

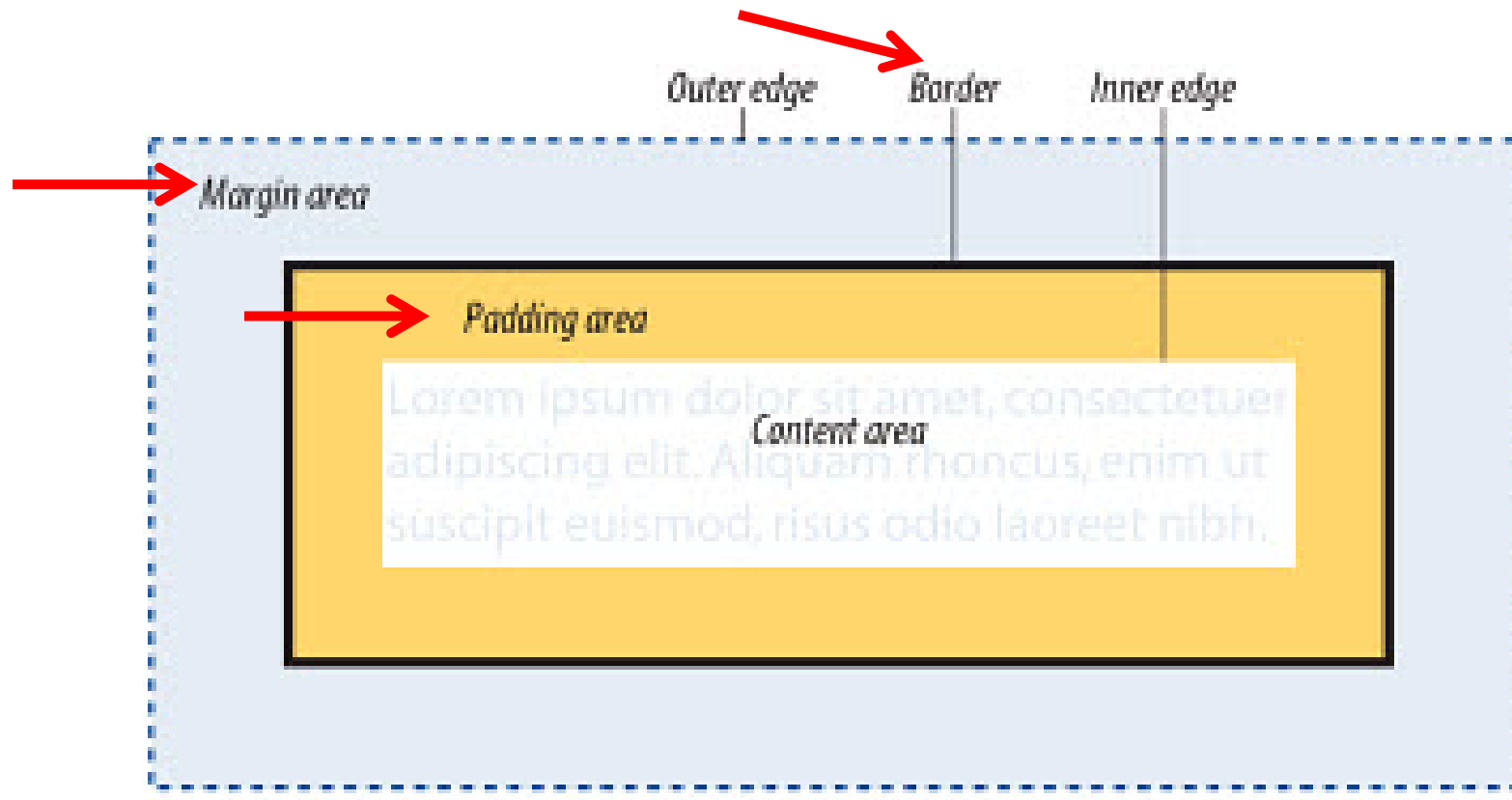
WEBD 162

Class 09

Chapter 14 - Box Model – Padding, Borders, Margins

Box model

CSS properties: margin – padding - border



Box dimensions – width / height

▶ **Width - default**

- ▶ **Block** level element – as wide as the **container**
- 100% unless specified
- ▶ **Inline** element – as wide as the **content**

▶ **Height - default**

- ▶ **Block** level element – as high as the content
- ▶ **Inline** element – as high as the content

Box dimensions – default width / height

```
<style>
```

```
  em {background-color: yellow;}
```

```
  p {background-color: aqua;}
```

```
</style>
```

```
<p><em>this is text here</em></p>
```



this is text here

Box dimensions – Setting width / height

▶ **Width**

- ▶ **Block** level element – **can set**
- ▶ **Inline** element – **cannot set**

▶ **Height**

- ▶ **Block** level element – **can set**
 - usually no need to set height
- ▶ **Inline** element – **cannot set**

Box dimensions – Setting width / height

Calculations

- **padding** and **border** are **added** to the width



```
<style>
```

```
  em {background-color: yellow;}
```

```
  p {background-color: aqua; width: 200px; padding: 10px;  
    border: 2px solid red;}
```

```
</style>
```

```
<p><em>this is text here</em></p>
```

What is real width?



CSS3 box-sizing Property – pg 307

box-sizing

– specifies if the width / height should include the border and padding (the default)

- ▶ **box-sizing: content-box;**

- ▶ width/ height **does not include** the border and padding

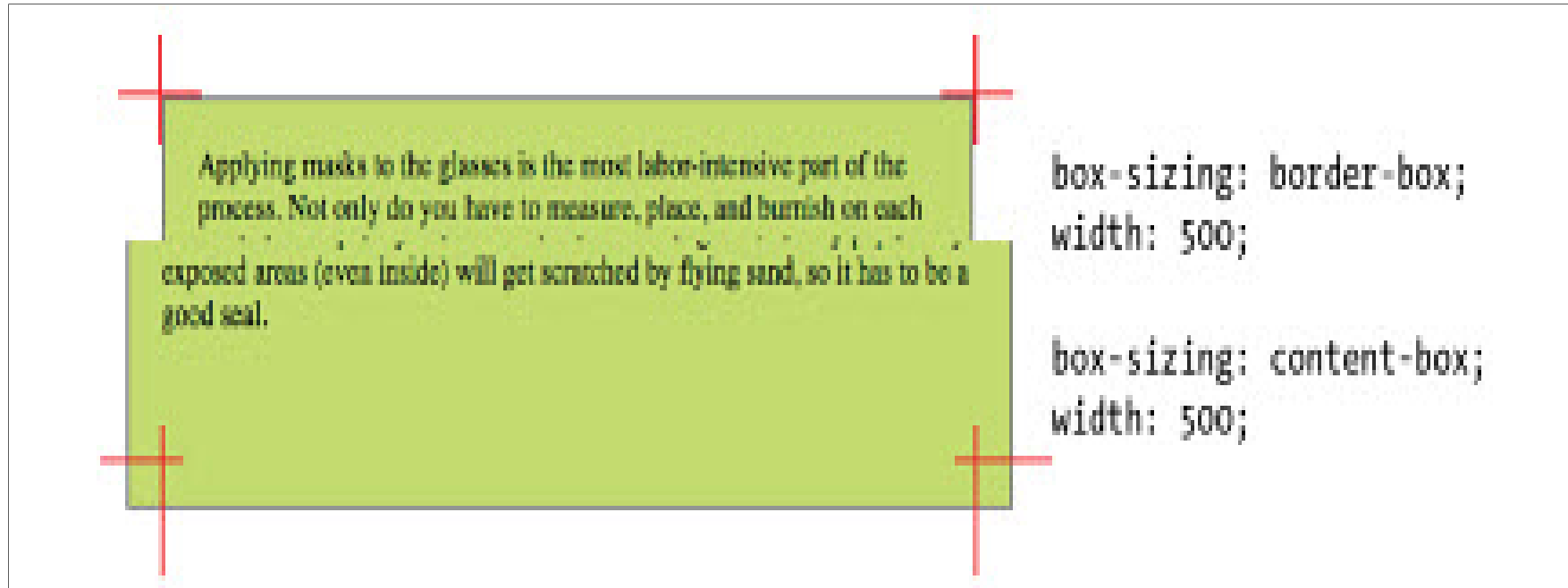
- ▶ **box-sizing: border-box;**

- ▶ width/ height **includes** the border and padding



- ▶ **margin** is on the outside and is never included

CSS3 box-sizing Property – pg 309



border-box – retains specified width

content-box – padding /border added to width

CSS3 box-sizing Property – pg 309

```
p {  
    -webkit-box-sizing: border-box;  
    -moz-box-sizing: border-box;  
    box-sizing: border-box;  
    width: 500px;  
    height: 150px;  
}
```

need Vendor / Browser prefixes
use Progressive Enhancement

max/min width/height – pg 309

Maximum and Minimum Dimensions

CSS2 introduced properties for setting minimum and maximum heights and widths for block elements. They may be useful if you want to put limits on the size of an element.

**max-height, max-width,
min-height, min-width**

Values: *percentage* | *length* | *none* | *inherit*

These properties work with block-level and replaced elements (like images) only.

When the content-box model is used, the value applies to the content area only, so if you apply padding, borders, or margins, it will make the overall element box larger, even if a max-width or max-height property has been specified. These properties are not supported by Internet Explorer 6 and earlier.

CSS2 for large screen browsers

CSS overflow Property – pg 312

- when content that exceeds the set width / height
- **by default a container is as big as it needs to be**

- ▶ **overflow: visible;**
 - ▶ default – shows all content
- ▶ **overflow: hidden;**
 - ▶ cuts off the content
- ▶ **overflow: scroll;**
 - ▶ adds horizontal and vertical scroll bars regardless
- ▶ **overflow: auto;**
 - ▶ adds horizontal / vertical scroll if needed



CSS overflow Property – pg 312

visible

Applying the masks to the glasses is the most labor-intensive part of the process. Not only do you have to measure, place, and burnish on each mask, but you also need to completely cover the remainder of the glass in heavy paper. Any exposed areas (even inside) will get scratched by the flying sand, so it has to be a good seal.

hidden

Applying the masks to the glasses is the most labor-intensive part of the process. Not only do you have to measure, place, and burnish on each mask, but you also need to completely

scroll

Applying the masks to the glasses is the most labor-intensive part of the process. Not only do you have to measure, place, and burnish on each mask, but you also

auto (short text)

Applying the masks to the glasses is the most labor-intensive part of the process.

auto (long text)

Applying the masks to the glasses is the most labor-intensive part of the process. Not only do you have to measure, place, and burnish on each mask, but you also

CSS padding property – pg 313

Adds space around the content / inside the container

Multiple sides / shorthand

Individual sides

- ▶ `padding-top: value;`
- ▶ `padding-bottom: value;`
- ▶ `padding-right: value;`
- ▶ `padding-left: value;`

1 value

`padding: 10px;`

Applied to all sides.

2 values

`padding: 10px 6px;`

First is top and bottom;
second is left and right.

3 values

`padding: 10px 6px 4px;`

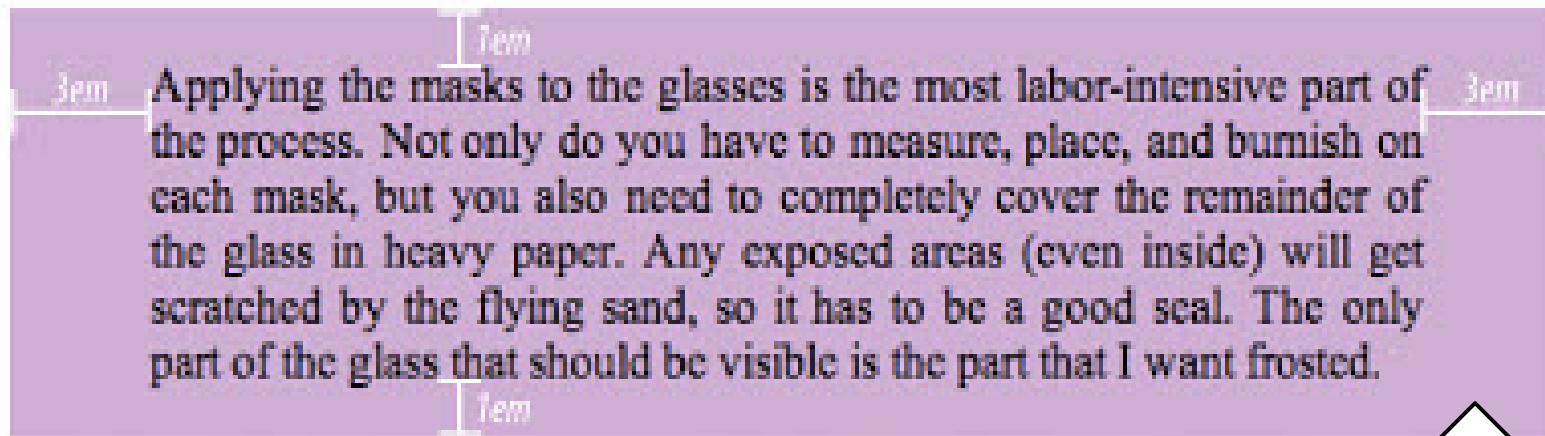
First is top;
second is left and right;
third is bottom.

4 values

`padding: 10px 6px 4px 10px;`

Applied clockwise to top, right,
bottom, and left edges consecutively

CSS padding property – pg 313



```
blockquote {  
  padding-top: 1em;  
  padding-right: 3em;  
  padding-bottom: 1em;  
  padding-left: 3em;  
  background-color: #D098D4;  
}
```

Hard to see without a
background color

CSS border property – pg 316

Line drawn around the content container

Three components of a border

- ▶ Border style
- ▶ Border width (thickness)
- ▶ Border color
- ▶ Border shorthand

CSS border-style property – pg 317

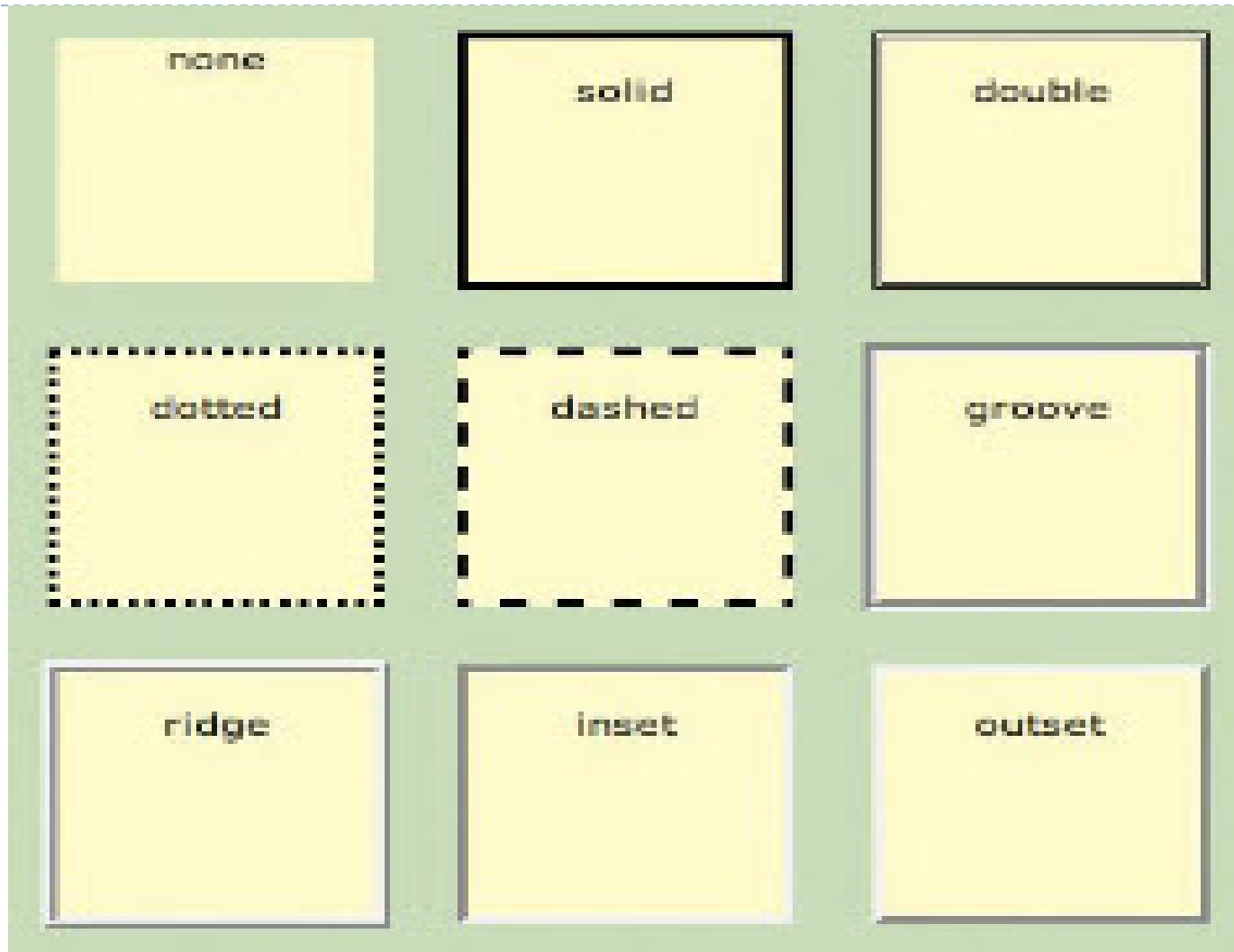
Individual sides

- ▶ border-top-style: value;
- ▶ border-bottom-style: value;
- ▶ border-right-style: value;
- ▶ border-left-style: value;

Multiple sides / shorthand

- ▶ border-style: value; (all 4 sides)
- ▶ border-style: value value; (top / bottom)
- ▶ border-style: value value value; (top / rl / bottom)
- ▶ border-style: value value value value; (t / r / b / l = clockwise)

CSS border styles



CSS border-width property – pg 318

Individual sides

- ▶ border-top-width: value;
- ▶ border-bottom-width: value;
- ▶ border-right-width: value;
- ▶ border-left-width: value;



Values:

- Unit measure
- Thin
- Medium
- Thick

Multiple sides / shorthand

- ▶ border-width: value; (all 4 sides)
- ▶ border-width: value value; (top / bottom)
- ▶ border-width: value value value; (top / rl / bottom)
- ▶ border-width: value value value value; (t / r / b / l = clockwise)

CSS border-width

```
div#help {  
  border-width: thin medium thick 12px;  
  border-style: solid;  
  width: 300px;  
  height: 100px;  
}
```



CSS border-color property – pg 320

Individual sides

- ▶ border-top-color: value;
- ▶ border-bottom-color: value;
- ▶ border-right-color: value;
- ▶ border-left-color: value;



Values:

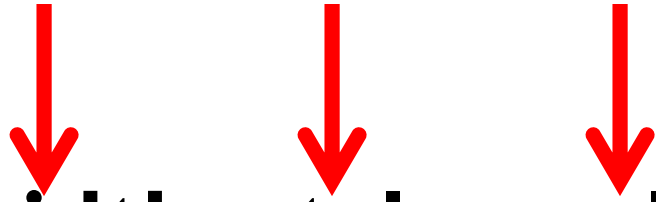
- name
- hexadecimal
- rgb
- etc

Multiple sides / shorthand

- ▶ border-color: value; (all 4 sides)
- ▶ border-color: value value; (top / bottom)
- ▶ border-color: value value value; (top / rl / bottom)
- ▶ border-color: value value value value; (t / r / b / l = clockwise)

CSS border shorthand

border: value value value;


width style color

- ▶ Sets all 4 side the same
- ▶ Values can go in any order

Use background-color / border to see where content and container are

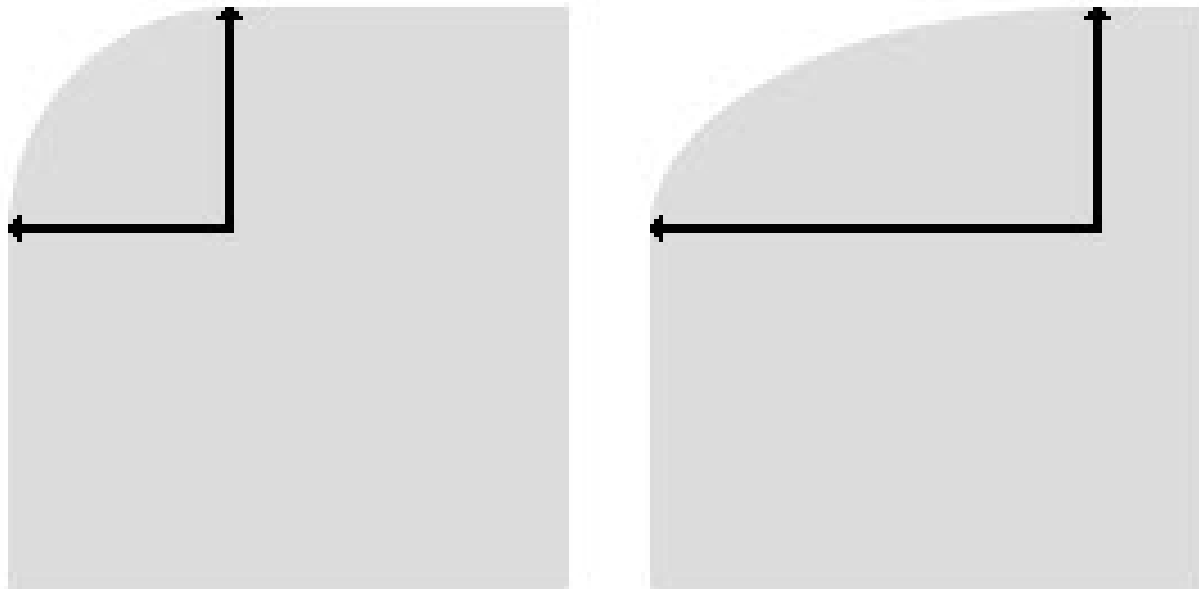


CSS3 border-radius – pg 322

Multiple sides / shorthand

border-radius: value; 4 sides same

border-radius: value / value; 4 sides same but skewed



CSS3 border-radius – pg 322

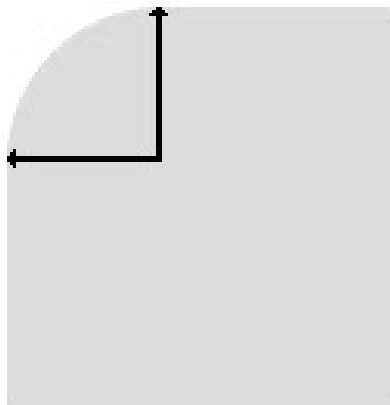
Individual sides

`border-top-right-radius: value;`

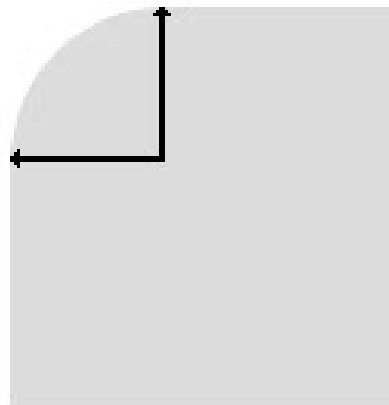
`border-top-left-radius: value;`

`border-bottom-right-radius: value;`

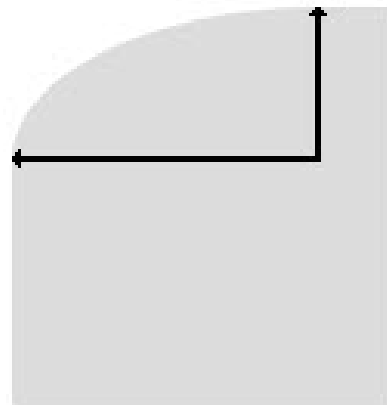
`border-bottom-left-radius: value;`



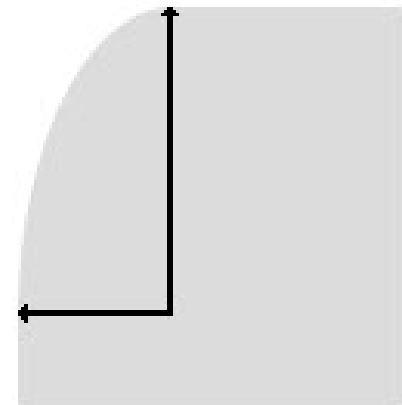
`border-top-left-radius:
50px;`



`border-top-left-radius:
50px 50px;`



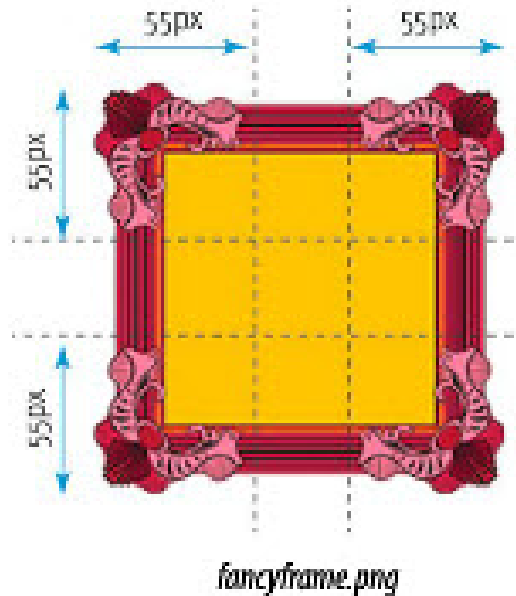
`border-top-left-radius:
100px 50px;`



`border-top-left-radius:
50px 100px;`

CSS3 border-image – pg 324

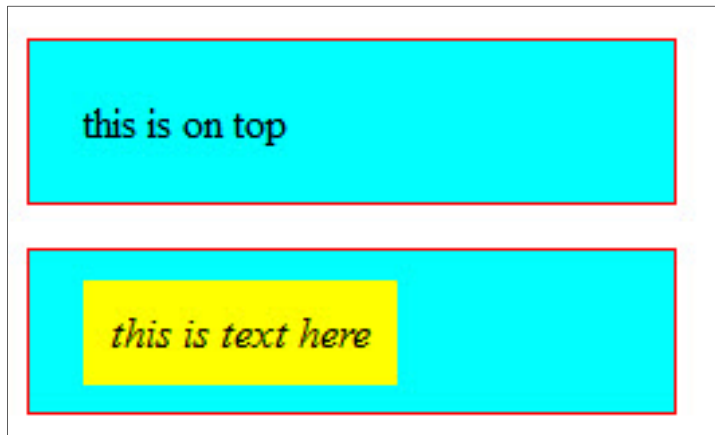
border-image: url(fancyframe.png) 55 55 55 55 stretch ;



CSS margin property – pg 328

The space between the block elements / containers

- ▶ You cannot set a top or bottom margin for an inline element but you can set padding and border



p, em { margin: 50px; }



CSS margin property – pg 328

Individual sides

- ▶ `margin-top: value;`
- ▶ `margin-bottom: value;`
- ▶ `margin-right: value;`
- ▶ `margin-left: value;`



Values:

- unit
- **auto**

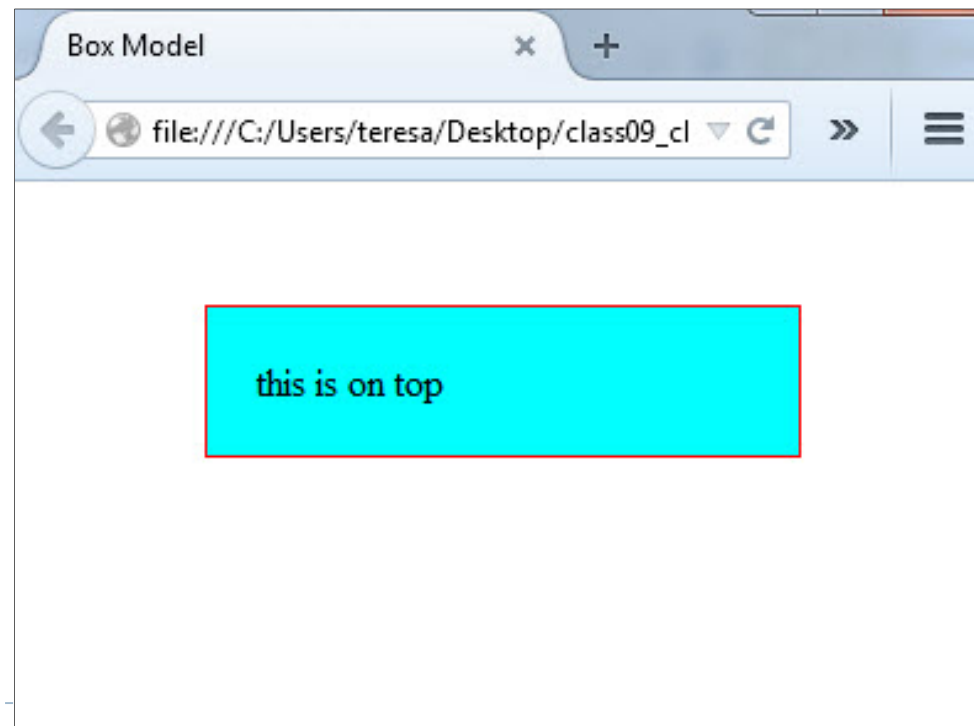
Multiple sides / shorthand

- ▶ `margin: value; (all 4 sides)`
- ▶ `margin: value value; (top / bottom)`
- ▶ `margin: value value value; (top / rl / bottom)`
- ▶ `margin: value value value value; (t / r / b / l = clockwise)`

CSS margin property

Center a block level container

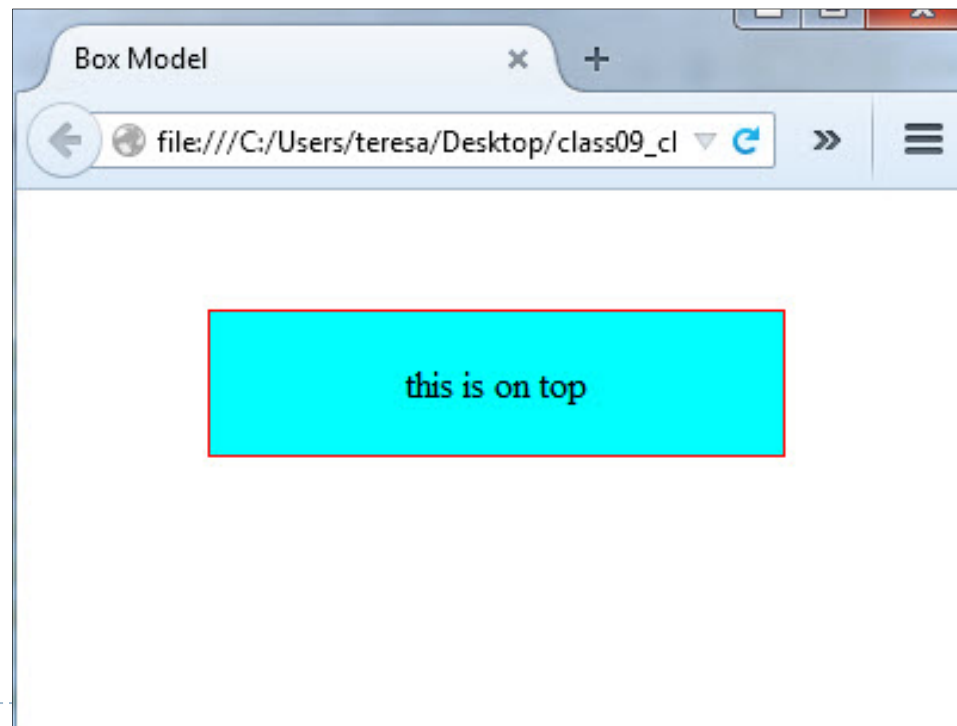
```
p { width: 200px;  
    margin-right: auto; margin-left: auto; }
```



CSS text-align property

Center content inside a block container

```
p { width: 200px; margin-right: auto; margin-left: auto;  
    text-align: center; }
```



CSS inherit value (children inherit from parents)

Box model properties are **not inherited** by default

```
section { background-color: aqua; width: 200px;
```

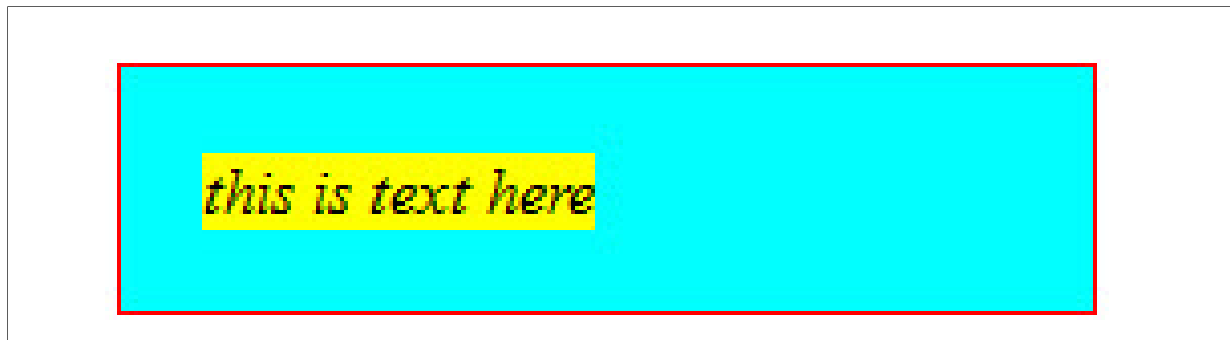
```
  margin: 20px;
```

```
  border: 1px solid red;
```

```
  padding: 20px; }
```

```
em { background-color: yellow; }
```

```
<section><em>this is text here</em></section>
```



CSS inherit value (children inherit from parents)

Box model properties are **not inherited** by default

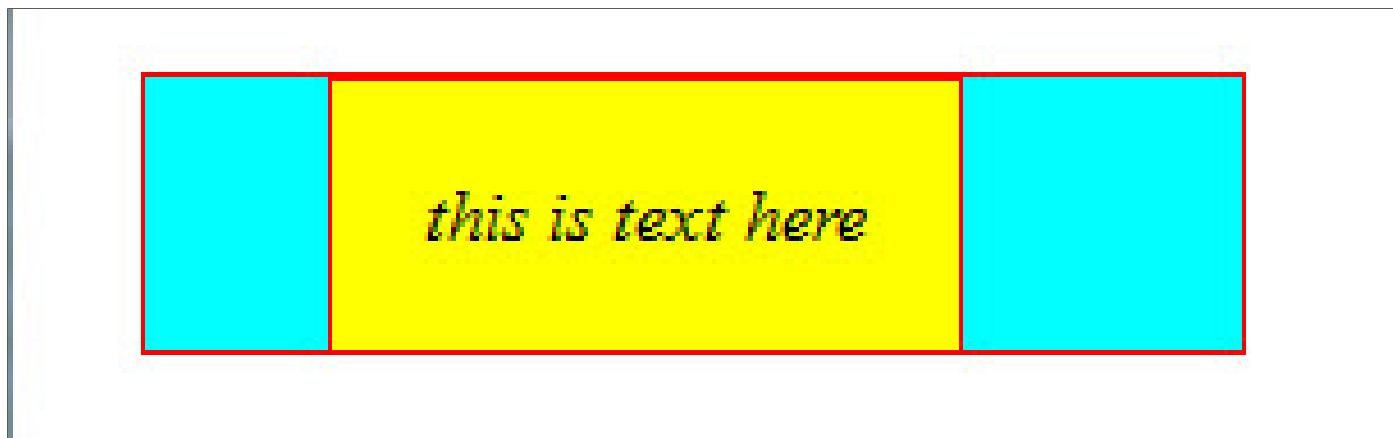
```
section { background-color: aqua; width: 200px;
```

```
    margin: 20px; border: 1px solid red; padding: 20px; }
```

```
em { background-color: yellow;
```

```
    margin: inherit; border: inherit; padding: inherit; }
```

```
<section><em>this is text here</em></section>
```



CSS display property – pg 334

- ▶ `display: inline;`
- ▶ `display: block;`

- ▶ Used in navigation

Navigation



```
<nav id="nav">
```

```
<ul>
```

```
<li><a href="">House</a></li>
```

```
<li><a href="">Baby</a></li>
```

```
<li><a href="">More</a></li>
```

```
<li><a href="">About</a></li>
```

```
</ul>
```

```
</nav>
```




```
#nav ul { list-style-type: none; padding-left: 0px;}
```

```
#nav a { display: block; padding: 10px; border: 1px solid #fff; }
```


CSS3 box-shadow property – pg 335

Firefox, Safari/Chrome, Opera and IE9 users should see a grey fading shadow under this box.

```
#example1 {  
box-shadow: 10px 10px 5px #888;  
}
```



horizontal vertical blur (optional spread) color