CS2204 Fundamentals of Internet Applications Development

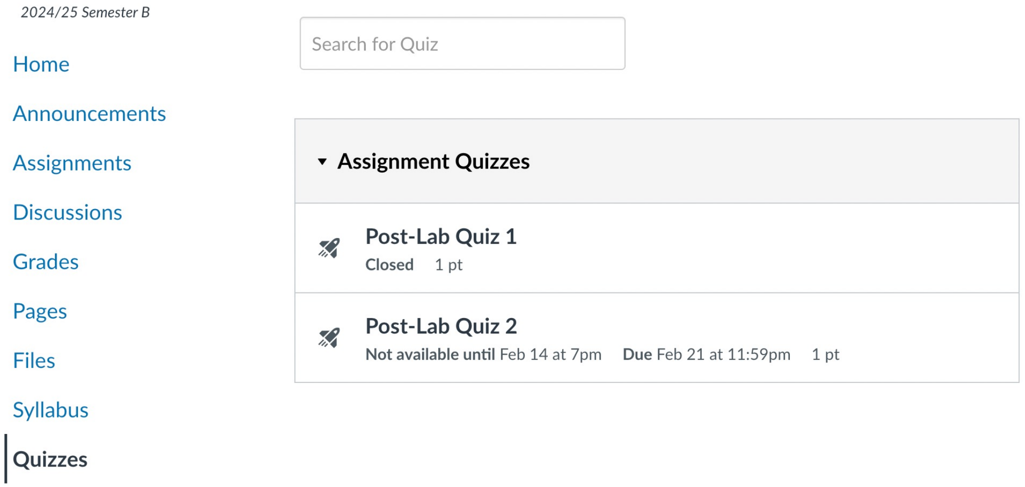
Lecture 4 CSS – Part 2

Computer Science, City University of Hong Kong

Semester B 2024-25

Announcement: Post-lab Quiz

Do not forget post-lab quiz that is available every Friday 7 PM when there is a lab session scheduled. Starting this week, they will be counted into your final grade



2024/25 Semester B

- Home
- Announcements
- Assignments
- Discussions
- Grades
- Pages
- Files
- Syllabus
- Quizzes

Search for Quiz

▾ Assignment Quizzes

Post-Lab Quiz 1
Closed 1 pt

Post-Lab Quiz 2
Not available until Feb 14 at 7pm Due Feb 21 at 11:59pm 1 pt

If you missed post-lab quiz for lab 1, please refer to the slides of Lec 01 (p. 10-11) for instructions.

Question Time

About the HTML element `<form>`

- What are the two key attributes within the `<form>`tag that handle the information entered by users, and what each of them are used for?

- How to add a text input area with descriptions of the area in a form?

- How to add clickable buttons in a form **without** creating a `<button>` element?

Question Time

`<input>` attributes

- How to limit the length of a text input?

- How to make an input required?

- Which of the followings are new input types?
  text, radio, checkbox, number, password, date, email, submit, reset

Agenda

CSS Basics

- How to apply style on HTML elements

CSS Selectors

- ID, Class, Group
- Contextual selector
- Advanced selector
- Cascading and inheritance

CSS Layout

Agenda

CSS Basics

- How to apply style on HTML elements

CSS Selectors

- ID, Class, Group
- Contextual selector
- Advanced selector
- Cascading and inheritance

CSS Layout

CSS

Cascading Style Sheets (**CSS**) describes how the HTML elements should be **displayed** by specifying the fonts, colors, layout and placement of these HTML elements

How to set/apply styles for Elements?

Inline Style

Set styles directly in the Web page using the style attribute of an element

```
<div style="color: #003366; font-size:.8em;">
```

Advantages:

- Good for diagnostics (testing or finding errors)
- A quick way to make sure the style is properly set

Disadvantage: extremely difficult to use or update because you need to find the elements one by one. It should not be used unless really necessary

Embedded Style

Put CSS rules in the `<head>` section.

- The `<style>` tag is used to enclose all rules, which will be applied to all elements selected in the entire page (but not other pages)

Advantages:

- Write once, apply to the whole page
- Easier to change
- Good for styles for one (this) page only
- Can use for overriding styles (to be covered later in cascading)
- Can use the media attribute in the <style> tag to specifies what media/device type the styles are optimized for

```
<head>
    <title>Page Title</title>

    <!-- Embedded stylesheet -->
    <style media="print">
        h2 {
            font-size: 1.5rem;
            color
            text-
        }

        p {
            font-variant: italic;
        }
    </style>
</head>
```

Intended to be viewed on a screen in **print preview mode**.

External Style

Put all CSS rules in a separate .css file (**a style sheet**)

Two ways to use an external style sheet

```
<head>
  <link rel="stylesheet" href="mystyle.css" type="text/css" media="all">
  <title>Page title</title>
</head>
<body>
```

Link style

```
<head>
     <style type="text/css" media="all">
       @import url("mystyle.css");
     </style>
   <title>Page title</title>
</head>
```

Import style

Disadvantages: need to be careful with the effect of multiple style sheets interaction

Advantages:

- Good for set up themes (consistent styles) applying to all pages in a Web site
- Easy to update
- Increase accessibility through the use of consistent styles
- Can use media attribute

Code Example: lec04-01-external.html

External style: Link

Linking to a style sheet means using the `<link>` tag in the head section to load all CSS rules and apply their effect to the page

- Note that more than one style sheet can be linked
- In the link tag, `rel` is **always** "`stylesheet`", but `type` is **not always** "`text/css`", because the initial design of style sheet allows other languages, not just CSS
- `media = "all"` is used for all media type devices

```
<head>
  <link rel="stylesheet" type="text/css" href="styles1.css" media="all">
  <link rel="stylesheet" type="text/css" href="styles2.css" media="all">
  <link rel="stylesheet" type="text/css" href="styles3.css" media="all">
</head>
```

External style: import

Rules are in `.css` file but links differently with `@import`

- Syntax: `@import` *`url|string list-of-mediaqueries;`*

```
  <head>
    <meta charset="utf-8">
    <style type="text/css" media="all">
      @import url("mystyle.css");
    </style>
</head>
```

Code Example: lec04-01-external.html

Value	Description
url|string	A url or a string representing the location of the resource to import. The url may be absolute or relative
list-of-mediaqueries	A comma-separated list of media queries conditioning the application of the CSS rules defined in the linked URL

Must be at the top of the document (but after any @charset declaration)

Supports media queries, so you can allow the import to be media-dependent.

We can also import style conditionally

`@import "printstyle.css"` print;

Import style only if the `media = "print"`

External style: Link vs. Import

- Link
  - Link one stylesheet to a html page
  - Load CSS style in parallel

- Import
  - Import one stylesheet to another stylesheet – convenient to link multiple stylesheets together and easy management of complex styles
  - Each @import statement sequentially results in a separate HTTP request, may slow down the loading time.
  - Cannot be handled by old browsers

- In general, link is preferred over import for better performance

A quick summary: different ways to apply styles

Inline - No CSS rules are required, only properties and values set in the style attribute

Embedded - put inside the `<head>` section with `<style>`

External

- Link: CSS rules are put in a `.css` file linked to Web pages using `<link>`
- Import - again rules are in `.css` file but links differently with `@import` directive/command

How to decide the ways to place CSS style?

Never use **Inline style** except in special circumstances (e.g., debugging)

If the styles are only used in one page: Embedded or Link style

When the styles will be used in more than one pages: Link style

If the styles are selectively used (e.g., on screen or on print): Embedded or Link style should be used because the media attribute can be used

Whether to use Import style depends on the strategy of the Website

Agenda

CSS Basics

- How to apply style on HTML elements

CSS Selectors

- ID, Class, Group
- Contextual selector
- Advanced selector
- Cascading and inheritance

CSS Layout

What is CSS selector?

The rule that is used to select and target specific HTML elements within a document.

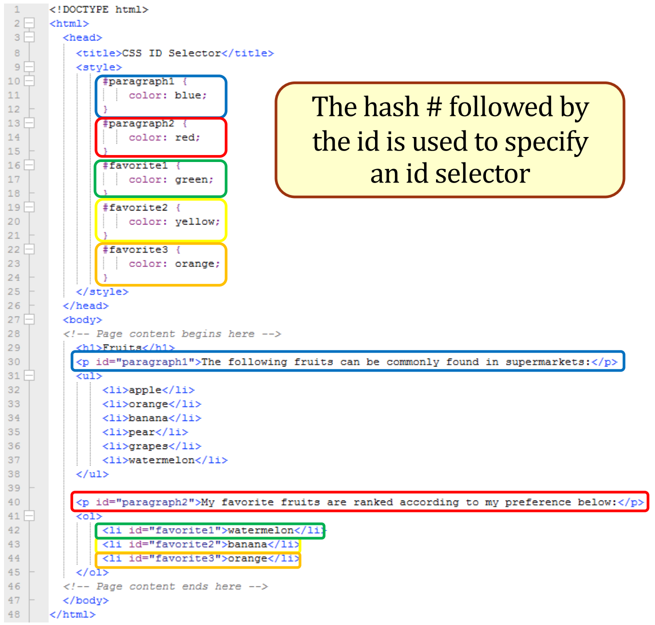
Each rule consists of:

```
selector {property1: value1; property2: value2; ...}
```

- A **selector** can be HTML tag(s), or class/id name(s)
- These rules are embeded in either the `<head>`section or an external file, **NOT** mixing with HTML

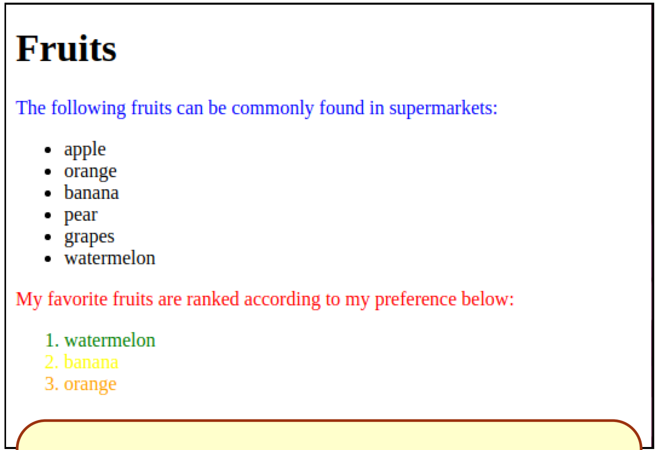
ID Selector (1)

Each html element can be labeled with an id. A CSS style can be applied to an element by specifying its #id



```
<!DOCTYPE html>
<html>
  <head>
    <title>CSS ID Selector</title>
    <style>
        #paragraph1 {
            color: blue;
        }
        #paragraph2 {
            color: red;
        }
        #favorite1 {
            color: green;
        }
        #favorite2 {
            color: yellow;
        }
        #favorite3 {
            color: orange;
        }
    </style>
  </head>
  <body>
  <!-- Page content begins here -->
    <h1>Fruits</h1>
    <p id="paragraph1">The following fruits can be commonly found in supermarkets:</p>
    <ul>
        <li>apple</li>
        <li>orange</li>
        <li>banana</li>
        <li>pear</li>
        <li>grapes</li>
        <li>watermelon</li>
    </ul>

    <p id="paragraph2">My favorite fruits are ranked according to my preference below:</p>
    <ol>
        <li id="favorite1">watermelon</li>
        <li id="favorite2">banana</li>
        <li id="favorite3">orange</li>
    </ol>
  <!-- Page content ends here -->
  </body>
</html>
```

The hash # followed by the id is used to specify an id selector



Fruits

The following fruits can be commonly found in supermarkets:

- apple
- orange
- banana
- pear
- grapes
- watermelon

My favorite fruits are ranked according to my preference below:

1. watermelon
2. banana
3. orange

Each id used to label an html element should be unique, i.e., it can only be used once, and no two html elements should have the same id

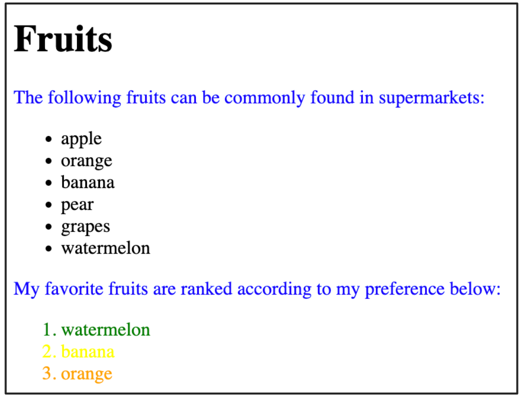
Code Example: lec04-02-CSS-id-selector.html

ID Selector (2)

What will happen if multiple elements shared the same id?

- Modern browsers are faculty-tolerant
- However, this practice is not encouraged
  - "id": identifying single element, particularly useful when applying animations/events, where shared ids among multiple elements can cause issues

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>CSS ID Selector</title>
    <style>
      #paragraph1 {
        color: blue;
      }
      #paragraph2 {
        color: red;
      }
      #favorite1 {
        color: green;
      }
      #favorite2 {
        color: yellow;
      }
      #favorite3 {
        color: orange;
      }
    </style>
  </head>
    <body>
    <!-- Page content begins here -->
      <h1>Fruits</h1>
      <p id="paragraph1">The following fruits can be commonly found in supermarkets:</p>
      <ul>
          <li>apple</li>
          <li>orange</li>
          <li>banana</li>
          <li>pear</li>
          <li>grapes</li>
          <li>watermelon</li>
      </ul>

      <p id="paragraph1">My favorite fruits are ranked according to my preference below:</p>
      <ol>
          <li id="favorite1">watermelon</li>
          <li id="favorite2">banana</li>
          <li id="favorite3">orange</li>
      </ol>
    <!-- Page content ends here -->
    </body>
</html>
```



Fruits

The following fruits can be commonly found in supermarkets:

- apple
- orange
- banana
- pear
- grapes
- watermelon

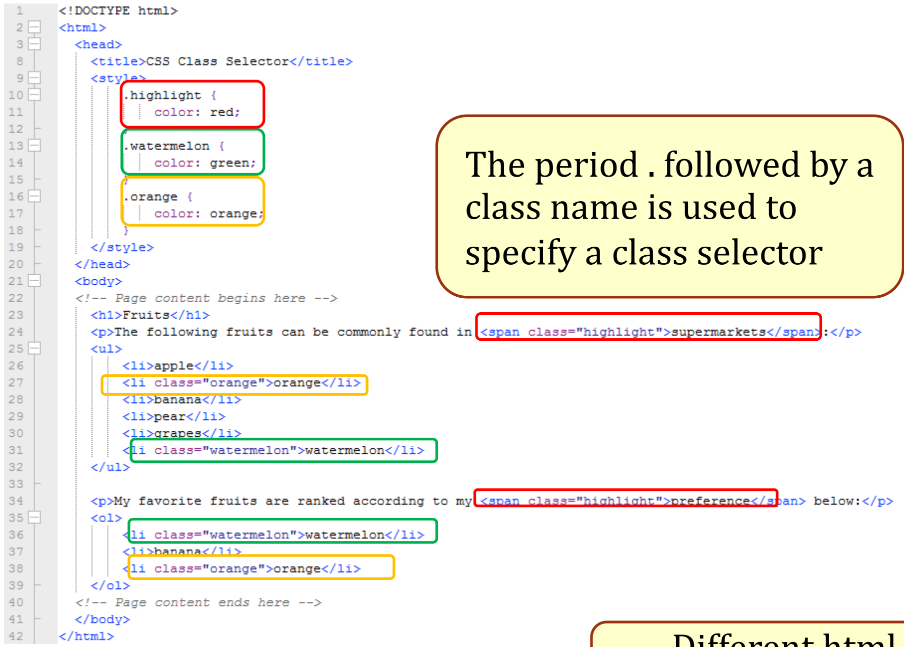
My favorite fruits are ranked according to my preference below:

1. watermelon
2. banana
3. orange

Code Example: lec04-02-CSS-id-selector.html

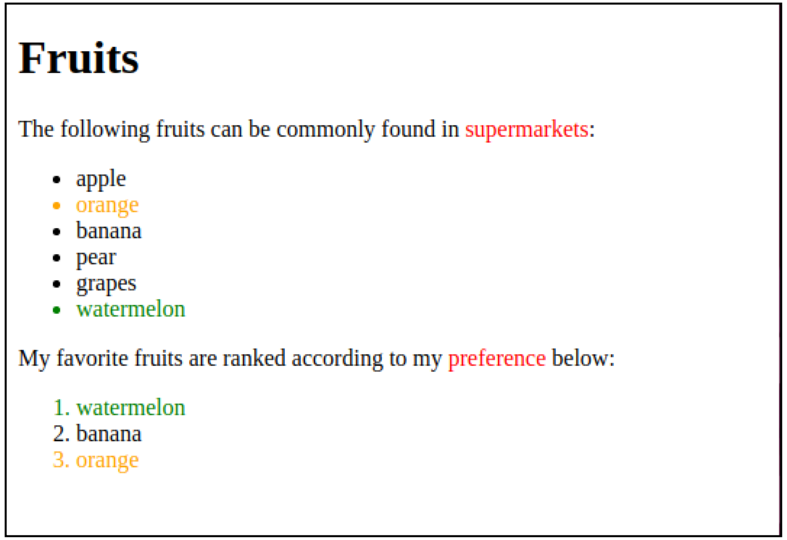
Class Selector

A CSS style can be applied to all elements with the same class name



```
<!DOCTYPE html>
<html>
  <head>
    <title>CSS Class Selector</title>
    <style>
      .highlight {
        color: red;

      .watermelon {
        color: green;

      .orange {
        color: orange;
      }
    </style>
  </head>
  <body>
  <!-- Page content begins here -->
    <h1>Fruits</h1>
    <p>The following fruits can be commonly found in <span class="highlight">supermarkets</span>:</p>
    <ul>
      <li>apple</li>
      <li class="orange">orange</li>
      <li>banana</li>
      <li>pear</li>
      <li>grapes</li>
      <li class="watermelon">watermelon</li>
    </ul>

    <p>My favorite fruits are ranked according to my <span class="highlight">preference</span> below:</p>
    <ol>
      <li class="watermelon">watermelon</li>
      <li>banana</li>
      <li class="orange">orange</li>
    </ol>
  <!-- Page content ends here -->
  </body>
</html>
```

The period . followed by a class name is used to specify a class selector



Fruits

The following fruits can be commonly found in supermarkets:

- apple
- orange
- banana
- pear
- grapes
- watermelon

My favorite fruits are ranked according to my preference below:

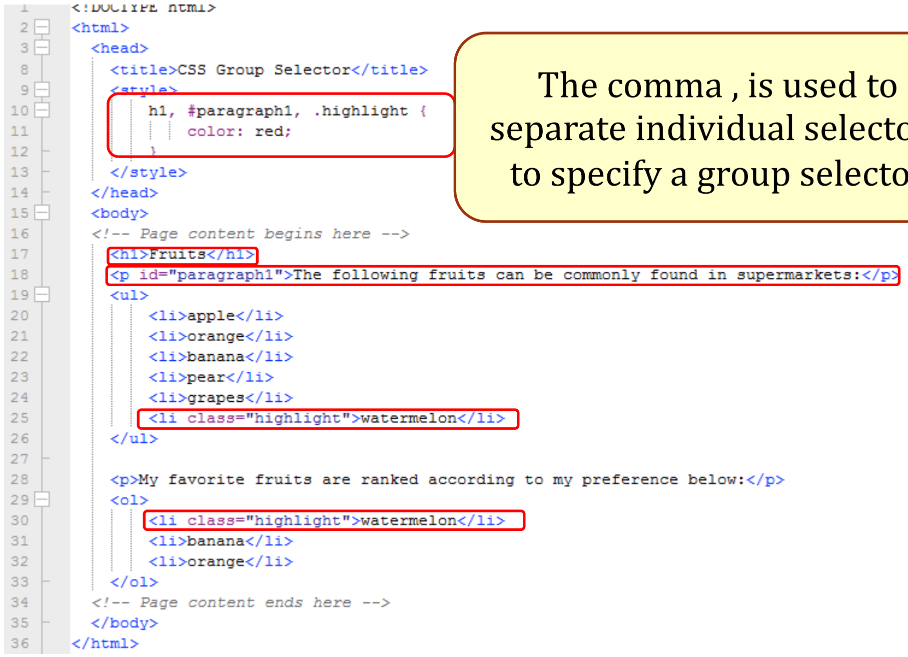
1. watermelon
2. banana
3. orange

Different html elements can be labeled with the same class name

Code Example: lec04-03-CSS-class-selector.html

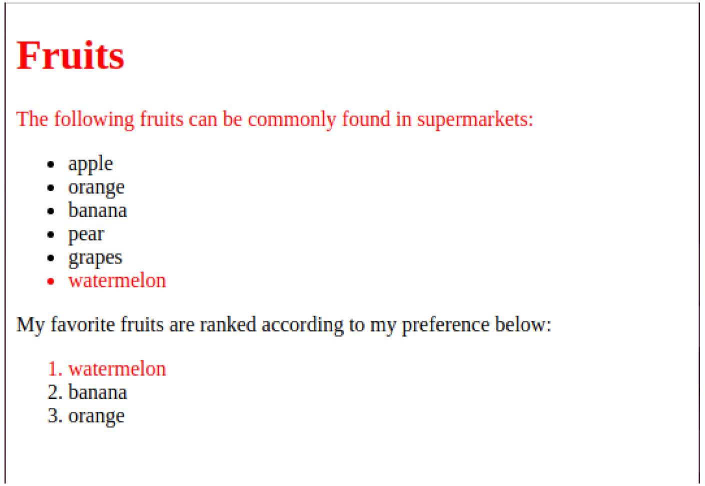
Group Selector

A CSS style can be applied to a group of elements



```
<!DOCTYPE html>
<html>
  <head>
    <title>CSS Group Selector</title>
    <style>
        h1, #paragraph1, .highlight {
            color: red;
        }
    </style>
  </head>
  <body>
  <!-- Page content begins here -->
    <h1>Fruits</h1>
    <p id="paragraph1">The following fruits can be commonly found in supermarkets:</p>
    <ul>
        <li>apple</li>
        <li>orange</li>
        <li>banana</li>
        <li>pear</li>
        <li>grapes</li>
        <li class="highlight">watermelon</li>
    </ul>

    <p>My favorite fruits are ranked according to my preference below:</p>
    <ol>
        <li class="highlight">watermelon</li>
        <li>banana</li>
        <li>orange</li>
    </ol>
  <!-- Page content ends here -->
  </body>
</html>
```

The comma , is used to separate individual selectors to specify a group selector



Fruits

The following fruits can be commonly found in supermarkets:

- apple
- orange
- banana
- pear
- grapes
- watermelon

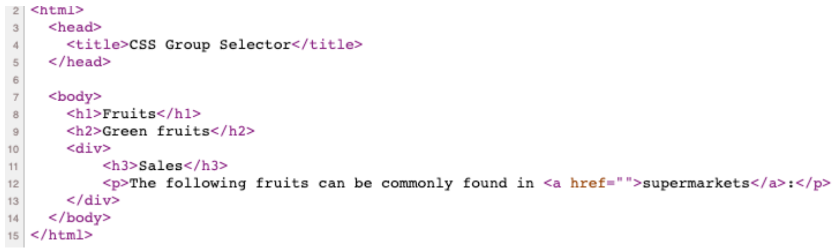
My favorite fruits are ranked according to my preference below:

1. watermelon
2. banana
3. orange

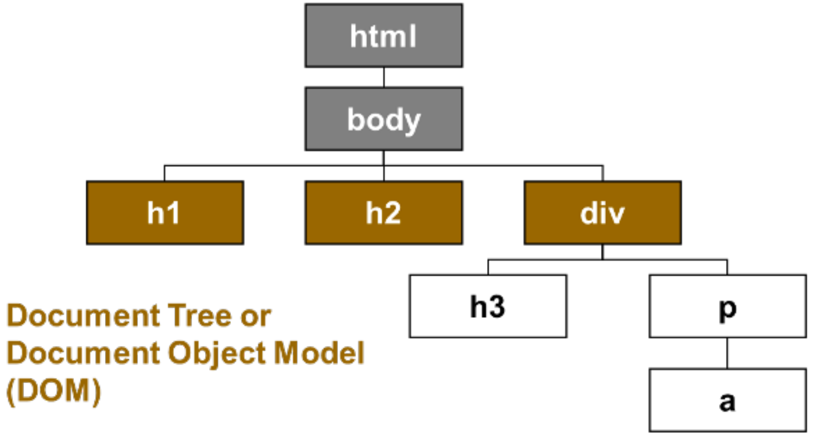
Code Example: lec04-04-CSS-group-selector.html

Contextual selector: What is it?

Also known as descendant selector, a type of CSS selector that targets elements based on their relationship to another element in the HTML structure.



```
<html>
  <head>
    <title>CSS Group Selector</title>
  </head>

  <body>
    <h1>Fruits</h1>
    <h2>Green fruits</h2>
    <div>
        <h3>Sales</h3>
        <p>The following fruits can be commonly found in <a href="">supermarkets</a>:</p>
    </div>
  </body>
</html>
```



Contexts: the position where you are relative to the surrounding
Document Tree: defines the context inside a Web page

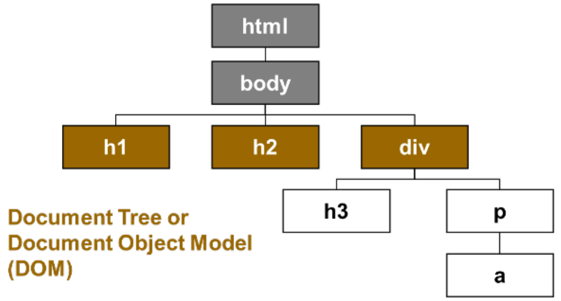
Contextual and document tree

Different kinds of relationships are defined in a document tree, known as **Document Object Model (DOM)**:

parent: an element directly above another element (e.g., html is the parent of body)

child: an element directly one level below another element (e.g., h1 is the child of body)

ancestor: parent, grandparent, great-grandparent, or higher within the tree



Which elements are h3's ancestors?

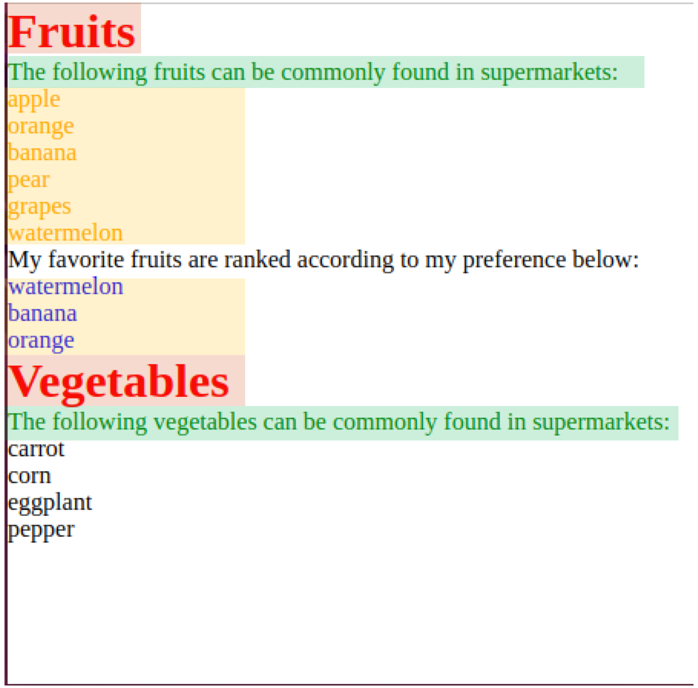
Which elements are html's descendants?

descendant: a child, grandchild, great-grandchild or further descendent down the line

sibling: elements that share the same parent (e.g., h1 and h2; h3 and p)

These relationships are used to form contextual selector

Contextual Selectors



Fruits

The following fruits can be commonly found in supermarkets:

apple
orange
banana
pear
grapes
watermelon

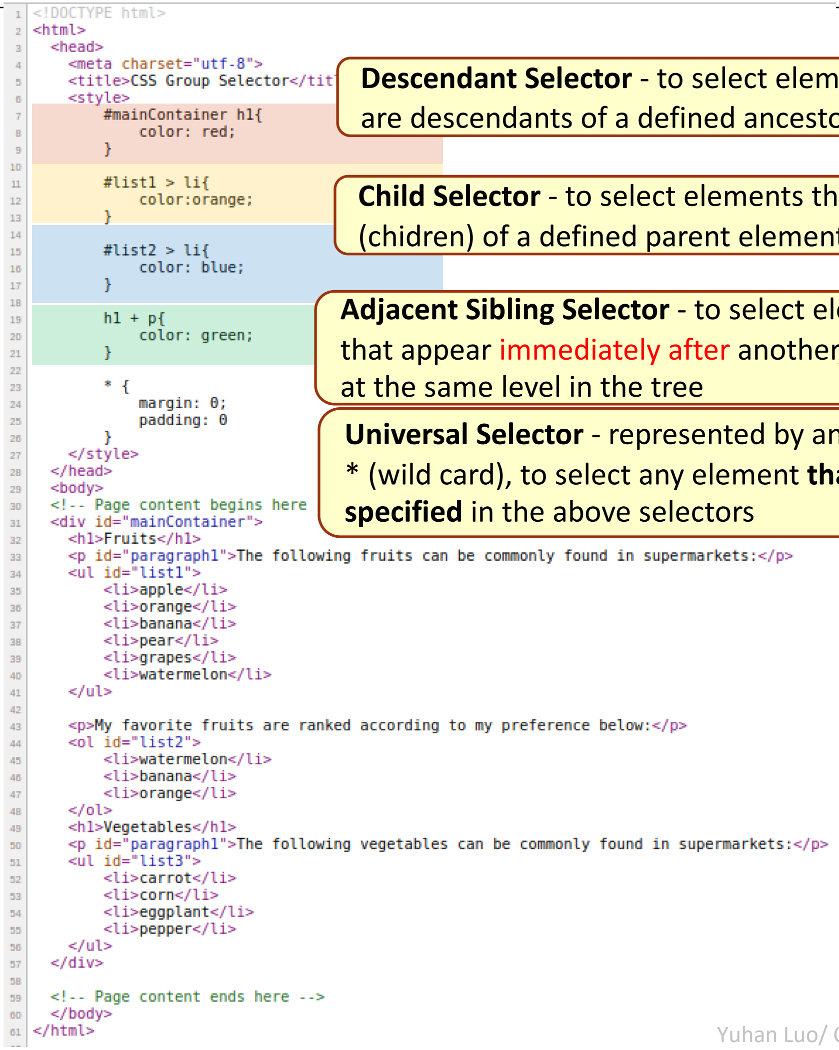
My favorite fruits are ranked according to my preference below:

watermelon
banana
orange

Vegetables

The following vegetables can be commonly found in supermarkets:

carrot
corn
eggplant
pepper



```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>CSS Group Selector</tit
    <style>
        #mainContainer h1{
            color: red;
        }

        #list1 > li{
            color:orange;
        }

        #list2 > li{
            color: blue;
        }

        h1 + p{
            color: green;
        }

        * {
            margin: 0;
            padding: 0
        }
    </style>
  </head>
  <body>
  <!-- Page content begins here
  <div id="mainContainer">
    <h1>Fruits</h1>
    <p id="paragraph1">The following fruits can be commonly found in supermarkets:</p>
    <ul id="list1">
        <li>apple</li>
        <li>orange</li>
        <li>banana</li>
        <li>pear</li>
        <li>grapes</li>
        <li>watermelon</li>
    </ul>

    <p>My favorite fruits are ranked according to my preference below:</p>
    <ol id="list2">
        <li>watermelon</li>
        <li>banana</li>
        <li>orange</li>
    </ol>
    <h1>Vegetables</h1>
    <p id="paragraph1">The following vegetables can be commonly found in supermarkets:</p>
    <ul id="list3">
        <li>carrot</li>
        <li>corn</li>
        <li>eggplant</li>
        <li>pepper</li>
    </ul>
  </div>

  <!-- Page content ends here -->
  </body>
</html>
```

Descendant Selector - to select elements that are descendants of a defined ancestor element

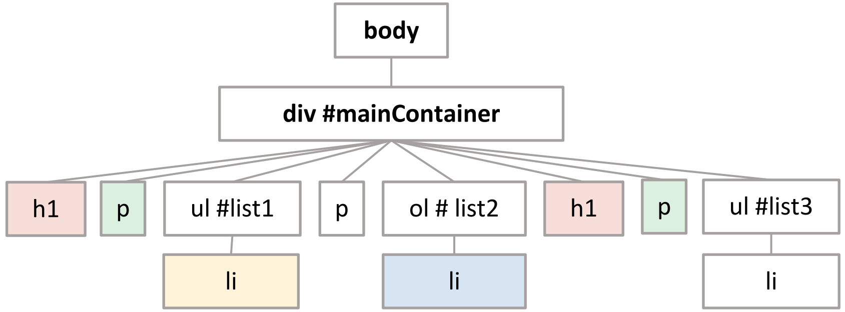
Child Selector - to select elements that are child (chidren) of a defined parent element

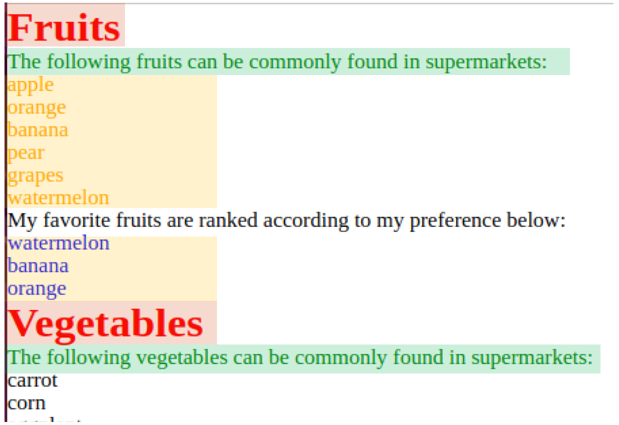
Adjacent Sibling Selector - to select elements that appear immediately after another, must be at the same level in the tree

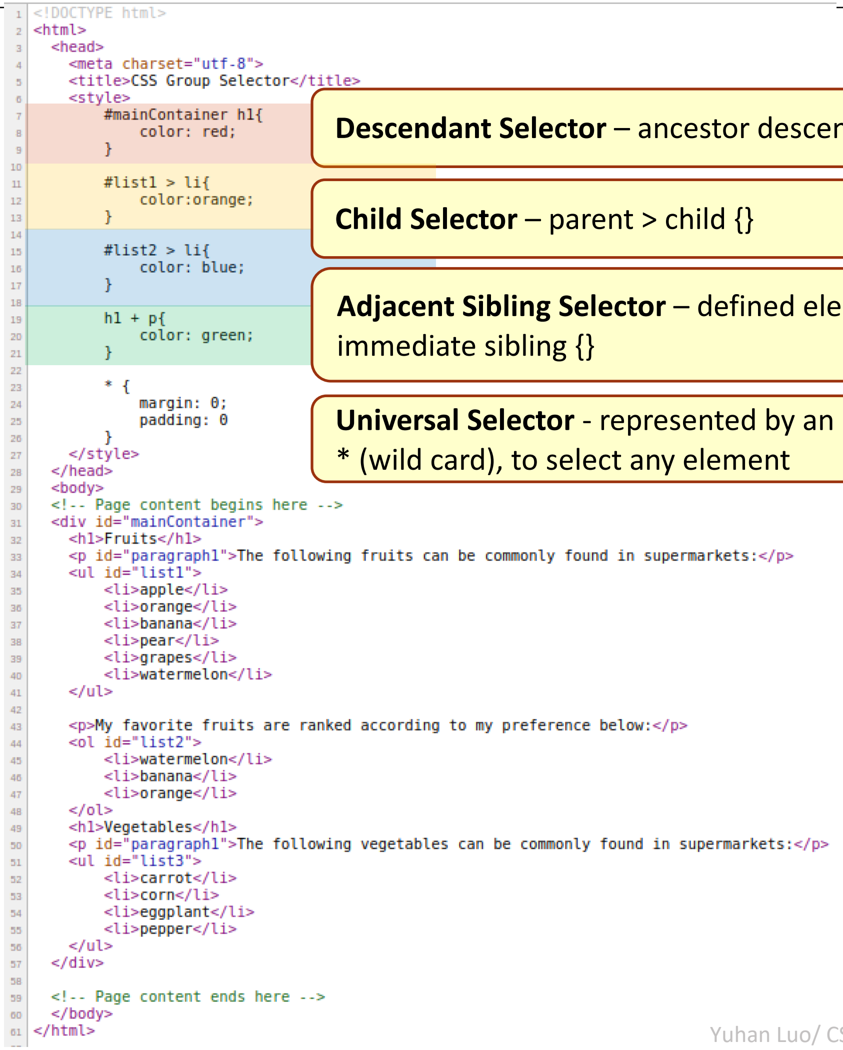
Universal Selector - represented by an asterisk * (wild card), to select any element **that are not specified** in the above selectors

Code Example: lec04-05-contextual-selector.html

Contextual Selectors explained




Fruits
The following fruits can be commonly found in supermarkets:
apple
orange
banana
pear
grapes
watermelon
My favorite fruits are ranked according to my preference below:
watermelon
banana
orange
Vegetables
The following vegetables can be commonly found in supermarkets:
carrot
corn


```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>CSS Group Selector</title>
    <style>
        #mainContainer h1{
            color: red;
        }

        #list1 > li{
            color:orange;
        }

        #list2 > li{
            color: blue;
        }

        h1 + p{
            color: green;
        }

        * {
            margin: 0;
            padding: 0
        }
    </style>
  </head>
  <body>
  <!-- Page content begins here -->
  <div id="mainContainer">
    <h1>Fruits</h1>
    <p id="paragraph1">The following fruits can be commonly found in supermarkets:</p>
    <ul id="list1">
        <li>apple</li>
        <li>orange</li>
        <li>banana</li>
        <li>pear</li>
        <li>grapes</li>
        <li>watermelon</li>
    </ul>

    <p>My favorite fruits are ranked according to my preference below:</p>
    <ol id="list2">
        <li>watermelon</li>
        <li>banana</li>
        <li>orange</li>
    </ol>
    <h1>Vegetables</h1>
    <p id="paragraph1">The following vegetables can be commonly found in supermarkets:</p>
    <ul id="list3">
        <li>carrot</li>
        <li>corn</li>
        <li>eggplant</li>
        <li>pepper</li>
    </ul>
  </div>

  <!-- Page content ends here -->
  </body>
</html>
```

Descendant Selector – ancestor descendant {}

Child Selector – parent > child {}

Adjacent Sibling Selector – defined element + immediate sibling {}

Universal Selector - represented by an asterisk * (wild card), to select any element

Code Example: lec04-05-contextual-selector.html

Advanced Selector

Also known as complex selectors, allow developers to target elements based on more specific criteria and relationships within the HTML structure.

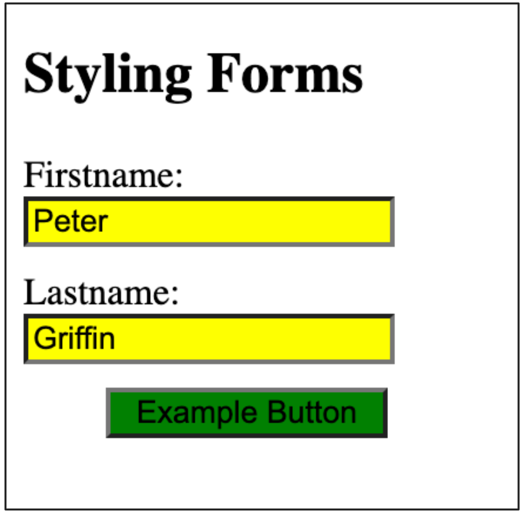
They provide more advanced and precise selection capabilities compared to basic selectors

- Attribute selector
- Pseudo class/element

Attribute Selector

Attribute Selector - select an element based on its attribute or attribute value.

```
<!DOCTYPE html>
<html>
<head>
<style>
input[type=text] {
  width: 150px;
  display: block;
  margin-bottom: 10px;
  background-color: yellow;
}

input[type=button] {
  width: 120px;
  margin-left: 35px;
  display: block;
  background-color: green;
}
</style>
</head>
<body>

<h2>Styling Forms</h2>

<form name="input" action="" method="get">
  Firstname:<input type="text" name="Name" value="Peter" size="20">
  Lastname:<input type="text" name="Name" value="Griffin" size="20">
  <input type="button" value="Example Button">
</form>

</body>
</html>
```

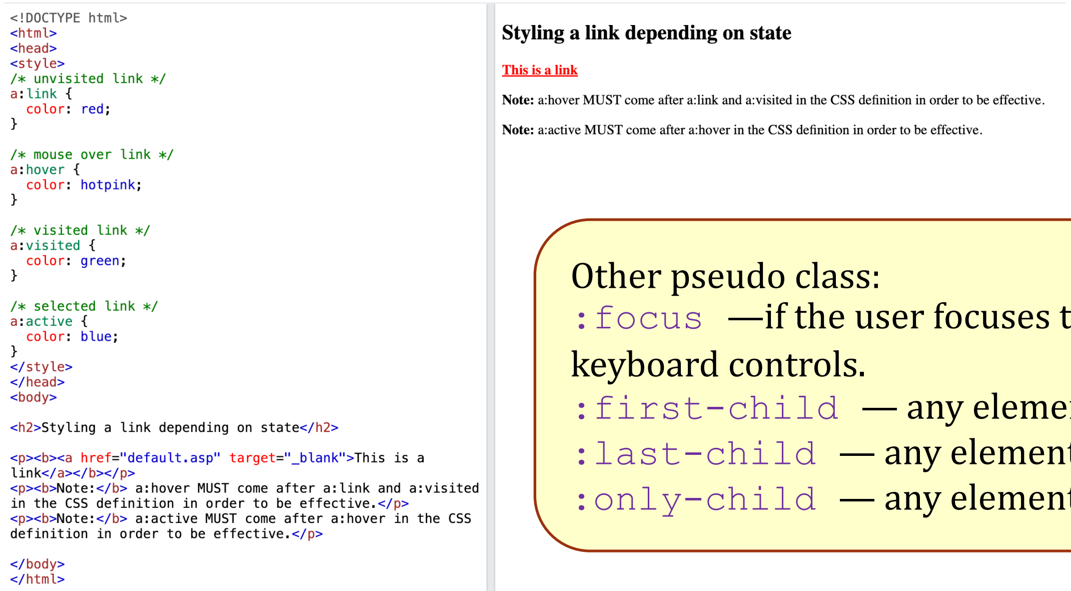


Code Example: lec04-06-CSS-attribute-selector.html

Pseudo class

A class not defined by using class name but by the **state** of the element

For example, `p:first-child {property: value;}` select all <p> that must be the first child, not just child, of its parent



```
<!DOCTYPE html>
<html>
<head>
<style>
/* unvisited link */
a:link {
  color: red;
}

/* mouse over link */
a:hover {
  color: hotpink;
}

/* visited link */
a:visited {
  color: green;
}

/* selected link */
a:active {
  color: blue;
}
</style>
</head>
<body>

<h2>Styling a link depending on state</h2>

<p><b><a href="default.asp" target="_blank">This is a
link</a></b></p>
<p><b>Note:</b> a:hover MUST come after a:link and a:visited
in the CSS definition in order to be effective.</p>
<p><b>Note:</b> a:active MUST come after a:hover in the CSS
definition in order to be effective.</p>

</body>
</html>
```

Styling a link depending on state

This is a link

Note: a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective.

Note: a:active MUST come after a:hover in the CSS definition in order to be effective.

Other pseudo class:
`:focus` —if the user focuses the element by clicking or using keyboard controls.
`:first-child` — any element that is the first child of an element
`:last-child` — any element that is the last child of an element
`:only-child` — any element that is the only child of an element

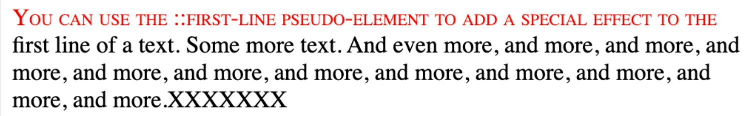
Code Example: lec04-07-CSS-pseudo-class-selector.html

Pseudo element

Similar to pseudo class, but is used to **style a specific part of an element**. It allows you to insert and style content that is not part of the HTML markup (e.g., dynamically generated content)

```
<!DOCTYPE html>
<html>
<head>
<style>
p::first-line {
  color: #ff0000;
  font-variant: small-caps;
}

p::after {
  content: "XXXXXXX";
}
</style>
</head>
<body>

<p>You can use the ::first-line pseudo-element to add
a special effect to the first line of a text. Some
more text. And even more, and more, and more, and
more, and more, and more, and more, and more, and
more, and more, and more, and more.</p>

</body>
</html>
```

Code Example: lec04-08-CSS-pseudo-element-selector.html



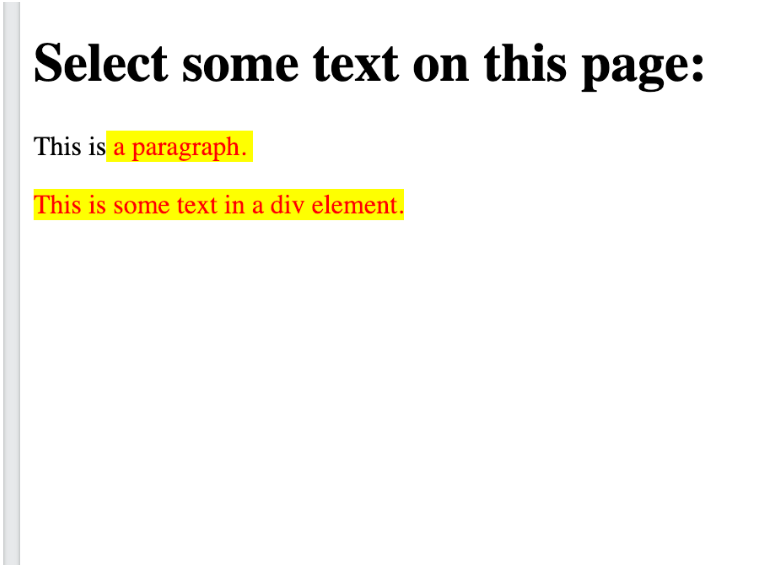
YOU CAN USE THE ::FIRST-LINE PSEUDO-ELEMENT TO ADD A SPECIAL EFFECT TO THE first line of a text. Some more text. And even more, and more, and more, and more, and more, and more, and more, and more, and more, and more, and more, and more.XXXXXXX

Other pseudo elements:
`::first-letter` – the first letter of the element
`::after` – insert content after an element
`::before` – insert content before an element
`::marker` – select the markers of the element (e.g., bullet points)
`::selection` – select the selected text

More Pseudo elements

The `::selection` pseudo-element matches the portion of an element that is selected by a user.

```
<!DOCTYPE html>
<html>
<head>
<style>
::selection {
  color: red;
  background: yellow;
}
</style>
</head>
<body>

<h1>Select some text on this page:</h1>
<p>This is a paragraph.</p>
<div>This is some text in a div element.</div>

</body>
</html>
```



Select some text on this page:

This is a paragraph.

This is some text in a div element.

Code Example: lec04-09-CSS-pseudo-element-selection.html

Question time

What if we want to add special effects to the second letter of a paragraph instead of the first letter?

Cascading and inheritance

So far we have covered so many ways to style a HTML element. What will happen if an element is stylized with different rules/selectors? For example …

```
<head>
<style>
h1 {color: red;}
.heading {color: blue;}
#text {color: pink;}
</style>
</head>

<body>
<h1 class = “heading” id= “text” style=“color: green;”>CS 2204</h1>
</body>
```

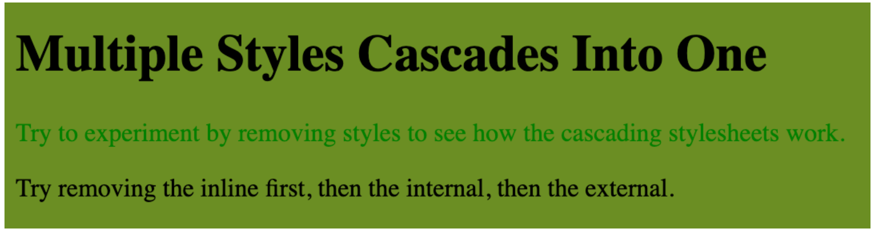
What color will the <h1> element “CS 2204” be displayed in the webpage?

Cascade Order

If different styles are specified for HTML elements, the styles will cascade into new styles with the following priority

- Priority 1: Inline styles
- Priority 2: External and internal style sheets
- Priority 3: Browser default

```html
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
<style>
body {background-color: lightblue;}
</style>
</head>
<body style="background-color: olivedrab">

<h1>Multiple Styles Cascades Into One</h1>
<p>Try to experiment by removing styles to see how the cascading
stylesheets work.</p>
<p>Try removing the inline first, then the internal, then the
external.</p>

</body>
</html>
```


Multiple Styles Cascades Into One

Try to experiment by removing styles to see how the cascading stylesheets work.

Try removing the inline first, then the internal, then the external.

Code Example: lec05-11-cascade.html

Inheritance

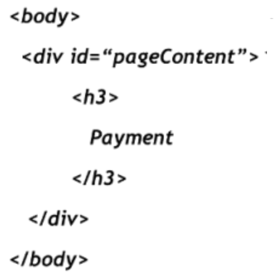
Consider the following CSS rules acting on the page in the diagram:

```
body { color: blue; }
#pageContent { font-size: 1em; }
h3 { text-transform: uppercase; }
```

The <`h3`> element has 3 rules affecting it for three properties although two rules are not explicitly used for it

CSS rules set for ancestors go on affecting descendants is called inheritance.

Note that **not all properties** would have inheritance effect



```
<body>
  <div id="pageContent">
      <h3>
        Payment
      </h3>
   </div>
</body>
```

Code Example: lec04-12-inheritance.html

Who Will Win In Cascading?

Origin of styles - **user agent** (i.e., browser default), **author** (i.e., you) or **user** (those who are looking at your Web page)

Types of style - inline, embedded or link

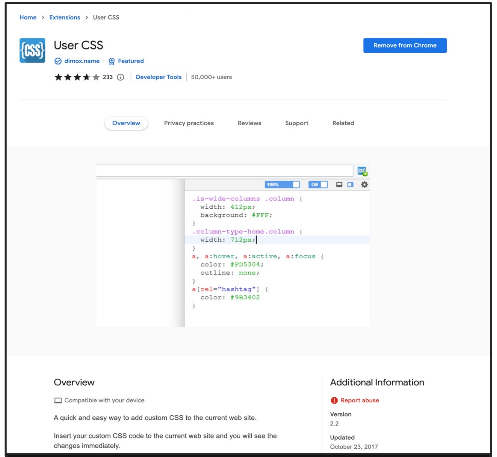
Selector - how the element is selected in the rule (specificity)

The **order** of applying CSS rules (typically style applied overrides previous style with the same specificity)

Origin of Styles

Consider the **origin** - according to the standard order of precedence is **author > user > user agent**

- Author style: supplied by the author of a webpage.
- User style sheets: supplied by the user of the browser.
- Default style (user agent): supplied by the browser vendor.


Home > Extensions > User CSS

User CSS

dimox.name Featured

★★★★★ 233 | Developer Tools | 50,000+ users

```
.is-wide-columns .column {
  width: 412px;
  background: #FFF;
}
.column-type-home.column {
  width: 712px;
}
a, a:hover, a:active, a:focus {
  color: #FD5304;
  outline: none;
}
a[rel="hashtag"] {
  color: #9B3402
}
```

Overview

Compatible with your device

A quick and easy way to add custom CSS to the current web site.

Insert your custom CSS code to the current web site and you will see the changes immediately.

Additional Information

Report abuse

Version
2.2

Updated
October 23, 2017

An example of user style

For any rule, it can be declared as **important** and results in higher priority, e.g., `p {margin-left: 5px !important}`

- with **`!important`** the precedence order of origin becomes - user important > author important > author normal > user normal > user agent. !
- important should **only be used with care** in author style; Otherwise, it becomes somewhat like Inline Style

Selector (Specificity)

Compare the priority of tag (type), id, and class

```
<!DOCTYPE html>
<html>
<head>
<style>
#demo {color: blue;}
.test {color: green;}
p {color: red;}
</style>
</head>
<body>

<p id="demo" class="test">Hello
World!</p>

</body>
</html>
```

Hello World!

Priority 0: In-line style
Priority 1: ids (e.g., `#demo`)
Priority 2: Classes, pseudo-classes, attribute selectors (e.g., `.test`, `:hover`, `[href]`)
Priority 3: Elements and pseudo-elements (e.g., `h1`, `:before`)

The universal selector (`*`) has no effect on specificity

Code Example: lec04-13-specificity.html

Order of Styles

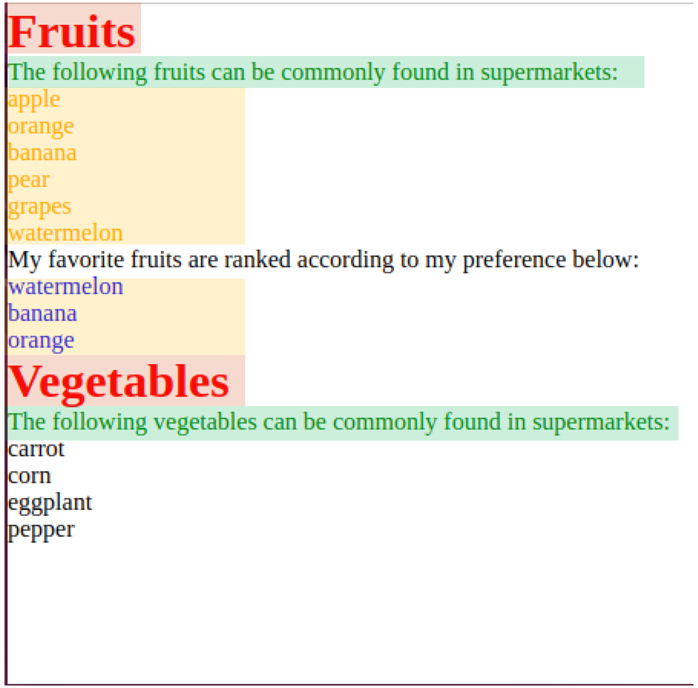
Author Styles (in a Web page):

- inline styles ALWAYS override other styles and are of the highest priority, **except** `!important`
- embedded and linked styles have no difference in priority and only depend on their order of application, i.e., whether the <link> tag before or after <style> tag

```
<!DOCTYPE html>
<html>
<head>
<style>
.test {color: green;}
.test {color: blue;}
</style>
</head>
<body>

<p class="test">Hello World!</p>

</body>
</html>
```

```
Hello World!
```

Back to the contextual selectors



Fruits

The following fruits can be commonly found in supermarkets:

apple
orange
banana
pear
grapes
watermelon

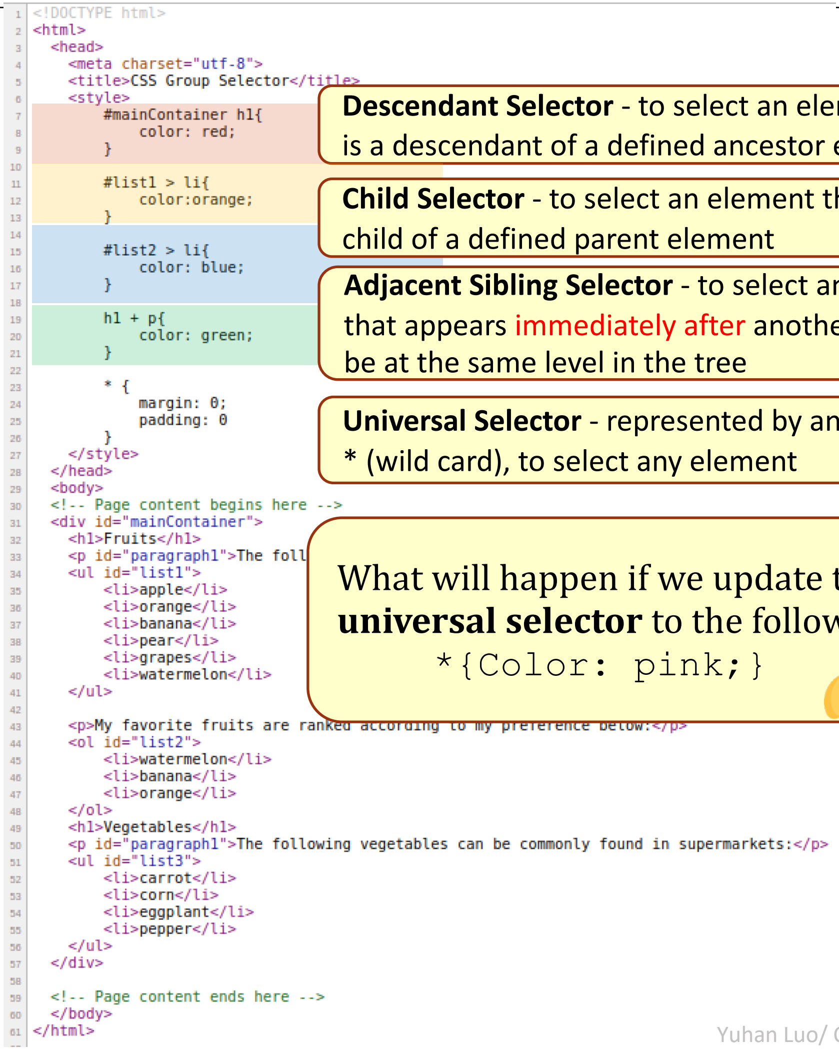
My favorite fruits are ranked according to my preference below:

watermelon
banana
orange

Vegetables

The following vegetables can be commonly found in supermarkets:

carrot
corn
eggplant
pepper



```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>CSS Group Selector</title>
    <style>
        #mainContainer h1{
            color: red;
        }

        #list1 > li{
            color:orange;
        }

        #list2 > li{
            color: blue;
        }

        h1 + p{
            color: green;
        }

        * {
            margin: 0;
            padding: 0
        }
    </style>
  </head>
  <body>
  <!-- Page content begins here -->
  <div id="mainContainer">
    <h1>Fruits</h1>
    <p id="paragraph1">The foll
    <ul id="list1">
        <li>apple</li>
        <li>orange</li>
        <li>banana</li>
        <li>pear</li>
        <li>grapes</li>
        <li>watermelon</li>
    </ul>

    <p>My favorite fruits are ranked according to my preference below:</p>
    <ol id="list2">
        <li>watermelon</li>
        <li>banana</li>
        <li>orange</li>
    </ol>
    <h1>Vegetables</h1>
    <p id="paragraph1">The following vegetables can be commonly found in supermarkets:</p>
    <ul id="list3">
        <li>carrot</li>
        <li>corn</li>
        <li>eggplant</li>
        <li>pepper</li>
    </ul>
  </div>

  <!-- Page content ends here -->
  </body>
</html>
```

Descendant Selector - to select an element that is a descendant of a defined ancestor element

Child Selector - to select an element that is a child of a defined parent element

Adjacent Sibling Selector - to select an element that appears immediately after another, must be at the same level in the tree

Universal Selector - represented by an asterisk * (wild card), to select any element

What will happen if we update the **universal selector** to the following:

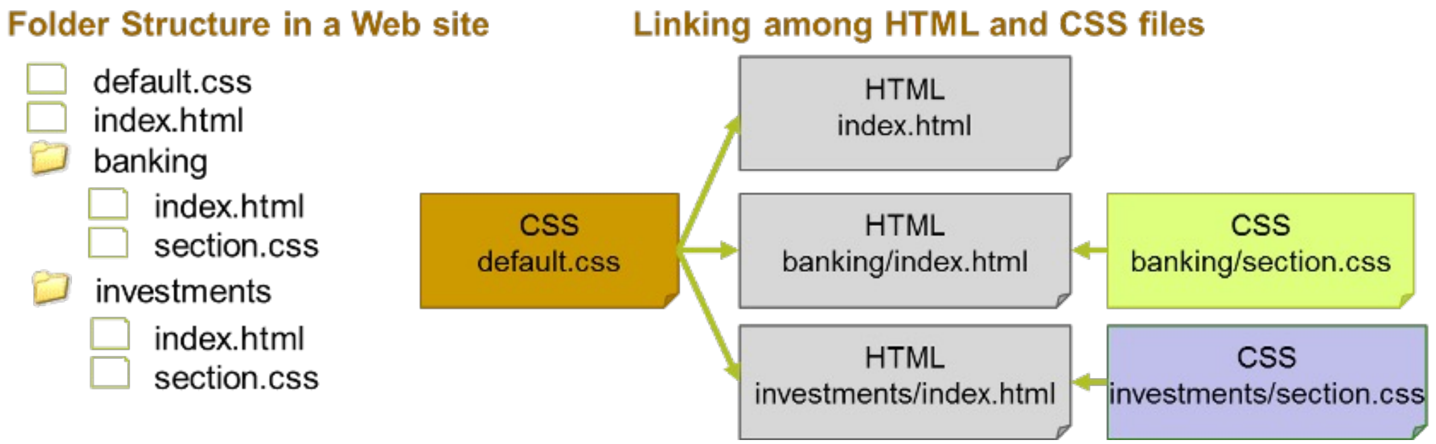
```
*{Color: pink;}
```

Code Example: lec04-05-contextual-selector.html

How to organize CSS styles? (1)

By application functions

- One highest level "*default.css*" - styles used in all pages
- One style sheet "*section.css*" - each for sub-systems grouping styles



How to organize CSS styles? (2)

By content (HTML) type - to maintain consistency in interfaces throughout the Website, we can consider

- setting up style sheets to control forms, tables, lists, etc.
- applying them to pages that have those HTML

When multiple style sheets are used, we need to think about the order of linking into a Web page

Media Type and Media Query (1)

Styles could be used selectively depending on different conditions (e.g., screen, print). Two ways to specify media dependencies for style sheet:

- Specify the target media inside a style sheet with the `@media` or `@import` at-rules
- Specify the target medium with external style sheet

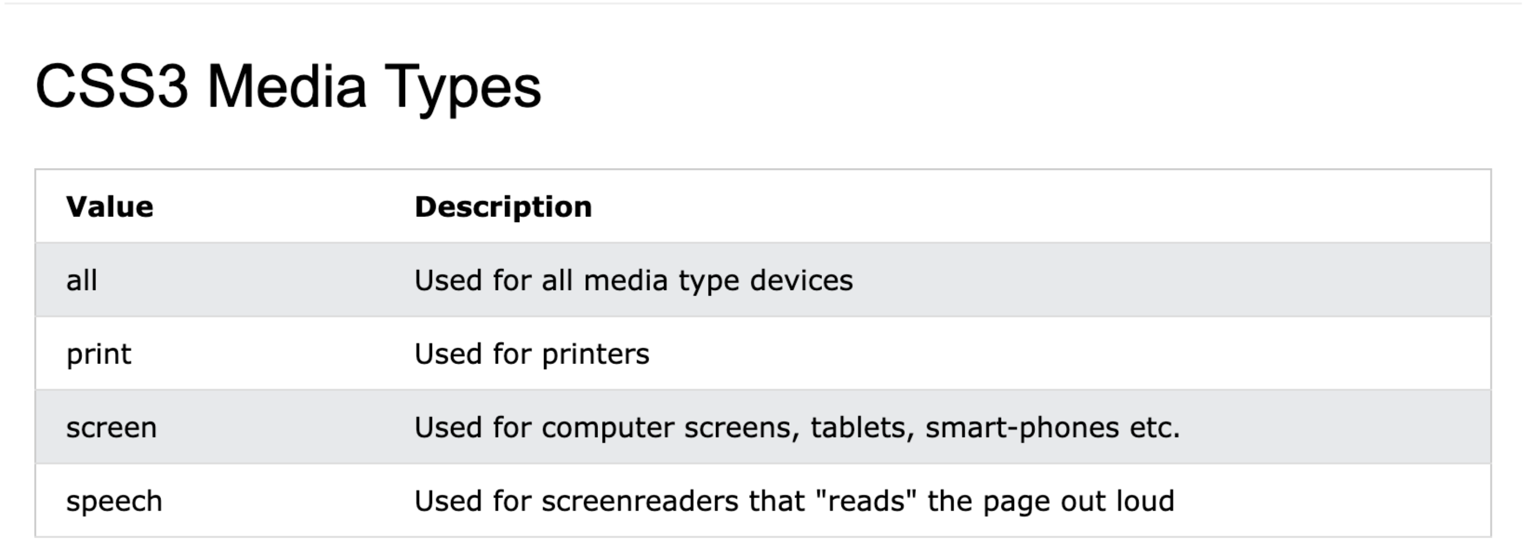
```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>ABC Banking Corporation Limited</title>

        <style type="text/css">
            @import url("screen.css") print;
        </style>

        <!-- <link rel="stylesheet" href="screen.css" type="text/css" media="screen"> -->
    </head>
```

How about removing "print"?

Code Example: lec04-14-media-type.html

Media Type And Media Query (2)

The media type has been further developed into more complicated as well as combined (e.g., and, or, etc.) conditions in CSS3 known as Media Queries



CSS3 Media Types

Value	Description
all	Used for all media type devices
print	Used for printers
screen	Used for computer screens, tablets, smart-phones etc.
speech	Used for screenreaders that "reads" the page out loud

Agenda

CSS Basics

- How to apply style on HTML elements

CSS Selectors

- ID, Class, Group
- Contextual selector
- Advanced selector
- Cascading and inheritance

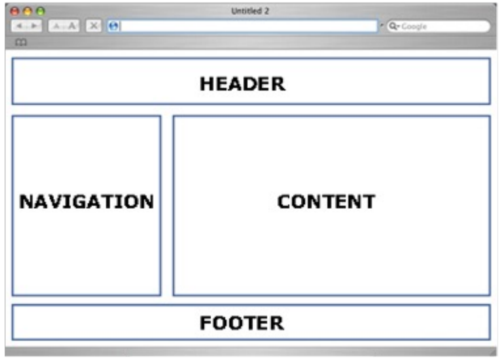
CSS Layout

CSS Layout

Arrangement/positioning of text and graphics.

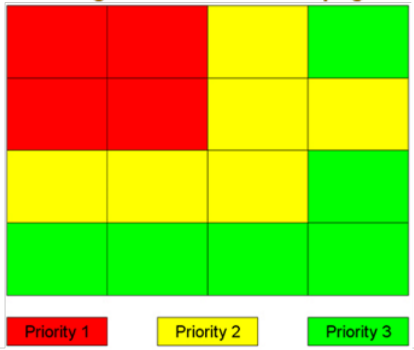
- **Viewing pattern**
  - The box model
  - Layout properties
  - Fixed layout
  - Liquid layout
  - Float properties
- **Responsive Design**

Common Web Page Layout



ADC. (2006). *Web Page Development: Best Practices.*

Viewing Patterns for Homepage



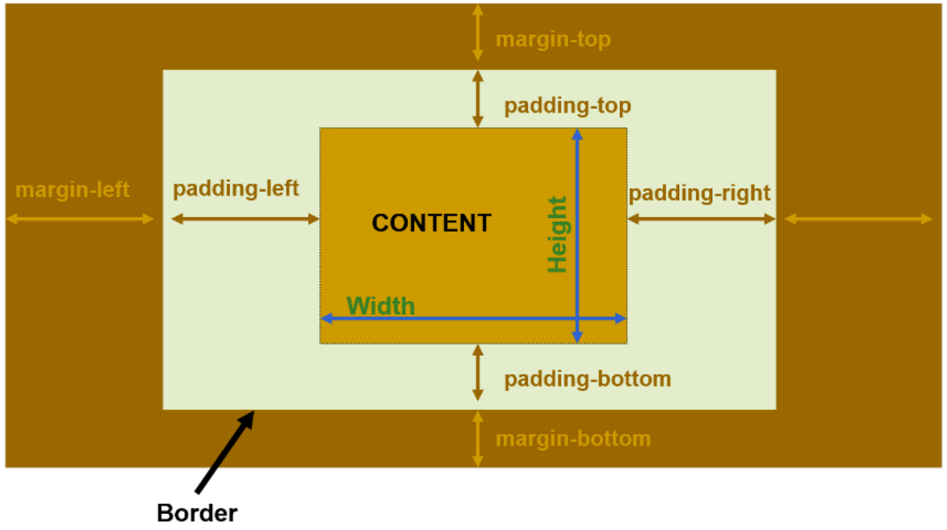
Ruel, L. and Outing, S. (2006). *Viewing Patterns for Homepages.*

The Box Model

What is a box?

- a box can be any HTML tag depending on at which level we are looking

Each tag can be treated as a discrete element box on the screen and controlled by CSS, with the following **properties**:

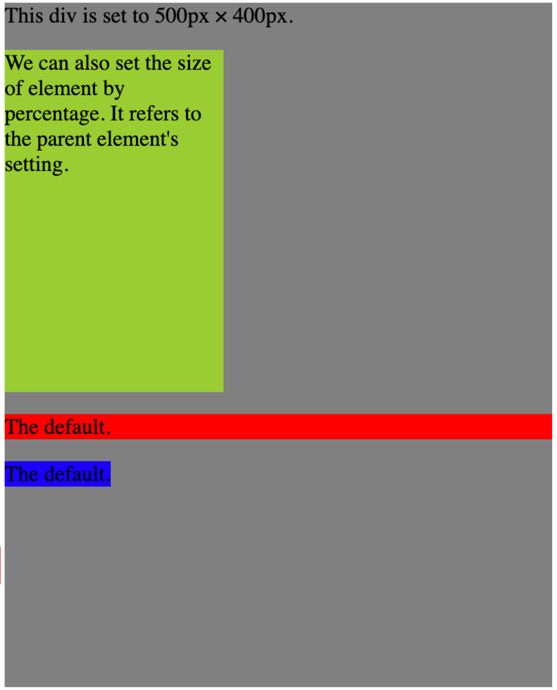


Width & Height

Each element can be specified their size using the **width** and **height** properties

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>CSS Width and Height</title>
    <style>
```

```
    </style>
</head>
<body>

    <!-- <div class="box">
        this is a box
    </div> -->

    <div class="length">
        This div is set to 500px × 400px.
        <p class="percentage">
            We can also set the size of element by percentage. It refers to the parent element's
setting.
        </p>
        <p class="auto">
            The default.
        </p>
        <span class="inline">The default.</span>
        <!-- <span class="inline">This is another inline example.</span> -->
    </div>

</body>
</html>
```



This div is set to 500px × 400px.

We can also set the size of element by percentage. It refers to the parent element's setting.

The default.

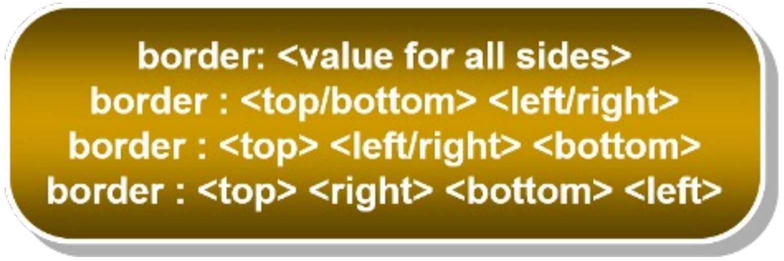
The default.

Code Example: lec04-15-CSS-width-and-height.html

Border

Border can be useful in knowing the exact position of a block when we work with complicate layout

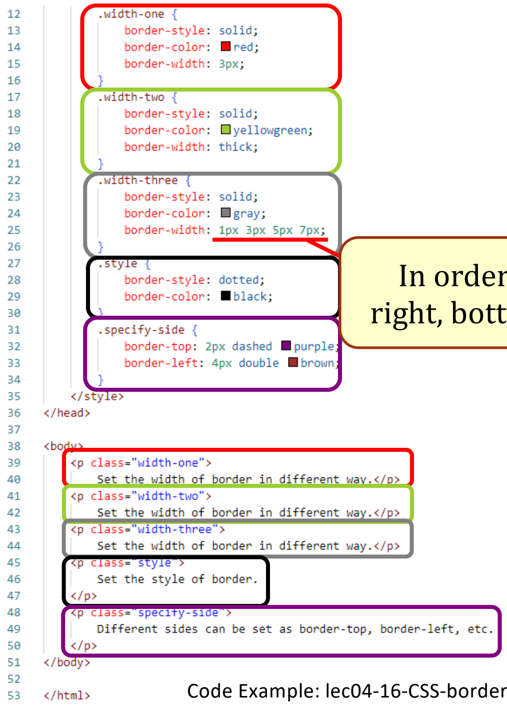
Different ways to specify border width



border: <value for all sides>
border : <top/bottom> <left/right>
border : <top> <left/right> <bottom>
border : <top> <right> <bottom> <left>

Border example

The width, color, and style of the border can be specified using border-width, border-color and border-style properties

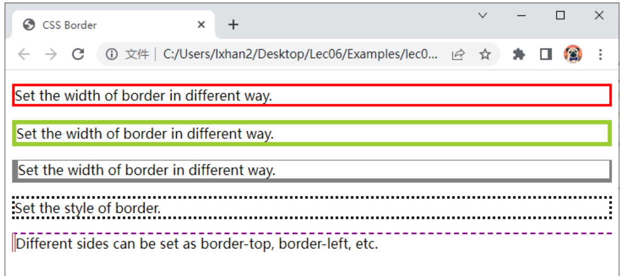


```
    .width-one {
        border-style: solid;
        border-color: red;
        border-width: 3px;
    }
    .width-two {
        border-style: solid;
        border-color: yellowgreen;
        border-width: thick;
    }
    .width-three {
        border-style: solid;
        border-color: gray;
        border-width: 1px 3px 5px 7px;
    }
    .style {
        border-style: dotted;
        border-color: black;
    }
    .specify-side {
        border-top: 2px dashed purple;
        border-left: 4px double brown;
    }
  </style>
</head>

<body>
  <p class="width-one">
      Set the width of border in different way.</p>
  <p class="width-two">
      Set the width of border in different way.</p>
  <p class="width-three">
      Set the width of border in different way.</p>
  <p class="style">
      Set the style of border.
  </p>
  <p class="specify-side">
      Different sides can be set as border-top, border-left, etc.
  </p>
</body>

</html>
```

In order of top, right, bottom & left

Code Example: lec04-16-CSS-border.html

- border-width : | thin | medium | thick | inherit
- border-color : | transparent | inherit
- border-style : dotted | dashed | solid | double | groove | ridge | inset | outset | none | inherit
  border-top, border-right, border-bottom, and border-left



Set the width of border in different way.

Set the width of border in different way.

Set the width of border in different way.

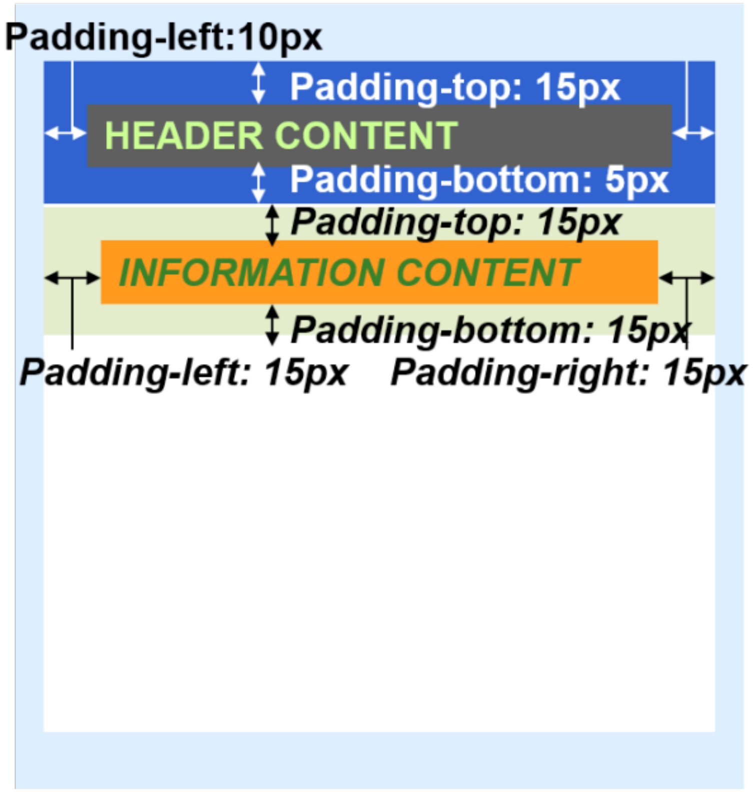
Set the style of border.

Different sides can be set as border-top, border-left, etc.

Question time

How to give all elements a border, so that we can easily check their positions? It should also be removed easily too after debugging

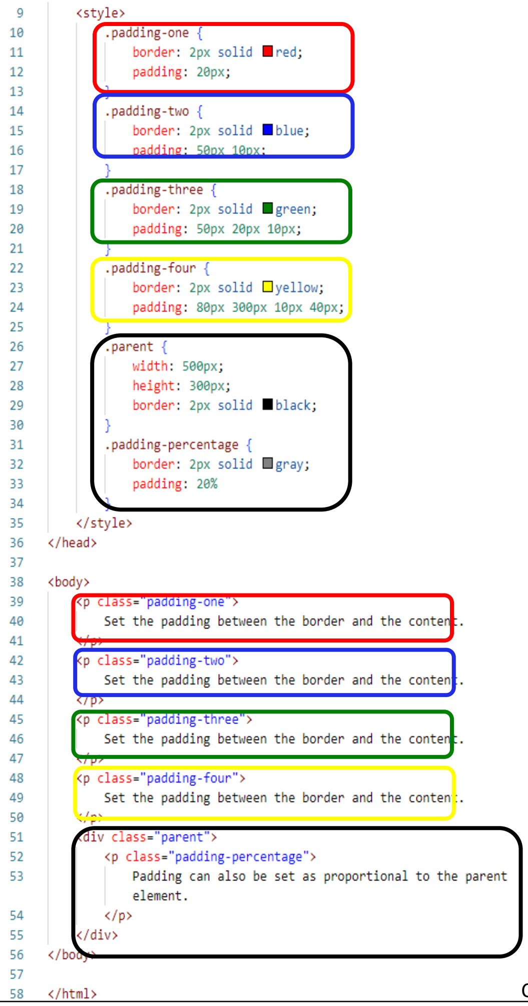
Padding



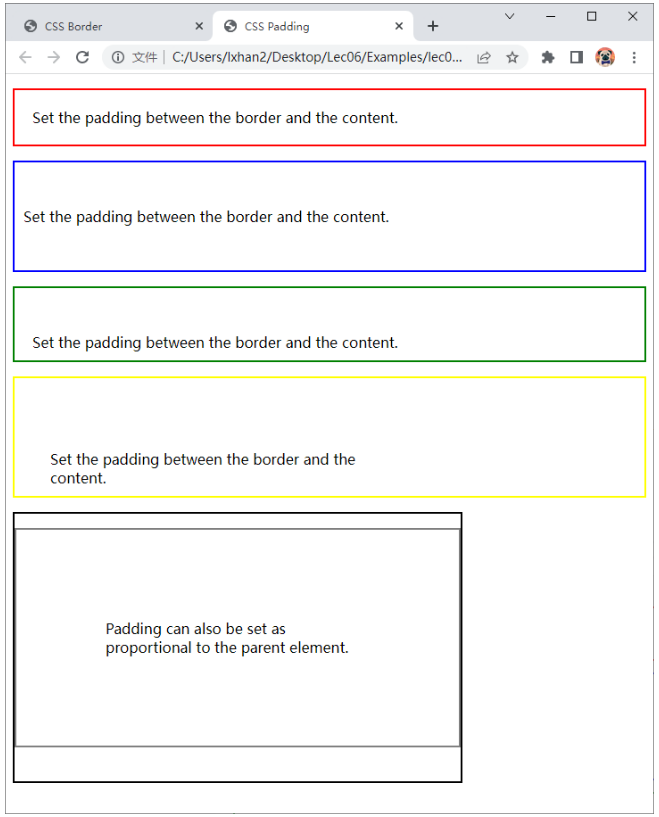
The padding property is used to add space between the border and the content

padding : <value for all sides>
padding : <top/bottom> <left/right>
padding : <top> <left/right> <bottom>
padding : <top> <right> <bottom> <left>

Padding example



```
    <style>
        .padding-one {
            border: 2px solid red;
            padding: 20px;
        }
        .padding-two {
            border: 2px solid blue;
            padding: 50px 10px;
        }
        .padding-three {
            border: 2px solid green;
            padding: 50px 20px 10px;
        }
        .padding-four {
            border: 2px solid yellow;
            padding: 80px 300px 10px 40px;
        }
        .parent {
            width: 500px;
            height: 300px;
            border: 2px solid black;
        }
        .padding-percentage {
            border: 2px solid gray;
            padding: 20%
        }
    </style>
</head>

<body>
    <p class="padding-one">
        Set the padding between the border and the content.
    </p>
    <p class="padding-two">
        Set the padding between the border and the content.
    </p>
    <p class="padding-three">
        Set the padding between the border and the content.
    </p>
    <p class="padding-four">
        Set the padding between the border and the content.
    </p>
    <div class="parent">
        <p class="padding-percentage">
            Padding can also be set as proportional to the parent
            element.
        </p>
    </div>
</body>

</html>
```



Set the padding between the border and the content.

Set the padding between the border and the content.

Set the padding between the border and the content.

Set the padding between the border and the
content.

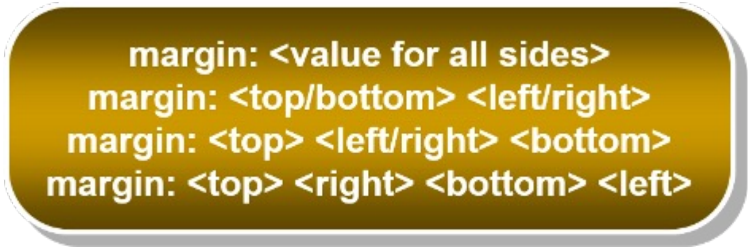
Padding can also be set as
proportional to the parent element.

Code Example: lec04-17-CSS-padding.html

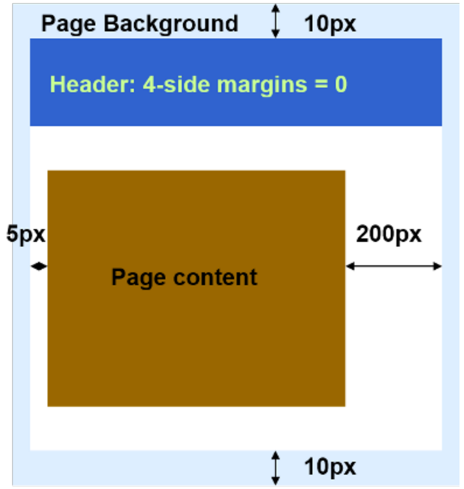
Margins

Value can be set in different ways:

- Length - in an **absolute** unit, such as px
- Percentage - with reference to the parent element's **width**
- Auto - leave to the browser's calculation, usually used for centering, e.g. `{margin: auto;}`
  - The element will then take up the specified width, and the remaining space will be split equally between the left and right margins



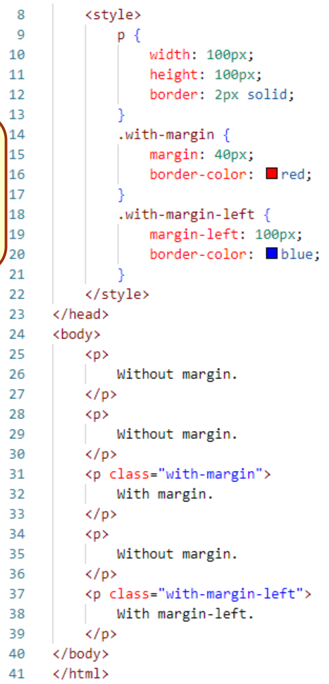
margin: <value for all sides>
margin: <top/bottom> <left/right>
margin: <top> <left/right> <bottom>
margin: <top> <right> <bottom> <left>



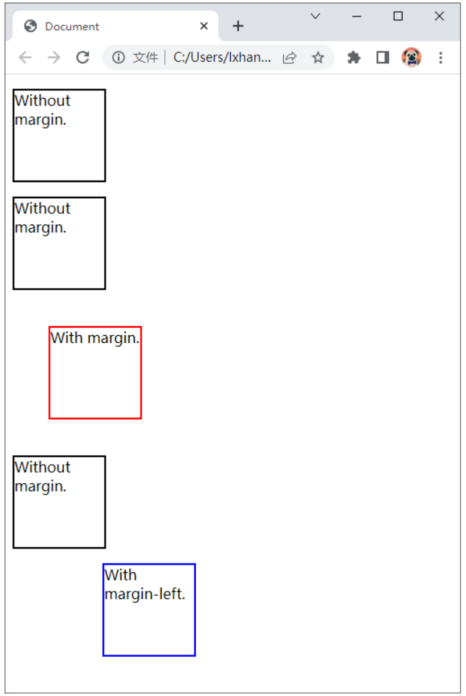
Margins Example

Use margin property to set the space **between** that element and other elements in the same container

The property can be a single <margin> or margins of 4 sides: <margin-left>, <margin-top>, etc.



```
    <style>
        p {
            width: 100px;
            height: 100px;
            border: 2px solid;
        }
        .with-margin {
            margin: 40px;
            border-color: red;
        }
        .with-margin-left {
            margin-left: 100px;
            border-color: blue;
        }
    </style>
</head>
<body>
    <p>
        Without margin.
    </p>
    <p>
        Without margin.
    </p>
    <p class="with-margin">
        With margin.
    </p>
    <p>
        Without margin.
    </p>
    <p class="with-margin-left">
        With margin-left.
    </p>
</body>
</html>
```



Without margin.

Without margin.

With margin.

Without margin.

With margin-left.

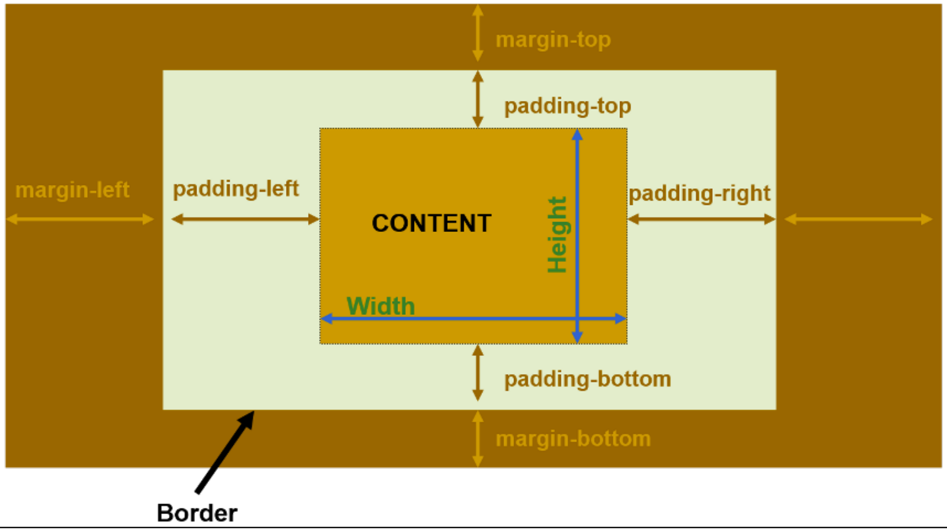
Code Example: lec04-18-CSS-margin.html

A Quick Summary of Box Model

A box can be any HTML tag depending on at which level we are looking

Set border: color, width, style

Differences in width/height, padding, and margin



Lecture summary

CSS style can be applied via inline style, embedded style, external link (`<link>` and `@import`)

CSS selector are rules to select target elements and apply style. In addition to tag, class, id, there are contextual selectors depending on the HTML structure and advanced selector for more specific styling

If different styles are specified for HTML elements, the styles will cascade into new styles with the following priority

When organize multiple CSS styles, we need to take the application function, consistency, display platform, etc, into account

All the HTML elements or a group of elements can be seen as a box with margin, padding, border, content, width and height