



HTML and CSS

8th Edition

Chapter 11: Layout with Styles

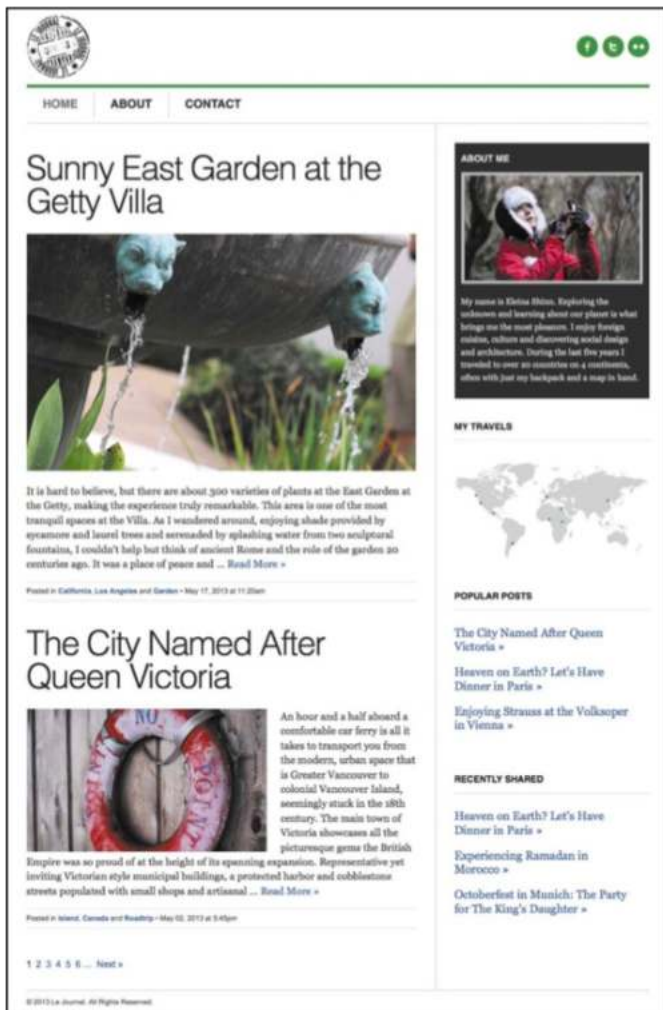


Objectives

- Structure a fixed layout page.
- Control the display type of elements
- Add margins and padding around an element.
- Float elements on a page.
- Position elements absolutely and in a stack.
- Align elements vertically.



Layout with Styles



- This page, with two columns, a header, and a footer, was laid out with CSS. CSS gives you tremendous versatility in how your pages look. This particular page design aims to be clean and content-focused. Building this layout is explained step by step throughout this chapter,



Considerations When Beginning a Layout

Separating content and presentation:

- Always separate content (HTML) and presentation (CSS) by linking to an external style sheet.
- Link all your pages, and they can all share the same layout and overall style.
- Also makes it easier to change the design of the whole site at a later date, simply by modifying the CSS file or files.



Layout Approaches

- Two main types of website designs: fixed-width and responsive.
- **Fixed-layout** has pixel-based widths for whole page and for each column of content.
 - Width doesn't change on smaller devices or browsers.
- **Responsive webpage layout**, also called fluid or liquid, uses percentages instead of pixels for widths.
 - Page shrinks and expands depending on viewing conditions.
 - Allows tailoring to mobile, tablet, and desktop users with the same HTML, not three different sites.
 - Ethan Marcotte coined the term *responsive web design*.



Structuring Your Pages

- **To structure your page:**
 - Divide sections of your page into **article**, **aside**, **main**, **nav**, **section**, **header**, **footer**, and **div** elements, as appropriate.
 - Put your content in an order that would be the most useful if CSS were not used.
 - Use **heading elements (h1–h2)** consistently to identify and prioritize information on page within sections.
 - Mark up the rest of your content with the appropriate semantics, such as **paragraphs**, **figures**, and **lists**.
 - Use comments as desired to identify different areas of your page and their contents.

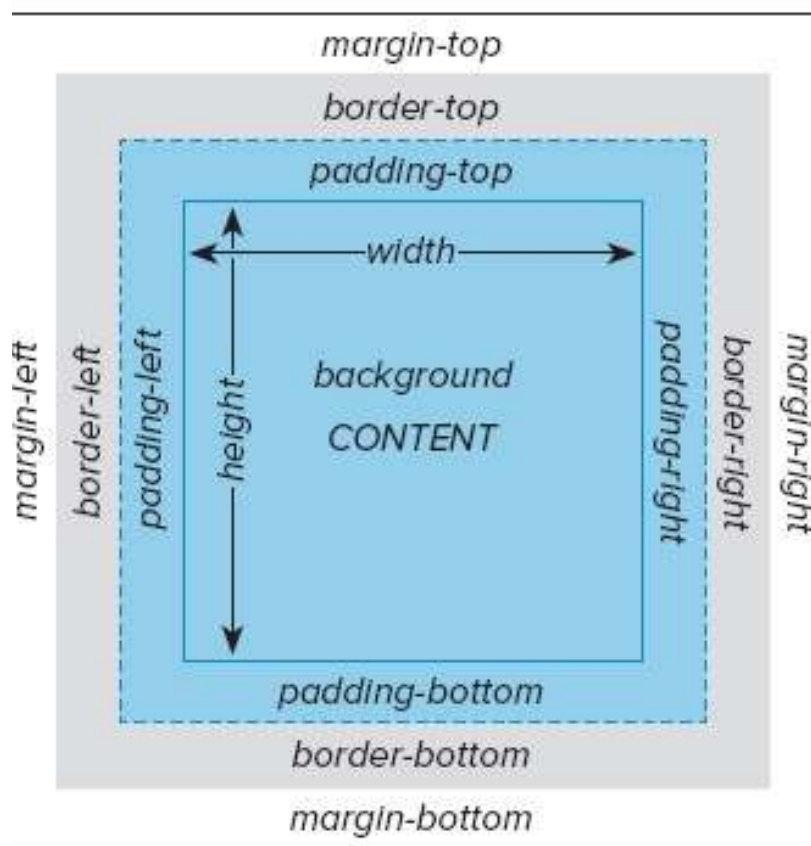


The Box Model

- **The box model:** Invisible box encloses each element.
- Four properties that determine its size:
 - Content area.
 - **Padding:** Space surrounding that area.
 - **Border:** Outside edge of the padding.
 - **Margin:** Invisible space around the border that separates one element from the next.
- Each property, and even parts of properties, can be controlled individually.



The Box Model in Context





The Box Model in Context

The space around content is the padding. (Here, there is padding on all four sides.) The background color fills both the content and padding areas

The margin is the space that separates an element from others. (Sometimes padding can be used for the same effect.)



The sidebar content area. It is set to a width of 300 pixels. No explicit height is specified so it is set automatically based on the content flow.

Each element may have a visible border. (In this case, the image has a border of 5 pixels on each side.)



Controlling the Display Type and Visibility of Elements

- Element display types:
 - **Block-level:** Starts on a new line like a new paragraph.
 - **In-line:** Doesn't generate a new line.
 - **None:** Element doesn't display at all.
- **Document flow:** Elements are displayed in same order they appear in HTML code, from top to bottom and with line breaks at beginning and end of each element that is block-level.



Inline Versus Block-Level

```
<p>This is the <em>first</em> paragraph.</p>  
<p>This is the <em>second</em> paragraph.</p>
```

Html file

Browser default

This is the *first* paragraph.

This is the *second* paragraph.

The default result

Block CSS rule

```
em {  
  display: block;  
}
```

em elements set to display: block

This is the
first
paragraph.

This is the
second
paragraph.

Inline CSS rule

```
p {  
  display: inline;  
}
```

Paragraphs set to display: inline

This is the *first* paragraph. This is the *second* paragraph.



To Specify How Elements Should Display

- In style sheet rule, type **display:**
- Type **block**, **inline**, or **none**.



By default, the **social icon links** and **main navigation links** are stacked (as shown in the figure on the left) because they are contained in list items (li). By overriding their default display: list-item; setting, the items in each list of links appear in a single row.

Changing the display to inline:

```
.social-sites li {  
  display: inline;  
}
```





Adding Padding Around an Element

- **Padding** is extra space around contents of an element, but inside the outer edge (border).
- Change padding's thickness, but not its color or texture.
- **To add padding around an element:**
 - Type **padding: x**
 - Where *x* is the amount of desired space to be added, expressed in units (typically in ems or pixels) or as a percentage of the width of the parent element.
 - Or apply padding to a single side by typing padding-top: x;, padding-right: x;, padding-bottom: x;, or padding-left: x;



Adding Padding

1

```
.about {
  background-color: #2b2b2b;
  padding: .3125em .625em .625em;
}
```

/*
 Similar to
 padding: 5px 10px 10px;
 because font-size is 16px and
 .3125 = 5px/16px
 .625 = 10px/16px
 */



2

```
.nav-main a {
  color: #292929;
  display: inline-block;
  font-family: 'Open Sans', sans-serif;
  font-size: 1.125em; /* 18px/16px */
  font-weight: 700;
  padding: .5em 1.15em .5em 1.4em;
  text-transform: uppercase;
}
```

/*
 Similar to
 padding: 9px 20.7px 9px 25.2px;
 because font-size is 18px and
 .5 = 9px/18px
 1.15 = 20.7px/18px
 1.4 = 25.2px/18px
 */

HOME ABOUT CONTACT

HOME

ABOUT

CONTACT



Shorthand Notation for Padding

- `padding: 5px;`—With one value, it applies to all sides.
- `padding: 5px 9px;`—With two values, the first value applies to the top and bottom and the second value applies to the right and left.
- `padding: 5px 9px 11px;`—With three values, the first applies to the top, the second to the right and left, and the third to the bottom.
- `padding: 5px 9px 11px 0;`—With four values, they are applied to the top, right, bottom, and left, in clockwise order.



Setting the Border

- Borders can be around an element or on individual sides of an element.
 - You can set its thickness, style, and color.
 - If you've specified any padding, the border encloses both the padding and the contents of the element.



```
.about img {  
  border: 5px solid #bebebe;  
}
```




Set One or More Border Properties with a Shortcut

- Type **border**.
 - If desired, type **-top**, **-right**, **-bottom**, or **-left** to limit the effect to a single side.
 - If desired, type **-*property***
 - Where *property* is style, width, or color, to limit the effect to a single property.
- Type **:** (a colon).
- Type appropriate values.



Set One or More Border Properties with a Shortcut

```
.nav-main {  
  /* green */  
  border-top: 5px solid #019443;  
  /* gray */  
  border-bottom: 1px solid #c8c8c8;  
}  
  
.nav-main li {  
  border-left: 1px solid #c8c8c8;  
  display: inline-block;  
}  
  
.nav-main li:first-child {  
  border-left: none;  
}
```





Border Styles in Firefox

```
<p class="ddd">A 5px border from top, right, bottom, left: <br />dotted, dashed, double, solid.</p>
<p class="ridge">Here's a paragraph with a 10px ridge border.</p>
<p class="groove">Here's one with a 10px groove border.</p>
<p class="inset">And here's one with a 10px inset border.</p>
<p class="outset">And here's one with a 10px outset border.</p>
```

```
p {
  border: 10px solid red;
  padding: 15px;
}

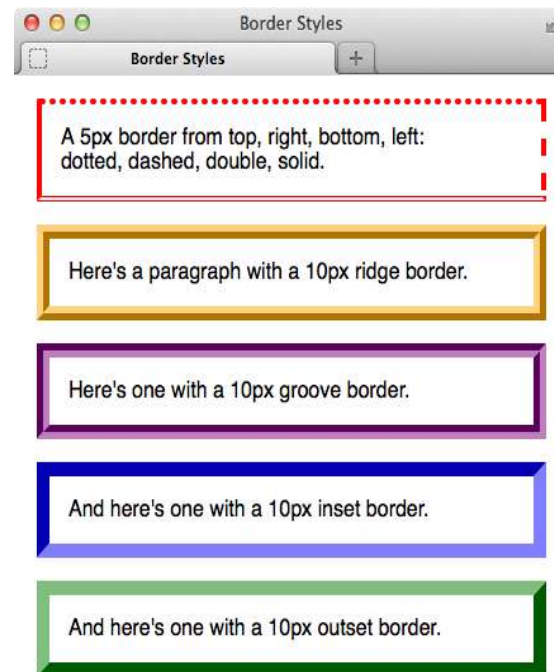
.ddd {
  border-width: 4px;
  border-style: dotted dashed double;
}

.ridge {
  border-style: ridge;
  border-color: orange;
}

.groove {
  border-style: groove;
  border-color: purple;
}

.inset {
  border-style: inset;
  border-color: blue;
}

.outset {
  border-style: outset;
  border-color: green;
}
```





Setting the Margins Around an Element

- **Margin** is the amount of transparent space between one element and the next.
- For padding and margins, em values are relative to the *element's* font size, *not* to its parent's font size.
- Formula to determine margin or padding em value is:
desired size / element's font size = value
- **To set an element's margins:**
 - Type **margin: x**
 - Where x is the amount of desired space to be added, expressed as a length, a percentage of the width of the parent element, or auto.



Adjusting Space Between Heading and Image

Sunny East Garden at the
Getty Villa



Sunny East Garden at the
Getty Villa



```
h1 {  
  color: #212121;  
  font-family: "Lato", sans-serif;  
  font-size: 3.25em; /* 36px/16px */  
  font-weight: 300;  
  letter-spacing: -2px;  
  line-height: .975;  
  margin-bottom: .4125em; /* 21.45px */  
}
```

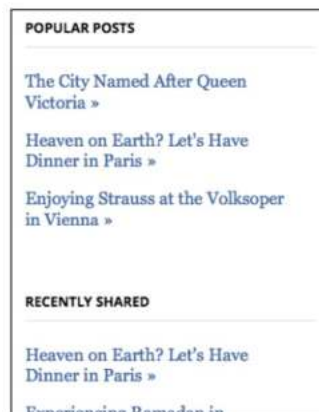


Setting the Margins Example

```
.map {  
  margin: 1.4375em 0 .8125em;  
  /* 23px 0 13px */  
}  
  
.links {  
  margin: 1.5em 0 4.125em;  
  /* 24px 0 66px */  
  padding: 0;  
}  
  
.links li {  
  margin-bottom: 1.1em;  
}
```



The map was crowded between the two headings (left), but the margins solved that (right).



There is more space between each group of links, as well as between the links themselves.



Making Elements Float

- **Float property** makes elements float in a sea of text (or other elements).
- Use to make text wrap around images or other elements to create multi-column layouts.

The City Named After Queen Victoria



An hour and a half aboard a comfortable car ferry is all it takes to transport you from the modern, urban space that is Greater Vancouver to colonial Vancouver Island, seemingly stuck in the 18th century. The main town of Victoria showcases all the picturesque gems the British

Empire was so proud of at the height of its spanning expansion. Representative yet inviting Victorian style municipal buildings, a protected harbor and cobblestone streets populated with small shops and artisanal ... [Read More »](#)

Posted in [Island](#), [Canada](#) and [Roadtrip](#) • May 02, 2013 at 5:45pm

```
.post-photo {  
    float: left;  
    margin-bottom: 2px;  
    margin-right: 22px;  
}
```



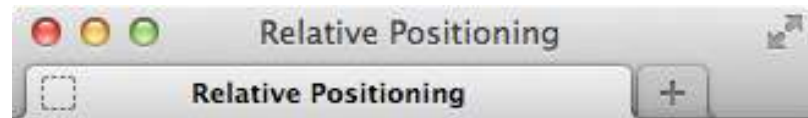
To Wrap Text Around Elements

- Type **float:**.
 - Type **left** if you want the element on the left and the rest of the content to flow to its right.
 - Or type **right** if you want the element on the right and the rest of the content to flow to its left.
 - Or type **none** if you don't want the element to float at all.
 - None is the default value.



Positioning Elements Relatively

- Each element has a natural location in a page's flow.
- **Relative positioning** means moving the element with respect to this original location. The surrounding elements are not affected at all.



Relative Positioning

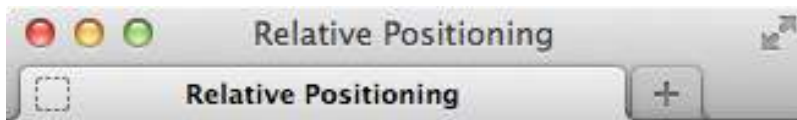
When you position an element relatively, you position it relative to its normal location.

```
...  
<h1>Relative Positioning</h1>  
  
<p>When you position an element relatively,  
→ you position it  
→ </span> relative to its normal location.  
→ </p>  
...
```



To Offset Elements Within the Natural Flow

```
.example {  
  position: relative;  
  top: 35px;  
  left: 100px;  
}
```



Relative Positioning

When you position an element relatively, you position it relative to its normal location.

position it

- Type `position: relative;`
- Type `top`, `right`, `bottom`, or `left`.
- Then type `:d`, where *d* is the desired distance that you want to offset the element from its natural location. Repeat for additional offsets.



Positioning Elements Absolutely

- Elements in a webpage generally flow in the order in which they appear in the HTML source code.
- **Absolute positioning** takes elements out of the normal flow entirely by specifying their precise position with respect to the body or to their nearest positioned ancestor element.
 - No space is left where an absolutely positioned element would have appeared normally.
 - Other elements do not flow around an absolutely positioned element.



To Position Elements Absolutely

- Type **position: absolute;**
- If desired, type **top**, **right**, **bottom**, or **left**.
- Then type : ***d***,
 - Where *d* is expressed as desired distance you want to offset the element from its ancestor or as percentage of ancestor.
- If desired, repeat step for additional offsets.
- If desired, add position: relative to *ancestor* element.
 - Skip this step and the element will be offset with respect to the body.

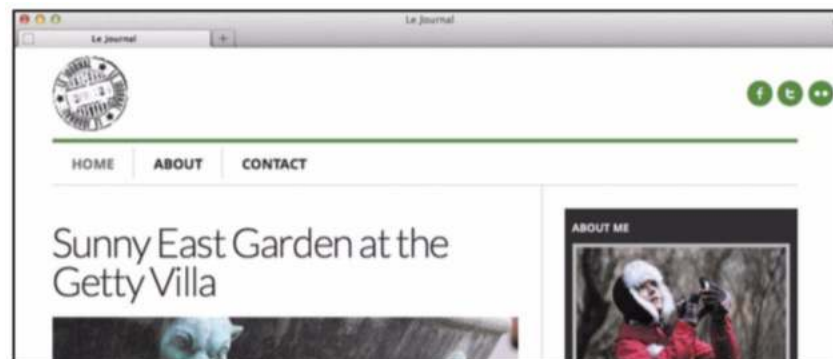


To Position Elements Absolutely

```
...  
<header class="masthead" role="banner">  
  <p class="logo"><a href="/"><img ... />  
  → </a></p>  
  
  <ul class="social-sites">  
    ...  
  </ul>  
  
  <nav role="navigation">  
    <ul class="nav-main">  
      ...  
    </ul>  
  </nav>  
</header>  
...
```



```
.masthead .social-sites {  
  position: absolute;  
  top: 41px;  
  right: 0;  
}
```

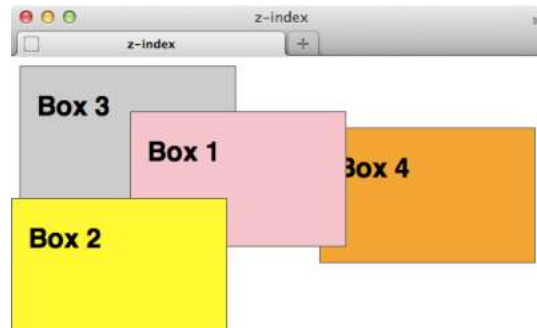




Positioning Elements in a Stack

- When elements overlap, you can choose which element should display on top.
- **To position elements in a stack:**
 - Type `z-index: n` , where n is a number that indicates the element's level in the stack of positioned objects.

```
<div class="box1"><p>Box 1</p></div>
<div class="box2"><p>Box 2</p></div>
<div class="box3"><p>Box 3</p></div>
<div class="box4"><p>Box 4</p></div>
```



```
div {
  border: 1px solid #666;
  height: 125px;
  position: absolute;
  width: 200px;
}

.box1 {
  background: pink;
  left: 110px;
  top: 50px;
  z-index: 120;
}

.box2 {
  background: yellow;
  left: 0;
  top: 130px;
  z-index: 530;
}

.box3 {
  background: #ccc;
  position: static;
  /* Static, so has no effect. */
  z-index: 1000;
}

.box4 {
  background: orange;
  left: 285px;
  top: 65px;
  z-index: 3;
}
```



Changing the Cursor

- **To change the cursor:**
 - Type **cursor**:
 - Type **pointer, arrow, crosshair, move, wait, help, text, or progress**.
 - Or type **auto** to get the cursor that usually appears in that situation.
 - Or type **x-resize** to get double-sided arrow, where *x* is the cardinal direction one of the arrows should point—that is, *n* (north), *nw* (northwest), *e* (east), and so on.



Changing the Cursor



```
.nav-main .current-page,  
.nav-main .current-page:hover {  
  color: #747474;  
  cursor: default;  
}
```




List styling: Choosing Your Markers

- When creating lists, choose markers (bullets, numbers, or images, none) to appear to left of each list item.



The Great American Novel

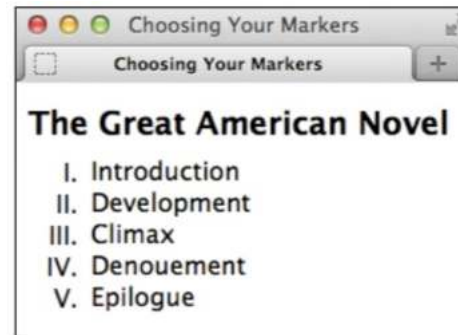
- I. Introduction
- II. Development
- III. Climax
- IV. Denouement
- V. Epilogue



To Choose Your Markers

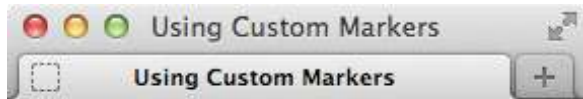
- **To choose your markers:**
 - In style sheet rule, *type list-style-type: marker*
 - *marker* is one of following values: disc, circle, square decimal (1, 2, 3, ...), upper-alpha (A, B, C, ...), lower-alpha (a, b, c, ...), upper-roman (I, II, III, IV, ...), lower-roman (i, ii, iii, iv, ...)
- **To display lists without markers:**
 - In the style sheet, type *list-style-type: none*

```
li {  
    list-style-type: upper-roman;  
}  
  
/* This would have the same effect because  
→ the li elements inherit list-style-type.  
  
ol {  
    list-style-type: upper-roman;  
}  
*/
```





Using Custom Markers



Product Features

- ✓ One-click page layout.
- ✓ Spell-checker for 327 languages.
- ✓ Image retouching tool.
- ✓ Unlimited undos and redos.

```
ul {  
  list-style: none;  
  margin-left: 0;  
  padding-left: 0;  
}  
  
li {  
  /* show image slightly down from top  
  → of item. */  
  background: url(../img/checkmark.png)  
  → no-repeat 0 .1em;  
  
  /* make line tall enough to show  
  → full checkmark */  
  line-height: 1.8;  
  
  /* bump the text over to make room  
  → for the checkmark */  
  padding-left: 1.75em;  
}
```

- Create a custom marker with an image.
- You don't have to change the HTML, just the CSS.



To Use Custom Markers

- In the rule for the desired list or list item, *type list-style: none;* to turn off normal markers.
- In the rule for the desired list, set *margin-left and/or padding-left* properties to dictate indentation.



To Use Custom Markers

- In the rule for li elements within the desired list, type *background: url(image.ext) repeat-type horizontal vertical;*
 - *image.ext* is path and file name of image;
 - *repeat-type* is value of no-repeat, repeat-x, or repeat-y;
 - *horizontal* and *vertical* are values for position of background.
- Type *padding-left: value;*
 - *value* is at least width of background image to prevent list item content from overlapping custom marker.



Setting All List-Style Properties at Once



- List-style properties can all be set at once.

Product Features

- One-click page layout. (This is particularly useful when you're under a heavy deadline. You just select whether you want the end product to be a book or a website, and poof, it's done!)
- Spell checker for 327 major languages.
- Image retouching tool.
- Unlimited undos and redos.

```
ul {  
    list-style: circle inside;  
}  
  
/*  
Use ol as the selector to style an unordered  
→ list.  
You may also use li instead for either.  
*/
```



To Set All the List-Style Properties at Once

- Type *list-style:*.
- If desired, specify kind of markers to appear next to list items, if any.
- If desired, specify custom marker that should be used for list items.
- If desired, specify whether markers should be hung outside the list paragraphs or flush with text.