

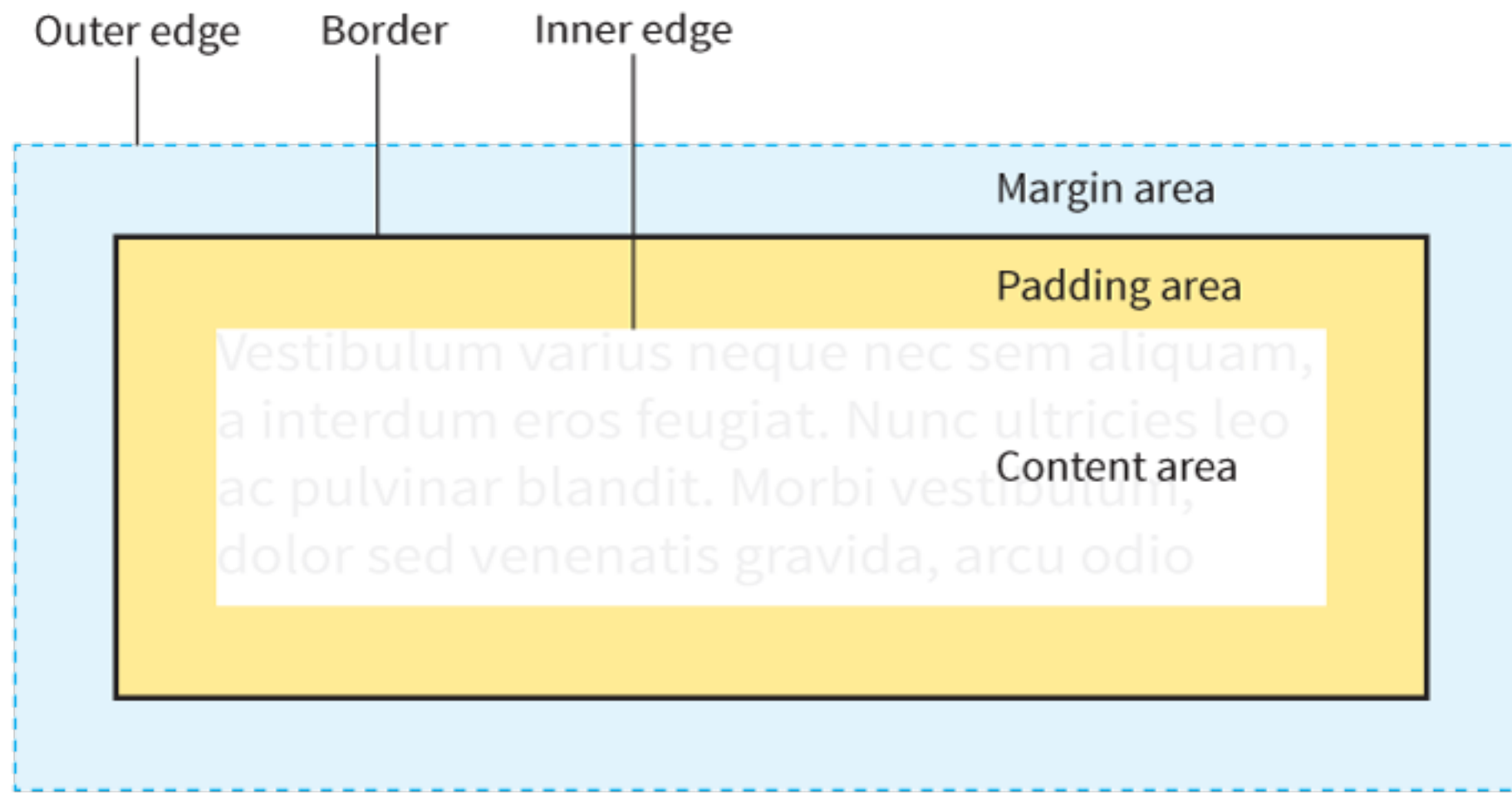
14

THINKING INSIDE THE BOX

OVERVIEW

- **The parts of an element box**
- **Box dimensions**
- **Padding**
- **Borders**
- **Outlines**
- **Margins**
- **Display roles**
- **Drop shadows**

The Parts of an Element Box



NOTE: The margin is indicated with a blue shade and outline, but is invisible in the layout.

Specifying Box Dimensions

width

Values: *Length, percentage*, auto

height

Values: *Length, percentage*, auto

Specify the dimensions of an element box with **width** and **height** properties

Box Sizing Models

`box-sizing`

Values: `content-box`, `border-box`

There are two methods for sizing an element box, specified with the `box-sizing` attribute:

Content-box sizing (default)

Applies `width` and `height` values to the content box only

Border-box sizing

Applies `width` and `height` values to the border box (including the padding and content)

Box Sizing Models Compared

← Total visible box width = 550px →
(50px of padding and 10px of border are added to 500px content box width)

```
box-sizing: content-box;  
width: 500px;
```

This week I am *extremely* excited about a new cooking technique called *sous vide*. In *sous vide* cooking, you submerge the food (usually vacuum-sealed in plastic) into a water bath that is precisely set to the target temperature you vacuum-sealed in plastic) into a water bath that is precisely set to the target temperature you want the food to be cooked to.

```
box-sizing: border-box;  
width: 500px;
```

← Total visible box width = 500px →
(50px of padding and 10px of border are included in the border-box size)

Overflow

overflow

Values: visible, hidden, scroll, auto

Specifies what to do when content doesn't fit in a sized element box:

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 cover the remainder of the glass

scroll

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

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 need to completely cover the remainder of the glass

Padding

**padding, padding-top, padding-right,
padding-bottom, padding-left**

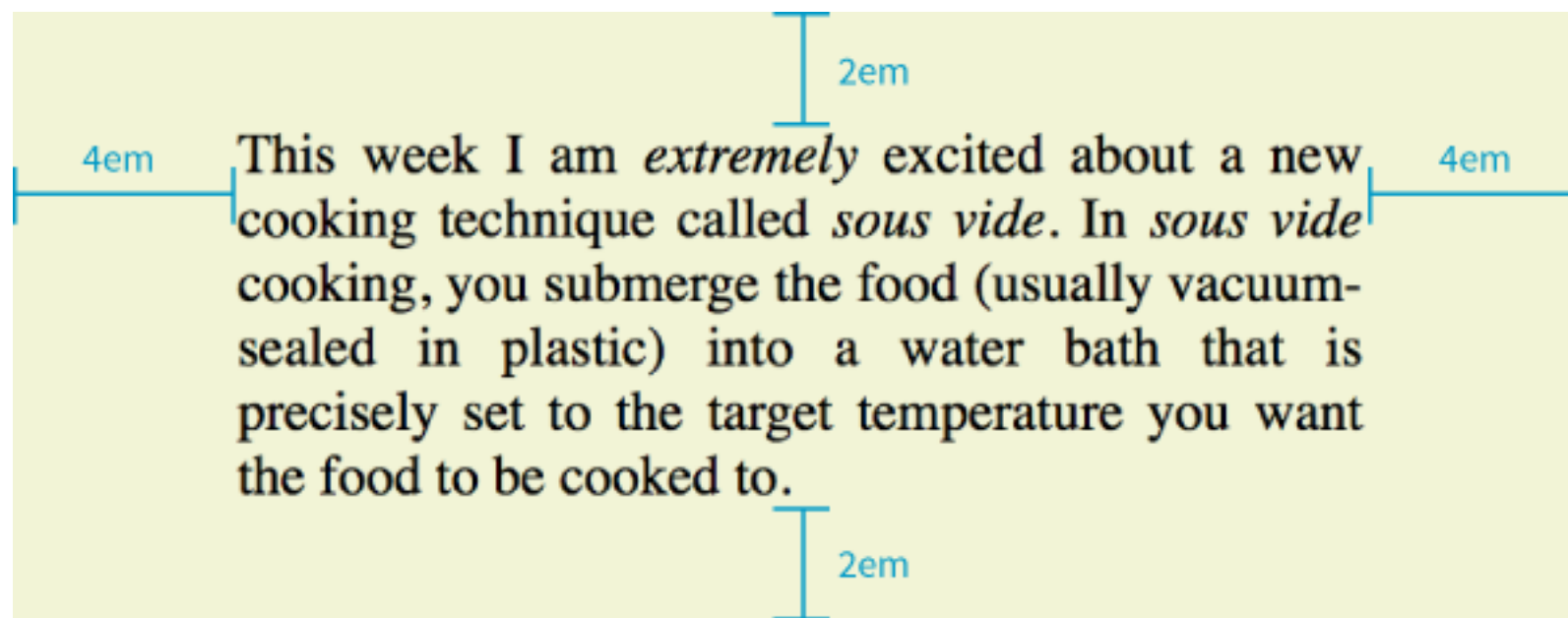
Values: *Length, percentage*

An amount of space between the content area and the border (or the space the border would be if one isn't specified).

You can add padding to one side at a time, or on all sides with the **padding** shorthand property.

Padding (cont'd)

```
blockquote {  
  padding-top: 2em;  
  padding-right: 4em;  
  padding-bottom: 2em;  
  padding-left: 4em;  
  background-color: #D098D4; /*light green*/  
}
```



Shorthand padding Property

The **padding** property adds space around 1, 2, 3, or 4 sides of the content using the clockwise top, right, bottom, left (TRouBLE) order:

```
padding: top right bottom left;
```

```
padding: 2em 4em 2em 4em;
```

(this shorthand produces the same result as the example on the previous slide)

Shorthand padding Property (cont'd)

If the left and right sides are the same, you can omit the last value, and the second value will be applied on both the left and right sides:

```
padding: top right+left bottom;
```

```
padding: 2em 4em 2em;
```

(this shorthand produces the same result as the examples on the two previous slides)

Shorthand padding Property (cont'd)

If the top and bottom sides are also the same, you can omit the third value, and the first value will be applied on both the top and bottom:

```
padding: top+bottom right+left;
```

```
padding: 2em 4em;
```

(same result as previous examples)

If all sides are the same, provide one value, and it's applied to all sides:

```
padding: all sides;
```

```
padding: 2em;
```

(2em padding all around)

Borders

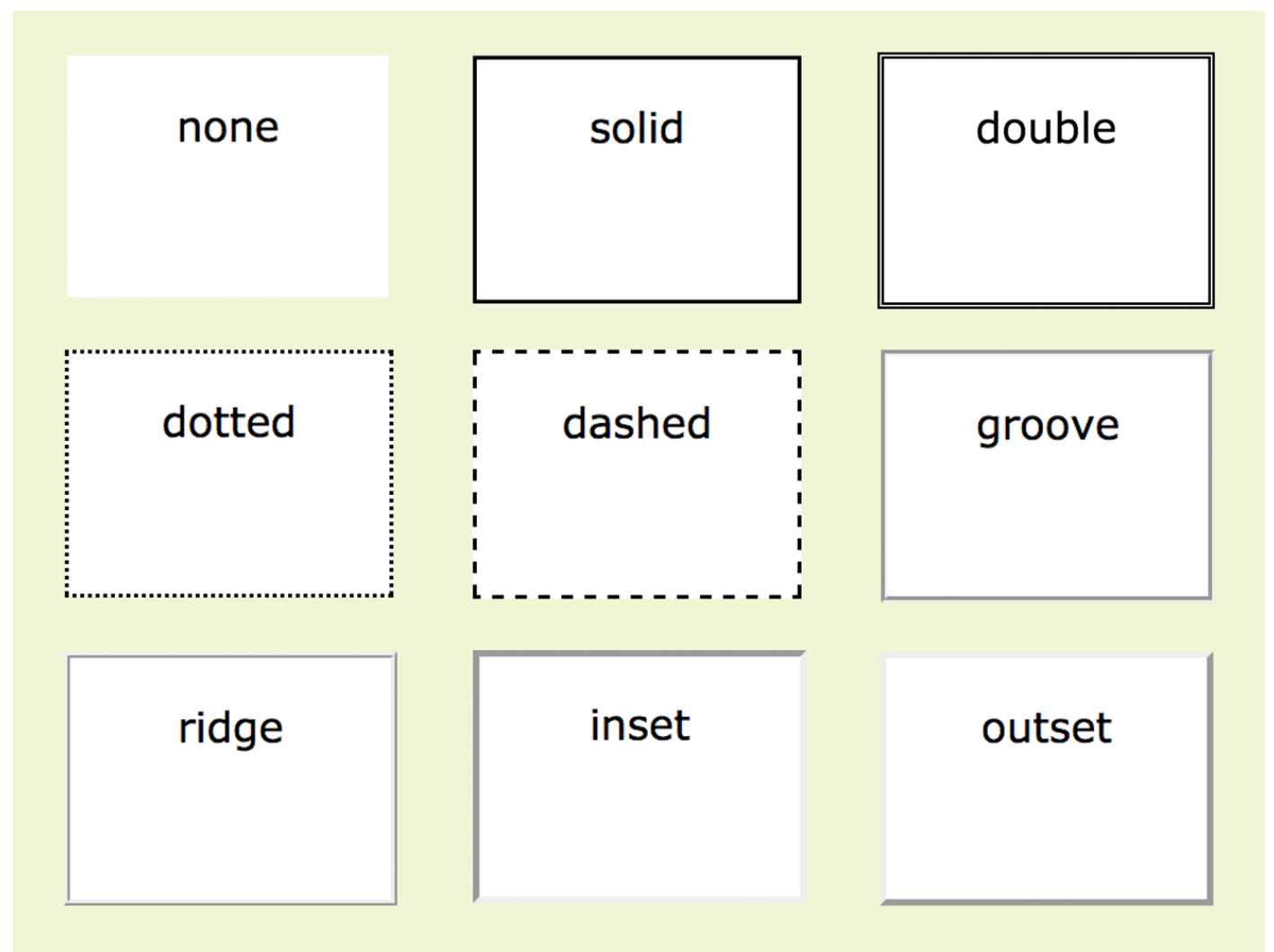
- A **border** is a line drawn around the content area and its (optional) padding.
- The thickness of the border is included in the dimensions of the element box.
- You define **style**, **width** (thickness), and **color** for borders.
- Borders can be applied to single sides or all around

Border Style

`border-style`,
`border-top-style`, `border-right-style`,
`border-bottom-style`, `border-left-style`

Values: none, solid, hidden, dotted, dashed, double, groove, ridge, inset, outset

NOTE: The default is none, so if you don't define a border style, it won't appear.



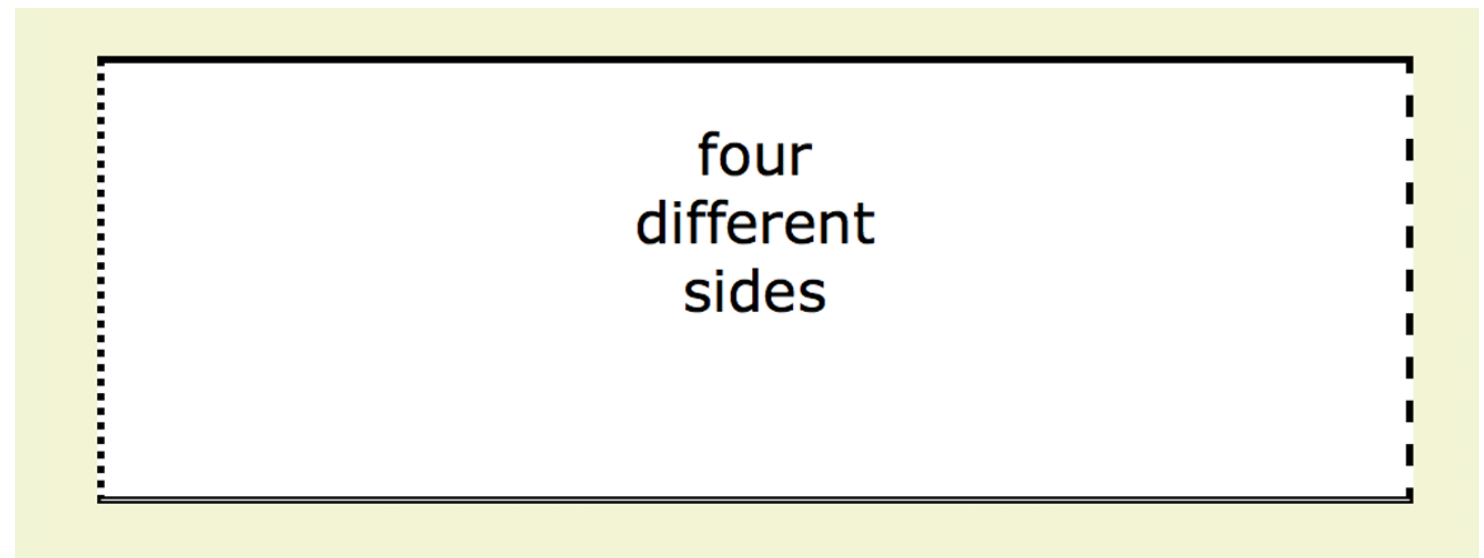
Border Style

`border-style`

The **`border-style`** shorthand uses the clockwise (TRouBLe) shorthand order. The following rules have the same effect:

```
div#silly {  
  border-top-style: solid;  
  border-right-style: dashed;  
  border-bottom-style: double;  
  border-left-style: dotted;  
  width: 300px;  
  height: 100px;  
}
```

```
div#silly {  
  border-style: solid dashed double dotted;  
}
```



Border Width

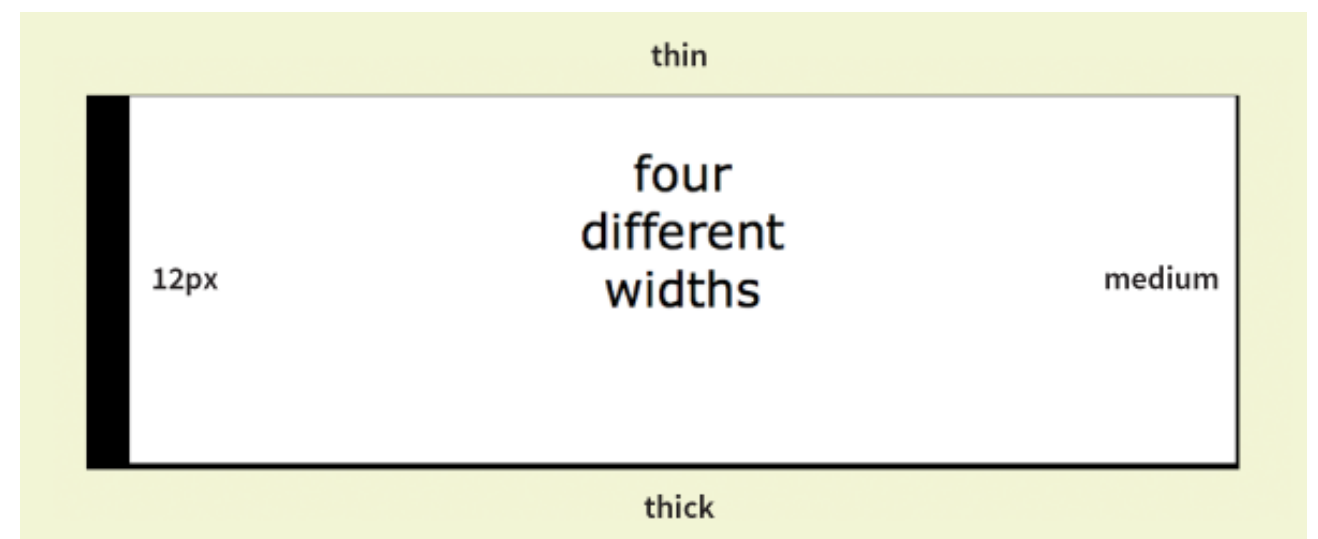
border-width,
border-top-width, border-right-width,
border-bottom-width, border-left-width

Values: *Length*, thin, medium, thick

The **border-width** shorthand uses the clockwise (TRouBLe) order:

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

NOTE: The **border-style** property is required for the border to be rendered.



Border Color

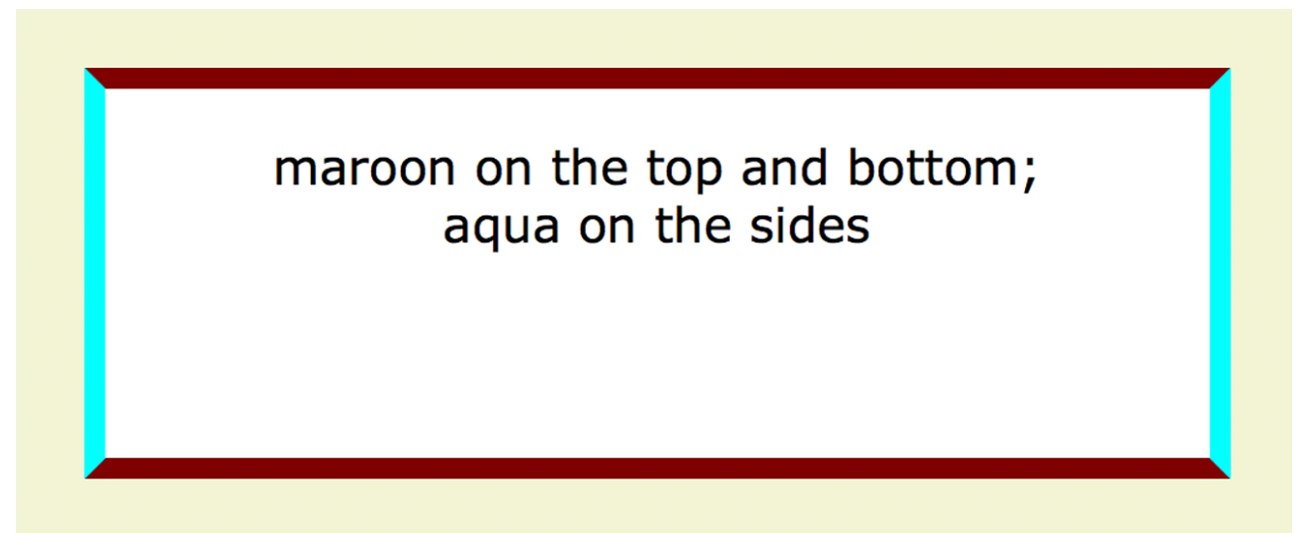
border-color,
border-top-color, border-right-color,
border-bottom-color, border-left-color

Values: *Color value (named or numeric)*

The **border-color** properties override the **color** property:

```
div#special {  
  border-color: maroon aqua;  
  border-style: solid;  
  border-width: 6px;  
  width: 300px;  
  height: 100px;  
}
```

NOTE: The **border-style** property is required for the border to be rendered.



Border Shorthand Properties

border,
border-top, border-right,
border-bottom, border-left

Values: *border-style border-width border-color*

Combine style, width, and color values in shorthand properties for each side or all around (**border**):

```
p.example {  
  border: 2px dotted aqua;  
}
```

NOTE: The **border-style** property must be included in the shorthand for the border to be rendered.

Border Radius (Rounded Corners)

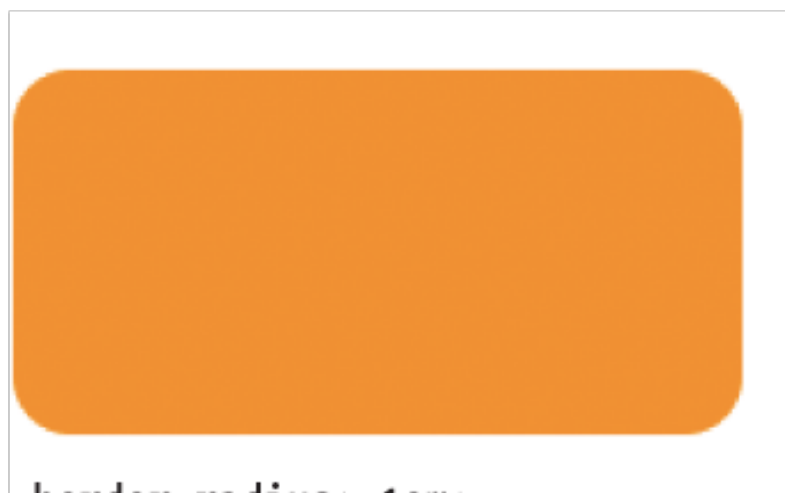
border-radius

Values: *1, 2, 3, or 4 length or percentage values*

- The **border-radius** property rounds off the corners of an element.
- The value is a length or percentage value reflecting the radius of the curve.
- Providing one value makes all the corners the same.
- Four values are applied clockwise, starting from the top-left corner.

Border Radius (cont'd)

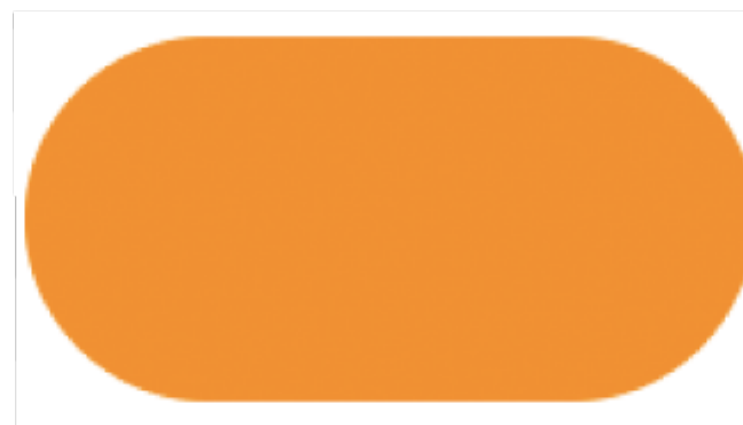
```
p {  
  width: 200px;  
  height: 100px;  
  background: darkorange;  
}
```



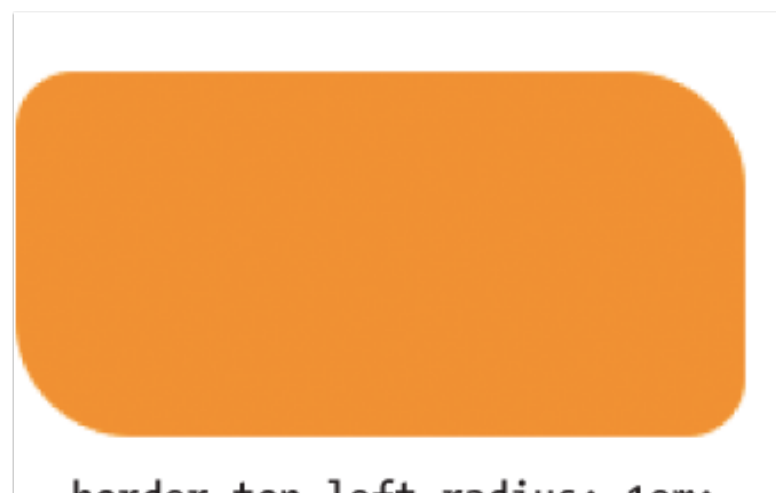
`border-radius: 1em;`



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



`border-radius: 50px;`



`border-top-left-radius: 1em;`
`border-top-right-radius: 2em;`
`border-bottom-right-radius: 1em;`
`border-bottom-left: 2em;`

or

`border-radius: 1em 2em;`

Margins

**margin, margin-top, margin-right,
margin-bottom, margin-left**

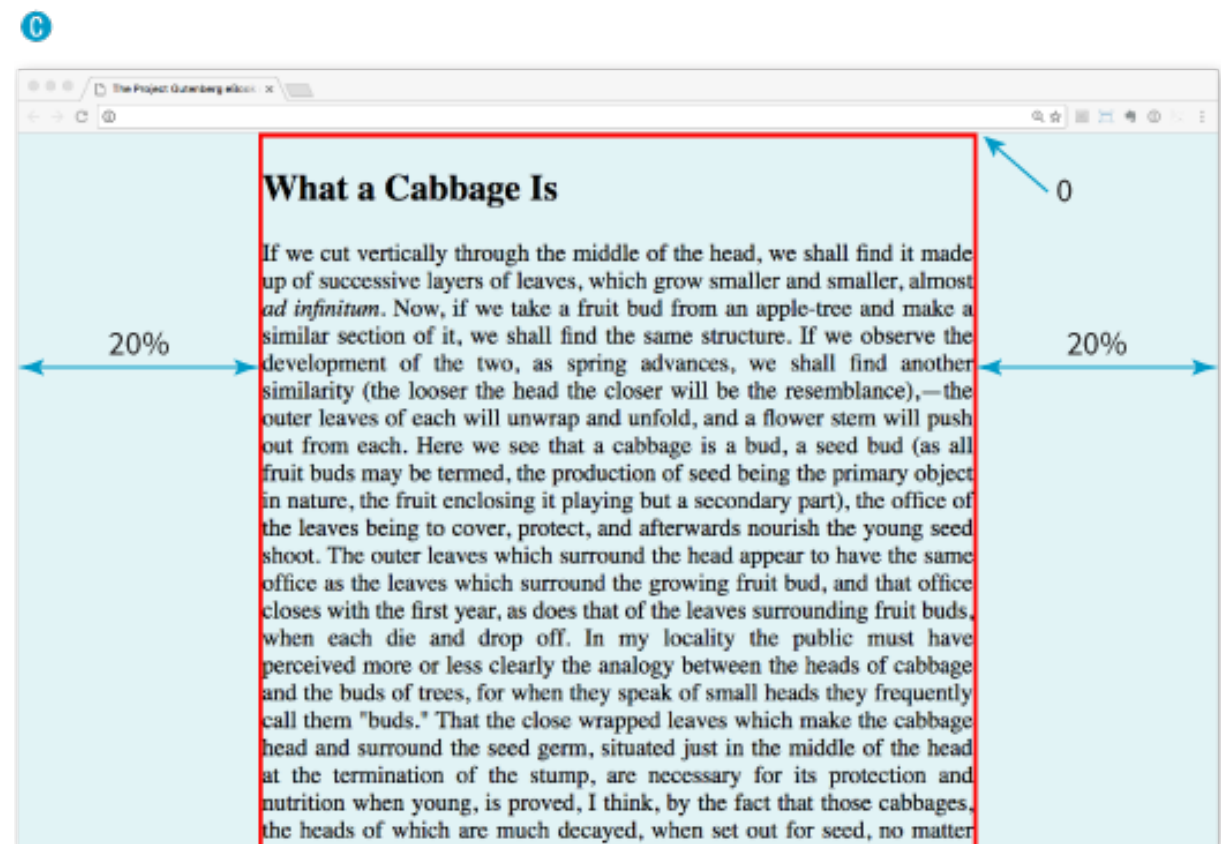
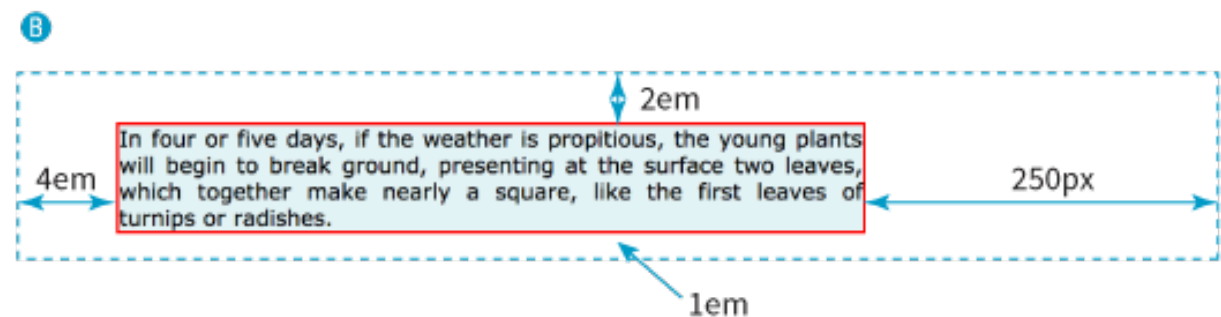
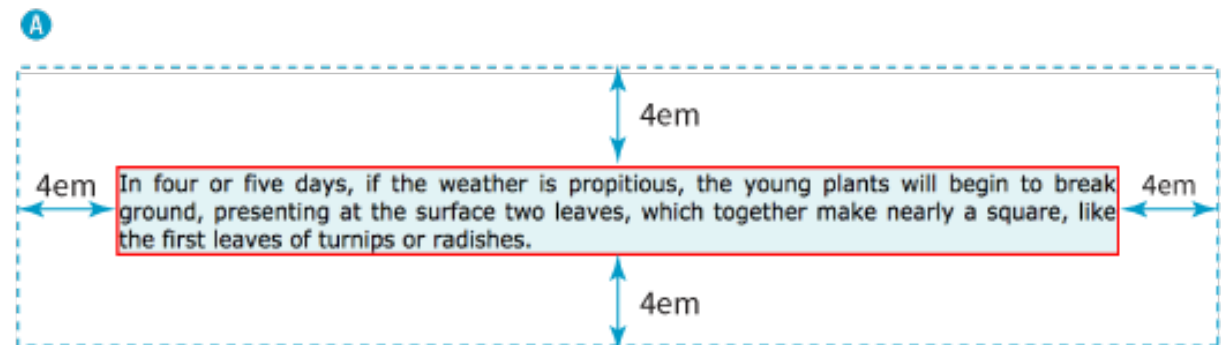
Values: *Length, percentage*

The **margin** is an amount of space added on the outside of the border. They keep elements from bumping into one another or the edge of the viewport.

The shorthand **margin** property works the same as the **padding** shorthand. Values are applied clockwise (TRouBLe order) and are repeated if fewer than 4 values are supplied.

Margins (cont'd)

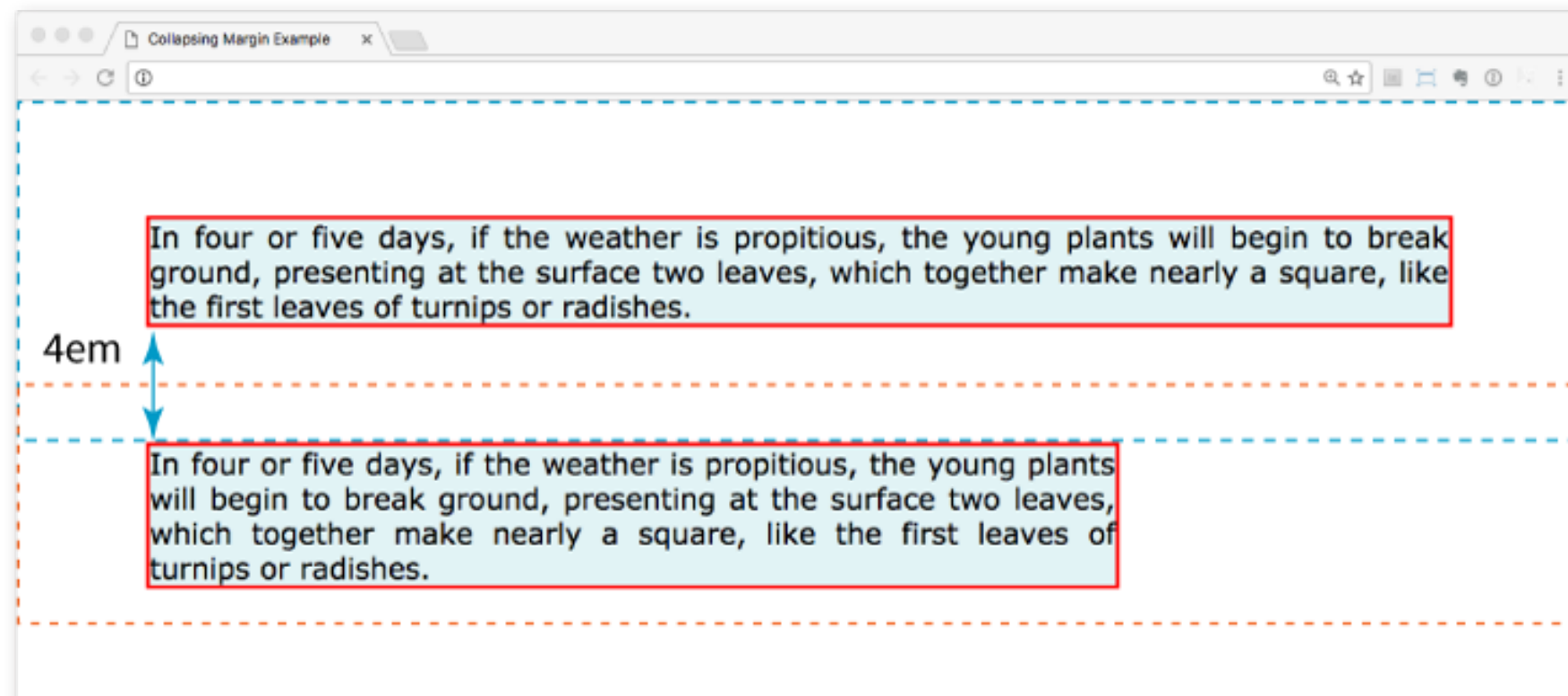
- A** `p#A {`
 margin: 4em;
 border: 2px solid red;
 background: #e2f3f5;
}
- B** `p#B {`
 margin-top: 2em;
 margin-right: 250px;
 margin-bottom: 1em;
 margin-left: 4em;
 border: 2px solid red;
 background: #e2f3f5;
}
- C** `body {`
 margin: 0 20%;
 border: 3px solid red;
 background-color: #e2f3f5;
}



Margins (cont'd)

Top and bottom margins of neighboring elements **collapse** (they overlap instead of accumulating).

The top element has a bottom margin of 4em. The bottom element has a top margin of 2em. The resulting margin is 4em (the largest value).



Assigning Display Types

display

Values: inline | block | run-in | flex | grid | flow | flow-root | list-item | table | table-row-group | table-header-group | table-footer-group | table-row | table-cell | table-column-group | table-column | table-caption | ruby | ruby-base | ruby-text | ruby-base-container | ruby-text-container | inline-block | inline-table | inline-flex | inline-grid | contents | none

Assigns a **display type** that determines how the element box behaves in layouts.

Examples:

- Make **li** (normally block elements) into inline elements so they line up in a horizontal menu: `nav li { display: inline; }`
- Make an anchor (**a**) element (normally inline) display as a block so you can give it a width and height: `nav li a { display: block; }`

Box Drop Shadows

box-shadow

Values: *horizontal-offset vertical-offset blur-distance spread-distance color inset, none*

Applies a drop shadow around the visible element box.

The values are a **horizontal** and **vertical offset**, optional **blur** and **spread** values (in pixels), a **color** value, and the option to make it appear **inset**.

Box Drop Shadows (cont'd)

A `box-shadow: 6px 6px gray;`

B `box-shadow: 6px 6px 5px gray; /* 5 pixel blur */`

C `box-shadow: 6px 6px 5px 10px gray; /* 5px blur, 10px spread */`

