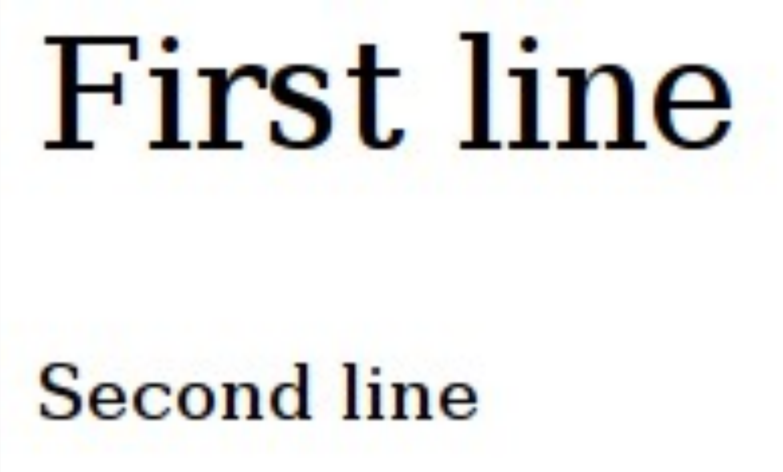
Font Size

You Write:

```
#first {
   font-size: 32px;
}


#second {
   font-size: 16px;
}
```

You See:



First line

Second line

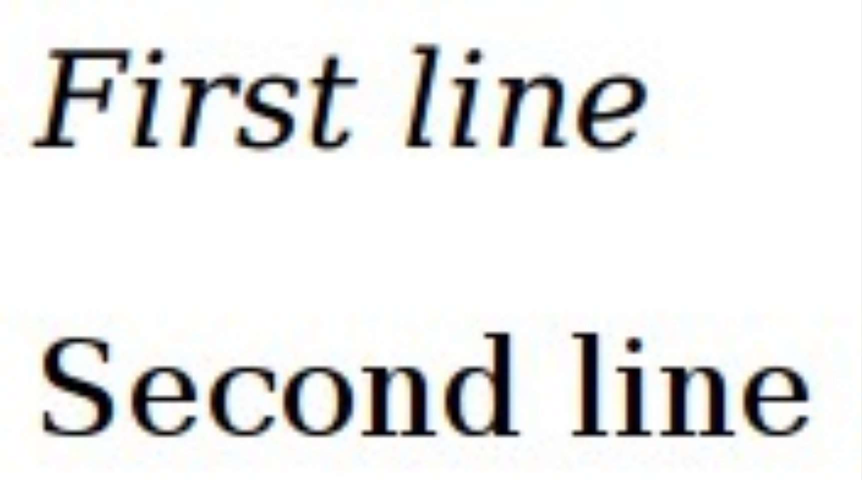
Font Style

You Write:

```
#first {
   font-style: italic;
}


#second {
   font-style: normal;
}
```

You See:



First line

Second line

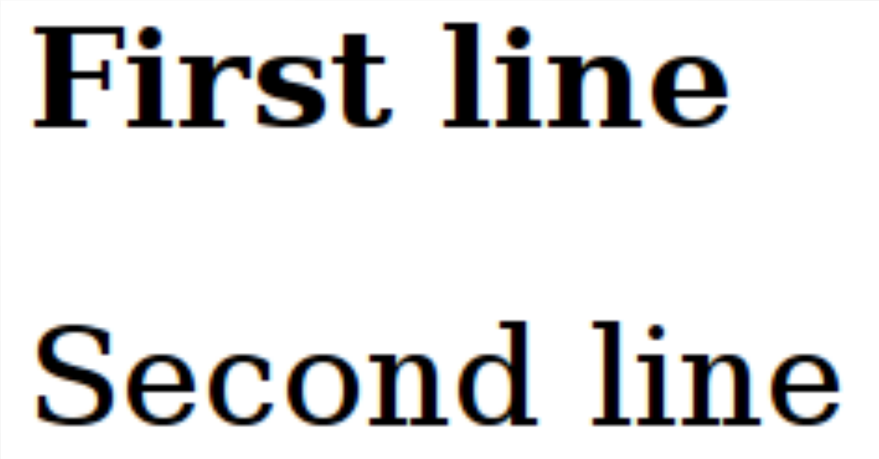
Font Weight

You Write:

```
#first {
   font-weight: bold;
}


#second {
   font-weight: normal;
}
```

You See:



First line

Second line

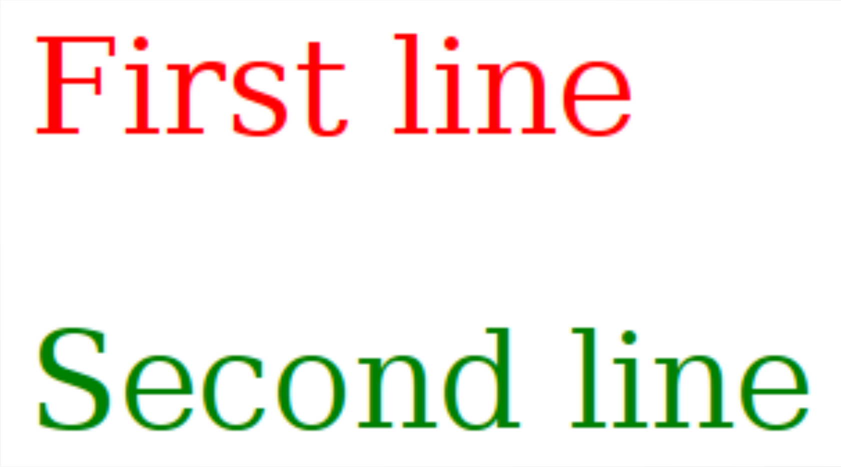
Text Color

You Write:

```
#first {
   color: red;
}


#second {
   color: green;
}
```

You See:



First line

Second line

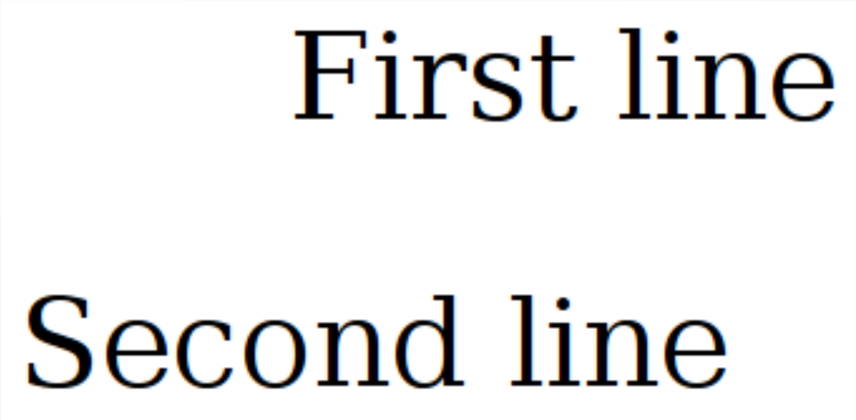
Text Color

You Write:

```
#first {
   text-align: right;
}


#second {
   text-align: left;
}
```

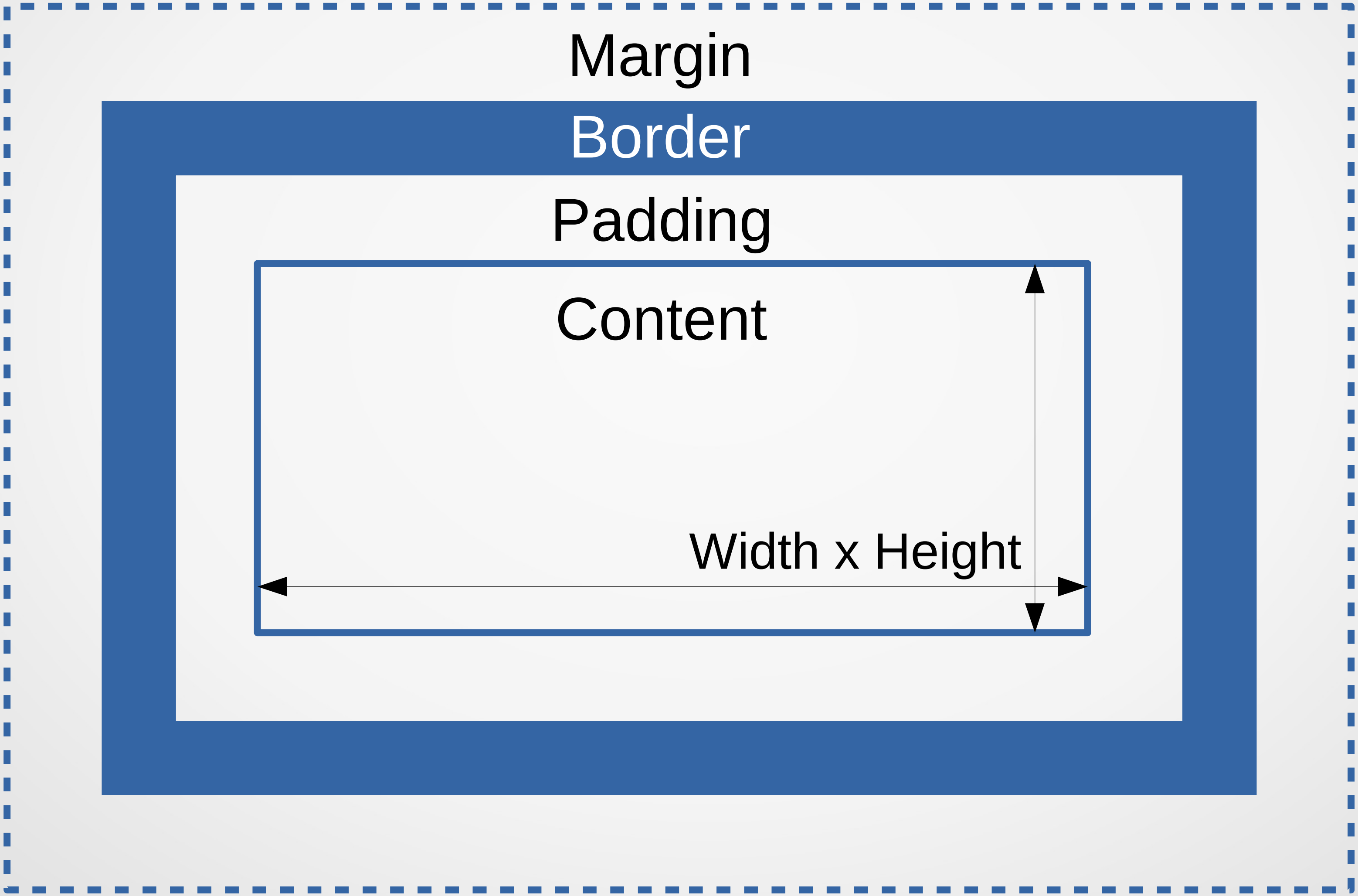
You See:



First line

Second line

Box Model


Margin
Border
Padding
Content
Width x Height