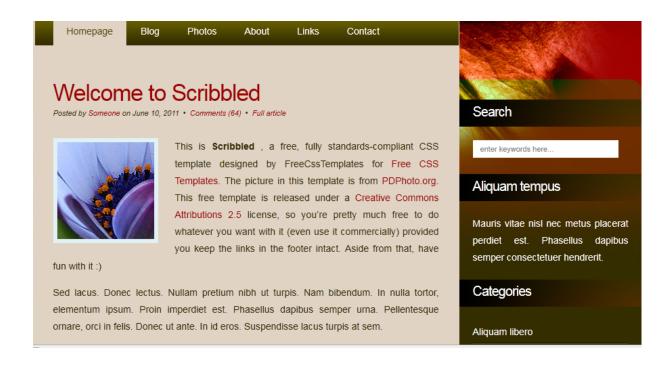
PAGE LAYOUT WITH CSS

- Styling Page Sections
- Introduction to Layout
- Floating Elements
- Sizing & Positioning

Styling Page Sections

Why do we need page sections?

- Style individual elements, groups of elements, sections of text or of the page
- Create complex page layouts



Sections of a page <div>

```
<div class="shout">
<h2>Coding Horror! Coding Horror!</h2>
See our special deal on Droids!
We'll beat any advertised price!
</div>

HTML
```

Coding Horror! Coding Horror!

See our special deal on Droids!

We'll beat any advertised price!

- □ Tag used to indicate a logical section or area of a page
- Has no appearance by default, but you can apply styles to it

Inline Sections

```
<h2>Coding Horror! Coding Horror!</h2>
See our <span class="special">spectacular</span> deal
on Droids!
We'll beat <span class="shout"> any advertised price</
span>!
HTML
```

Coding Horror! Coding Horror!

See our spectacular deal on Droids!

We'll beat any advertised price!

output

has no onscreen appearance, but you can apply a style or ID to it, which will be applied to the text inside the span

CSS context selectors

```
selector1 selector2 {
properties
}
```

applies the given properties to selector2 only if it is inside a selector1 on the page

```
selector1 > selector2 {
properties
}
```

applies the given properties to selector2 only if it is directly inside a selector1 on the page

Context selector example

```
Eat at <strong>Greasy's Burger</strong>...

The <strong>greasiest</strong> burgers in town!
Yummy and greasy at the same time!
```

HTML

Eat at Greasy's Burger...

- The **greasiest** burgers in town!
- Yummy and greasy at the same time!

More complex example

```
<div id="ad">
Eat at <strong>Greasy's Burger</strong>...

cli class="important">The <strong>greasiest</strong> burgers in town!
Yummy and <strong>greasy at the same time </strong>!

</div>
```

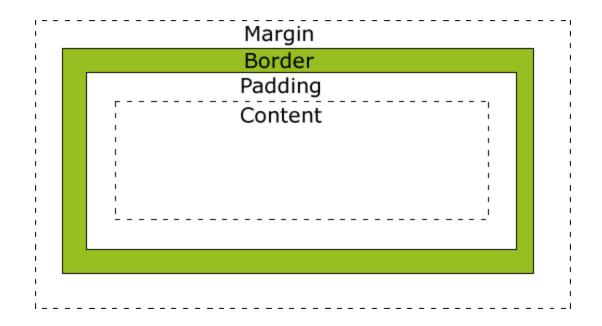
Eat at Greasy's Burger...

- The **greasiest** burgers in town!
- Yummy and greasy at the same time!

Introduction to Layout

The CSS Box Model

- Every element composed of:
 - content
 - a border around the element
 - padding between the content and the border
 - a margin between the border and other content

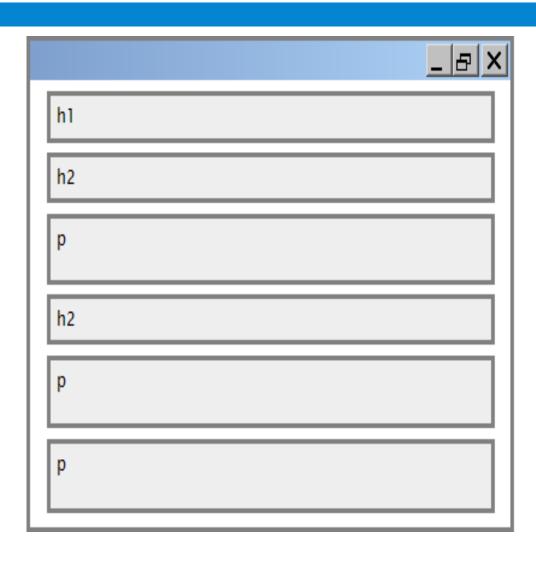


The CSS Box Model (cont.)

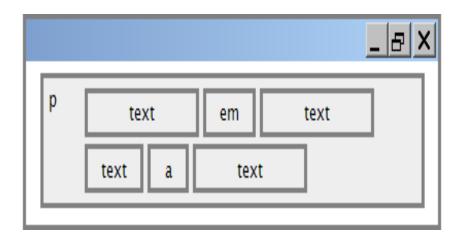
width = content width + L/R padding + L/R border + L/R margin height = content height + T/B padding + T/B border + T/B margin



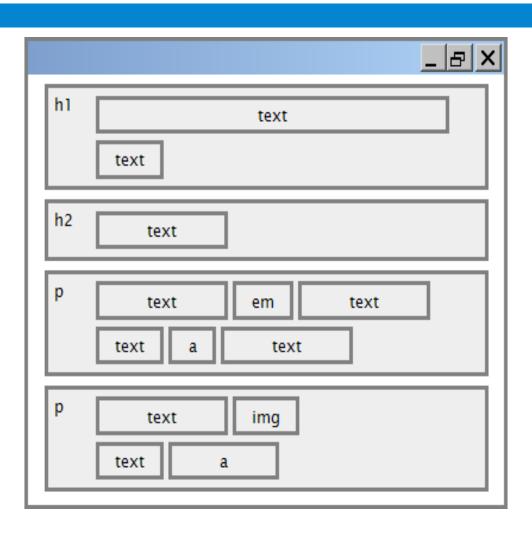
Document Flow – block elements



Document flow - inline elements



Document flow - a larger example



CSS properties for borders

```
h2 { border: 5px solid red; }

CSS
```

This is a heading.

property	description
border	thickness/style/size of border on all 4 sides

- □ Thickness: px, pt, em, or thin, medium, thick
- □ Style: none, hidden, dotted, dashed, double, groove, inset, outset, ridge, solid
- color

More border properties

property	description
border-color, border-width, border-style	specific properties of border on all 4 sides
border-bottom, border-left, border-right, border-top	all properties of border on a particular side
border-bottom-color, border-bottom- style, border-bottom-width, border-left- color, border-left-style, border-left-width, border-right-color, border-right-style, border-right-width, border-top-color, border-top-style, border-top-width	properties of border on a particular side
Complete list of border properties http://www.w3schools.com/css/css border.asp	

Another border example

```
h2 {
border-left: thick dotted #CC0088;
border-bottom-color: rgb(0, 128, 128);
border-bottom-style: double;
}

CSS
```

This is a heading.

- each side's border properties can be set individually
- if you omit some properties, they receive default

CSS properties for padding

property	description
padding	padding on all 4 sides
padding-bottom	padding on bottom side only
padding-left	padding on left side only
padding-right	padding on right side only
padding-top	padding on top side only
Complete list of padding properties	
http://www.w3schools.com/css/css_padding.asp	

Padding example 1

```
p { padding: 20px; border: 3px solid black; }
h2 { padding: 0px; background-color: yellow; }
```

This is a first paragraph.

This is a second paragraph.

This is a heading

Padding example 2

```
p {
padding-left: 200px; padding-top: 30px;
background-color: fuchsia;
}
```

This is a first paragraph

This is a second paragraph

- each side's padding can be set individually
- notice that padding shares the background color of the element

CSS properties for margins

property	description
margin	margin on all 4 sides
margin-bottom	margin on bottom side only
margin-left	margin on left side only
margin-right	margin on right side only
margin-top	margin on top side only

Complete list of margin properties

http://www.w3schools.com/css/css_margin.asp

Margin example 1

```
p {
margin: 50px;
background-color: fuchsia;
}

This is a first paragraph

This is a second paragraph

output
```

notice that margins are always transparent

Margin example 2

```
p {
margin-left: 8em;
background-color: fuchsia;
}

This is a first paragraph

This is a second paragraph

output
```

each side's margin can be set individually

CSS properties for dimensions

```
p { width: 350px; background-color: yellow; }
h2 { width: 50%; background-color: aqua; }

CSS
```

This paragraph uses the first style above

An h2 heading

property	description
width, height	how wide or tall to make this element (block elements only)
max-width, max-height, min-width, min-height	max/min size of this element in given dimension

Floating Elements

The CSS float property

```
img.headericon {
float: right; width: 130px;
}
```

Ghostbusters is a 1984 American science fiction comedy film written by co-stars Dan Aykroyd and Harold Ramis about three eccentric New York City parapsychologists-turned-ghost capturers.

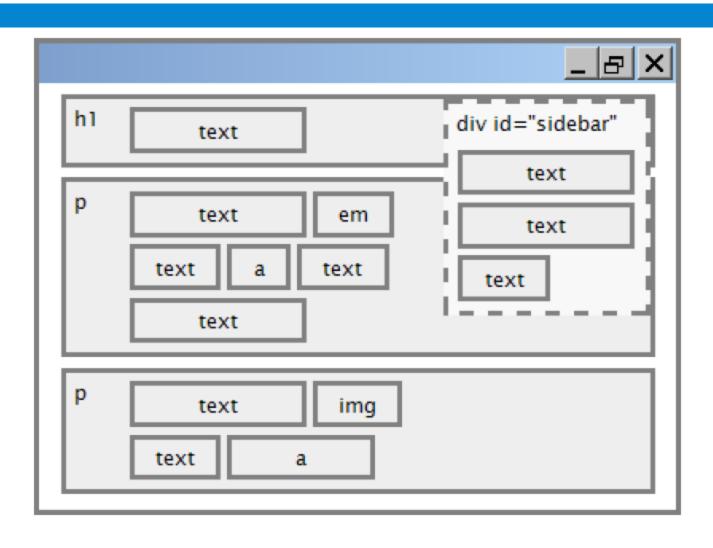


output

property	description
float	side to hover on; can be left, right, or none (default)

removed from normal document flow; underlying text wraps around as necessary

Floating elements diagram



Common float bug: missing width

I am not floating, no width

I am floating right, no width

I am not floating, 45% width

I am floating right, 45% width

 often floating block elements must have a width property value

The clear property

```
p { background-color: fuchsia; }
h2 { clear: right; background-color: yellow; }
```

Mario is a fictional character in his video game series. Serving as Nintendo's mascot and the main protagonist of the series, Mario has appeared in over 200 video games since his creation



Super Mario Fan Site!

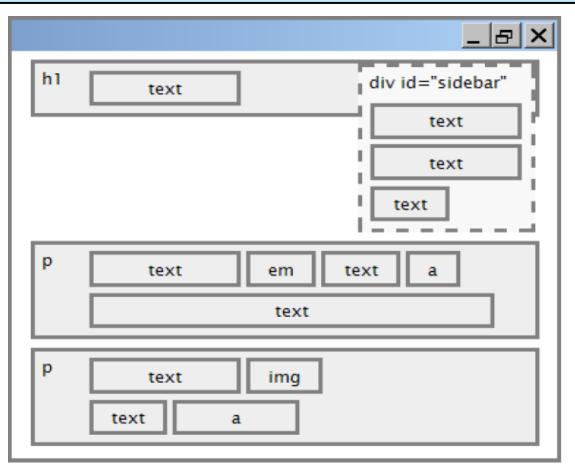
The clear property (cont.)

property	description
clear	disallows floating elements from overlapping this element; can be left, right, or none
	(default)

Clear diagram

```
div#sidebar { float: right; }
p { clear: right; }

CSS
```



Common error: container too short

```
<img src="images/mario.png" alt="super mario" />
Mario is a fictional character in his video game
series.....
HTML
```

Mario is a fictional character in his video game series. Serving as Nintendo's mascot and the main protagonist of the series, Mario has appeared in over 200 video games since his creation.



output

The overflow property

```
p { border: 2px dashed black; overflow: hidden; }
```

Mario is a fictional character in his video game series. Serving as Nintendo's mascot and the main protagonist of the series, Mario has appeared in over 200 video games since his creation.



The overflow property (cont.)

property	description
overflow	specifies what to do if an element's content is too large; can be auto, visible, hidden, or scroll

Multi-column layouts

Some other text that is important

third paragraph second paragraph first paragraph

Sizing and Positioning

The position property (examples)

```
div#ad {
    position: fixed;
    right: 10%;
    top: 45%;
}
```

property	value	description
position	static	default position
	relative	offset from its normal static position
	absolute	a fixed position within its containing element
	fixed	a fixed position within the browser window
top, bottom, left, right	positions of box's corners	

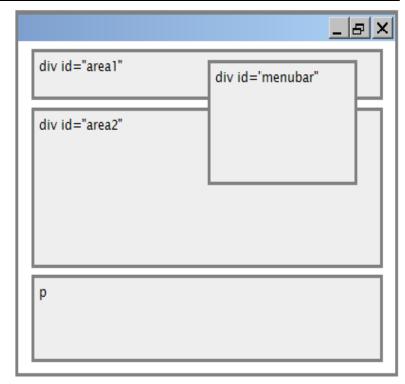
Absolute positioning

```
#menubar {
position: absolute;
left: 400px;
top: 50px;
}
```

- removed from normal flow
- positioned relative to the blockelement containing them
- actual position determined by

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

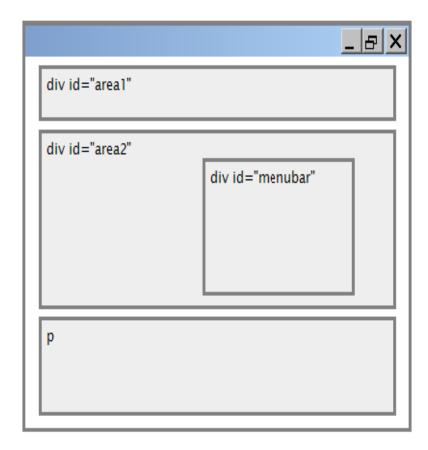
should often specify a width property as well



Relative positioning

```
#area2 { position: relative; }
CSS
```

- absolute-positioned elements are normally positioned at an offset from the corner of the overall web page
- to make the absolute element to position itself relative to some other element's corner, wrap the absolute element in an element whose position is relative

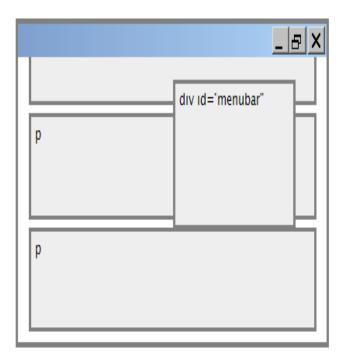


Fixed positioning

```
#menubar {
position: fixed;
left: 400px;
top: 50px;
}

CSS
```

- removed from normal flow
- positioned relative to the browser window even when the user scrolls the window, element will remain in the same place



Alignment vs. float vs. position

- □ If possible, lay out an element by *aligning* its content
 - horizontal alignment: text-align
 - vertical alignment: vertical-align
- If alignment won't work, try floating the element
- If floating won't work, try positioning the element
 - absolute/fixed positioning are a last resort and should not be overused

Details about inline elements

- Size properties (width, height, min-width, etc.) are ignored
- margin-top and margin-bottom are ignored,
- but margin-left and margin-right are not
 ignored

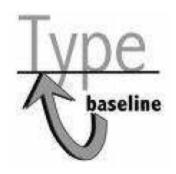
Details about inline elements

- the containing block element's text-align property controls horizontal position of inline elements within it
 - text-align does not align block elements within the page
- ach inline element's vertical-align property
 aligns it vertically within its block element

The vertical-align property

property	description
vertical-align	specifies where an inline element should be aligned vertically, with respect to other content on the same line within its block element's box

can be top, middle, bottom, baseline (default), sub, super, text-top, text-bottom, or a length value or % (baseline means aligned with bottom of non-hanging letters)



The display property

```
h2 { display: inline; background-color: yellow; }
CSS
```

This is a heading This is another heading

output

property	description
display	sets the type of CSS box model an element is displayed with

- values: none, inline, block, run-in, compact,...
- use sparingly, because it can radically alter the page layout

The display property

```
#topmenu li {
display: inline;
border: 2px solid gray;
margin-right: 1em;
}
```



- lists and other block elements can be displayed inline
- flow left-to-right on same line
- width is determined by content

The visibility property

```
p.secret {
    visibility: hidden;
}
```

output

- hidden elements will still take up space onscreen, but will not be shown
 - to make it not take up any space, set display to none instead
- can be used to show/hide dynamic HTML content on the page in response to events