```
<!DOCTYPE html>
<html>
<head>
    <title>CSCI 111 Web Programming and Problem Solving</title>
</head>
<body>
    <h1>Week-3-Lecture</h1>
    <h2>Introduction to CSS Part II</h2>
    <ul>
        <li Talgat Manglayev</li>
        <li Irina Dolzhikova</li>
        <li Aigerim Yessenbayeva</li>
    </ul>
</body>
</html>
```

outline

```
<ol>
   <li>CSS selectors</li>
   <ol>
      <li>DOM based</li>
      <li>Pseudo class</li>
      <li>Pseudo elements</li>
   </ol>
   <li>Conflict Resolution</li>
   <ol>
      <li>Cascade</li>
      <li>Inheritance</li>
      <li>Specificity</li>
   </ol>
</ol>
```

CSS Syntax

What if we want to change a particular paragraph? How do we select a specific element?

What if one element is changed in several places? How to resolve the conflicts?

```
<style>
p { color : blue; }
h1 { color : red; }
h2 {color: green;}
</style>
```

Questions and Answers related to HTML and CSS

Do I really need to learn HTML and CSS?

HTML and CSS, unlike programming languages, have no competitors. If you want to create web pages and start a career in web design, web development, or even web journalism, you must have a basic understanding of HTML and CSS. Fortunately, the fundamentals of HTML and CSS are simple.

Is HTML CSS enough to get a job?

Can I get a job with just HTML and CSS? It's a question we get here at Skillcrush all the time, and the short answer is yes, with caveats. If you want to start working in tech, the first thing you should do is learn HTML and CSS.

What is the fastest way to learn HTML and CSS?

As with any new skill, hands-on practice is the best way to learn HTML and build these foundational skills. Once you've gone through some introductory tutorials and feel comfortable with the basics, challenge yourself to start building simple websites from scratch.

CSS selectors

To distinguish between elements, we use selectors:

- Element Types (Tags)
- Element Classes
- Element Attributes
- Element IDs
- DOM based
- pseudo-class
- pseudo-elements

Element Selectors

Selection of one or more **elements**:

```
body
{
    margin: 0;
    padding: 0;
}
h1, p
{
    color: blue; font-size: 12pt;
}
```

Selects and changes the properties of body element

Selects and changes the properties of h1 and p elements (*Note the comma* in between)

Class Selectors

Class - identifier that can group together multiple elements.

```
<p class="second"> … </p>

<li class="second item"> … </li>
```

Definition of the class **second** for two elements. Elements can belong to several classes.

```
.second { color: red; }

li.item { color: purple; }
```

Selects the elements with the specified class (second or item)
Note a dot before class name

Attribute Selectors

```
h1[style]
{
    text-align: center;
}
li[name]
{
    color: gray;
}
li[name="item1"]
{
    color: orange;
}
```

Selects all h1 tags with their style attribute defined

Selects all **li** elements with their name attribute defined as well as those which have

specific values for name attribute

ID Selectors

ID is an identifier of an element unique within the document

Assignment of **ID** to the element

```
<p id="last"> … </p>
```

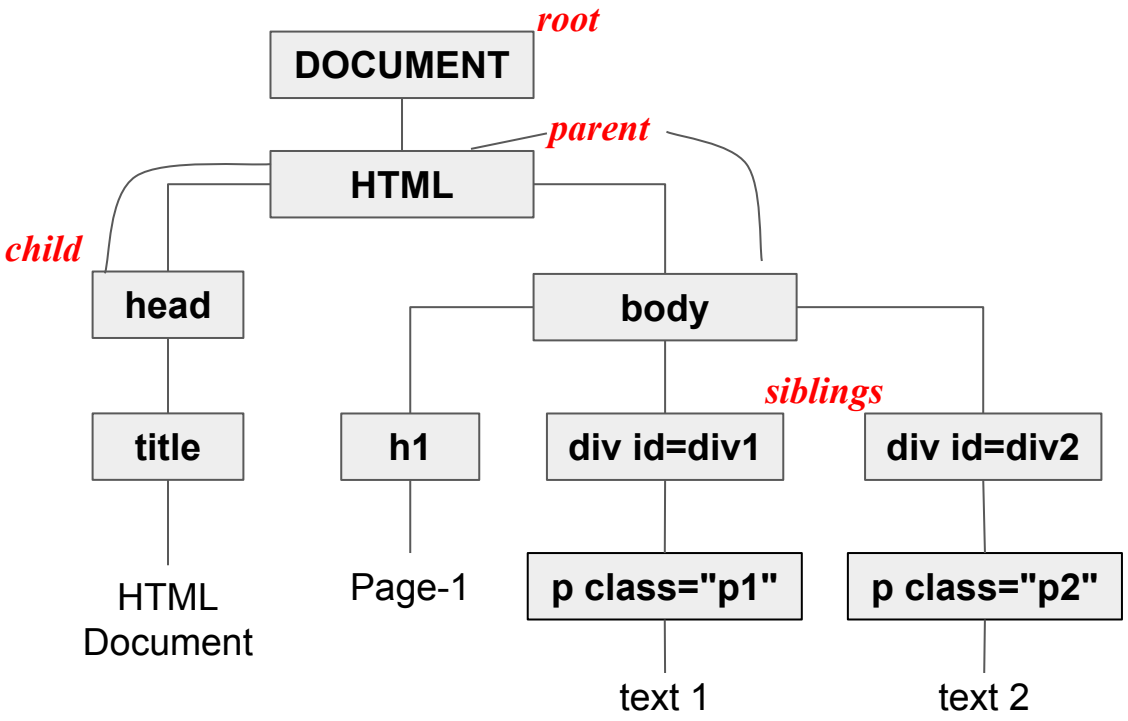
Selects the elements by its ID *Note a hash* before the ID

```
#last
{
    color: red;
    font-size: 15pt;
}
```

DOM based

- HTML document can be viewed as a tree-like structure.
- This structure is represented as Document Object Model (DOM) in memory
- Elements are called nodes

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <title>HTML Document</title>
5  </head>
6  <body>
7      <h1>Page-1</h1>
8      <div id = "div1">
9          <p class="p1">text 1</p>
10     </div>
11     <div id = "div2">
12         <p class="p2">text 2</p>
13     </div>
14 </body>
15 </html>
```



Children selectors

```
/*selects all li elements of div */
div li
{
     font-size: large;
}
/*selects only the direct h2 elements of div */
div > h2
{
     font-size: xx-large;
}
/*selects only the direct h2 elements of div */
body > h2
{
     color: green;
}
```

```
<body>
<h1>Global Rankings</h1>
     <div id="div-1">
          <h2>Ranking of countries by area</h2>
          <ol class="countries area">
               <li>Russia</li>
               <li>Canada</li>
                    <li>China</li>
                    <li>United States</li>
                    <li>Brazil</li>
               </ol>
     </div>
<h2>Resources:</h2>
<ul>
     <li><a
href="https://www.worldometers.info">Largest
Countries in the World (by area)</a>
     </li>
</ul>
</body>
```

page-6.html

Pseudo-Class Selectors

A pseudo-class is used to define a special state of an element.

```
selector:pseudo-class
{
    property: value;
}
```

https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-classes

page-7.html

states of tag a

```
/* unvisited link */
a:link
{
     color: red;
}
/* visited link */
a:visited
{
     color: green;
}
/* mouse over link */
a:hover
{
     color: hotpink;
}
/* selected link */
a:active
{
     color: blue;
}
```

Pseudo-Class Selectors

To select the first child of an element:

```
ul li:first-child
{
    color: blue;
}
```

To select the n-th child of an element:

```
ul li:nth-child(3)
{
    color: blue;
}
```

https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-classes

Pseudo-Element Selectors

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

```
selector::pseudo-element
{
    property: value;
}
```

https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-elements

```
p::first-letter
{
color: red;
}
p::first-line
{
color: green;
}
p::selection
{
color: red;
background: yellow;
}
::marker
{
color: orange;
}
```

page-9.html

Conflict Resolution

Application of CSS rules depends on three main concepts:

- Cascade
- Specificity
- Inheritance

To resolve the conflicts, we need to understand them well.

https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/Styling_basics/Handling_conflicts

Cascade

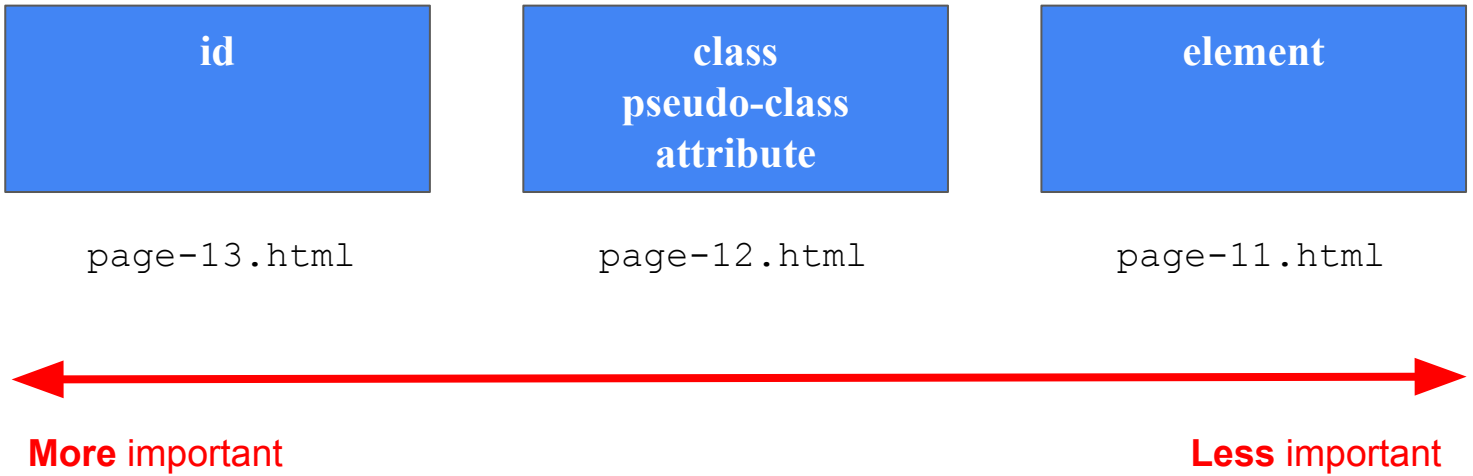
concept which means that the origin and the order of CSS rules matter, i.e. the latest rule is applied.

```
<head>
    <style>
    h1
    {
        color: blue;
    }
    /* this rule overrides  the previous rule */
    h1
    {
        color: green;
    }
    </style>
</head>
<body>
    <h1>How to quickly master novel skills</h1>
</body>
```

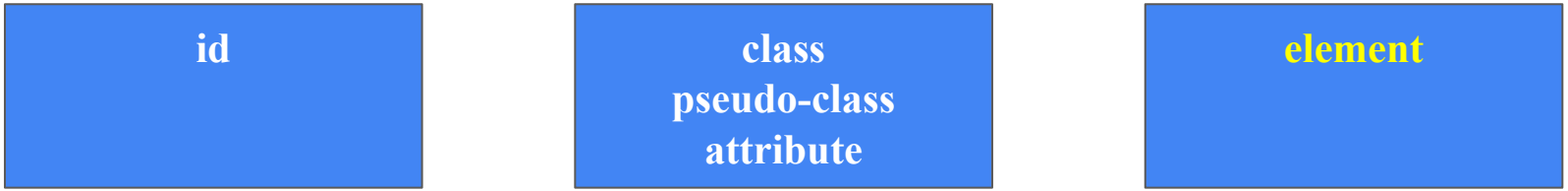
page-10.html

Specificity

weight that the browser uses to decide which property value is applied to an element.
The weight is composed of 3 numbers based on the location of a rule and the number of appearance of the selectors.
Inline style has more weight than id, and not included into calculation.



Specificity



page-11.html

```
<style>
/*p > span has more weigh as it has two element selectors p and span*/
p > span
{
     color: red;
}
/*span has less weigh as it has only one element selectors span*/
span
{
     color: blue;
}
</style>
```

0 0 2

0 0 1

Specificity

id	class pseudo-class attribute	element

page-12.html

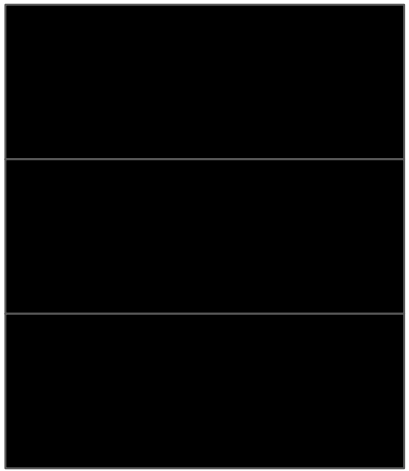
```
<style>
     .list-1 { font-size: x-large; }
     ul { font-size: small;    }
     ul li:last-child { color: blue; }
     li:last-child { color: yellow; }
     li[name]{ color: red; }
     li{color: green; }
</style>
```

Specificity

id	class pseudo-class attribute	element

page-13.html

```
<style>
#last
{
      color: blue;
}
.spanClass
{
      color: red;
}
p > span
{
      color: green;
}
</style>
```



Cascade works if Specificity weight is the same

Inheritance

Elements can inherit the properties defined in their parents or ancestors.

- Some properties can’t be inherited like weight or margin.
- CSS provides special property values for elements:
  - inherit – turn on inheritance
  - initial – property’s default
  - revert – browser's default
  - unset – set to inherit or initial

```
body
{
    color: blue;
}
/* revert to browser’s default value*/
h1
{
    color: revert;
}
```

page-14.html, page-15.html, page-16.html

!important

Importance is the mechanism to apply a rule no matter what the order specificity or inheritance of other rules.

```
h1
{
    color: blue !important;
}
/* this rule is not applied */
h1
{
    color: green;
}
```

However, it is not recommended to use it unless really necessary

Summary

The selection can be done:

- using element's type, class, attributes and ID
- based on DOM (structure of HTML)
- using pseudo-classes and pseudo-elements

Three concepts are important in conflict resolution

- Cascade
- Specificity
- Inheritance

Use **important** keyword only when really necessary

bonus info