CSS

IN DEPTH (ISH)

`<p style="font-size:12px;">`

```
<style>
p {
    font-size:12px;
}
</style>
```

\<head\>

\<meta charset="UTF-8"\>

\<link rel="stylesheet" type="text/css" href="main.css"**\>**

\<title\>Untitled Document\</title\>

\</head\>

TYPE ATTRIBUTE

RELATION ATTRIBUTE

MIME (COMBINATION) Type

RELATION ATTRIBUTE

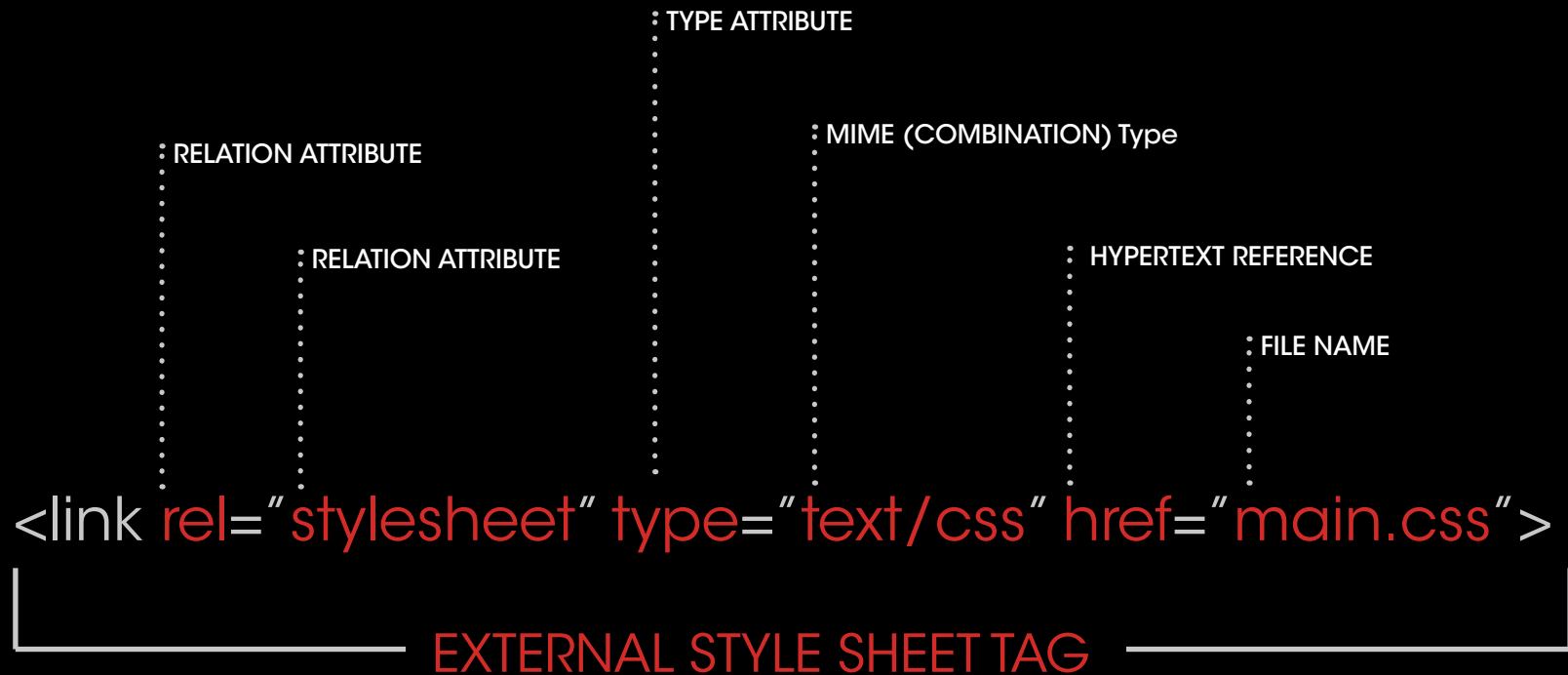HYPERTEXT REFERENCE

FILE NAME

`<link rel="stylesheet" type="text/css" href="main.css">`

EXTERNAL STYLE SHEET TAG

Folder Structure:

The way files follow each other or nest within one another creates relationships between them

root_folder
  - css
  - img
  - js
  - index.html

root_folder
  - css
      - main.css
  - img
      - logo.jpg
  - js
      - main.js
  - index.html

Inheritance & Cascading:

Certain properties are derived (**inherited**) from one's parents or ancestors.

Style information is passed down (**cascades** down) until it is overridden by a style command with more weight.

Inheritance:

Relationships deteremined from HTML structure/nesting

```
<section>
    <header>
        <h1>Header One</h1>
        <p>Lorem <em>Ipsum</em></p>
    </header>
    <p></p>
</section>
```
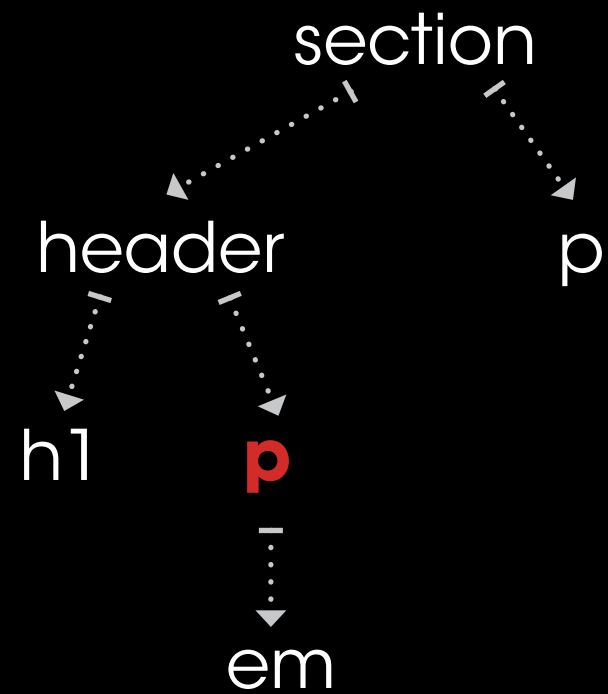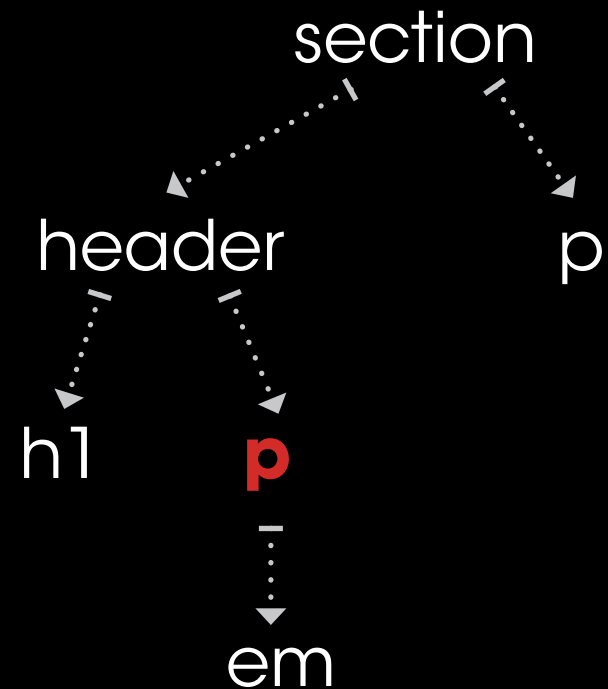
ANCESTOR

<section>

CHILD

<header>

SIBLING

<h1>Header One</h1>

<p>Lorem Ipsum**</p>**

PARENT

</header>

<p>Body Copy</p>

</section>

```css
section {
    color: black;
}
```

section
├── header
│   ├── h1
│   └── p
│   └── em
└── p

```
section {
    color: black;
}


header {
    color: white;
}
```

section
header   p
h1   p
em

```css
section {
    color: black;
}

header {
    color: white;
}

p {
    color: blue;
}
```

section
├─ header
│ ├─ h1
│ └─ p
│   └─ em
└─ p
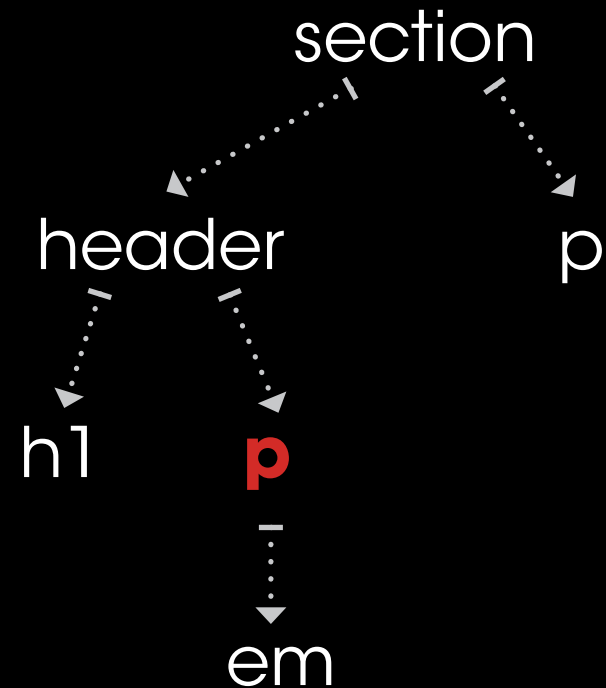
```css
section {
    color: black;
}

header {
    color: white;
}

p {
    color: blue;
}

header > p {
    color: purple;
}
```

section

header   p

h1   p

em
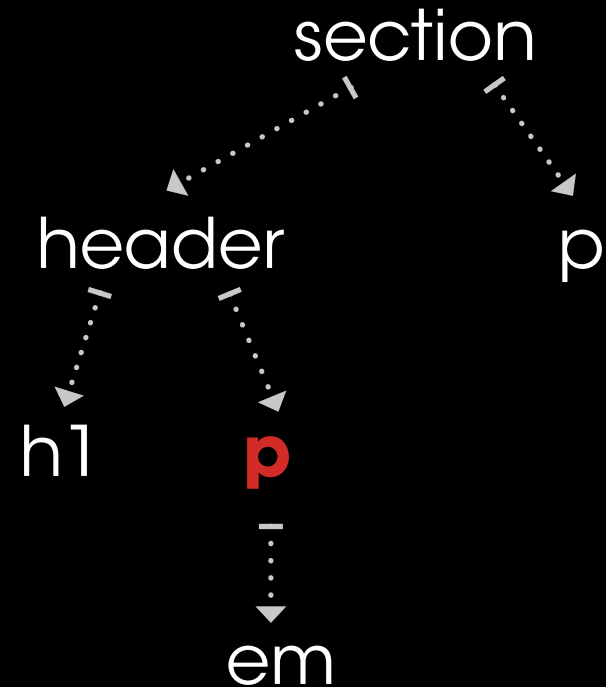
Cascading:
Order in which CSS is read, top to bottom and specificity.

```
<section>
  <header>
    <h1>Header One</h1>
    <p>Lorem <em>Ipsum</em></p>
  </header>
  <p></p>
</section>
```

top to bottom

```css
p {
    color: blue;
}

section {
    color: black;
}

header {
    color: white;
}

p {
    color: red;
}
```

```css
section {
    color: black;
}

header {
    color: white;
}

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

```
<link rel="stylesheet"

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

```
<style>

p {
    color: orange;
}

</style>
```

```
<link rel="stylesheet"

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

```
<style>

p {
    color: orange;
}

</style>
```

```
<p style="color:yellow;"> Lorem Ipsum </p>

<p> Lorem Ipsum </p>

<p style="color:pink;"> Lorem Ipsum </p>
```

specificity with elements

ELEMENT

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

CONTEXTUAL ELEMENT

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

section > header > p {
    color: purple;
}
```

specificity with classes
classify elements into a conceptual group

```
<section>
    <header>
        <h1>Header One</h1>
        <p class="lead">Lorem <em>Ipsum</em></p>
    </header>
    <p></p>
</section>
```

`<p `**`class="lead"`**`>`Lorem Ipsum`</p>`

CAN BE USED MULTIPLE TIMES

`<h1 `**`class="lead indent"`**`>`Lorem Ipsum`</h1>`

SPACE

`<p `**`class="lead" `**`class=" indent"`**`>`Lorem Ipsum`</p>`

PERIOD

NAME

TARGET ALL TAGS
WITH LEAD CLASS

```css
.lead {
    color: orange;
}
```

ELEMENT

TARGET ALL PARAGRAPH
TAGS WITH LEAD CLASS

```css
p.lead {
    color: purple;
}
```

specificity with ids
unique identifying name

```
<section>
    <header>
        <h1>Header One</h1>
        <p id="context">Lorem <em>Ipsum</em></p>
    </header>
    <p></p>
</section>
```

`<p `**`id="context"`**`>`Lorem Ipsum`</p>`

CAN ONLY HAVE ONE

`<h1 `**`class="lead"`**` `**`id="context"`**`>`Lorem Ipsum`</h1>`

SPACE

`<p `**`id="context" id="indent"`**`>`Lorem Ipsum`</p>`

HASHTAG

NAME

TARGET ALL TAGS
WITH CONTEXT ID

```css
#context {
    color: orange;
}
```

ELEMENT

UNNECESSARY OVER
IDENTIFICATION

```css
p#context {
    color: purple;
}
```

Least Amount of CSS:
Faster, cleaner code

Element

Target through cascade

Group styles

Class

Class + Element

ID

TARGETING ALL
PARAGRAPHS ··············

```
p {
    color: orange;
}
```

ALL PARAGRAPHS
CHILD OF SECTION ··············

```
section > p {
    color: orange;
}
```

ALL PARAGRAPHS,
ANCHORS, & HEADLINE 3 ··············

```
p, a, h3 {
    color: orange;
}
```

ALL CLASSES WITH
THIS VALUE ··············

```
.class-value {
    color: orange;
}
```

ALL PARAGRAPH ELEMENTS
WITH THIS CLASS VALUE ··············

```
p.class-value {
    color: orange;
}
```

ALL IDS WITH
THIS VALUE ··············

```
#id-value {
    color: orange;
}
```

Universal Selector:

Targets everything - very slow.

```
*  {
      color: orange;
}
```

Styling the Anchor:

```
p {
    color: blue;
}

a {
    color: orange;
}
```

Pseudo Selectors:

Used to define a special state of an element.

a or **a:link** /* unvisited link */

a:visited /* visited link */

a:hover /* mouse over link */

a:active /* selected link */

YAY :D

```css
a {
    color: blue;
}

a:visited {
    color: orange;
}

a:hover {
    color: orange;
}

a:active {
    color: orange;
}
```

NAY D:

```css
a:hover {
    color: blue;
}

a {
    color: orange;
}

a:visited {
    color: orange;
}

a:active {
    color: orange;
}
```

YAY :D

```css
a {
    color: blue;
}

a:hover {
    color: orange;
}
```

NAY D:

```css
a:hover {
    color: blue;
}

a {
    color: orange;
}
```

CSS Shorthand:
Combining multiple CSS declarations into one.

```css
p {
    font-style: italic;
    font-weight: bold;
    font-size: 12px;
    line-height: 16px;
    font-family: Arial, sans-serif;
}
```

```css
p {
    font: italic bold 12px/16px Arial, sans-serif;
}
```

SEPERATED WITH A SPACE

```
p {
    font: style weight size/line-height family;
}
```

MUST BE IN THIS
ORDER

REQUIRED

REQUIRED

```css
p {
    font: 12px Arial, sans-serif;
}
```

```css
p {
    font: bold 12px Arial, sans-serif;
}
```

```css
p {
    margin-top: 20px
    margin-right: 40px;
    margin-bottom: 20px;
    margin-left: 40px;
}
```

```
p {
    margin: 20px 40px;
}
```

TOP & BOTTOM

LEFT & RIGHT

ALL SIDES

```css
p {
    margin: 20px;
}
```

THERE ARE MORE!

REFER TO YOUR BOOK AND THE INTERNET. YOU WILL NEED TO KNOW THEM.

OO CSS:

Object Oriented CSS is about separating styles to be easily reusable and less repeatable

```
#btn1 {
    width: 300px;
    height:100px;
    color: red;
}
```

`<p id="btn1" > </p>`

```
#btn2 {
    width: 300px;
    height:100px;
    color: blue;
}
```

`<p id="btn2" > </p>`

```css
.btn {
    width: 300px;
    height:100px;
}

.red {
    color: red;
}

.blue {
    color: blue;
}
```

```html
<p class="btn red" > </p>


<p class="btn blue" > </p>
```

Conventional Names:

Common OO-CSS class names

BUTTON	btn
LEAD IN PARAGRAPH	lead
BORDER	brdr
BRAND COLOR	brd
NAVIGATION BAR	navbar
MODULE	mod
ROW	row
TEXT	txt
MARGIN TOP	mt
MARGIN BOTTOM	mb
SUB NAVIGATION	subnav