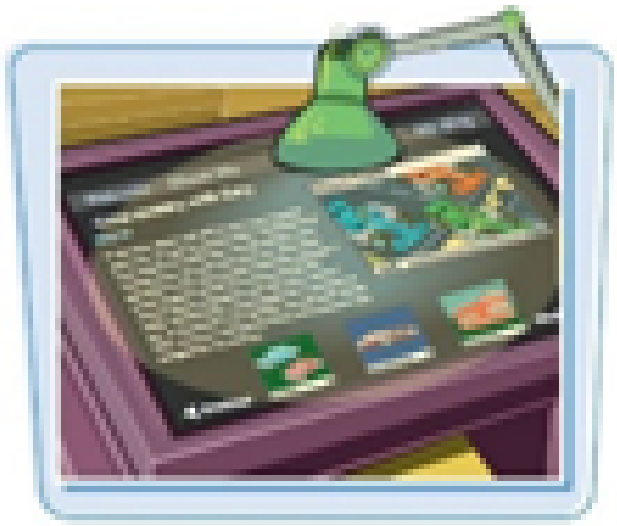
Lesson Proper for Week 11

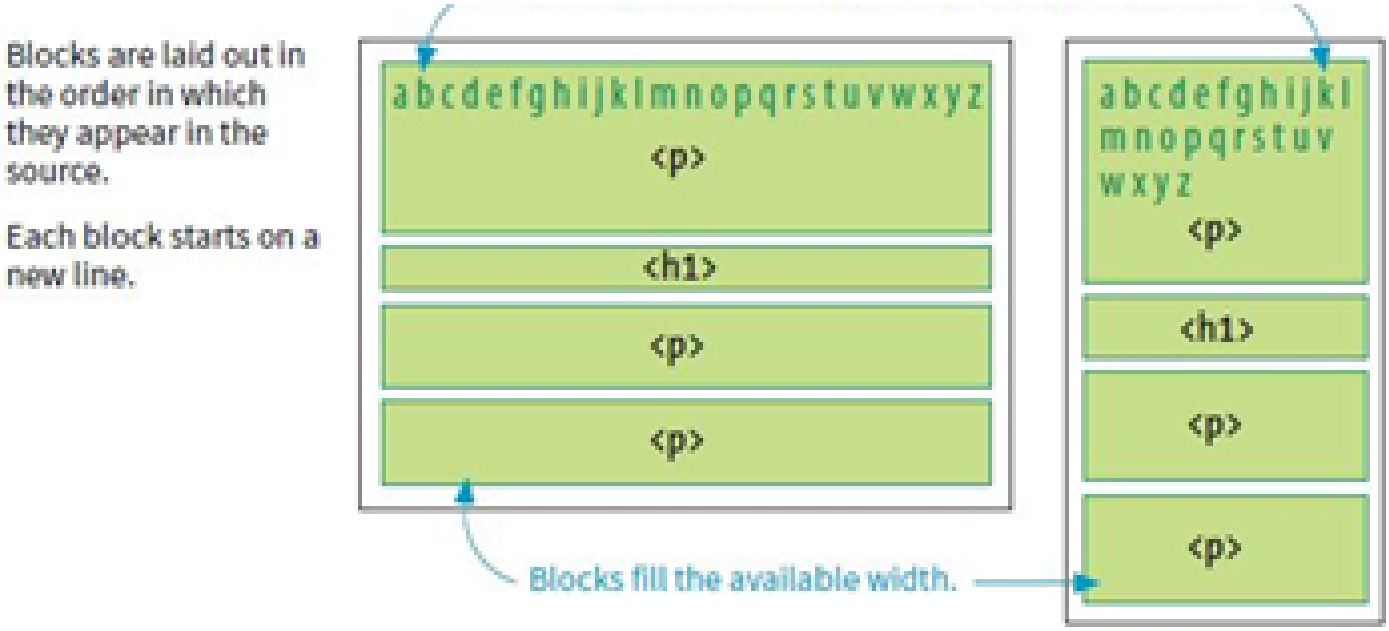


You can use cascading style sheets, or CSS, to organize text, images, and other elements on your Web page in precise ways. This enables you to create layouts that are more complicated than those you can create with HTML. Style sheets allow you to specify where in a page to put different types of content by defining coordinates within the browser window. You can also precisely control the space around different elements and even overlap content on your pages.

Normal Flow

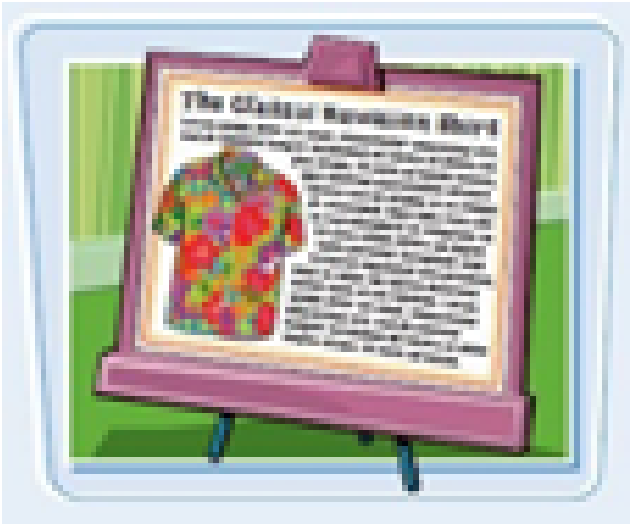
Block elements stack up on top of one another and fill the available width of the browser window or other containing element. Inline elements and text characters line up next to one another to fill the block elements.

When the window or containing element resizes, the block elements expand or contract to the new width, and the inline content reflows to fit.



Objects in the normal flow affect the layout of the objects around them. This is the behavior you've come to expect in web pages—elements don't overlap or bunch up. They make room for one another. Floating and positioning change the relationship of elements to the normal flow in different ways.

Floating Content



The float CSS property takes a box out of the normal flow of your page and moves it to the right or left side of the enclosing box. Content that follows then wraps around the floated element. Floating allows you to align images, paragraphs, tables, and other content, similar to how you can align images using the **align** attribute in HTML.

You can use the **float** property to control how text wraps around the elements on your Web page. The left value controls the left side of an element, and the right value controls the right side of an element. To ensure proper text wrapping, place the floating element right before the text you want to wrap.
float property is used to put content side-by-side.

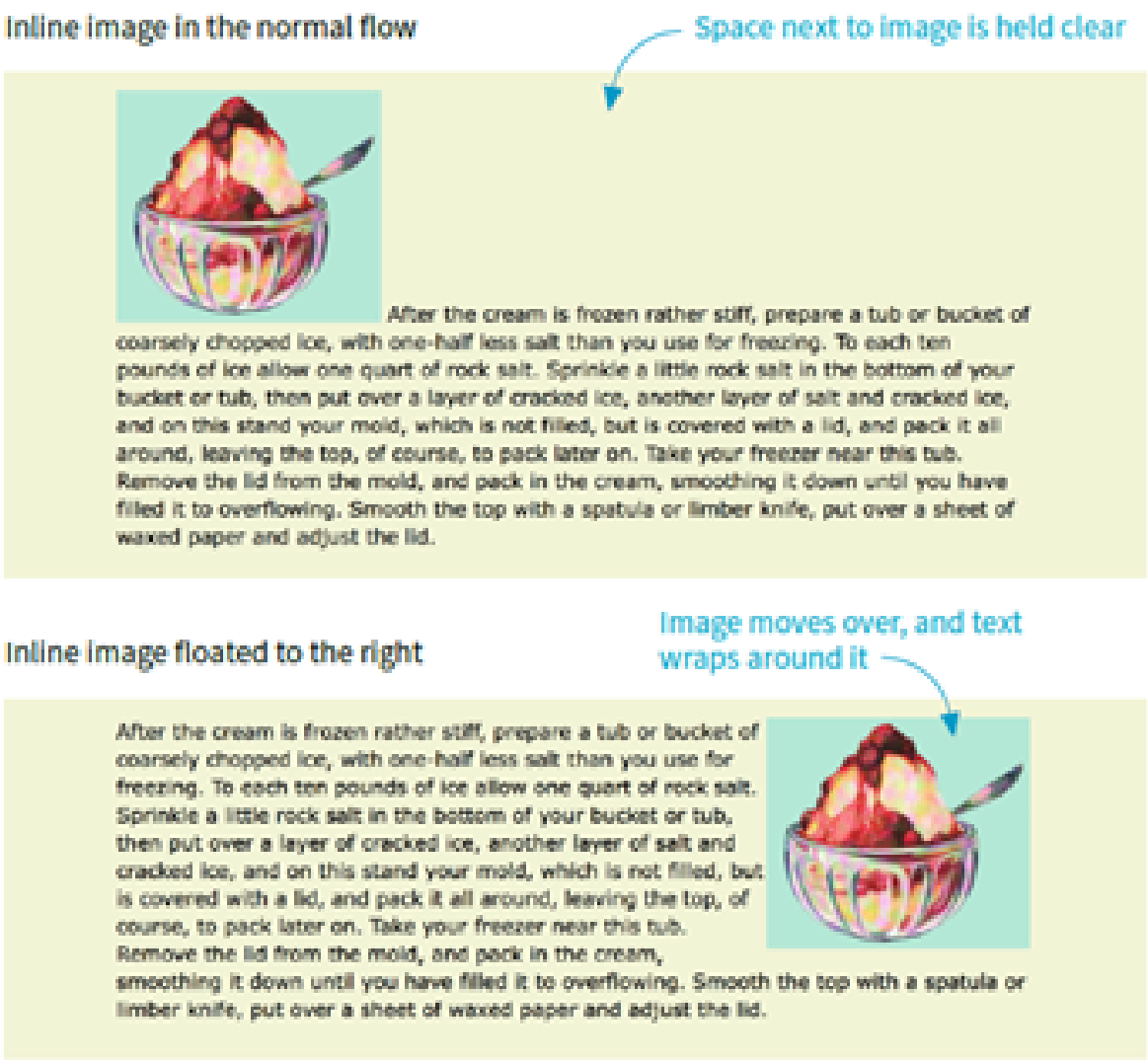
The float property does not work with elements for which you have assigned an absolute or fixed position.

THE MARKUP

```
<p><img src="icecreambowl.png" alt=""> After the cream is frozen rather
stiff,…
```

THE STYLES

```
img {
  float: right;
}
```



Example:

The float property is applied to an img element to float it to the right. Example shows how the paragraph and the contained image are rendered by default (top) and how it looks when the float property is applied (bottom).

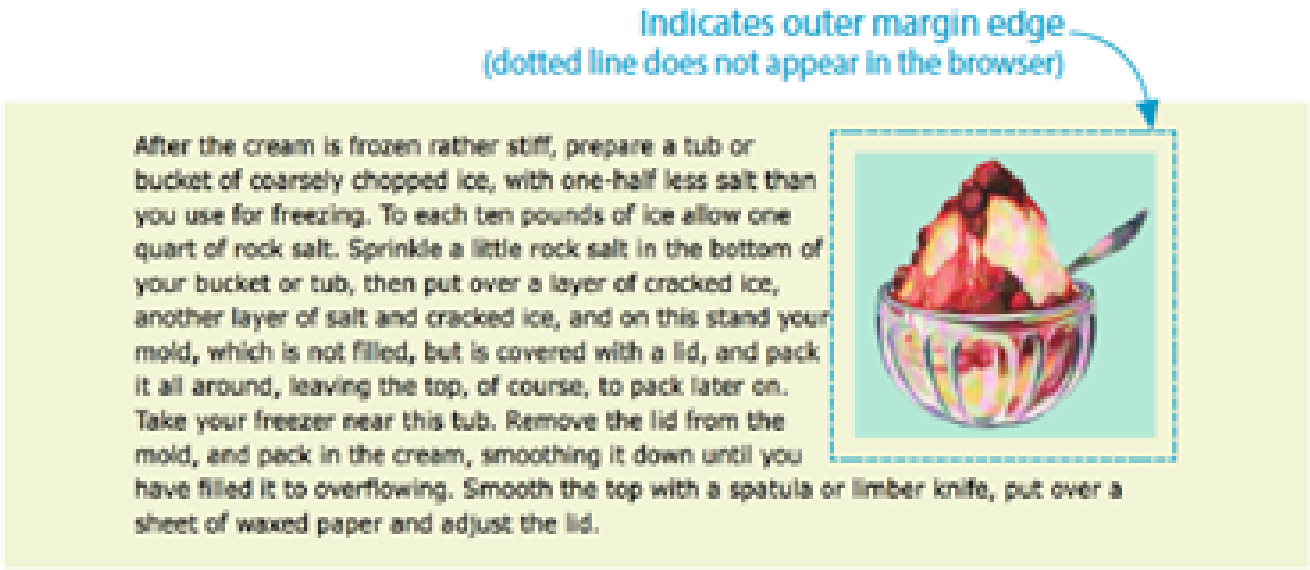
That's a nice effect. We've gotten rid of a lot of wasted space on the page, but now the text is bumping right up against the image. How do you think you would add some space between the image element and the surrounding text? If you guessed, "add a margin," you're absolutely right. I'll add 1em of space on all sides of the image with the

margin property, see example below

margin property, see example below.

You can begin to see how the box properties work together to improve page layout.

```
img {
  float: right;
  margin: 1em;
}
```



Floating Inline and Block elements

It is possible to float any HTML element, both inline and block-level, as we'll see in the following examples.

THE MARKUP

```
<p><span class="tip">TIP: Make sure that your packing tub or bucket
has a hole below the top of the mold so  the  water will drain
off.</span>After the cream is frozen rather stiff, prepare a tub or
bucket of…</p>
```

THE STYLES

```
span.tip {
  float: right;
  margin: 1em;
  width: 200px;
  color: #fff;
  background-color: lightseagreen;
  padding: 1em;
}
```

After the cream is frozen rather stiff, prepare a tub or bucket of coarsely chopped ice, with one-half less salt than you use for freezing. To each ten pounds of ice allow one quart of rock salt. Sprinkle a little rock salt in the bottom of your bucket or tub, then put over a layer of cracked ice, another layer of salt and cracked ice, and on this stand your mold, which is not filled, but is covered with a lid, and pack it all around, leaving the top, of course, to pack later on. Take your freezer near this tub. Remove the lid from the mold, and pack in the cream, smoothing it down until you have filled it to overflowing. Smooth the top with a spatula or limber knife, put over a sheet of waxed paper and adjust the lid.

TIP: Make sure that your packing tub or bucket has a hole below the top of the mold so the water will drain off.

Floating block elements

THE MARKUP

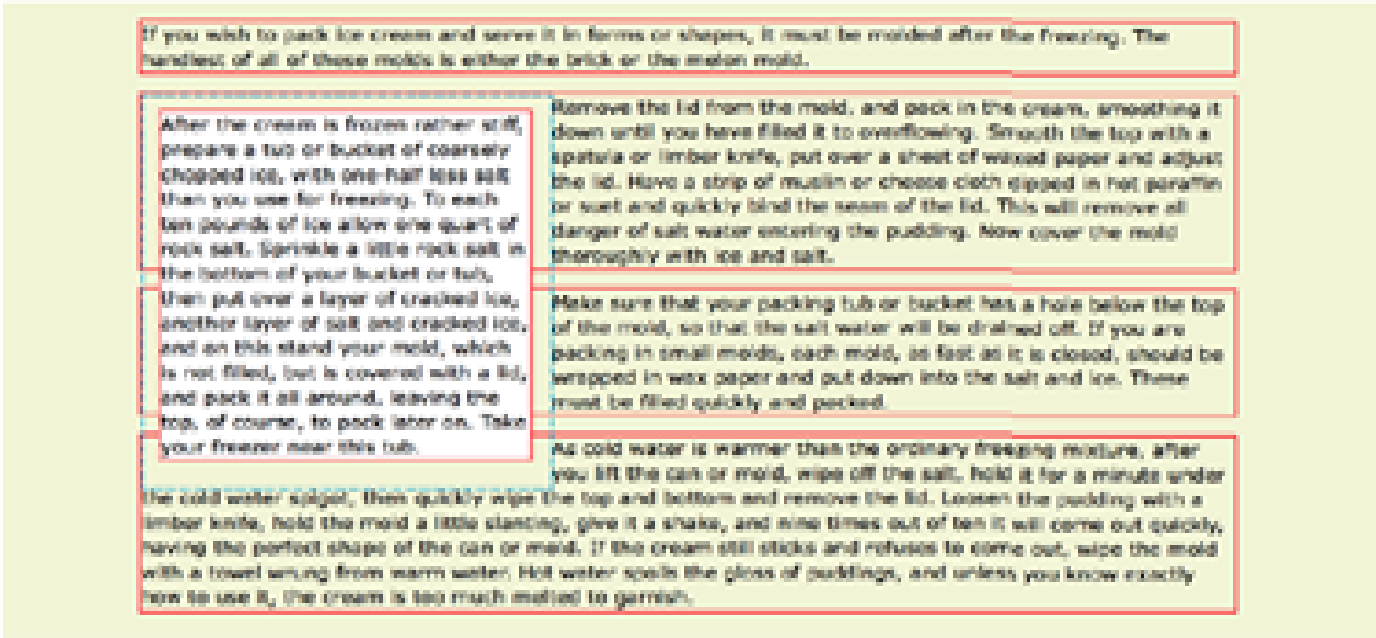
THE MARKUP

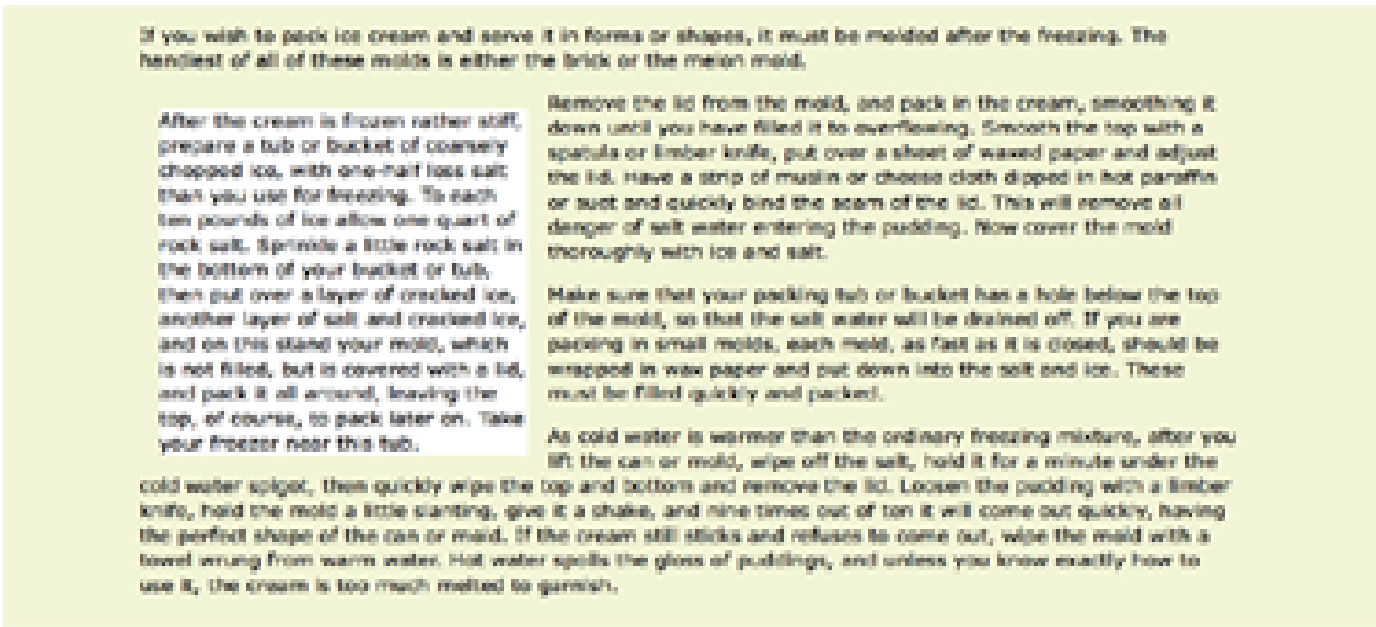
```
<p>If you wish to pack ice cream...</p>
<p id="float">After the ice cream is rather stiff,...</p>
<p>Make sure that your packing tub or bucket...</p>
<p>As cold water is warmer than the ordinary...</p>
```

THE STYLES

```
p {
  border: 2px red solid;
}

#float {
  float: left;
  width: 300px;
  margin: 1em;
  background: white;
}
```





Let's look at what happens when you float a block within the normal flow. In this example, the whole second paragraph element is floated to the left. Just as we saw with the image, the paragraph moves off to the side (left this time), and the following content wraps around it, even though blocks normally stack on top of one another.

Clearing Floated Elements

PROPERTY	VALUE
clear	none | left | right | both Initial value: none

If you're going to be floating elements around, it's important to know howto turn the text wrapping *off* and get back to normal flow as usual. You do this by clearing the element that you want to start below the float. Applying the **clear** property to an element prevents it from appearing next to a floated element and forces it to start against the next available "clear" space below the float.

The clear property is used to control floating content. The following table outlines the clear property and its possible values.

```
img {
  float: left;
  margin-right: .5em;
}
h2 {
  clear: left;
  margin-top: 2em;
}
```

If pure raw cream is stirred rapidly, it swells and becomes frothy, like the beaten whites of eggs, and is "whipped cream." To prevent this in making Philadelphia Ice Cream, one-half the cream is scalded, and when it is very cold, the remaining half of raw cream is added. This gives the smooth, light and rich consistency which makes these creams so different from others.

USE OF FRUITS

Use fresh fruits in the summer and the best canned unsweetened fruits in the winter. If sweetened fruits must be used, cut down the given quantity of sugar. Where acid fruits are used, they should be added to the cream after it is partly frozen.

The time for freezing varies according to the quality of cream or milk or water; water ices require a longer time than ice creams. It is not well to freeze the mixtures too rapidly; they are apt to be coarse, not smooth, and if they are churned before the mixture is icy cold they will be greasy or "buttery."

Keep in mind that you apply the clear property to the element you want to start below the floated element, not the

Keep in mind that you apply the clear property to the element you want to start below the floated element, not the floated element itself. The left value starts the element below any elements that have been floated to the left. Similarly, the right value makes the element clear all floats on the right edge of the containing block. If there are multiple floated elements, and you want to be sure an element starts below all of them, use the both value to clear floats on both sides.

In this example, the clear property has been used to make h2 elements start below left-floated elements. Example shows how the h2 heading starts at the next available clear edge below the float.

Positioning Content

You can use different types of positioning to place the boxes of content on your pages. Relative positioning places content on the page relative to the normal flow of the other content on the page. Absolute positioning places content on absolute points on the page relative to the containing block. Fixed positioning places content relative to the browser window and keeps it fixed as a user scrolls.

Positioning Basics

PROPERTY	VALUE
position	static | relative | absolute | fixed Initial value: static
top	<length> | <percentage> | auto Initial value: auto
right	<length> | <percentage> | auto Initial value: auto
bottom	<length> | <percentage> | auto Initial value: auto
left	<length> | <percentage> | auto Initial value: auto

CSS provides several methods for positioning elements on the page. They can be positioned relative to where they would normally appear in the flow, or removed from the flow altogether and placed at a particular spot on the page. You can also position an element relative to the viewport, and it will stay put while the rest of the page scrolls.

The position property indicates that an element is to be positioned and specifies which positioning method to use.

The following table outlines the position property and its values, and the four offset properties, top, right, bottom, and left, and their possible values.

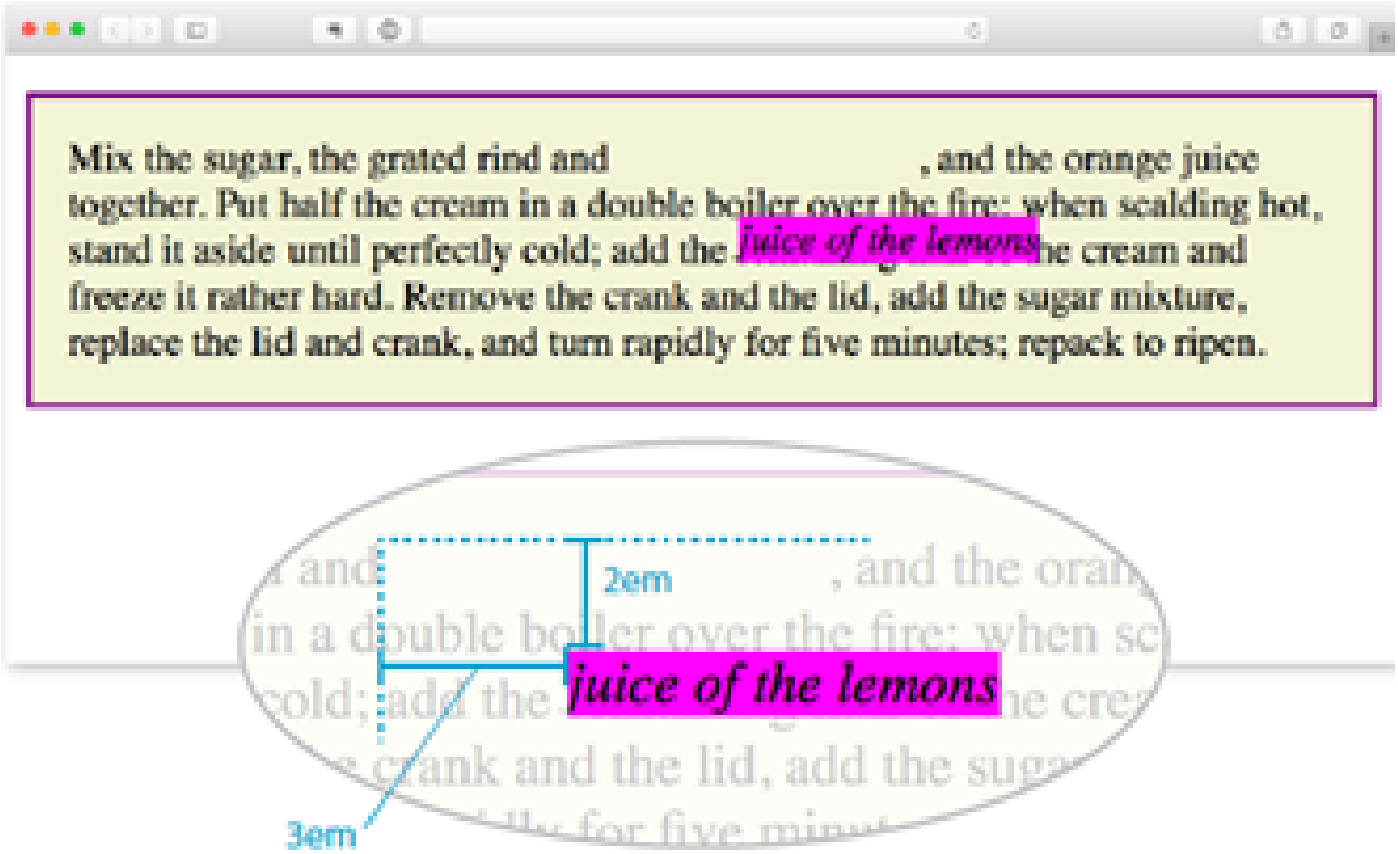
Types of Positioning

static - This is the normal positioning scheme in which elements are positioned as they occur in the normal document flow.

relative - Relative positioning moves the element box relative to its original position in the flow. The distinctive

Relative - Relative positioning moves the element box relative to its original position in the flow. The distinctive behavior of relative positioning is that the space the element would have occupied in the normal flow is preserved as empty space.

```
em {
  position: relative;
  top: 2em; /* moves element down */
  left: 3em; /* moves element right */
  background-color: fuchsia;
}
```



Example:

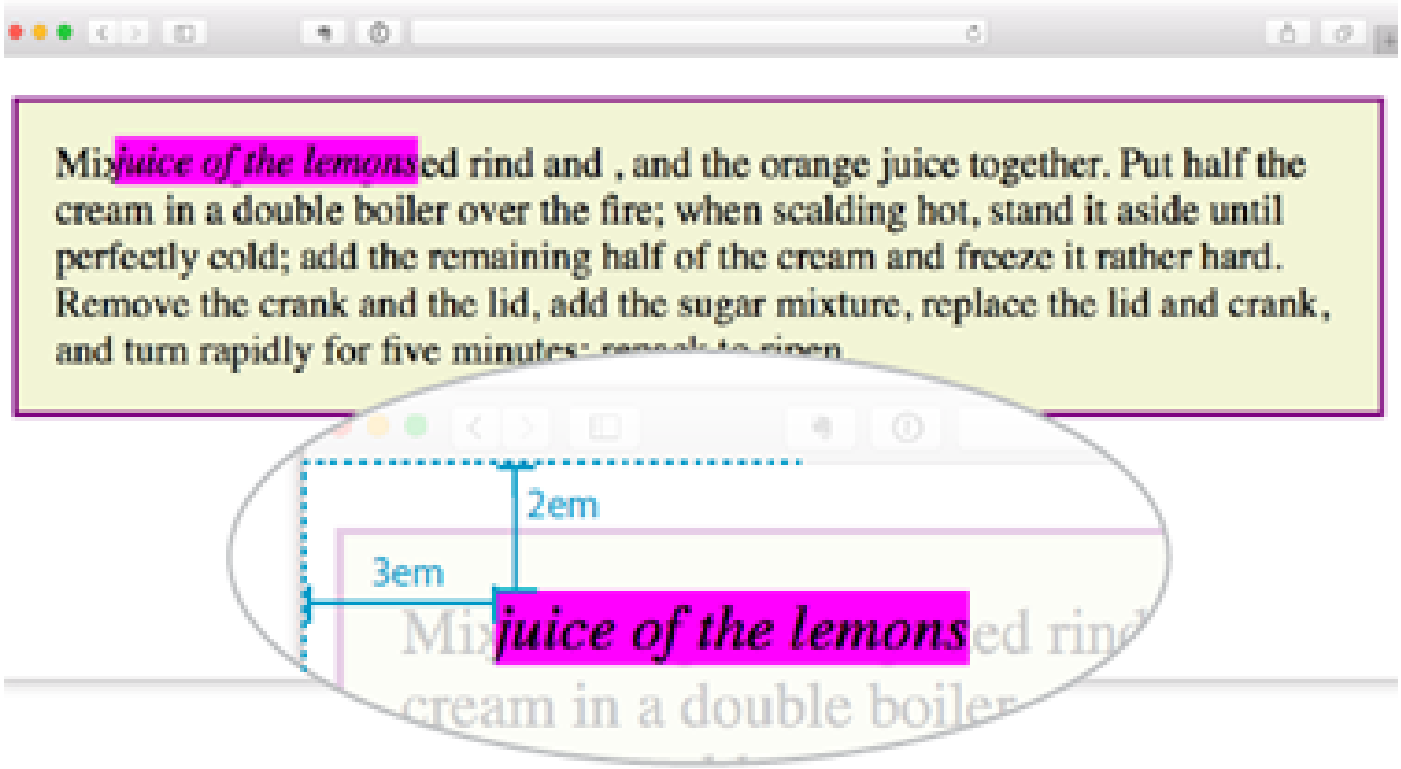
Here I've positioned an inline **em** element. A bright background color on the ***em*** and a border on the containing paragraph make their boundaries apparent. First, I used the position property to set the method to relative, and then I used the top offset property to move the element ***2em*** down from its initial position, and the left property to move it ***3em*** to the right. Remember, offset property values move the element away from the specified edge, so if you want something to move to the right, as I did here, you use the left offset property.

NOTE: *When an element is relatively positioned, the space it once occupied is preserved.*

absolute - Absolutely positioned elements are removed from the document flow entirely and positioned with respect to the viewport or a containing element. Unlike relatively positioned elements, the space they would have occupied is closed up. In fact, they have no influence at all on the layout of surrounding elements.

Now that you've seen how relative positioning works, let's take the same example, only this time we'll change the value of the position property to absolute.

```
em {
  position: absolute;
  top: 2em;
  left: 3em;
  background-color: fuchsia;
}
```



Example:

NOTE: *When an element is absolutely positioned, it is removed from the flow and the space is closed up.*

fixed - The distinguishing characteristic of fixed positioning is that the element stays in one position in the viewport even when the document scrolls. Fixed elements are removed from the document flow and positioned relative to the viewport rather than another element in the document.

sticky - Sticky positioning is a combination of relative and fixed in that it behaves as though it is relatively positioned, until it is scrolled into a specified position relative to the viewport, at which point it remains fixed.

Specifying Position - Once you've established the positioning method, the actual position is specified with some combination of up to four offset properties. **top, right, bottom, left.**

The value provided for each offset property defines the distance the element should be moved away from that

The value provided for each offset property defines the distance the element should be moved away from that respective edge. For example, the value of top defines the distance the top outer edge of the positioned element should be offset from the top edge of the browser or other containing element. A positive value for top results in the element box moving down by that amount. Similarly, a positive value for left would move the positioned element to the right (toward the center of the containing block) by that amount.

◀ Preliminary Activity for Week 11

Jump to...

Analysis, Application, and Exploration for Week 11 ▶

Navigation

Home
Dashboard
Site pages
My courses
121 - CC106
121 - BPM101 / DM103
121 - OAELEC2
121 - ITE3
121 - MUL101
121 - ITSP2B
121 - WEB101 / CCS3218
Participants
Grades
General
01 Introduction to Internet and World Wide Web
02 Full Stack Development Overview
03 The Basics of HTML
04 Mark Up Text
05 HTML Link, Image and Table
06 - Preliminary Examination
07 HTML Block, Inline Element and Forms Control
08 Introduction To Cascading Style Sheets
09 The Font Style Properties
10 The List and Table style properties
11 Advanced CSS
Preliminary Activity for Week 11
Lesson Proper for Week 11
Analysis, Application, and Exploration for Week 11

Fair Warning

NOTICE: Please be reminded that it has come to the attention of the Publishing Team of eLearning Commons that learning materials published and intended for ***free use only by students and faculty members within the eLearning Commons network were UNLAWFULLY uploaded in other sites without due and proper permission***.

PROSECUTION: Under Philippine law (Republic Act No. 8293), copyright infringement is punishable by the following: Imprisonment of between 1 to 3 years and a fine of between 50,000 to 150,000 pesos for the first offense. Imprisonment of 3 years and 1 day to six years plus a fine of between 150,000 to 500,000 pesos for the second offense.

COURSE OF ACTION: Whoever has maliciously uploaded these concerned materials are hereby given an ultimatum to take it down within 24-hours. Beyond the 24-hour grace period, our Legal Department shall initiate the proceedings in coordination with the National Bureau of Investigation for IP Address tracking, account owner identification, and filing of cases for prosecution.

2nd Semester Enrollment

Activities

- Assignments
- Forums
- Quizzes
- Resources