CS2204 Fundamentals of Internet Applications Development

Lecture 6 CSS – Part 3

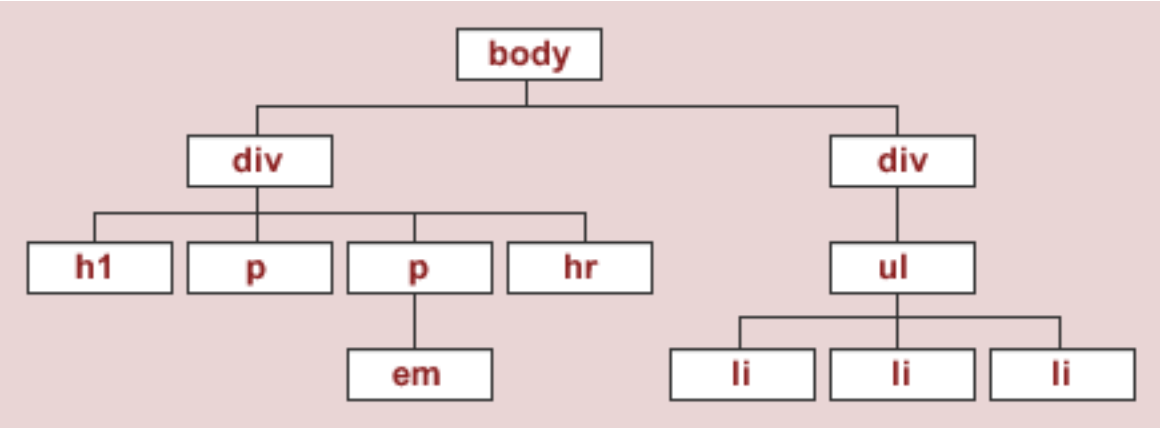
Computer Science, City University of Hong Kong

Semester A 2023-24

Review: CSS Advanced Selector (1)

1. Which elements are selected with the following selector?

- div > p {…}
- h1 + p
- h1 ~ p



Review: CSS Advanced Selector (2)

2. What's the difference between pseudo class and pseudo element? Is first-letter a pseudo class or a pseudo element?

3. Among ID, class, attribute, pseudo-class, and universal selector, which one has the highest priority (specificity)? Which one has no effect on specificity?

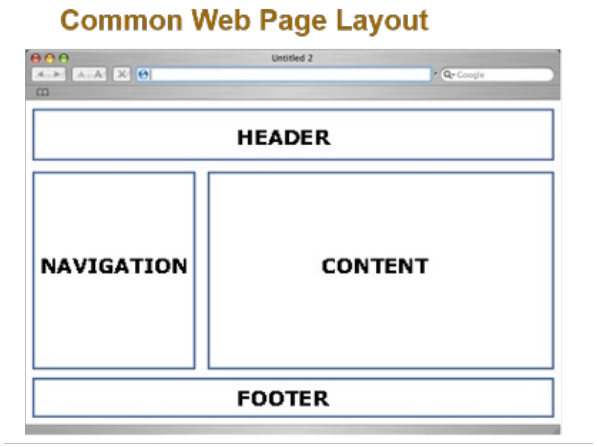
Outline

- CSS Layout
  - The box model
- Element display
- Float property

CSS Layout

Arrangement/positioning of text and graphics.

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

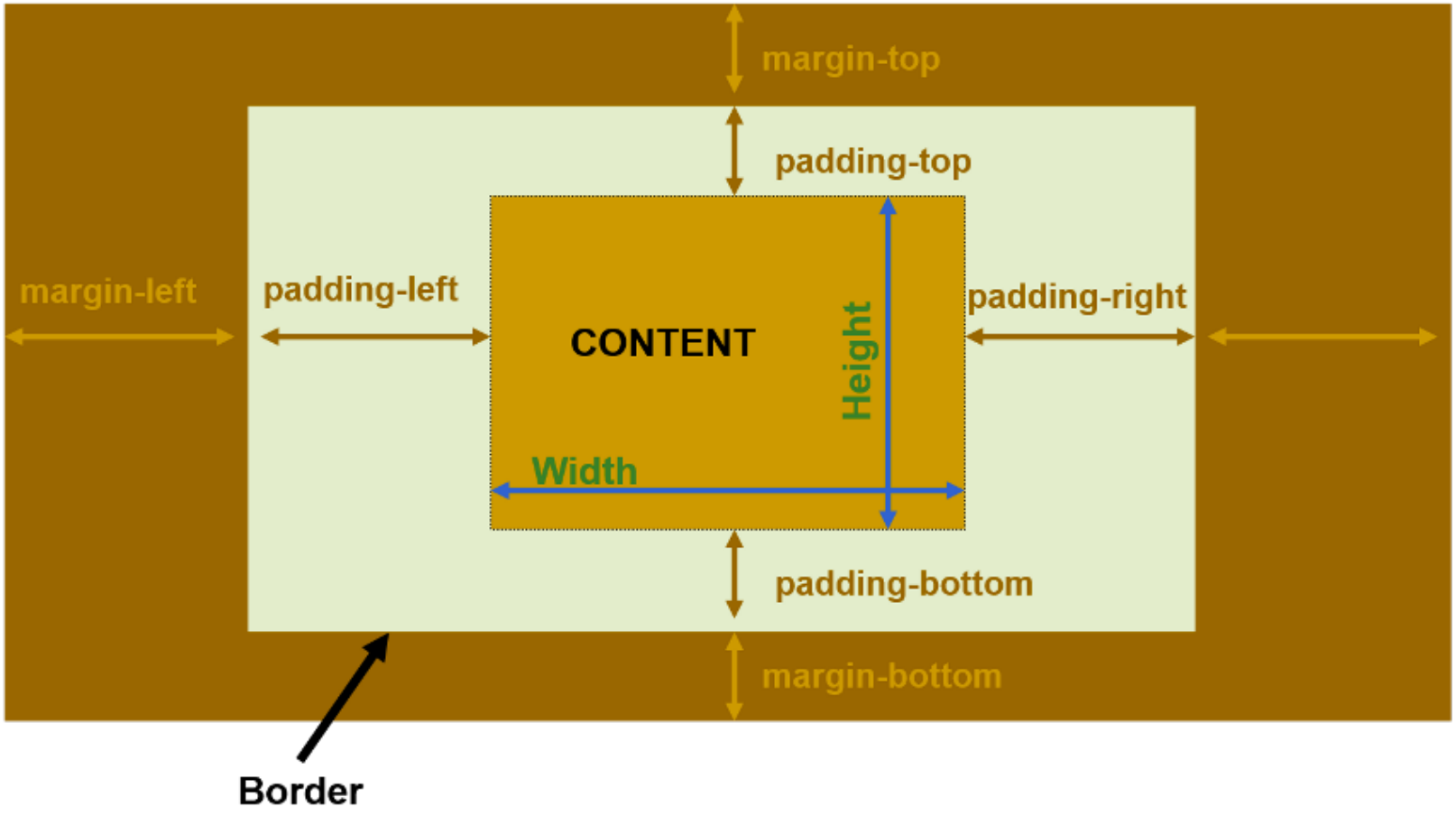


The Box Model

- What is a block? Recalling in a Web page, a HTML element can contain other element(s). Therefore, a block 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**:
  - Content: at the center of the box; includes all **descendant** tags
  - Width & Height: the dimensions of the content area
  - Border: a line that surrounds the element; invisible unless its color, width, and style are set
  - Padding: the space between the border and the content of the element; background colors and images will also fill this space
  - Margin: the space between the border of the element and other elements in the window or container

Example

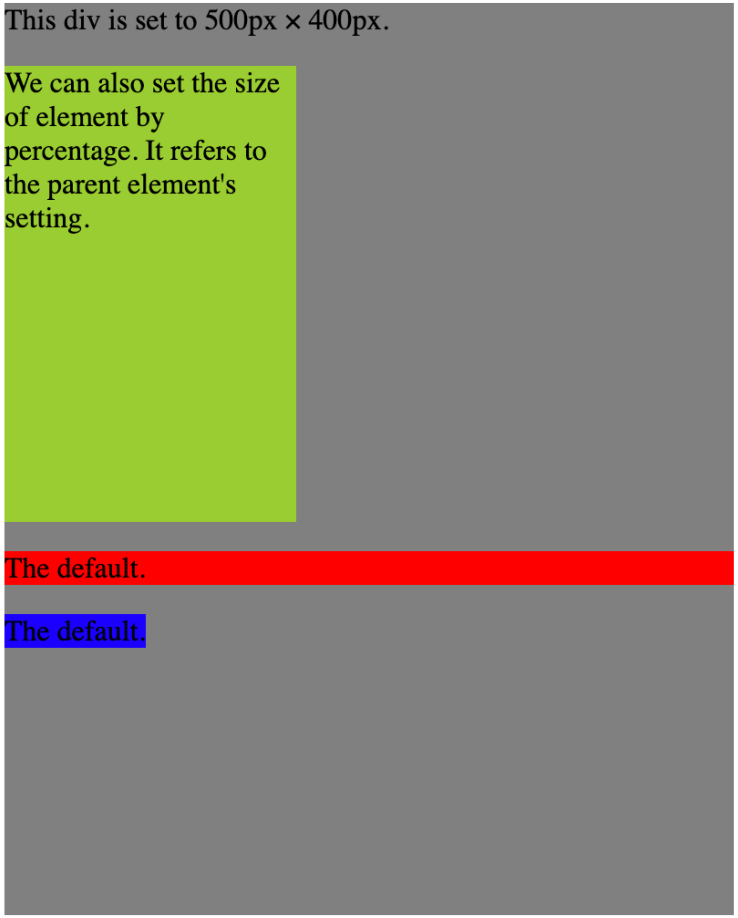
The Box Model



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>
```



Code Example: lec06-01-CSS-width-and-height.html

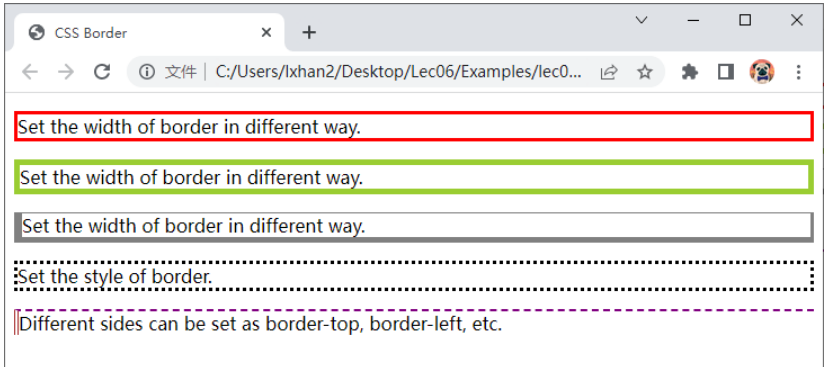
Border

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

```
12      .width-one {
13          border-style: solid;
14          border-color: red;
15          border-width: 3px;
16      }
17      .width-two {
18          border-style: solid;
19          border-color: yellowgreen;
20          border-width: thick;
21      }
22      .width-three {
23          border-style: solid;
24          border-color: gray;
25          border-width: 1px 3px 5px 7px;
26      }
27      .style {
28          border-style: dotted;
29          border-color: black;
30      }
31      .specify-side {
32          border-top: 2px dashed purple;
33          border-left: 4px double brown;
34      }
35  </style>
36  </head>
37
38  <body>
39      <p class="width-one">
40          Set the width of border in different way.</p>
41      <p class="width-two">
42          Set the width of border in different way.</p>
43      <p class="width-three">
44          Set the width of border in different way.</p>
45      <p class="style">
46          Set the style of border.
47      </p>
48      <p class="specify-side">
49          Different sides can be set as border-top, border-left, etc.
50      </p>
51  </body>
52
53  </html>
```

In order of top, right, bottom & left

- border-width : | thin | medium | thick | xxpx
- border-color : | transparent | a color
- border-style : dotted | dashed | solid | | none
- border-top, border-right, border-bottom, and border-left



Code Example: lec06-02-CSS-border.html

Border (2)

- For each property, we can set its value for all sides together, or separately as follows
  - property: <one value for all sides>; (one value)
  - property: <top/bottom> <left/right>; (two values)
  - property: <top> <left/right> <bottom>; (three values)
  - property: <top> <right> <bottom> <left>; (four values)
- Compound writing for different properties
  - `border: width style color;` (for all four borders)
  - `border-top: width style color;` (for one border)
  - others: `border-left | border-right | border-bottom`

```
div {
  border: 1px solid black;
  border-top: 1px solid red;
}
```

Border (3)

- Border width can increase the size of the box
- Border collapse

```css
div {
  width: 200px;
  height: 100px;
  border: 10px solid black;
  background-color: blue;
}
```

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
    <style>
        .myTable {
            width: 200px;
            text-align: center;
        }
        .myTable,
        td {
            border: 1px solid black;
            border-collapse: collapse;
        }
    </style>
</head>
<body>

    <table class="myTable" cellspacing="0" align="center">
        <tr>
            <td>1</td>
            <td>2</td>
            <td>3</td>
        </tr>
        <tr>
            <td>4</td>
            <td>5</td>
            <td>6</td>
        </tr>
        <tr>
            <td>7</td>
            <td>8</td>
            <td>9</td>
        </tr>
    </table>

</body>
</html>
```

1	2	3
4	5	6
7	8	9

- `border-collapse: collapse | separate;`
- it sets whether table borders should collapse into a single border or be separated as in standard HTML

Code Example: lec06-03-CSS-border-collapse.html

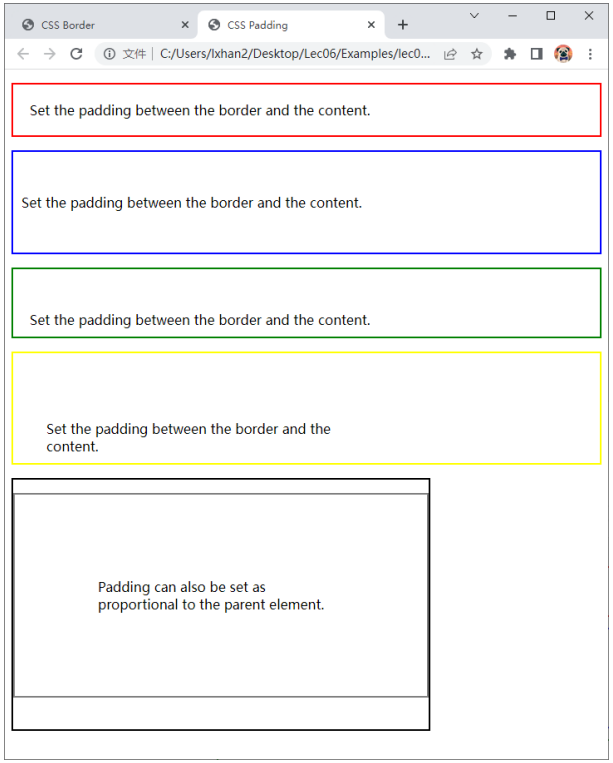
Critical Thinking

- 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

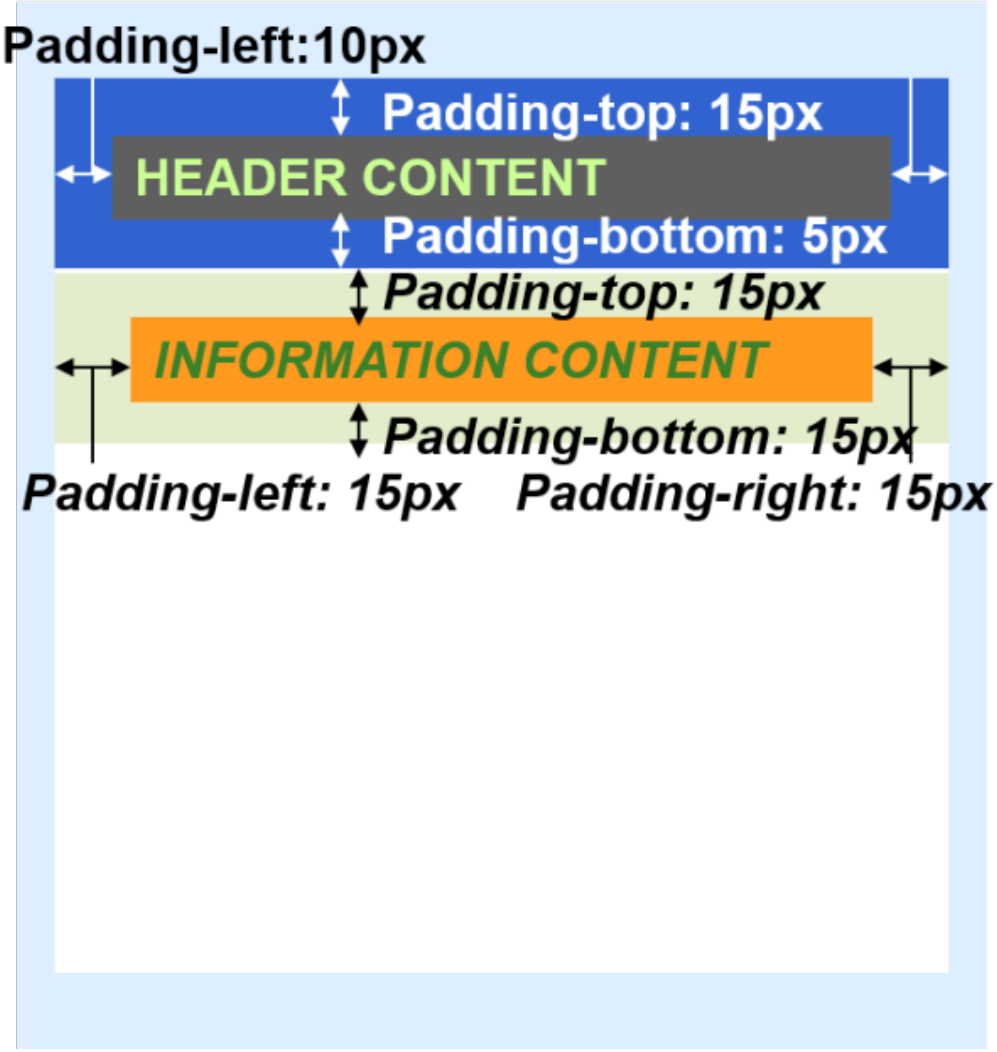
```
9     <style>
10        .padding-one {
11            border: 2px solid red;
12            padding: 20px;
13        }
14        .padding-two {
15            border: 2px solid blue;
16            padding: 50px 10px;
17        }
18        .padding-three {
19            border: 2px solid green;
20            padding: 50px 20px 10px;
21        }
22        .padding-four {
23            border: 2px solid yellow;
24            padding: 80px 300px 10px 40px;
25        }
26        .parent {
27            width: 500px;
28            height: 300px;
29            border: 2px solid black;
30        }
31        .padding-percentage {
32            border: 2px solid gray;
33            padding: 20%
34        }
35    </style>
36 </head>
37
38 <body>
39    <p class="padding-one">
40        Set the padding between the border and the content.
41    </p>
42    <p class="padding-two">
43        Set the padding between the border and the content.
44    </p>
45    <p class="padding-three">
46        Set the padding between the border and the content.
47    </p>
48    <p class="padding-four">
49        Set the padding between the border and the content.
50    </p>
51    <div class="parent">
52        <p class="padding-percentage">
53            Padding can also be set as proportional to the parent
              element.
54        </p>
55    </div>
56 </body>
57
58 </html>
```



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

Code Example: lec06-04-CSS-padding.html

Padding (2)


Padding-left:10px
Padding-top: 15px
HEADER CONTENT
Padding-bottom: 5px
Padding-top: 15px
INFORMATION CONTENT
Padding-bottom: 15px
Padding-left: 15px
Padding-right: 15px

Padding (3)

- Padding could change the size of the box
- But if we do not specify the width/height of a box, padding will not change the width/height of the box
  - E.g., the width of the following

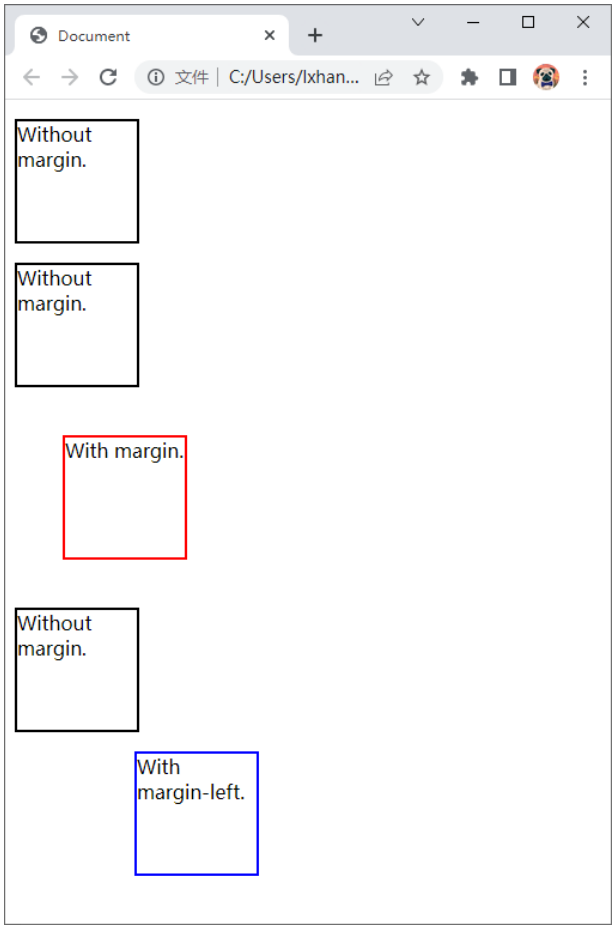
```
div {
  height: 100px;
  background-color: blue;
  padding: 10px;
}
```

Margins

- 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.

```
8     <style>
9         p {
10            width: 100px;
11            height: 100px;
12            border: 2px solid;
13        }
14        .with-margin {
15            margin: 40px;
16            border-color: red;
17        }
18        .with-margin-left {
19            margin-left: 100px;
20            border-color: blue;
21        }
22    </style>
23</head>
24<body>
25    <p>
26        Without margin.
27    </p>
28    <p>
29        Without margin.
30    </p>
31    <p class="with-margin">
32        With margin.
33    </p>
34    <p>
35        Without margin.
36    </p>
37    <p class="with-margin-left">
38        With margin-left.
39    </p>
40</body>
41</html>
```



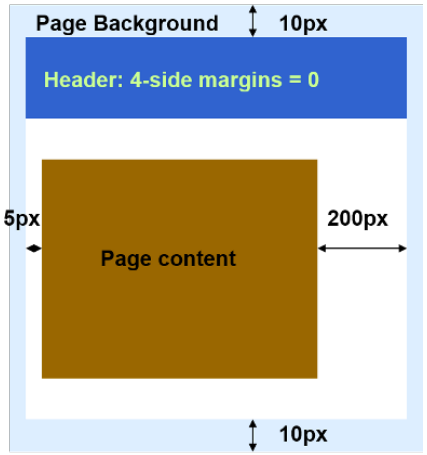
Code Example: lec06-05-CSS-margin.html

Margins (2)

- For the value, it can be set in different ways:
  - Length - in an **absolute** unit, such as px
  - Percentage - with reference to the parent element's **width**
    - What if the parent's width is also in percentage?
    - The value will be calculated by going up one level to the grand parent, or until to the ultimate ancestor (Window) to get a value setting
  - 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

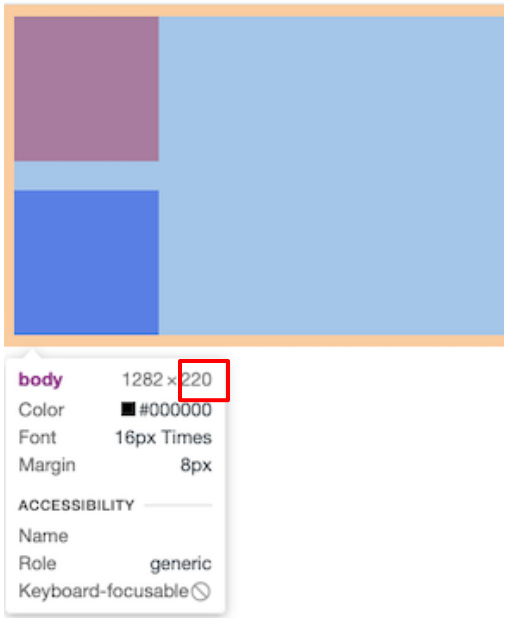
margin: <value for all sides>
margin: <top/bottom> <left/right>
margin: <top> <left/right> <bottom>
margin: <top> <right> <bottom> <left>



Margins (3)

- Emerged margins of two boxes
  - Two adjacent boxes

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
    <style>
        .one {
            width: 100px;
            height: 100px;
            background-color: red;
            margin-bottom: 20px;

        }
        .two {
            width: 100px;
            height: 100px;
            background-color: blue;
            margin-top: 10px;
        }
    </style>
</head>
<body>
    <div class="one"></div>
    <div class="two"></div>
</body>
</html>
```



Code Example: lec06-06-CSS-margin-merging.html

Margins (4)

- Emerged margins of two boxes
  - Two nested boxes

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
    <style>
        .one {
            width: 100px;
            height: 100px;
            background-color: red;
            margin-bottom: 20px;

        }
        .two {
            width: 100px;
            height: 100px;
            background-color: blue;
            margin-top: 10px;
        }
        .three {
            width: 100px;
            height: 100px;
            background-color: green;
            margin-top: 20px;
        }
        .four {
            width: 20px;
            height: 20px;
            background-color: purple;
            margin-top: 20px;
        }
    </style>
</head>
<body>
    <div class="one"></div>
    <div class="two"></div>

    <div class="three">
        <div class="four"></div>
    </div>
</body>
</html>
```

Code Example: lec06-06-CSS-margin-merging.html

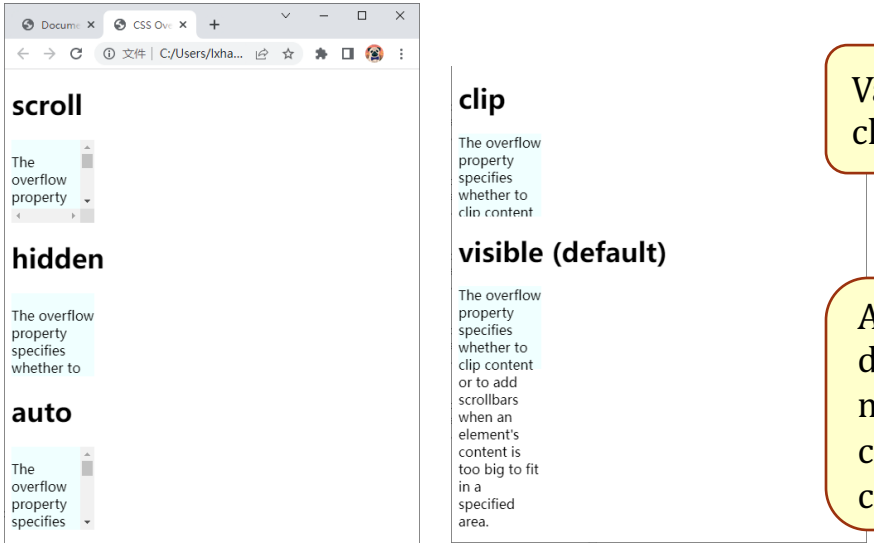
Margins (5)

- Many elements have their default margin and padding values, which are usually removed by CSS initially
  - Usually, it is the first CSS rule for a webpage

```
* {
  margin: 0px;
  padding: 0px;
}
```

The Overflow Property

- The overflow property can be used to specify the behavior of the browser when the size of the element is larger than the container




```
8     <style>
9         div {
10            height: 100px;
11            width: 100px;
12            background-color: azure;
13        }
14        .scroll {
15            overflow: scroll;
16        }
17        .hidden {
18            overflow: hidden;
19        }
20        .auto {
21            overflow: auto;
22        }
23        .clip {
24            overflow: clip;
25        }
26        .visible {
27            overflow: visible;
28        }
29    </style>
30</head>
31<body>
32    <h1>scroll</h1>
33    <div class="scroll">
34        <p>The overflow property specifies whether to clip content or to add scrollbars when an element's content is too big to fit in a specified area.</p>
35    </div>
36    <h1>hidden</h1>
37    <div class="hidden">
38        <p>The overflow property specifies whether to clip content or to add scrollbars when an element's content is too big to fit in a specified area.</p>
39    </div>
40    <h1>auto</h1>
41    <div class="auto">
42        <p>The overflow property specifies whether to clip content or to add scrollbars when an element's content is too big to fit in a specified area.</p>
43    </div>
44    <h1>clip</h1>
45    <div class="clip">
46        <p>The overflow property specifies whether to clip content or to add scrollbars when an element's content is too big to fit in a specified area.</p>
47    </div>
48    <h1>visible (default)</h1>
49    <div class="visible">
50        <p>The overflow property specifies whether to clip content or to add scrollbars when an element's content is too big to fit in a specified area.</p>
51    </div>
52</body>
53</html>
```

Values: scroll | hidden| auto | clip | visible

Auto – allows the browser to decide whether scroll bars need to be displayed and the content is shown inside the container

Code Example: lec06-07-CSS-overflow.html

Types of Elements

- **Block-level** element
  - always starts on a new line and always takes up the full width available, e.g., `<div>`, `<p>`, `<ul>`, `<h1~6>`, etc.
  - its width, height, padding and margin can be adjusted
  - it usually can contain other elements
    - special case: text-related elements usually **cannot** contain other **block-level** elements, e.g., <p> cannot contain other block-level elements, such as

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
    <style>
        .one {
            height: 100px;
            /* width: 100px; */
            background-color: red;
        }
    </style>
</head>
<body>
    <div class="one"></div>

    <p>this is a <div></div> box</p>

</body>
</html>
```

Code Example: lec06-08-CSS-type-block.html

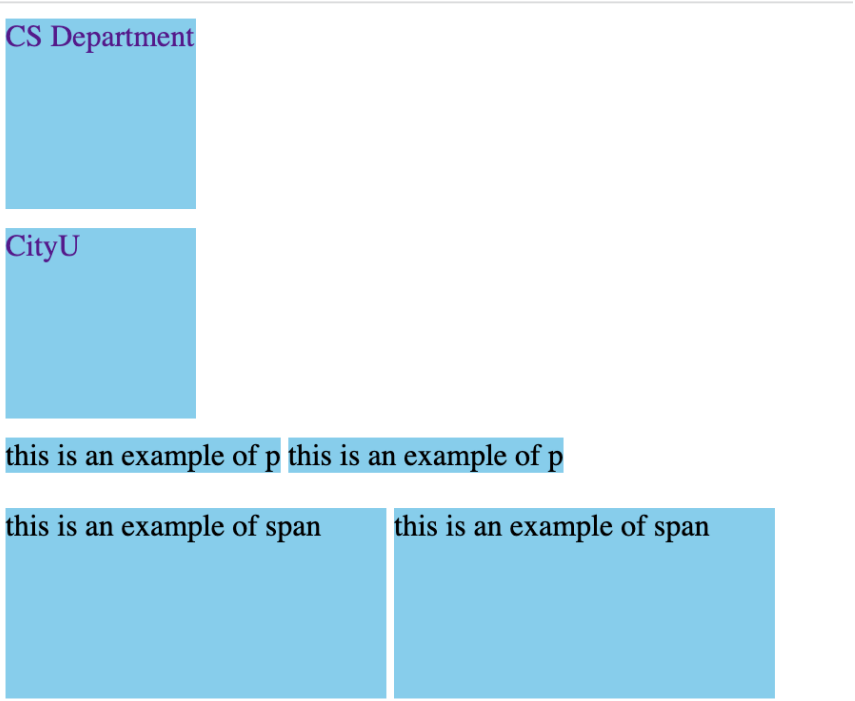
Types of Elements

- **Inline-level** element
  - does **not** start on a new line and only takes up as much width as necessary, e.g., `<a>`, `<span>`, etc.
  - set width and height is invalid
  - multiple inline-level elements can be in one line
  - it can contain only texts or other inline-level elements (no block-level elements)
    - special case: <a> can contain another block-level element
- **Inline-block** element
  - E.g., `<img>`, `<input>`, `<td>`, etc.
  - multiple such elements can be in one line, and their default width is the same as their enclosed content (similar to inline-level elements)
  - but their width, height, padding and margin can be adjusted (similar to block-level elements)

Types of Elements (2)

- Conversion of different types
  - `display: block | inline | inline-block;`

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
    <style>
        a {
            text-decoration: none;
        }
        a {
            width: 100px;
            height: 100px;
            background-color: skyblue;
            display: block;
            margin-bottom: 10px;
        }
        p {
            width: 200px;
            height: 100px;
            background-color: skyblue;
            display: inline;
        }
        span {
            width: 200px;
            height: 100px;
            background-color: skyblue;
            display: inline-block;
        }
    </style>
</head>
<body>
    <a href="#">CS Department</a>
    <a href="#">CityU</a>

    <p>this is an example of p</p>
    <p>this is an example of p</p>

    <br><br>

    <span>this is an example of span</span>
    <span>this is an example of span</span>

</body>
</html>
```



Code Example: lec06-09-CSS-type-conversion.html

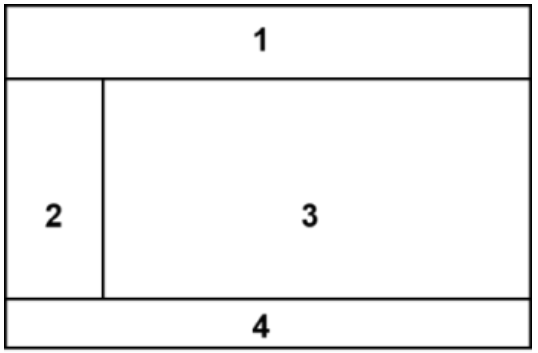
Types of Elements (3)

- Design the following webpage
  - The whole area of each green block is clickable
    - Each block is a pair of `<a>` tag
  - Two blocks in the middle of the page
    - Each block: width (100px) and height (50px)
    - Background: yellowgreen
    - Text: in the center *horizontally* and *vertically*
    - **Remove** underline

Code Example: lec06-10-CSS-display.html

Structuring Your Page For Layout

- Consider a typical layout with 4 sections:
  - Header (<header>)
  - Content (<section>)
  - Navigation (<nav>)
  - Footer (<footer>)
- Three typical layouts
  - normal flow
  - floating
  - positioning



Normal Flow

- The default way to place elements
  - block-level elements
    - 1) one element occupies **one line**
    - 2) placed from **top to down**
  - inline or inline-block elements
    - 1) from **left to right** in one line
    - 2) if there is no enough space, it is placed in the next line

Floating

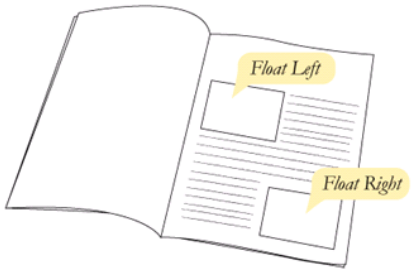
Definition in HTML standard : "A float is a box that is shifted to the left or right on the current line

```
<!DOCTYPE html>
<html>
<head>
<style>
img {
  float: right;
}
</style>
</head>
<body>
<h1>The float Property</h1>

<p>In this example, the image will float to the
right in the text, and the text in the paragraph
will wrap around the image.</p>

<p><img src="pineapple.jpg" alt="Pineapple"
style="width:170px;height:170px;margin-
left:15px;">
Lorem ipsum dolor sit amet, consectetur
adipiscing elit. </p>

</body>
</html>
```

The float Property

In this example, the image will float to the right in the text, and the text in the paragraph will wrap around the image.

Lorem ipsum dolor sit amet, consectetur adipiscing elit.



https://css-tricks.com/all-about-floats/

Set multi-column designs: {float: value;}

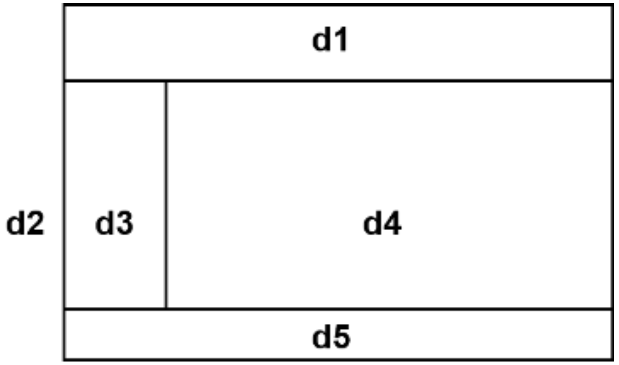
- The value here is the float property (left, right or none)
- The box is taken out of the normal flow

Float – 2 Columns Design

```
<div id="d1"></div>
<div id="d2">
   <div id="d3"></div>
   <div id="d4"></div>
</div>
<div id="d5"></div>
```

Float **left** for menu

- #d3 {float: left;}
- #d4 {float: left;}



Float **right** for menu

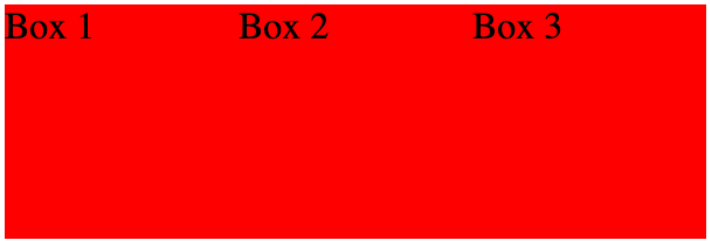
- #d3 {float: right;}
- do we need to float #d4?

Code Example: lec06-11-CSS-float.html

Floating

- Place three div boxes in the same line

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
    <style>
        div {
            width: 100px;
            height: 100px;
            background-color: red;
            /* display: inline-block; */
            float: left;
        }
    </style>
</head>
<body>
    <div class="one">Box 1</div>
    <div class="two">Box 2</div>
    <div class="three">Box 3</div>

</body>
</html>
```

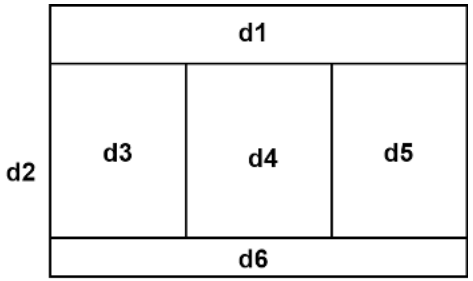


Code Example: lec06-12-CSS-float-three-box.html

Float – 3 columns design

- Float left:
  - #d3, #d4, #d5 {float: left;}

```
<div id="d1"></div>
<div id="d2">
   <div id="d3"></div>
   <div id="d4"></div>
   <div id="d5"></div>
</div>
<div id="d6">
</div>
```



Code Example: lec06-13-CSS-float2.html

Summary of Float Properties

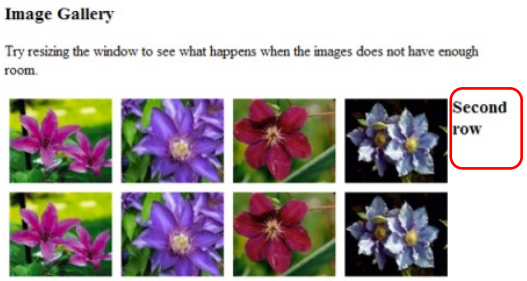
- For each floating element
  - it will leave the **normal** flow
  - it does not occupy the **original position** in the normal flow
  - its width and height can be set (inline-block property)
- For **multiple** floating elements
  - they will be displayed in the same line and aligned on top
  - if the parent box cannot hold all boxes, they will occupy multiple lines
- Element placement
  - Vertical direction: normal flow
  - Horizontal direction: floating

Clear

- To stop floating sequence at some point, we can clear the float property using:

  {clear: left | right | both; }

- **No** element should appear on its left, right or both sides



Code Example: lec06-14-CSS-clear.html

- One common use of clear is to make sure an element starts on a new line, e.g., a footer to appear always in new row regardless how other blocks are floating (second diagram below)

Source: https://css-tricks.com/all-about-floats/

Clear

- Another common use of clear property: if a parent box does not have height specified, all its child boxes are floating

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
    <style>
        .box {
            width: 300px;
            background-color: red;
            border: 1px solid black;
            margin: 0 auto;
        }
        .one {
            width: 100px;
            height: 100px;
            background-color: yellow;
            float: left;
        }
        .two {
            width: 100px;
            height: 100px;
            background-color: green;
            float: left;
        }
        .three {
            clear: both;
        }
    </style>
</head>
<body>
    <div class="box">
        <div class="one">one</div>
        <div class="two">two</div>
        <div class="three"></div>
    </div>
</body>
</html>
```

Code Example: lec06-15-CSS-clear2.html

calc()

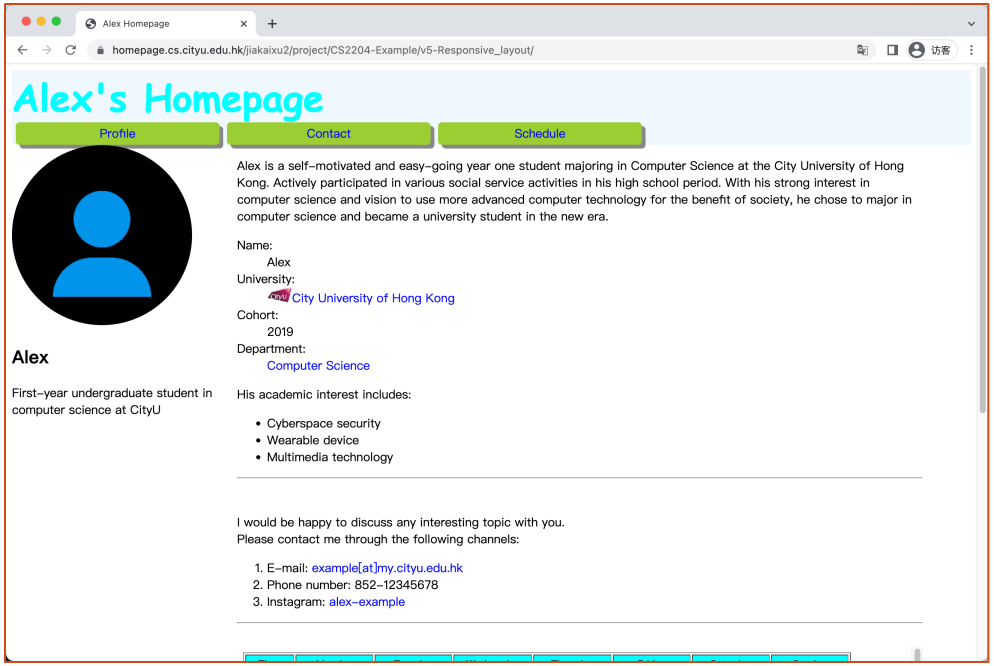
- it is function to be used as the property value
  - for example, box-1 contains box-2, and we want to set the width of box-2 to be always less than that of box-1 by 100px
  - `width: calc(100% - 100px);`

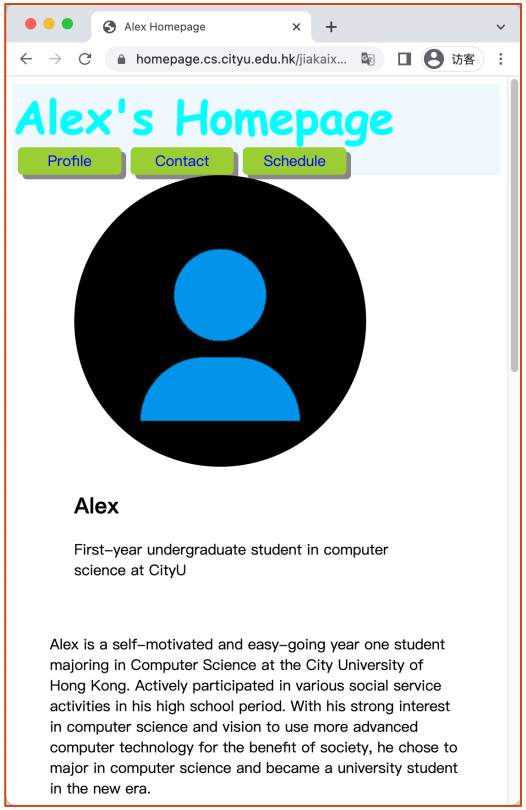
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
    <style>
        .one {
            width: 80%;
            height: 100px;
            background-color: skyblue;
            margin: 0 auto;
        }
        .two {
            width: calc(100% - 100px);
            height: 50px;
            background-color: red;
        }
    </style>
</head>
<body>
    <div class="one">
        <div class="two"></div>
    </div>
</body>
</html>
```

Code Example: lec06-16-CSS-calc.html

Personal Homepage (Version-05)

- **Task**: Detect orientation using media query and add responsive layout for *landscape* and *portrait* mode





Personal Homepage (Version-06)

- **Task 1**: Create "*basicLayout.css*" and fix the position of header and footer at the top and bottom of the page.
  - Header and footer will not move when scrolling the page
- **Task 2**: Create "*basic.css*" and import "*basicLayout.css*" and "*basicStyle.css*". Then link "*basic.css*" directly to the main html file instead of linking other two .css files respectively