

# 15

## FLOATING AND POSITIONING

---

# OVERVIEW

---

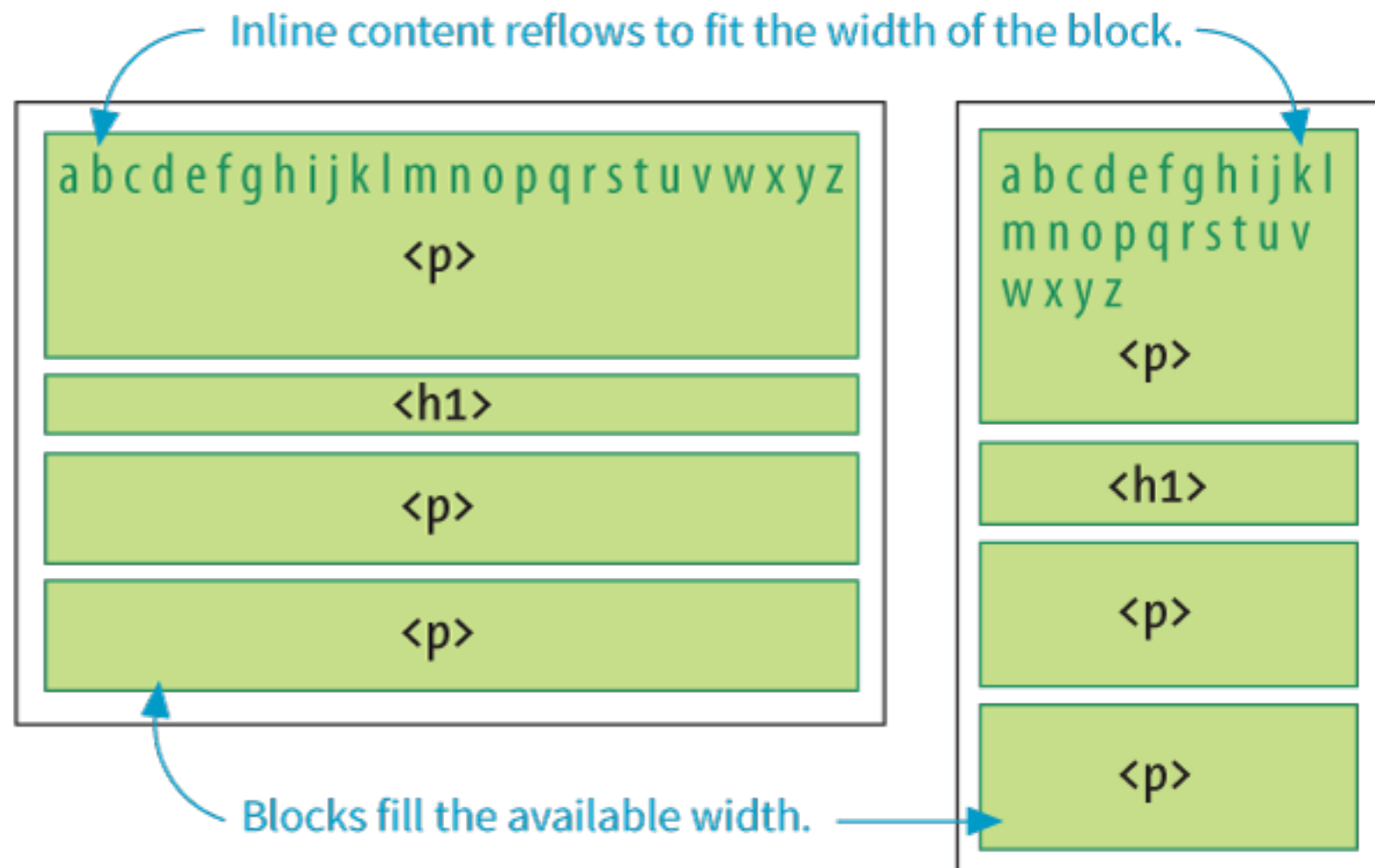
- **Understanding normal flow**
- **Floating elements to the left and right**
- **Clearing and containing floated elements**
- **Text wrap shapes**
- **Positioning: Absolute, relative, fixed**

# Normal Flow

In the **normal flow**, elements are laid out from **top to bottom** in the order in which they appear in the source and from **left to right** (in left-to-right reading languages).

Blocks are laid out in the order in which they appear in the source.

Each block starts on a new line.



# Floating

## float

**Values:** left, right, none

Moves an element as far as possible to the left or right and allows the following content to wrap around it:

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

Inline image floated to the right

Image moves over, and text wraps around it

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.



## Floating (cont'd)

- Floated elements are removed from the normal flow but influence the surrounding content.
- Floated elements stay within the content area of the element that contains it.
- Margins are always maintained (they don't collapse) on all sides of floated elements.
- You must provide a width for a floated text element (because default width is `auto`).
- Elements don't float higher than their reference in the source.

# Clearing Floated Elements

## clear

**Values:** left, right, both, none

Prevents an element from appearing next to a floated element and forces it to start against the next available “clear” space

```
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."

(The h2 is “cleared” and starts below the floated element.)

# Floating Multiple Elements

- When you float multiple elements, browsers follow rules in the spec to ensure they don't overlap.
- Floated elements will be placed as far left or right (as specified) and as high up as space allows.

Elements floated to the same side line up.

If there is not enough room, subsequent elements move down and as far left as possible.

[PARAGRAPH 1] ONCE upon a time there lived in the village of Montignies-sur- Roc a little cow-boy, without either father or mother. His real name was Michael, but he was always called the Star Gazer, because when he drove his cows over the commons to seek for pasture, he went along with his head in the air, gaping at nothing.

[P2] As he had a white skin, blue eyes, and hair that curled all over his head, the village girls used to cry after him, 'Well, Star Gazer, what are you doing?' and Michael would answer, 'Oh, nothing,' and go on his way without even turning to look at them.

[P3] The fact was he thought them very ugly, with their sun-burnt necks, their great red hands, their coarse petticoats and their wooden shoes. He had heard that somewhere in the world there were girls whose necks were white and whose hands were small, who were always dressed in the finest silks and laces, and were called princesses.

[P4] One morning about the middle of August, just at mid-day when the sun was hottest, Michael ate his dinner of a piece of dry bread, and went to sleep under an oak. And while he slept he dreamt that there appeared before him a beautiful lady, dressed in a robe of cloth of gold, who said to him: 'Go to the castle of Belocil, and there you shall marry a princess.'

[P6] The following day, to the great astonishment of all the village, about two o'clock in the afternoon a voice was heard singing:

[P7] 'Raleo, raleo, How the cattle go!'

[P8] It was the little cow-boy driving his herd back to the byre.

[P9] The farmer began to scold him furiously, but he answered quietly, 'I am going away,' made his clothes into a bundle, said good-bye to all his friends, and boldly set out to seek his fortunes.

[P5] That evening the little cow-boy, who had been thinking a great deal about the advice of the lady in the golden dress, told his dream to the farm people. But, as was natural, they only laughed at the Star Gazer.

[P10] There was great excitement through all the village, and on the top of the hill the people stood holding their sides with laughing, as they watched the Star Gazer trudging bravely along the valley with his bundle at the end of his stick.

# CSS Shapes (Text Wrap)

## **shape-outside**

**Values:** `none`, `circle()`, `ellipse()`, `polygon()`, `url()`,  
`[margin-box | padding-box | content-box ]`

Changes the shape of the text wrap to a circle or ellipse, a complex path, or based on transparent areas in an image

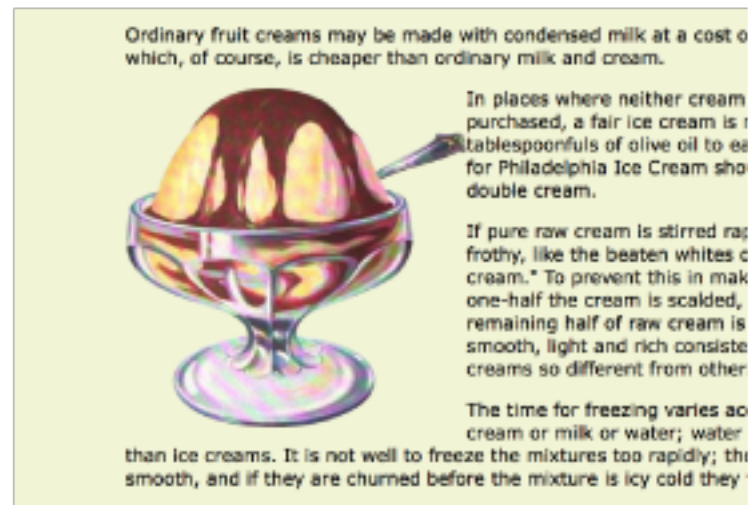
## **shape-margin**

**Values:** *length*, *percentage*

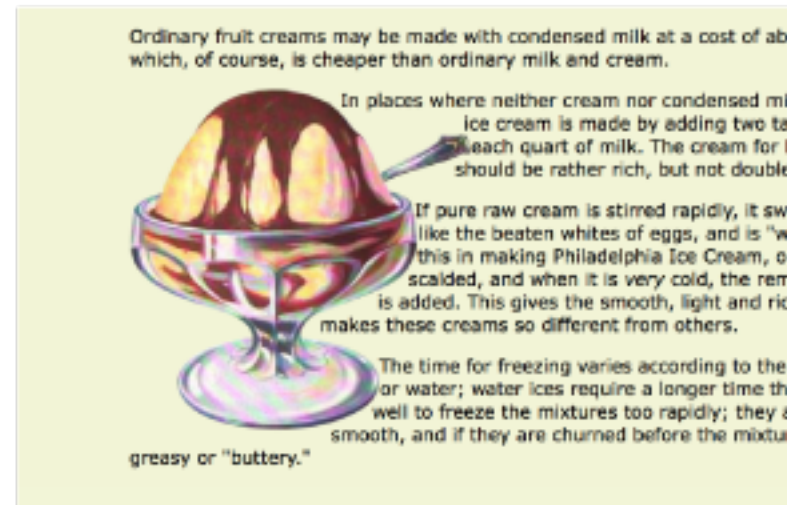
Specifies an amount of space to hold between the image and the wrapped text



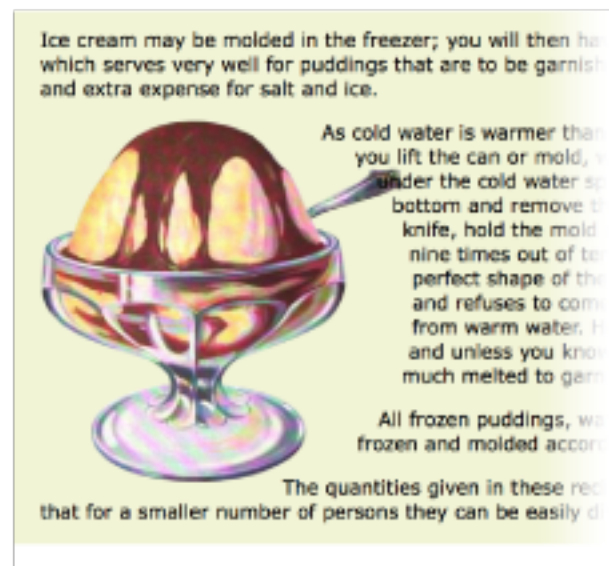
# CSS Shapes (cont'd)



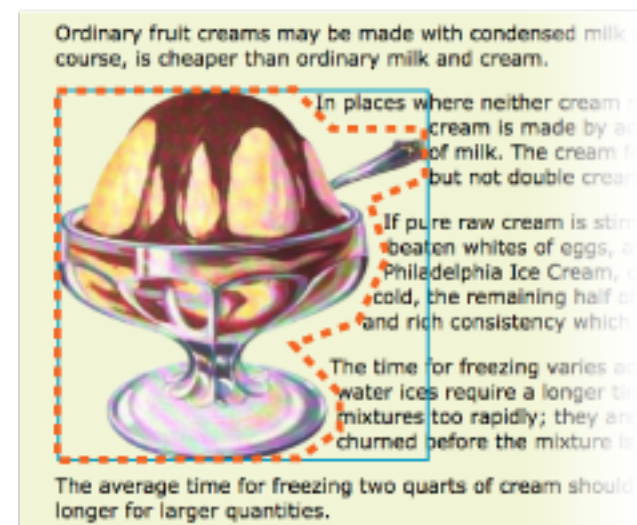
Default text wrap



Using the transparent areas of the image as a guide:  
**shape-outside: url(sundae.png);**



Using circle() notation:  
**shape-outside: circle(200px);**



The edges of the image (blue) and polygon path (dotted orange) revealed.

Using a path:  
**shape-outside: polygon(0px 0px, 186px 0px, 225px 34px, 300px 34px, 300px 66px, 255px 88px, 267px 127px, 246px 178px, 192px 211px, 226px 236px, 226px 273px, 209px 300px, 0px 300px);**

# Positioning

## `position`

**Values:** `static`, `relative`, `absolute`, `fixed`, `sticky`

Indicates that an element is to be positioned and specifies which positioning method to use

## `top`, `right`, `bottom`, `left`

**Values:** *length*, *percentage*, `auto`

Offset properties that provide the distance the element should be moved *away* from that respective edge

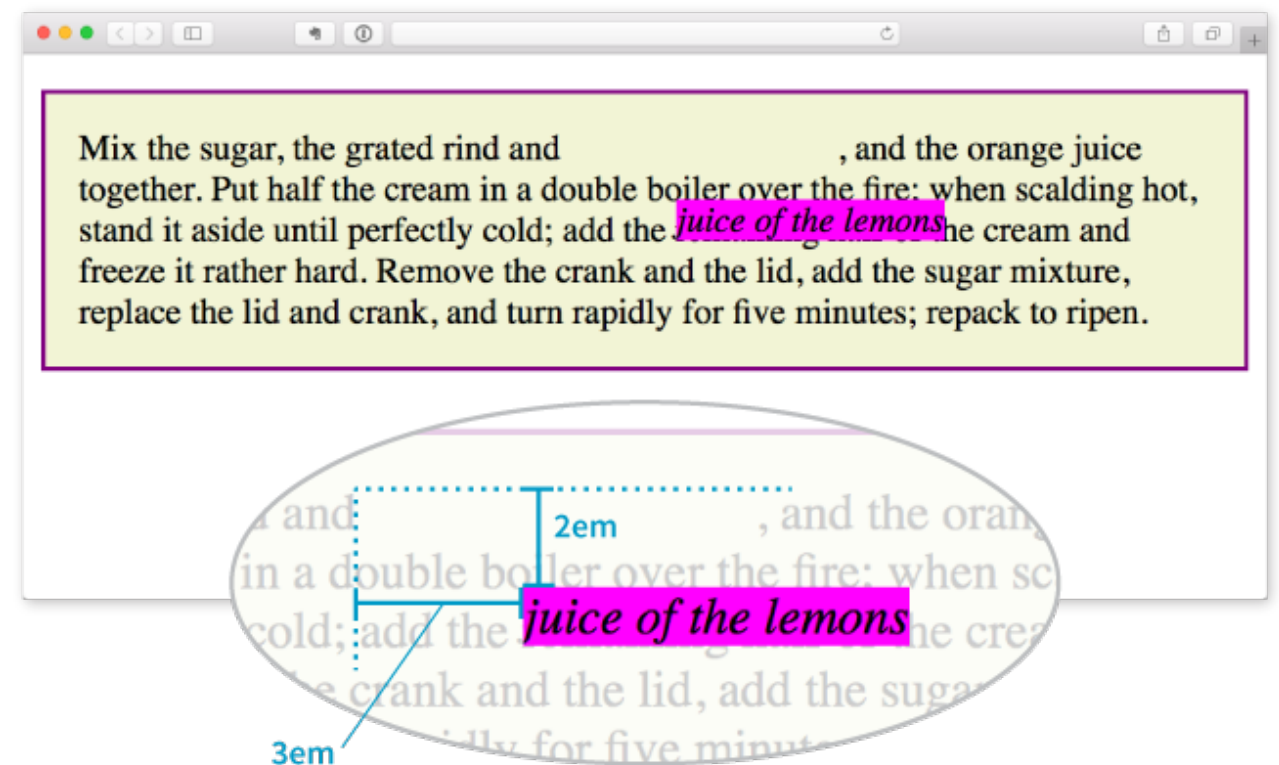
# Types of Positioning

- **Static:** The default position in the flow
- **Relative:** Moves the element relative to its original position
- **Absolute:** Removes the element from the flow and places it relative to the viewport or other containing element
- **Fixed:** Element stays in one position relative to the viewport
- **Sticky:** Element starts in the flow but stays fixed once it scrolls to a particular position relative to the viewport

# Relative Positioning

- Moves the element relative to its original position
- The space it originally occupied is preserved.

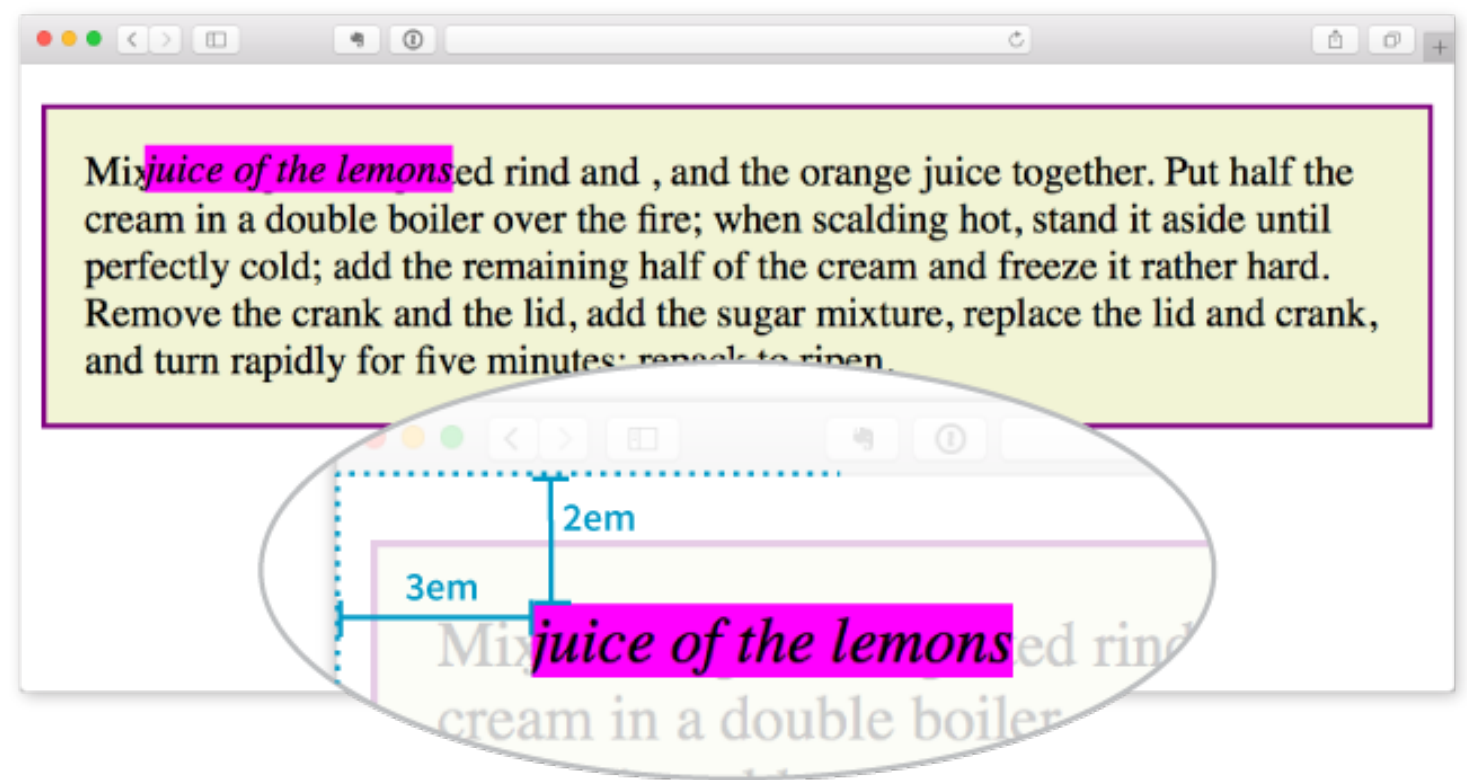
```
em {  
  position: relative;  
  top: 2em; /* moves it down */  
  left: 3em; /* moves it right */  
  background-color: fuchsia;  
}
```



# Absolute Positioning

- Moves the element relative to the **viewport** or **containing block** element
- The space it originally occupied is closed up.

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



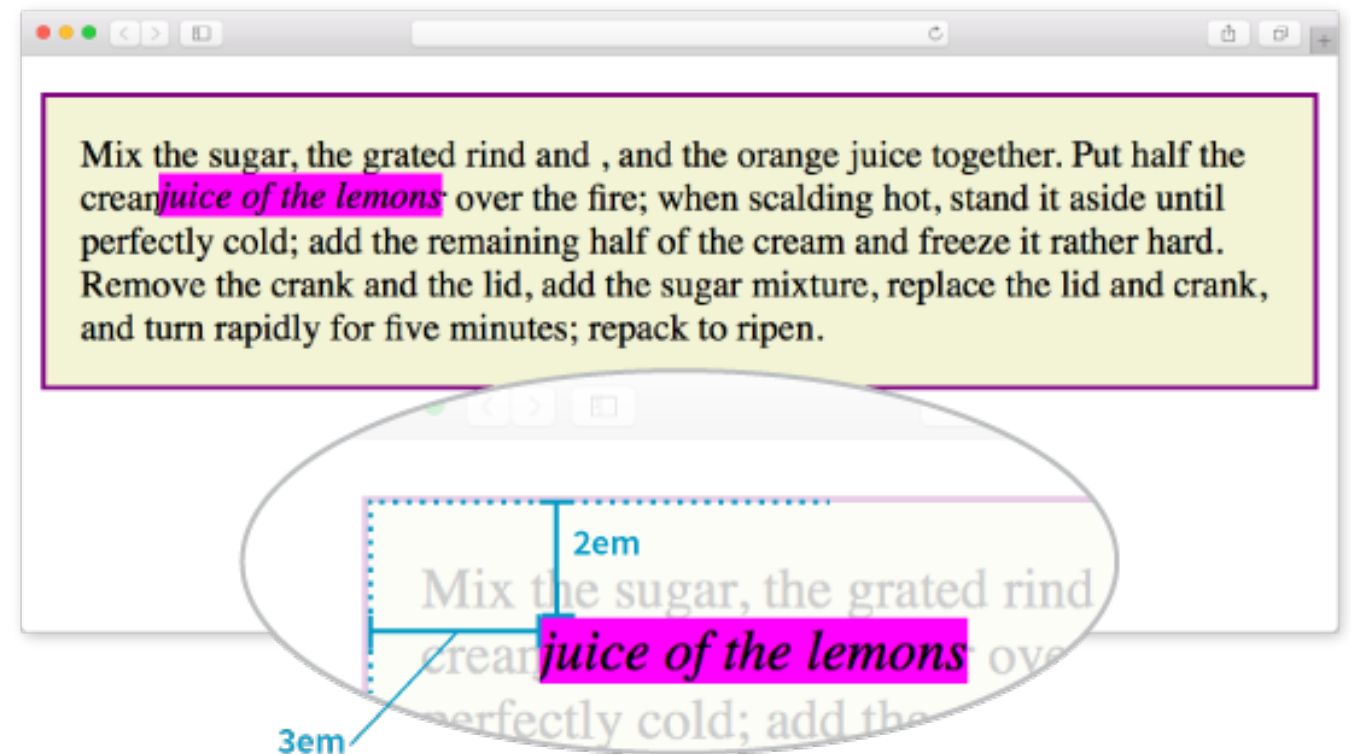
# Containing Blocks

- A positioned element serves as a **containing block** (or *positioning context*) for the elements it contains.
- If a positioned element has an ancestor that has its `position` set to `relative`, `absolute`, or `fixed`, then its position will be relative to that containing block element.
- If a positioned element is *not* contained within another positioned element, then it is placed relative to the initial containing block (the **html** element) and the viewport.



# Containing Blocks (cont'd.)

```
p {  
  position: relative;  
  padding: 15px;  
  background-color: #F2F5D5;  
  border: 2px solid purple;  
}  
em {  
  position: absolute;  
  top: 2em;  
  left: 3em;  
  background-color: fuchsia;  
}
```



The relatively positioned **p** element acts as a containing block for the **em** element.

# Specifying Position

- Position can be specified in length measurements (like pixels) or percentages.
- The measurement moves it *away* from the positioning offset property provided (i.e., **top: 200px;** moves the element *DOWN* from the top edge).
- Be careful not to overspecify. Two offset properties are usually enough.



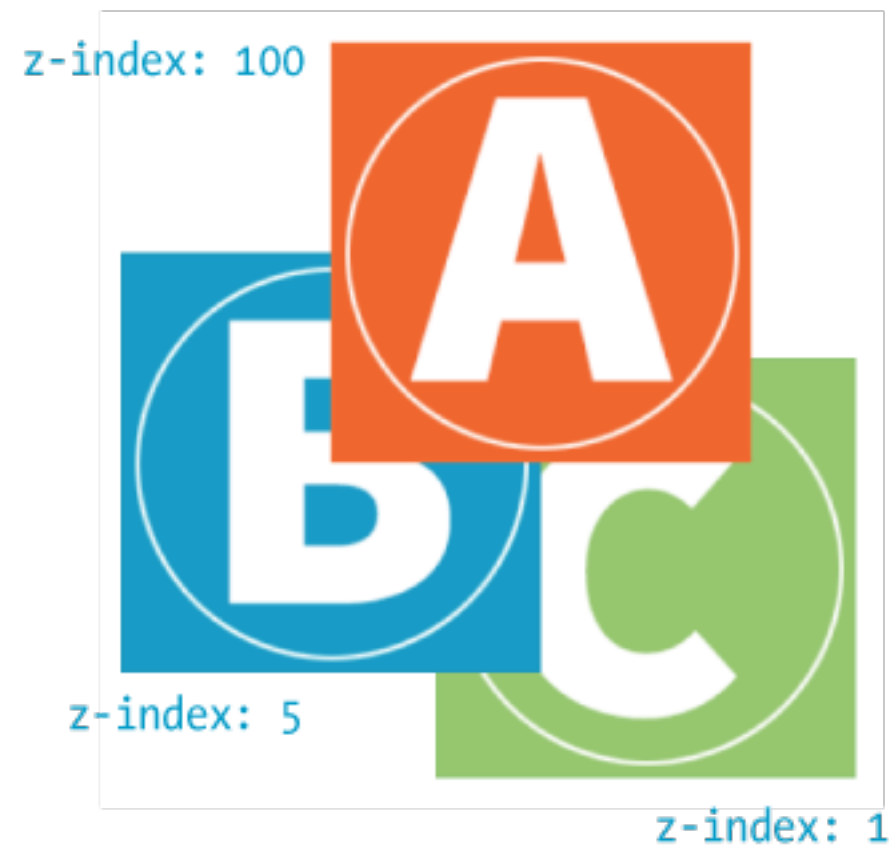
# Stacking Order

## z-index

**Values:** *Number*, *auto*



By default, elements later in the document source order stack on top of preceding elements.



You can change the stacking order with the z-index property. Higher values stack on top of lower values.