Floating and positioning

Web development I: Front-end engineering

Normal layout flow

Inline content reflows to fit the width of the block.

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

Each block starts on a new line.

a b c d e f g h i j k l m n o p q r s t u v w x y z
<p>

<h1>

<p>

<p>

a b c d e f g h i j k l
m n o p q r s t u v
w x y z
<p>

<h1>

<p>

<p>

Blocks fill the available width.
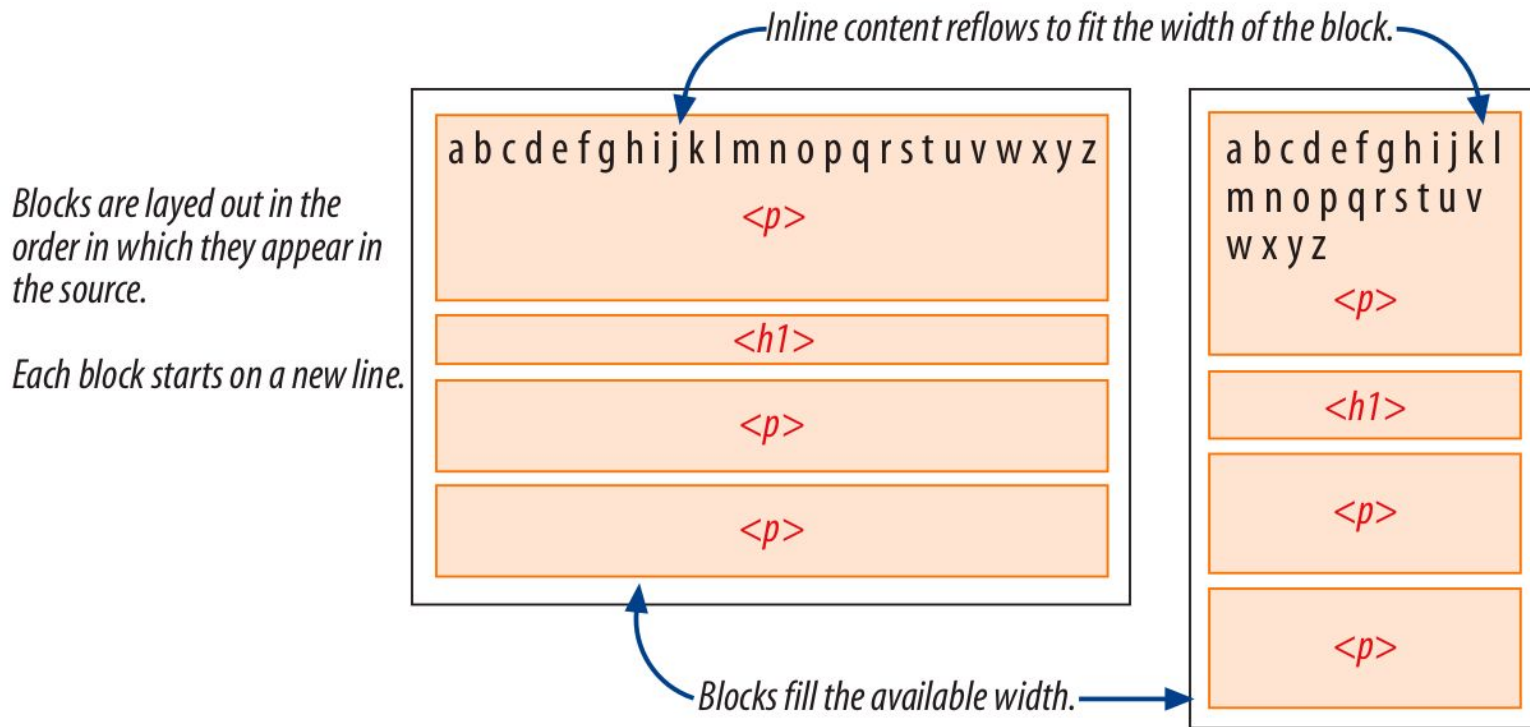
Floating

float: left | right | none | inherit

```
img {

  float: right;

}
```

A floated element is like an island in a stream.

Floats stay in the content area of the containing element.

Margins are maintained.

Inline image in the normal flow

space next to the image is held clear

They went down, down, down, till at last they came to a passage with a door at one end, which was only fastened with a latch. The eldest Princess opened it, and they found themselves immediately in a lovely little wood, where the leaves were spangled with drops of silver which shone in the brilliant light of the moon. They next crossed another wood where the leaves were sprinkled with gold, and after that another still, where the leaves glittered with diamonds.

Inline image floated to the right.

They went down, down, down, till at last they came to a passage with a door at one end, which was only fastened with a latch. The eldest Princess opened it, and they found themselves immediately in a lovely little wood, where the leaves were spangled with drops of silver which shone in the brilliant light of the moon. They next crossed another wood where the leaves were sprinkled with gold, and after that another still, where the leaves glittered with diamonds.

image moves over and text wraps around it

Floating notes

Inline elements:

- Provide a width for floated text elements.
- Floated inline elements behave as block elements.
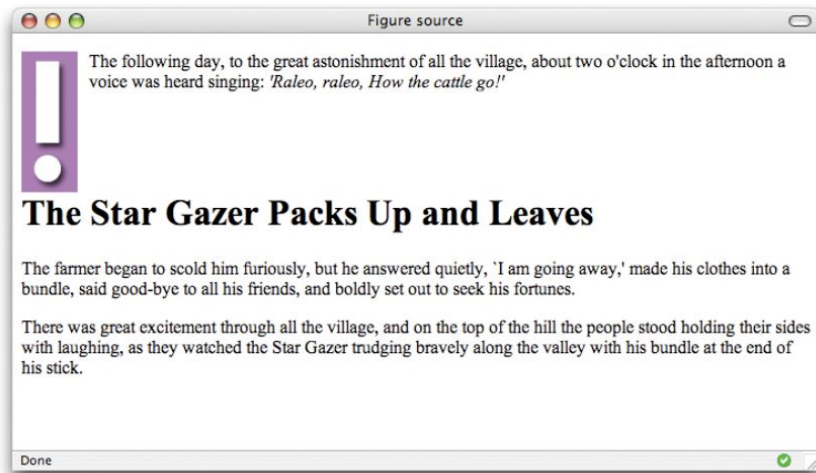- Margins on floated elements do not collapse.

Block elements:

- Provide a width for floated block elements.
- Elements do not float higher than their reference in the source.

Clearing floats

clear: left | right | both | none | inherit

```
img {

  float: left;

}

h2 {

  clear: left;

}
```
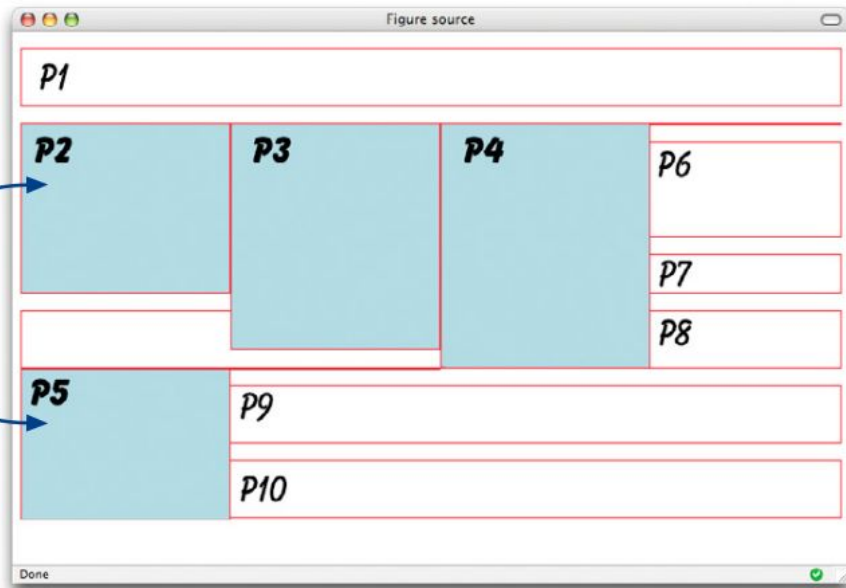
Floating multiple elements

```
<p>P1</p>

<p class="float">P2</p>

<p class="float">P3</p>

<p class="float">P4</p>

<p class="float">P5</p>

<p>P6</p>

<p>P7</p>

<p>P8</p>

<p>P9</p>

<p>P10</p>
```


Elements floated to the same side line up. If there is not enough room, subsequent elements move down.

Containing floats

```
#container {

  float: left;

}
```

```
#container {

  overflow: auto;

}
```


Hand-stitched iPhone motif
More info

Hand-stitched iPhone motif
More info

Positioning

position: static | relative | absolute | fixed | inherit

Also **top**, **right**, **bottom**, and **left** properties: length measurement | percentage | auto | inherit
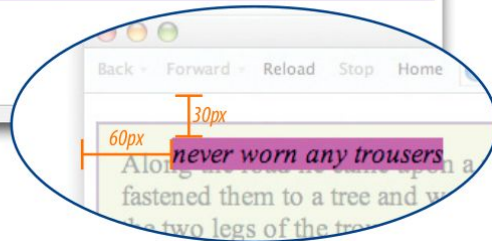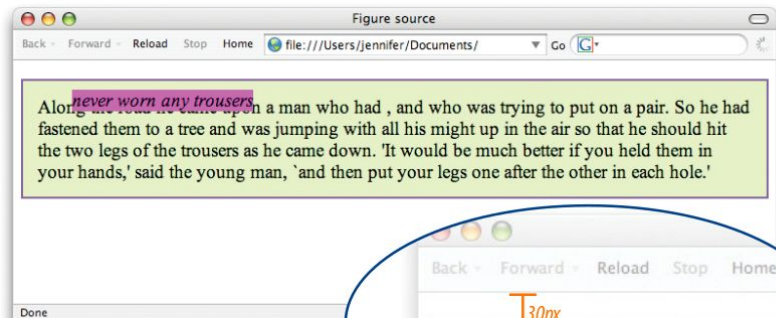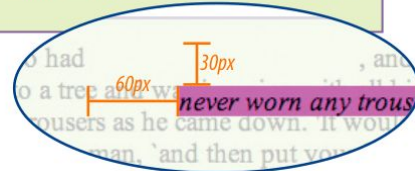
Positioning

```css
div#a {

  position: relative;

  height: 120px;

  width: 300px;

}
div#b {

  position: absolute;

  top: 20px;

  right: 30px;

  bottom: 40px;

  left: 50px;

}
```

div#a (*width: 300px; height: 120px;*)

top: 20px;

left: 50px;

div#b
(*calculated at 220 pixels wide x 60 pixels high*)

right: 30px;

bottom: 40px;
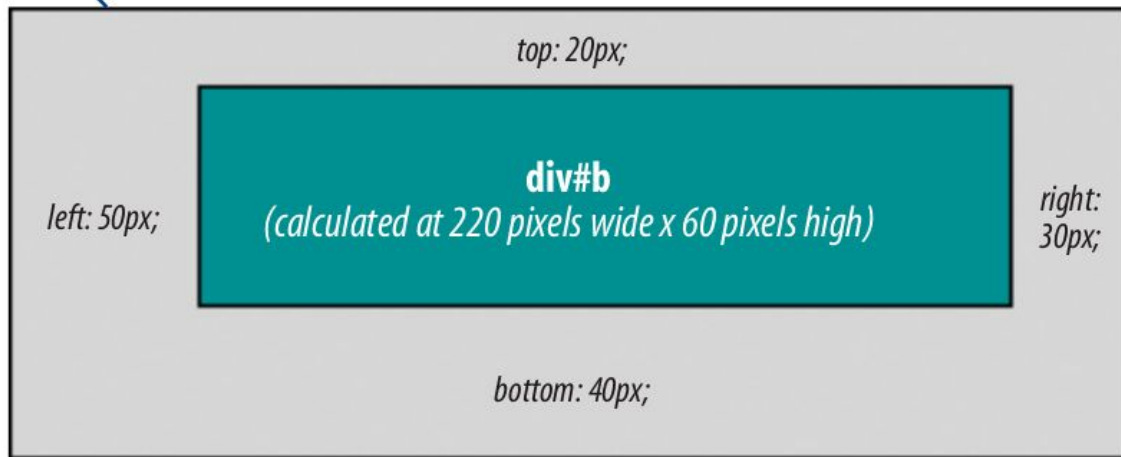
Fixed positioning

Relative to the viewport.

Even when scrolling!

Stacking order

z-index: number | auto | inherit

Painter's Algorithm:

sort *polygons* by *depth*

for each *polygon p*:

    for each *pixel* that *p* covers:

        paint *p.color* on *pixel*

z-index: 10;
z-index: 1;
z-index: 5;

By default, elements later in the document stack on top of preceding elements.
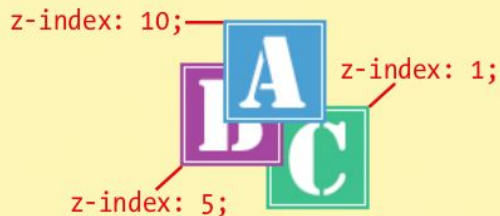
You can change the stacking order with the z-index property. Higher values stack on top of lower values.