HOA SEN UNIVERSITY

www.hoasen.edu.vn

CSS

Phạm Thị Kim Ngôn

ngon.phamthikim@hoasen.edu.vn

Content

- Introduction to CSS
- CSS Fundamentals
- Basic Anatomy of a Style Sheet
- Summary of the Selector syntax
- Calculating a selector's specificity
- Css in action
- Box Model
- Layout

CSS ?

- Cascading Style Sheets
- CSS is a style sheet language used to determine the formatting of an HTML document.
- Before we had CSS (and before it was widely adopted) all of this formatting information was embedded directly in the document- either in the form of attributes like width or bgcolor (background color) or in the form of purely presentational tags like font.
- Combined with the abuse of the table tag to create complicated layouts, the landscape for layout and design on the web was an unmanageable mess.
- CSS fixed all that (kind of.)

CSS ?

- Using separate style sheets for an entire site, leveraging semantic markup and identifiers like ids (for unique page elements) and classes (for multiple, like elements) a developer can apply styles across a whole site while updating a single (cacheable) file.

What It Looked Life Before

```
<p align="center">
        <font face="Papyrus"><img border="0" src=“fancy-header.png” width="207" height="279"></font>
</p>
<p align="center">
        <font face="Papyrus“> Welcome to The Fancy lad Site!</font>
</p>
<p align="center">
        <font face="Papyrus">This web-page is the semi-official home of Fancylads on the World Wide Web.</font>

</p>
```

Not So Bad? Try This.

```
<table width="158" border="0" align="center" cellpadding="0" cellspacing="0">
  <tr bgcolor="#006699">
    <td valign="top" bgcolor="#000066"><div align="center"> <strong> <font color="#FFFFFF" size="-1"
face="Verdana, Arial, Helvetica, sans-serif"> Sponsors: </font> </strong> </div></td>
  </tr>
  <tr bgcolor="#CCCCCC">
    <td class="body-small"><div align="center"><font color="#666666" size="-2"><img src="spacer.gif"
width="1" height="15" border="0"><a href="http://www.packaginggraphics.net/packaging-design.html"
target="_blank">Packaging Graphics Co.</a></font></div></td>
  </tr>
  <tr bgcolor="#CCCCCC">
    <td height="22" class="body-small"><div align="center"><img src="images/spacer.gif" width="1"
height="15" align="absmiddle"> <font color="#666666" size="-2"><a href="http://www.brochure-
design.com" target="_blank">Brochure Design &amp;amp; Printing</a></font></div></td>
  </tr>
  <tr>
    <td height="10" bgcolor="cccccc"></td>
  </tr>
</table>
```

CSS Fundamentals

- CSS1

  December 1996

- CSS 2

  Became a W3C Recommendation in May 1998

- CSS 3

  CSS level 3 has been under development since December 15, 2005

CSS Version (1)

- CSS 1
  - Font properties such as typeface and emphasis
  - Color of text, backgrounds, and other elements
  - Text attributes such as spacing between words, letters, and lines of text
  - Alignment of text, images, tables and other elements
  - Margin, border, padding, and positioning for most elements
  - Unique identification and generic classification of groups of attributes

CSS Version (2)

- CSS2

  includes a number of new capabilities like
  - absolute, relative, and fixed positioning of elements and z-index,
  - the concept of media types
  - support for aural style sheets and bidirectional text
  - new font properties such as shadows.

CSS Version (3)

- CSS3
  - Modules include:
  - Borders (border-radius, box-shadow)
  - Backgrounds (multiple backgrounds)
  - Color (HSL colors, HSLA colors, opacity, RGBA colors)
  - Also:
  - media queries
  - multi-column layout
  - Web fonts

Simple Example (1)

```
<h1>Fancy Lads</h1>
<p>Welcome to The Fancy lad Site!</p>
<p>This web-page is the semi-official home of
    Fancy lads on the World Wide Web.</p>

<!--How much simpler is that? -->

<p align="center">
    <font face="Papyrus">
        <img border="0" src="fancy-header.png" width="207" height="279">
    </font>
</p>
<p align="center">
    <font face="Papyrus"> Welcome to The Fancy lad Site!</font>
</p>
<p align="center">
    <font face="Papyrus">This web-page is the semi-official home
    of Fancylads on the World Wide Web.</font>
</p>
```

Simple Example (2)

```
h1{
    background:url(fancy-header.png) no-repeat;
    width:207px;
    height:279px;
}
p {
    text-align:center;
    font-family:papyrus;
}
```

So, How Does It Work?

- You create a style sheet, the browser downloads it, parses it and then the browser:

Matches elements on the page

And then it ->

Styles Them

Getting the Style Sheet on the Page

```
<!- This is in the HEAD of your document -- >

<!- xHTML -- >
<link rel="stylesheet" type="text/css"
  href="/_assets/styles/style.css" />
<!- HTML5-- >
<link type="text/css" href="/_assets/styles/style.css" >
```

Basic Anatomy of a Style Sheet (1)

```
/* A single tag */
/* Many elements will inherit from this tag, since it's high up in the
document */
body {
        background: #CCC url(/_assets/styles/images/page-bg.png) repeat-x;
        font: normal .825em/1.65 Verdana, Arial, Helvetica, sans-serif;
        color: #333;
}
/*an ID */
 #container {
        background:#fff;
        height:auto;
        margin:auto;
        overflow:auto;
        position:relative;
        width:980px;
}
```

Basic Anatomy of a Style Sheet (2)

```
 /* A single tag */
h1 {
        color: #999;
        font-size: 200%;
        text-transform: uppercase;
        font-weight:normal;
}
 /* A series of ID/tag combinations, with the same rules applied */
#main h2, #main h3, #main h4, #main h5 {
        font-weight:normal;
        line-height:1.4;
        margin:7px auto;
}
```

Basic Anatomy of a Style Sheet (3)

```
/* A class */
.more-link {
        font-weight:bold;
        text-transform:uppercase;
        font-size:110%;
        text-decoration:none !important;
}
/* An ID/class combo */
#main .share {
        margin-top:7px;
}
/* An ID/class/tag combo */
#main .share strong {
        background: url(/_assets/styles/images/share.png) 0px 3px no-
repeat;
        color:#393;
        padding-left: 19px;
        text-transform:uppercase;
}
```

```
.selector {
    /*declaration*/
    property:value
}
```

HOA SI
UNIVE

Summary of the Selector syntax (1)

Pattern	Meaning
*	any element
E	an element of type E
E[foo]	an E element with a "foo" attribute
E[foo="bar"]	an E element whose "foo" attribute value is exactly equal to "bar"
E[foo~="bar"]	an E element whose "foo" attribute value is a list of whitespace-separated values, one of which is exactly equal to "bar"
E[foo^="bar"]	an E element whose "foo" attribute value begins exactly with the string "bar"
E[foo$="bar"]	an E element whose "foo" attribute value ends exactly with the string "bar"
E[foo*="bar"]	an E element whose "foo" attribute value contains the substring "bar"
E[foo|="en"]	an E element whose "foo" attribute has a hyphen-separated list of values beginning (from the left) with "en"

HOA S
UNIVE

Summary of the Selector syntax (2)

Pattern	Meaning
E:root	an E element, root of the document
E:nth-child(n)	an E element, the n-th child of its parent
E:nth-last-child(n)	an E element, the n-th child of its parent, counting from the last one
E:nth-of-type(n)	an E element, the n-th sibling of its type
E:nth-last-of-type(n)	an E element, the n-th sibling of its type, counting from the last one
E:first-child	an E element, first child of its parent
E:last-child	an E element, last child of its parent
E:first-of-type	an E element, first sibling of its type
E:last-of-type	an E element, last sibling of its type
E:only-child	an E element, only child of its parent
E:only-of-type	an E element, only sibling of its type
E:empty	an E element that has no children (including text nodes)

https://jsfiddle.net/hms3xzgp/2/

HOA UNIV

Summary of the Selector syntax (3)

Pattern	Meaning
E:link E:visited	an E element being the source anchor of a hyperlink of which the target is not yet visited (:link) or already visited (:visited)
E:active E:hover E:focus	an E element during certain user actions
E:target	an E element being the target of the referring URI
E:lang(fr)	an element of type E in language "fr" (the document language specifies how language is determined)
E:enabled E:disabled	a user interface element E which is enabled or disabled
E:checked	a user interface element E which is checked (for instance a radio-button or checkbox)
E::first-line	the first formatted line of an E element
E::first-letter	the first formatted letter of an E element

Summary of the Selector syntax (4)

Pattern	Meaning
E::before	generated content before an E element
E::after	generated content after an E element
E.warning	an E element whose class is "warning" (the document language specifies how class is determined).
E#myid	an E element with ID equal to "myid".
E:not(s)	an E element that does not match simple selector s
E F	an F element descendant of an E element
E > F	an F element child of an E element
E + F	an F element immediately preceded by an E element
E ~ F	an F element preceded by an E element

https://jsfiddle.net/yoauhrj3/4/

The Selector Classification (1)

1. Universal selector

*[lang=fr] and [lang=fr] are equivalent.

*.warning and .warning are equivalent.

*#myid and #myid are equivalent.

2. Type selectors

```
h1 { font-family: sans-serif }
```

3. Descendant selectors

```
h1 { color: red }
em { color: red }
h1 em { color: blue }
```

```
<H1>This <SPAN class="myclass">headline is
<EM>very</EM> important</SPAN></H1>
```

The Selector Classification (2)

4. Child selectors

```
body > P { line-height: 1.3 }
div ol>li p
```

5. Adjacent sibling selectors

```
h1 + h2 { margin-top: -5mm }
```

6. Attribute selectors

```
a[href="http://www.w3.org/"] { color: blue; }
```

7. Class selectors

```
.pastoral { color: green }  /* all elements with
class~=pastoral */
```

The Selector Classification (3)

8. ID selectors

- The following ID selector represents an h1 element whose ID-typed attribute has the value "chapter1":

  h1#chapter1

- The following ID selector represents any element whose ID-typed attribute has the value "chapter1":

  #chapter1

- The following selector represents any element whose ID-typed attribute has the value "z98y".

  *#z98y

The Selector Classification (4)

9. Pseudo-classes

- The pseudo-class concept is introduced to permit selection based on information that lies outside of the document tree or that cannot be expressed using the other simple selectors.
- A pseudo-class always consists of a "colon" (:) followed by the name of the pseudo-class and optionally by a value between parentheses

```
a:link    /* unvisited links */
a:visited /* visited links */
a:hover   /* user hovers */
a:active  /* active links */
```

Examples – Attribute Selector (1)

General sibling combinator: **E ~ F**

targets an element (F) preceded by another (E) e.g.: each p following h1.english, both have the same parent (body there)

```
h1.english ~ p{
        font-weight:bold;
        color:#777;
        font-size:1.4em;
        padding:1em;
}
```

Examples – Attribute Selector (2)

Attribute selector: **E[foo]**

targets an element (E) with a "foo" attribute e.g.: each img with an alt attribute

```
img[alt]{
        border:10px rgb(147,103,94) solid;
        border-radius:10px;
}
```

Examples – Attribute Selector (3)

Attribute selector: **E:not(s)**

targets an element (E) that does not match "s"

e.g.: each img with NO alt attribute

```
img:not([alt]){
        border:10px rgb(147,103,94) solid;
        border-radius:10px;
}
```

Examples – Attribute Selector (4)

Attribute selector: **E[foo*="bar"]**

targets an element (E) whose "foo" attribute contains the string "bar“

e.g.: each a whose "href" attribute contains the substring "glossaire"

```
a[href*="glossaire"]{
        color:gray;
}
```

Examples – Attribute Selector (5)

Attribute selector: **E[foo^="bar"]**

targets an element (E) whose "foo" attribute begins with the string "bar“

e.g.: each a whose "href" attribute begins with the substring "http"

```
a[href^="http"]{
        color:#6996d3;
        padding-left:1em;
        text-decoration:none;
}
```

Examples – Attribute Selector (6)

Attribute selector: **E[foo$="bar"]**

targets an element (E) whose "foo" attribute ends with the string "bar"

e.g.: each a whose "href" attribute ends with the substring ".pdf"

```
a[href$=".pdf"]{
        background:url(../images/
   selectors/pictopdf.gif) no-repeat 100% 50%;
}
```

Examples – Attribute Selector (7)

E:first-child

targets an element (E) that is the first child of its parent

e.g.: each <th> that is the first child of a <tr>

```
th:first-child{
        border-left:none;
}
th:nth-child(1){ /* is similar to :first-child */
        border-left:none;
}
```

Examples – Attribute Selector (8)

E:last-of-type

targets an element (E) that is the last child of its type in its parent

e.g.: each <th> that is the last th of a <tr>

```
th:last-of-type{
        border-bottom:none;
}
th:first-child{
        border-left:none;
}
th:nth-child(1){ /* is similar to :first-child */
        border-left:none;
}
```

Priority	CSS Source Type	Description
1	Importance	The ‘!important’ annotation overwrites the previous priority types
2	Inline	A style applied to an HTML element via HTML ‘style’ attribute
3	Media Type	A property definition applies to all media types, unless a media specific CSS is defined
4	User defined	Most browsers have the accessibility feature: a user defined CSS
5	Selector specificity	A specific contextual selector (#heading p) overwrites generic definition
6	Rule order	Last rule declaration has a higher priority
7	Parent inheritance	If a property is not specified, it is inherited from a parent element
8	CSS property definition in HTML document	CSS rule or CSS inline style overwrites a default browser value
9	Browser default	The lowest priority: browser default value is determined by W3C initial value specifications

Calculating a selector's specificity (1)

- A selector's specificity is calculated as follows:
  - count the number of ID selectors in the selector (= a)
  - count the number of class selectors, attributes selectors, and pseudo-classes in the selector (= b)
  - count the number of type selectors and pseudo-elements in the selector (= c)
  - ignore the universal selector

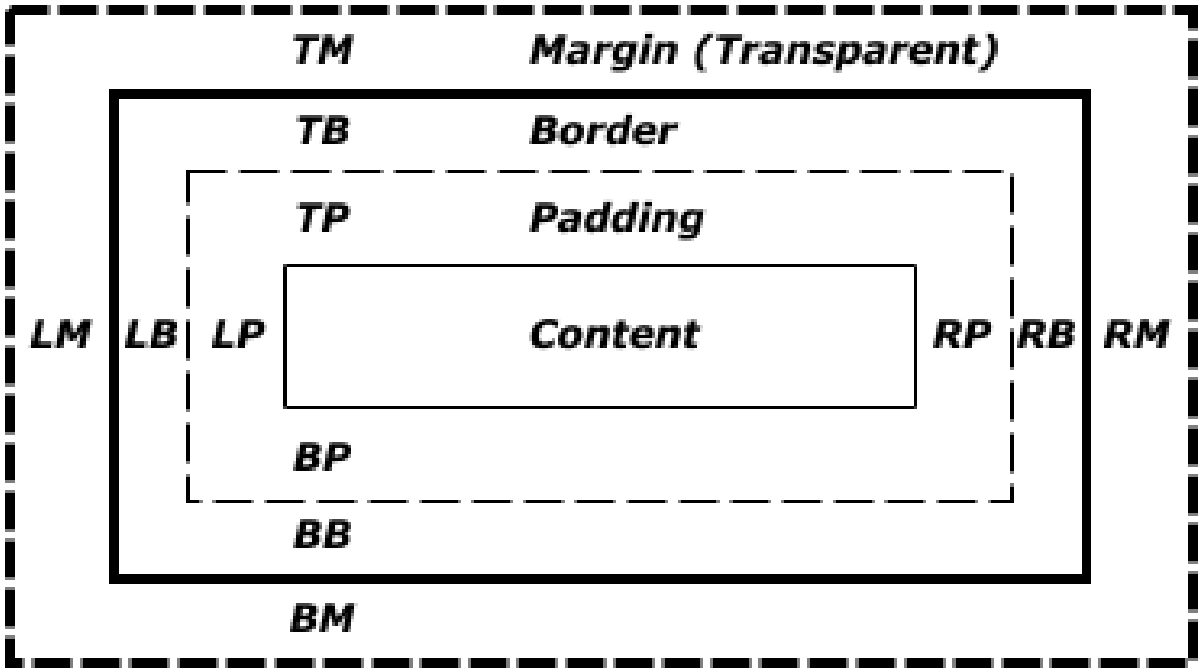
Calculating a selector's specificity (2)

Examples:

```
*                 /* a=0 b=0 c=0 -> specificity =   0 */
LI                /* a=0 b=0 c=1 -> specificity =   1 */
UL LI             /* a=0 b=0 c=2 -> specificity =   2 */
UL OL+LI          /* a=0 b=0 c=3 -> specificity =   3 */
H1 + *[REL=up]    /* a=0 b=1 c=1 -> specificity =  11 */
UL OL LI.red      /* a=0 b=1 c=3 -> specificity =  13 */
LI.red.level      /* a=0 b=2 c=1 -> specificity =  21 */
#x34y             /* a=1 b=0 c=0 -> specificity = 100 */
#s12:not(FOO)     /* a=1 b=0 c=1 -> specificity = 101 */
```

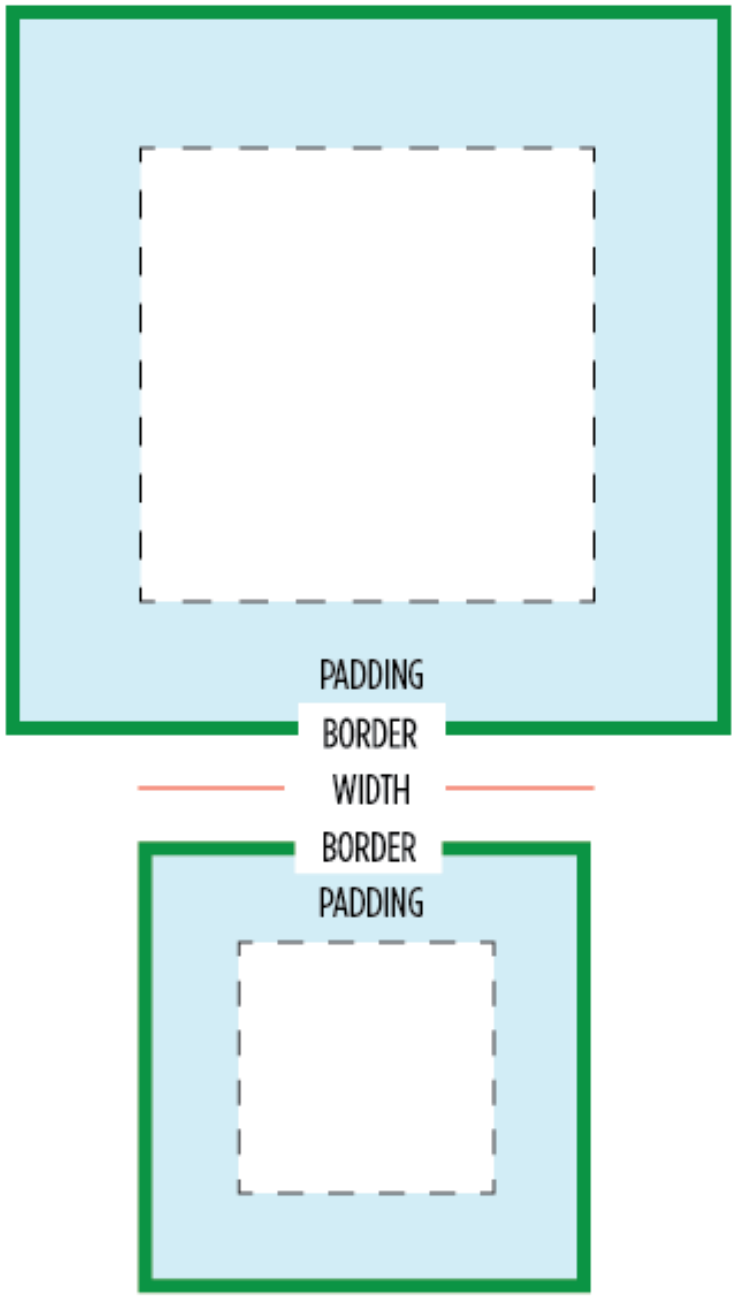
LET'S SEE IT IN ACTION

Box model

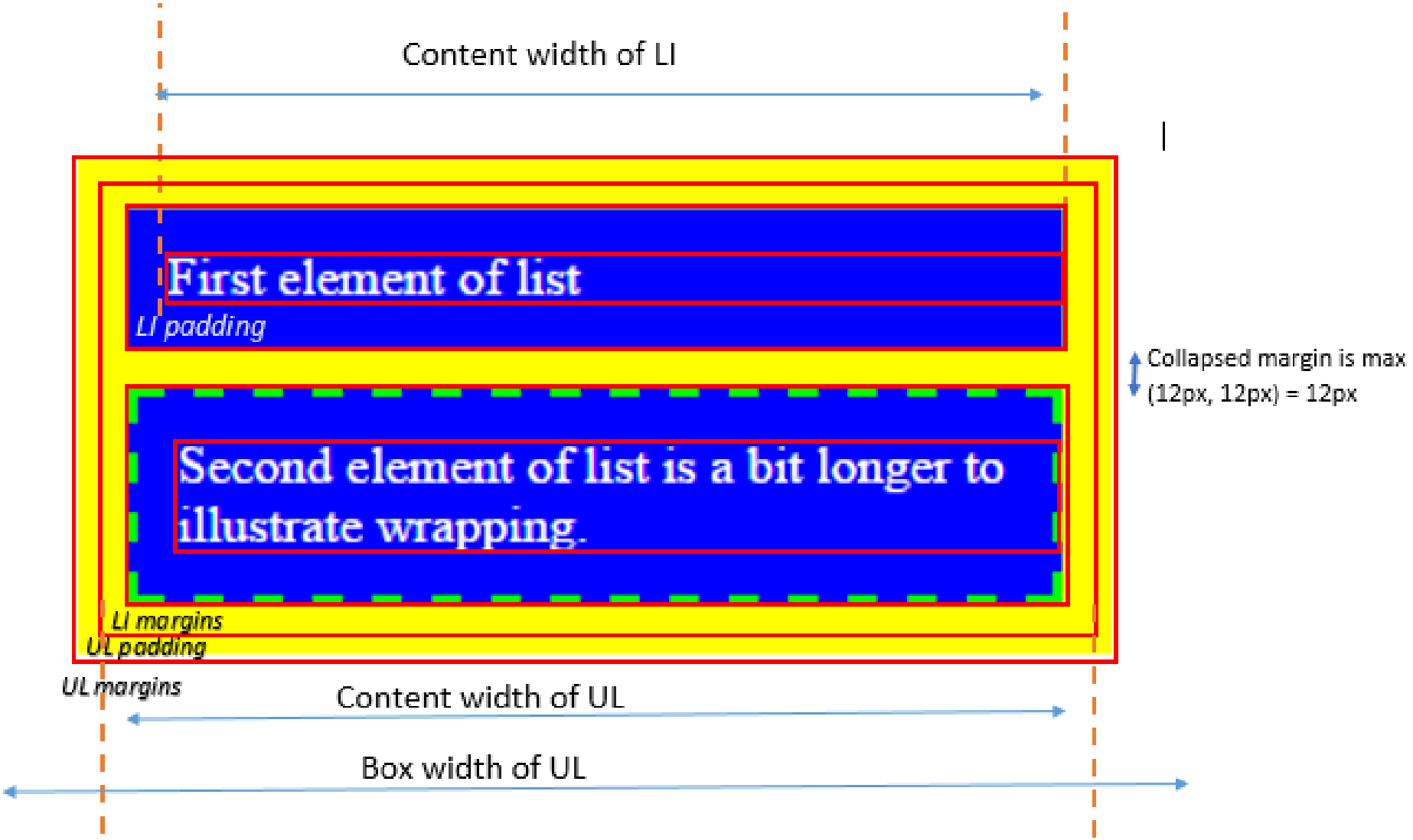


Margin edge
Border edge
Padding edge
Content edge

Box model

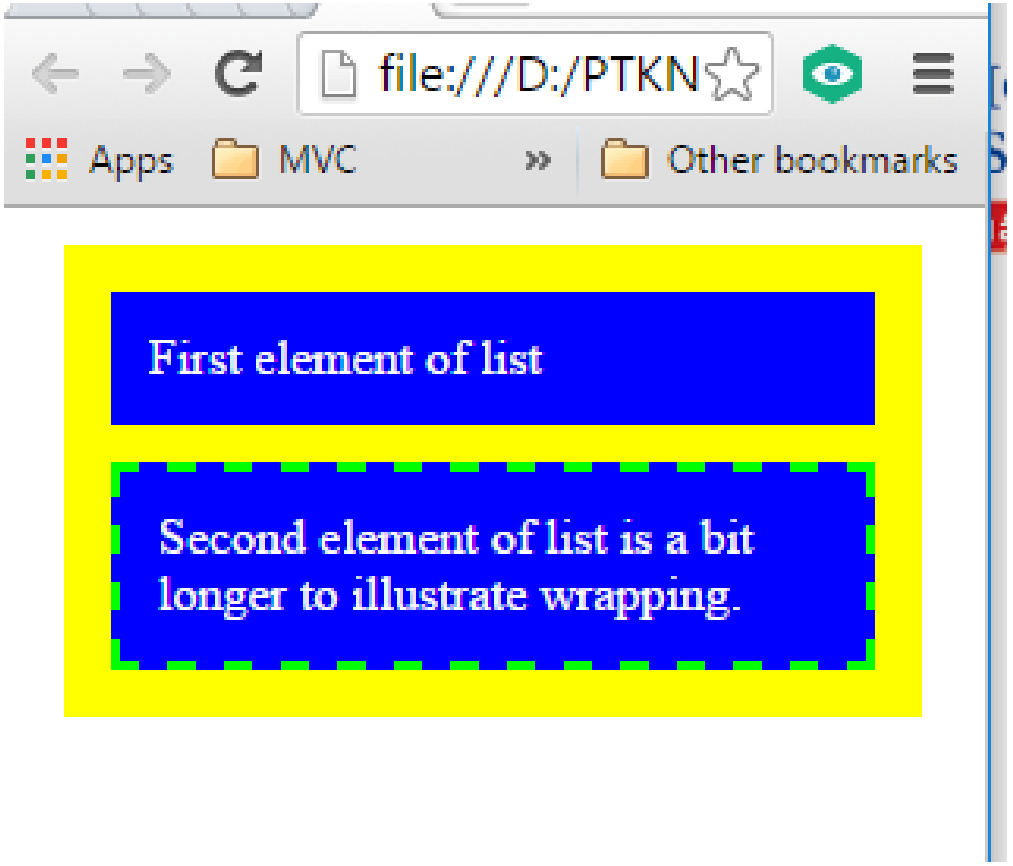


Box model


Content width of LI
First element of list
LI padding
Collapsed margin is max
(12px, 12px) = 12px
Second element of list is a bit longer to illustrate wrapping.
LI margins
UL padding
UL margins
Content width of UL
Box width of UL

Box model

Example



Box Sizing

- Since the dawn of CSS, the box model has worked like this by default:

  width + padding + border = actual visible/rendered width of an element's box

  height + padding + border = actual visible/rendered height of an element's box

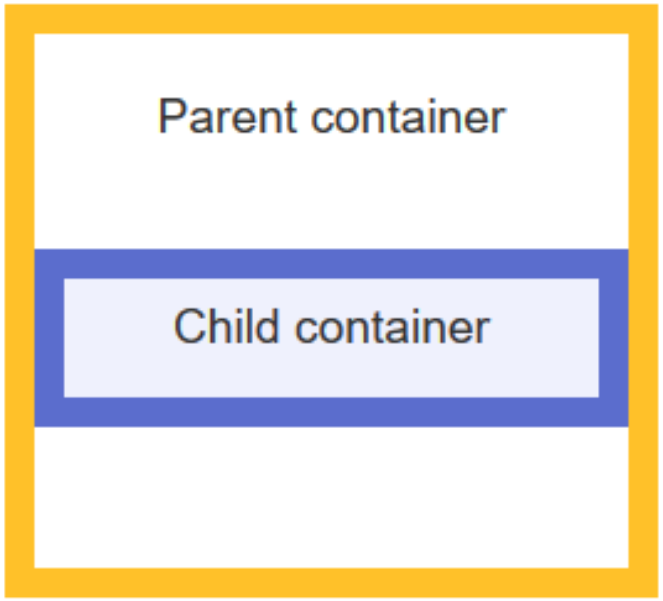
- The box-sizing property was introduced in CSS3.
- Though box-sizing has three possible values (**content-box, padding-box, and border-box**), the most popular value is **border-box**.

https://codepen.io/team/css-tricks/pen/970f26f621cfa3ae3eec7e2a6b0e8c97

```
box-sizing: content-box;
width: 100%;
```

```
box-sizing: content-box;
width: 100%;
border: solid #5B6DCD 10px;
padding: 5px;
```

```
box-sizing: border-box;
width: 100%;
border: solid #5B6DCD 10px;
padding: 5px;
```



Box-Sizing

Fonts/Backgrounds/Borders

- https://www.w3schools.com/css/css_border.asp
- http://jsfiddle.net/7pj0Lmwe/

Position (1)

Value:	static | relative | absolute | fixed | inherit
Initial:	static
Applies to:	all elements
Inherited:	no

- **Static:** The box is a normal box, laid out according to the normal flow. The 'top', 'right', 'bottom', and 'left' properties do not apply.
- **Relative:** The box's position is calculated according to the normal flow (this is called the position in normal flow). Then the box is offset relative to its normal position. When a box B is relatively positioned, the position of the following box is calculated as though B were not offset. The effect of 'position:relative' on table-row-group, table-header-group, table-footer-group, table-row, table-column-group, table-column, table-cell, and table-caption elements is undefined.

Position (2)

- **Absolute**: The box's position (and possibly size) is specified with the 'top', 'right', 'bottom', and 'left' properties. These properties specify offsets with respect to the box's containing block. Absolutely positioned boxes are taken out of the normal flow. This means they have no impact on the layout of later siblings. Also, though absolutely positioned boxes have margins, they do not collapse with any other margins.
- **Fixed**: The box's position is calculated according to the 'absolute' model, but in addition, the box is fixed with respect to some reference. As with the 'absolute' model, the box's margins do not collapse with any other margins.

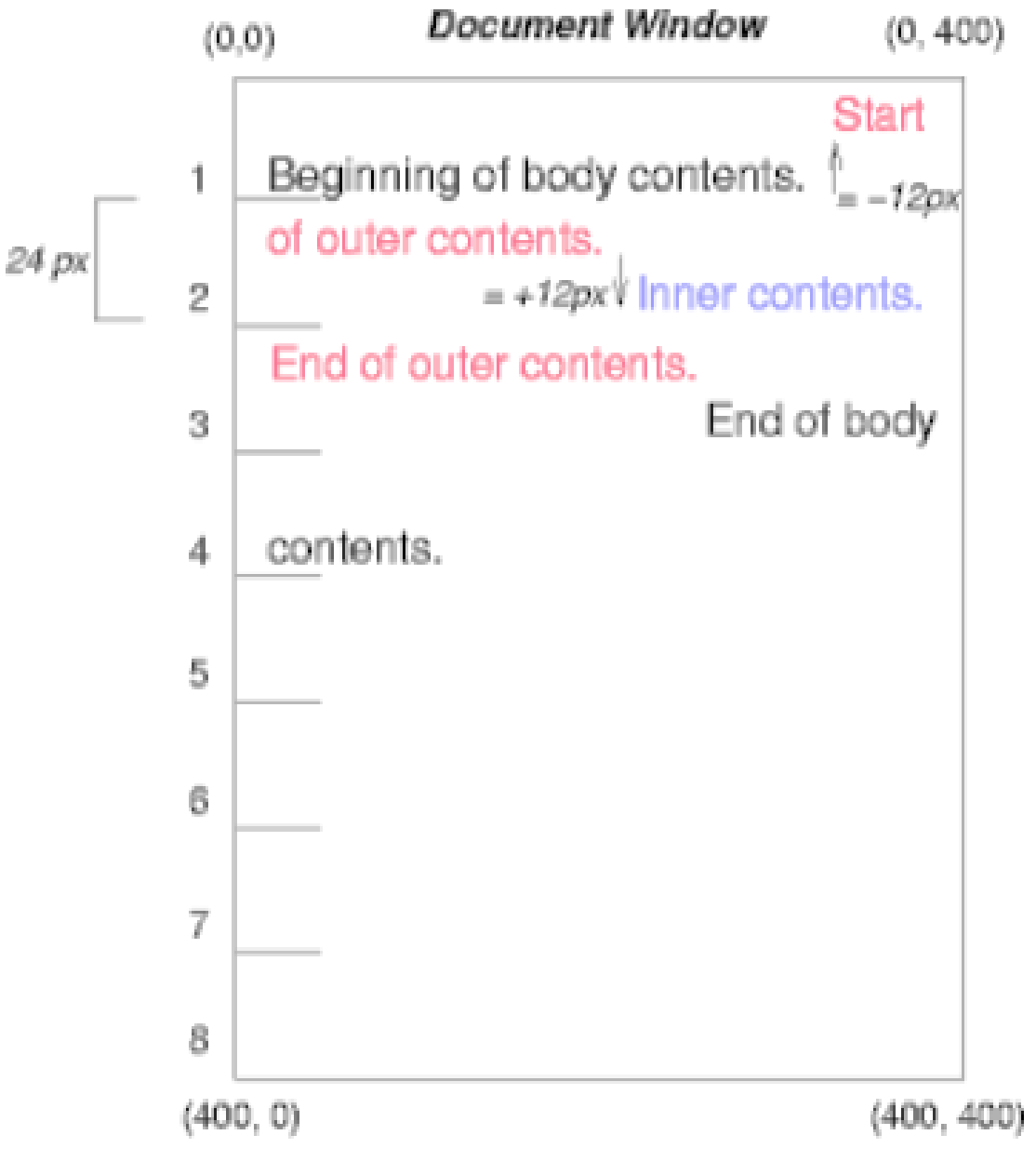
Demo - Position

- **Absolute, fixed**

  http://jsfiddle.net/3uNsN/41/

- **Relative**

http://jsfiddle.net/3uNsN/55/



Float (1)

- A float is a box that is shifted to the left or right on the current line. The most interesting characteristic of a float (or "floated" or "floating" box) is that content may flow along its side (or be prohibited from doing so by the 'clear' property). Content flows down the right side of a left-floated box and down the left side of a right-floated box. The following is an introduction to float positioning and content flow; the exact rules governing float behavior are given in the description of the 'float' property.

Float (2)

Result

A float is a box that is shifted to the left or right on the current line. The most interesting characteristic of a float (or "floated" or "floating" box) is that content may flow along its side (or be prohibited from doing so by the 'clear' property). Content flows down the right side of a left-floated box and down the left side of a right-floated box. The following is an introduction to float positioning and content flow; the exact rules governing float behavior are given in the description of the 'float' property.

A float is a box that is shifted to the left or right on the current line. The most interesting characteristic of a float (or "floated" or "floating" box) is that content may flow along its side (or be prohibited from doing so by the 'clear' property). Content flows down the right side of a left-floated box and down the left side of a right-floated box. The following is an introduction to float positioning and content flow; the exact rules governing float behavior are given in the description of the 'float' property.

I'm floating, yo!

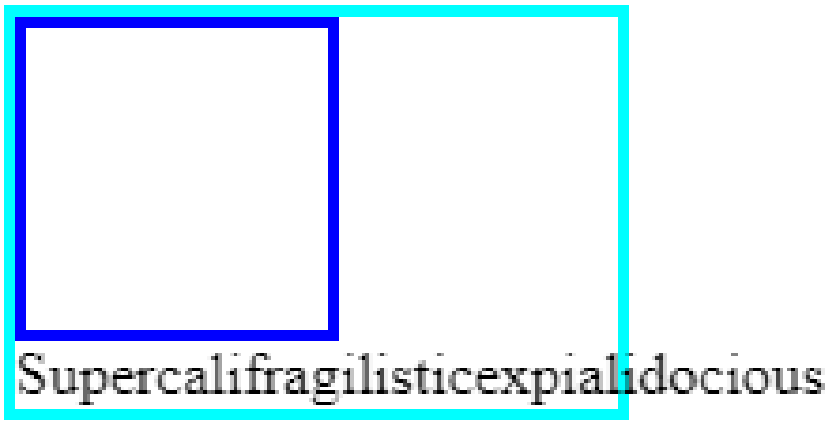
A float is a box that is shifted to the left or right on the current line. The most interesting characteristic of a float (or "floated" or "floating" box) is that content may flow along its side (or be prohibited from doing so by the 'clear' property). Content flows down the right side of a left-floated box and down the left side of a right-floated box. The following is an introduction to float positioning and content flow; the exact rules governing float behavior are given in the description of the 'float' property.

http://jsfiddle.net/sMjJq/

Float (3)

```
p {
        width: 10em; border: solid aqua;
}
span {
        float: left; width: 5em; height: 5em; border: solid blue;
}
```

```
<p>
  <span> </span>
  Supercalifragilisticexpialidocious
</p>
```

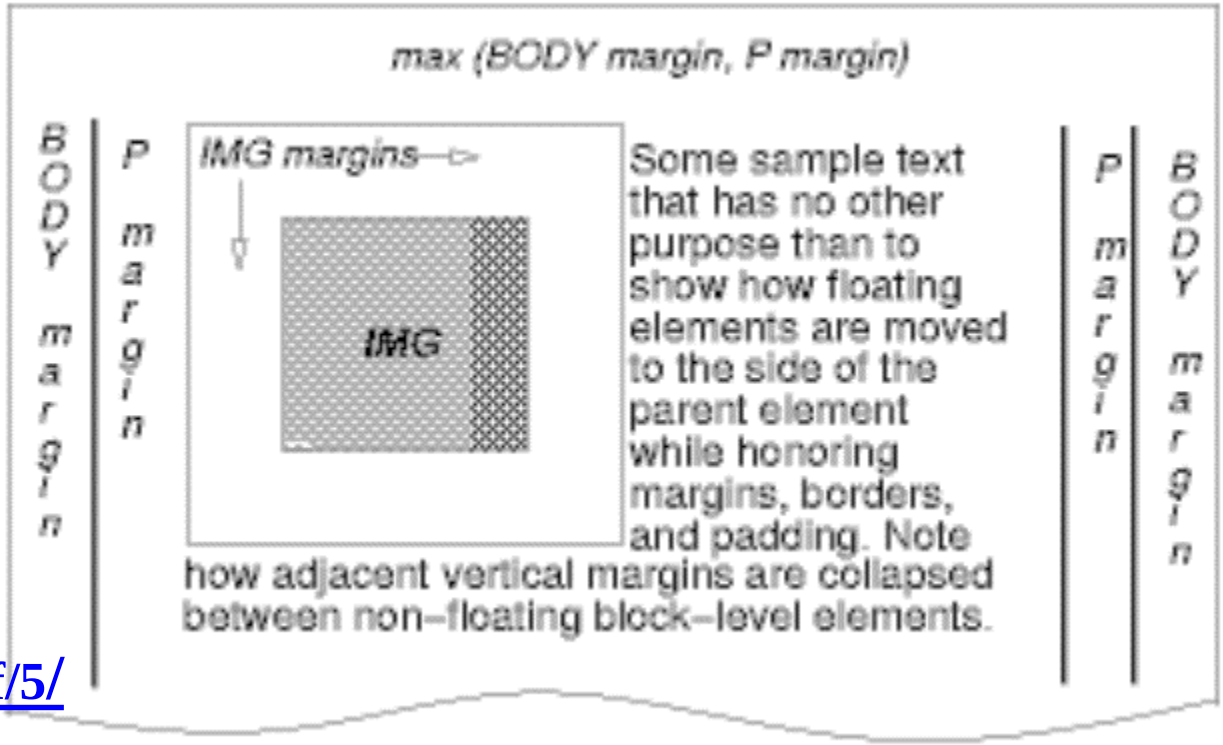


http://jsfiddle.net/tx41sm5f/

Float (4)

```
<P><IMG src=img.png alt="This image will illustrate floats“ />
    Some sample text that has no other... </P>
```

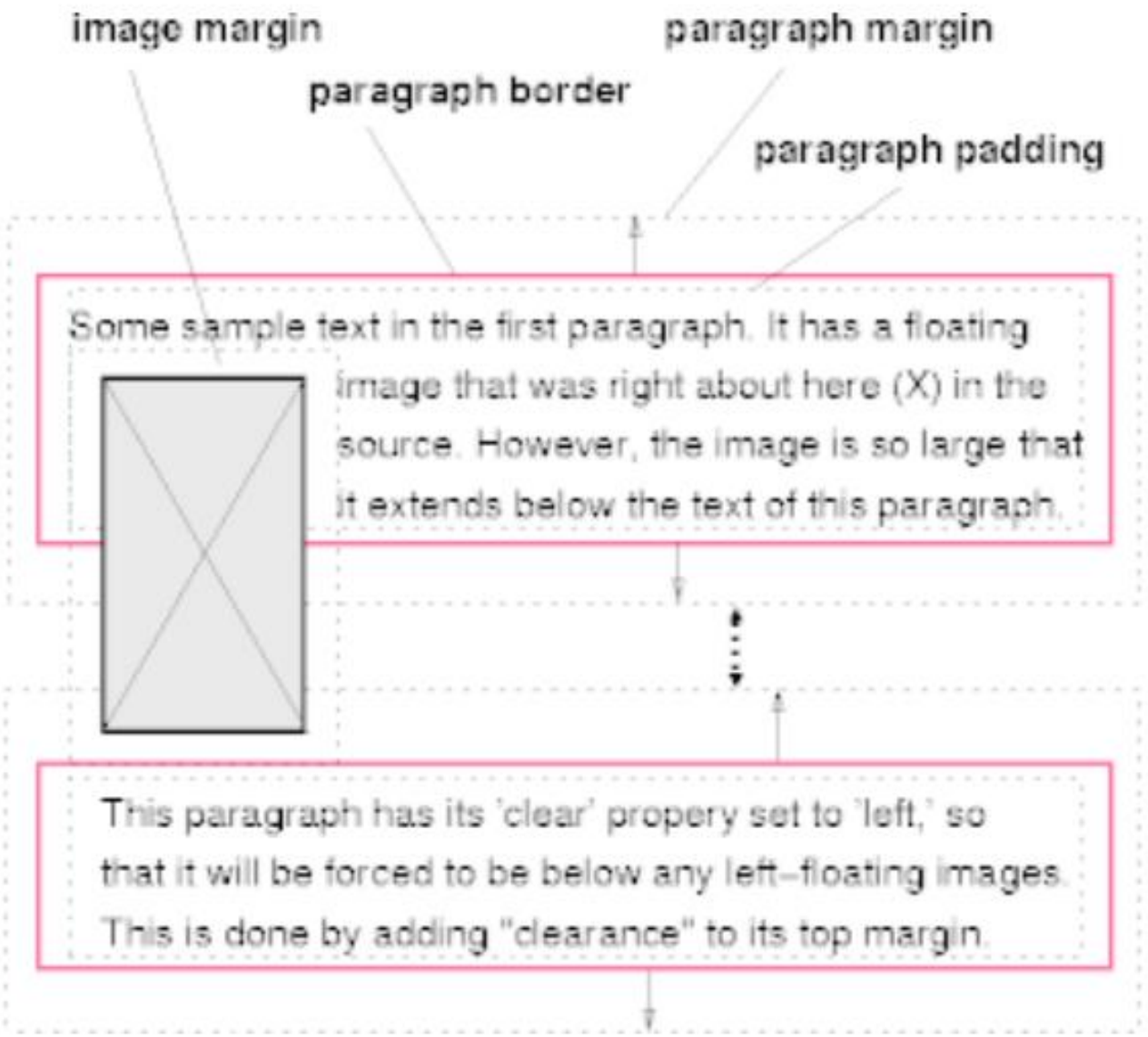
```
IMG { float: left; width:80px; height:80px;}
BODY, P, IMG { margin: 2em }
```



http://jsfiddle.net/tx41sm5f/5/

Float (5)

```
p { clear: left }
```



http://jsfiddle.net/tx41sm5f/11/

Layout

- http://jsfiddle.net/95qmesah/1/

References

- https://www.w3.org/TR/css3-selectors/#structural-pseudos
- https://www.w3schools.com