COMP 1950

Web Development and Design 2

Day 04

CSS3

- Some Advantages to CSS3
  - Create better layout design using just styles as opposed to images
  - Create more advanced transitions without the use of JavaScript
- Some Dis-Advantages to CSS3
  - No support in older versions of IE

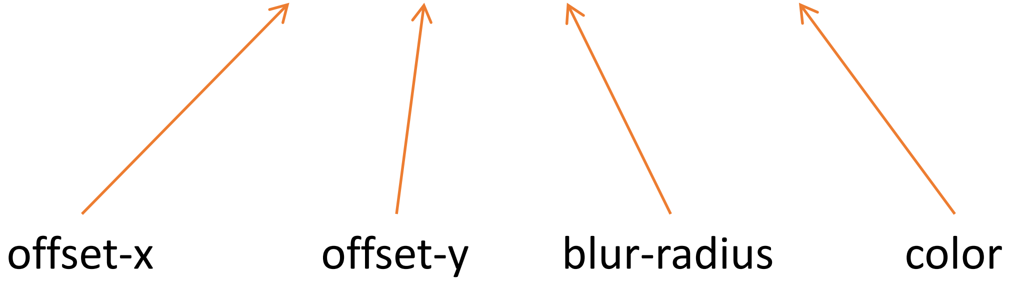
Text Shadow

- Text shadow adds a drop shadow to text
- You can have multiple text shadows applied to the same text
- Text shadow parameters
  - Color (optional)
    - Can be placed at the beginning or end of the style declaration
  - offset-x (required)
    - Sets the x axis offset of the drop shadow
  - offset-y (required)
    - Sets the y axis offset of the drop shadow
  - blur-radius (optional)
    - Sets the blur of the shadow

Text Shadow

Text Shadow

```
.box_main .h_01 {
     text-shadow: 10px 10px 10px #363636;
}
```



Text Shadow

Text Shadow - Multiple

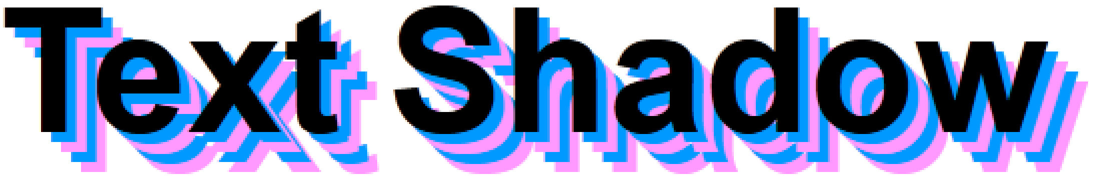
```
.box_main .h_02 {
     text-shadow: 5px 5px 0 #09F,
                  10px 10px 0 #F9F,
                  15px 15px 0 #09F,
                  20px 20px 0 #F9F;
}
```

shadow 01

shadow 02

shadow 03

shadow 04



Text Shadow

- Browser compatibility for text shadow:

IE10+

Firefox 3.1+

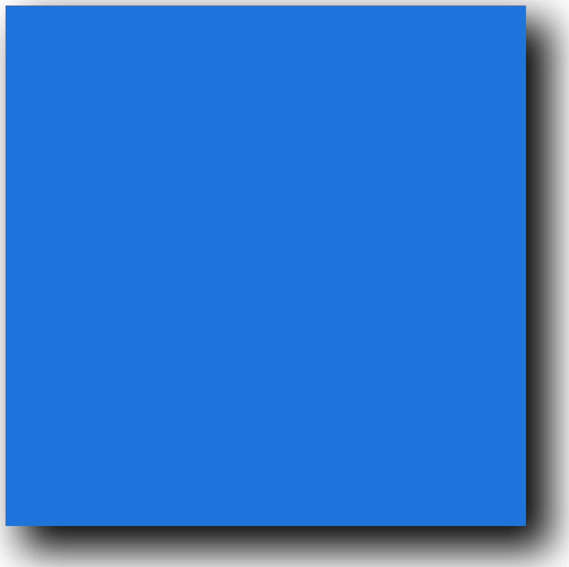
Chrome 2+

Safari 1.1+

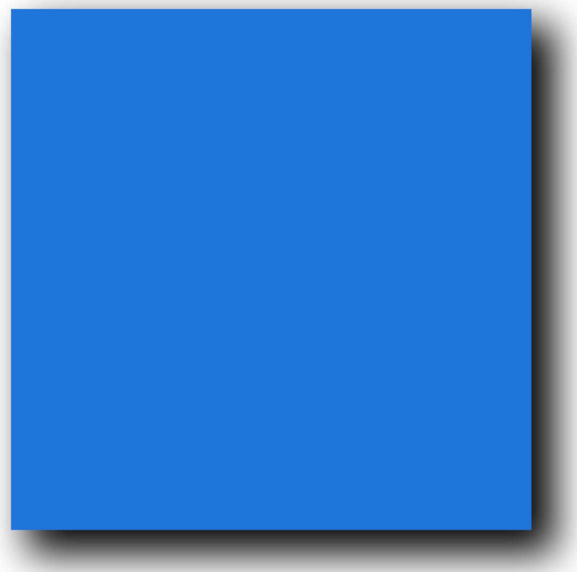
Opera 9.5+

Box Shadow

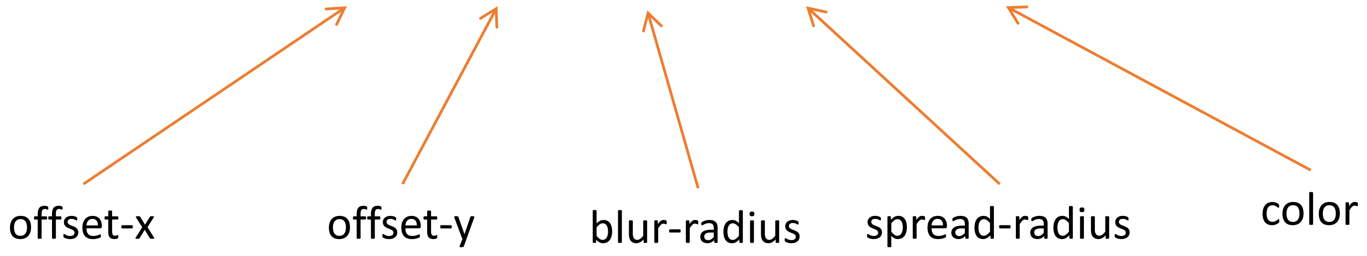
- Box Shadow adds a drop shadow to HTML elements
- You can set inset box shadow styles
- Box Shadow parameters
  - Inset (optional)
    - Determines whether the shadow is inset or not
    - If omitted the browser will assume that the box shadow is a regular drop shadow
  - offset-x (required)
    - Sets the x axis offset of the drop shadow
  - offset-y (required)
    - Sets the y axis offset of the drop shadow
  - blur-radius (optional)
    - Sets the blur of the shadow
  - spread-radius (optional)
    - Sets how much the shadow will expand or shrink
    - Positive values will cause the shadow to expand
    - Negative values will cause the shadow to shrink



Box Shadow



```
.box_01 {
        box-shadow: 10px 10px 19px 2px #000000;
}
```



offset-x

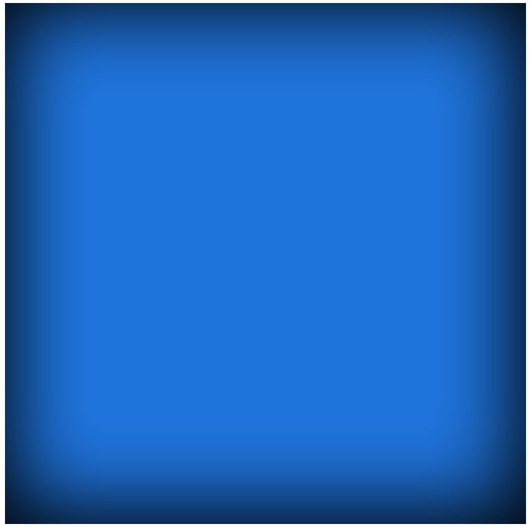
offset-y

blur-radius

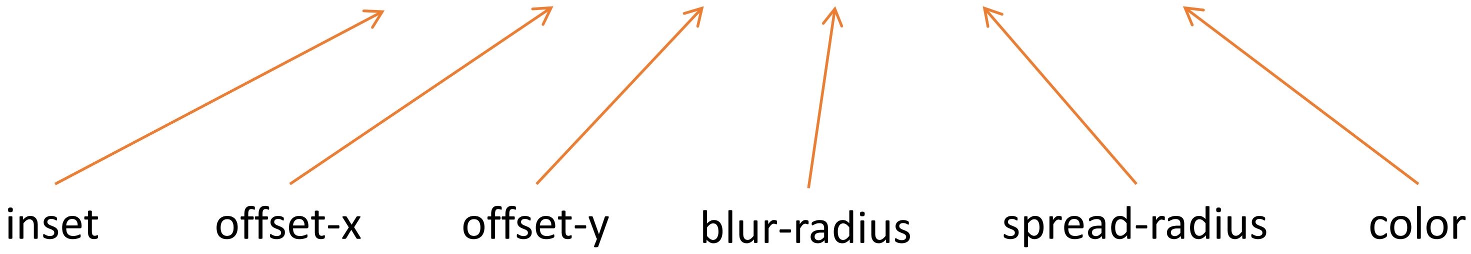
spread-radius

color

Box Shadow - Inset



```
.box_02 {
        box-shadow: inset -1px -2px 41px 5px #000000;
}
```



inset offset-x offset-y blur-radius spread-radius color

Box Shadow

- Browser compatibility for box shadow:

IE9+ Firefox 3.5+ Chrome 3+ Safari 3+ Opera 10.5+

Border Radius

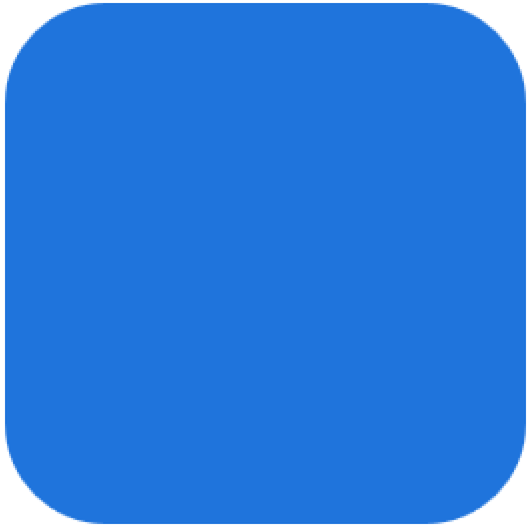
- Border radius adds rounded corners to elements
- You can have different values set for each corner
- You can have different values for the horizontal and vertical radii of a quarter ellipse which can create a more elliptical appearance in the corners as opposed to more a more circular appearance when the horizontal and vertical radii are of the same value
- Border radius can use a shorthand or long hand syntax
- Border radius parameters
  - length
    - Determines the size of the rounded corner based on a CSS unit (px, ems etc)
  - percentage
    - Determines the size of the rounder corner based on a percentage of the size of the HTML element

Border Radius

- Shorthand syntax where all the corners have the same value

```
.box_04 {
        border-radius: 50px;
}
```

length



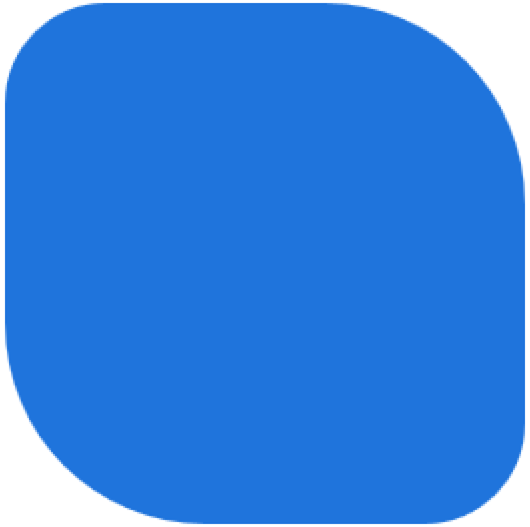
Border Radius

- Shorthand syntax where the adjacent corners have different values

```
.box_05 {
        border-radius: 50px 100px;
}
```

top left and bottom right length

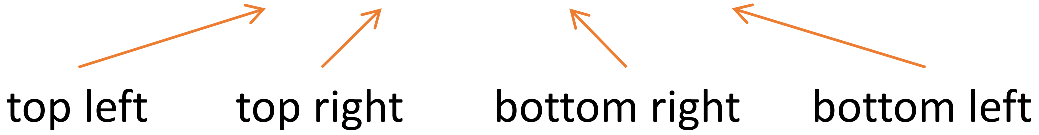
top right and bottom left length



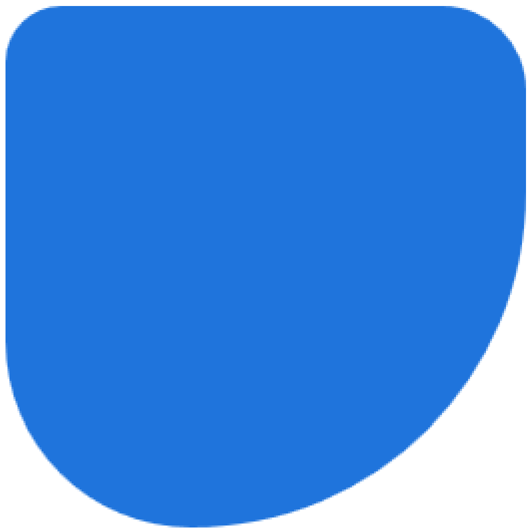
Border Radius

- Shorthand syntax where all corners have different values

```
.box_06 {
        border-radius: 30px 45px 180px 100px;
}
```



top left top right bottom right bottom left



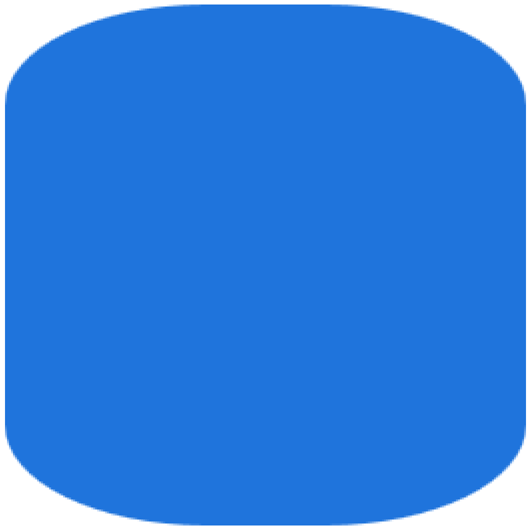
Border Radius

- Shorthand syntax for elliptical corners where all corners have the same values

```
.box_07 {
        border-radius: 100px / 50px;
}
```

horizontal radii

vertical radii



Border Radius

- Browser compatibility for border radius:

IE9+ Firefox 1+ Chrome 3+ Safari 3+ Opera 10.5+

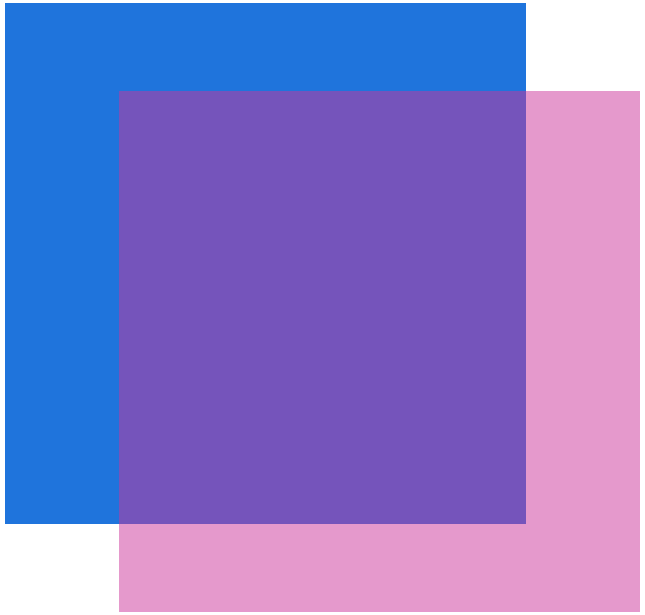
Opacity

- Sets the opacity level of an element
- The opacity style applies to the element and all its descendant elements
- Child elements can not override an opacity level if an opacity level is set on any of its ascendant elements (parents, grandparents, great grandparents, etc)
- Opacity parameters
  - Alpha value
    - Takes a value between 0 and 1
      - 0 = fully transparent
      - 1 = fully opaque

Opacity

```
.box_09 {
     opacity: 0.5;
}
```

alpha value



Opacity

- Browser compatibility for opacity:

IE9+

Firefox 1.5+

Chrome 1+

Safari 1.2+

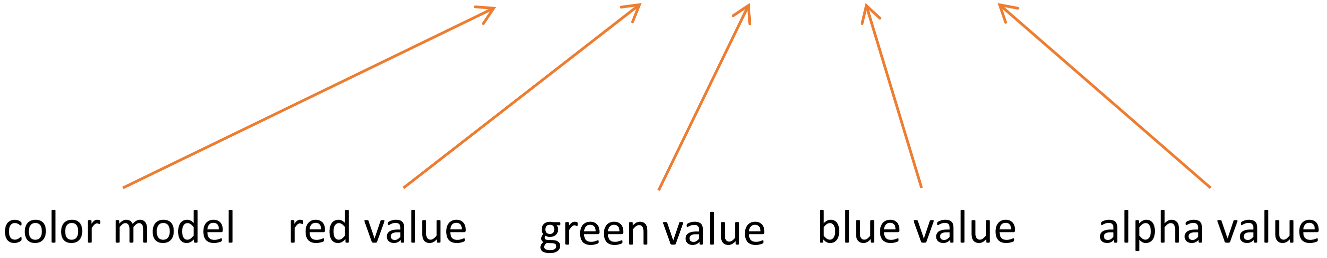
Opera 9+

RGBA

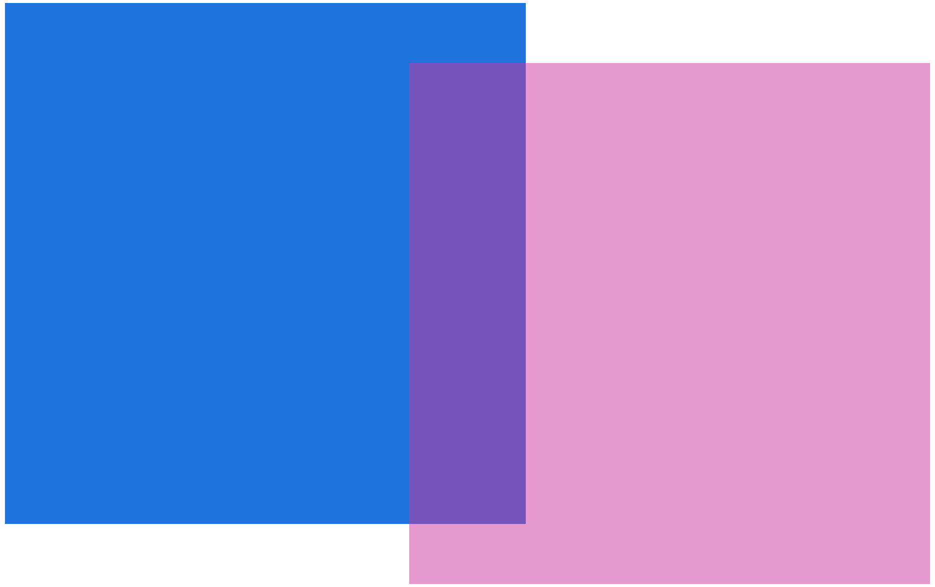
- RGBA adds an alpha channel to the RGB color format for setting color in CSS
- The alpha channel controls the transparency of the color
- The alpha channel value takes a value between 0 and 1
  - 0 = transparent
  - 1 = opaque

RGBA

```
.box_11 {
     background-color: rgba(204,51,153,0.5);
}
```



color model red value green value blue value alpha value



RGBA

- Browser compatibility for rgba:

IE9+ Firefox 3+ Chrome 3+ Safari 3.2+ Opera 10+

CSS3 Gradients

- CSS3 Gradients allow you to create gradient backgrounds using CSS and without images
- You can create linear or radial backgrounds using the CSS3 gradient style
- CSS gradients are considered backgrounds so they are actually written as a value of the background property
- You can create multiple gradients for the same background property by separating each gradient with a comma
    - This is the same method as adding multiple background images to an element which we will cover in the next HTML course
- The syntax for CSS gradients can be quite complex so below is a link to a CSS gradient generator that will help you generate the CSS code for creating CSS3 gradients
        - http://www.colorzilla.com/gradient-editor

CSS3 Gradients

- Browser compatibility for CSS3 gradients:

IE10+ Firefox 3.6+ Chrome 10+ Safari 5.1+ Opera 11.6+

Advanced CSS Selectors

- CSS 2.1 and CSS3 contain many advanced selectors that allow web designers to select elements on the page using fewer custom classes or ID’s
- Some more advanced CSS selectors include:
  - Attribute selectors
  - Child selectors
  - Adjacent Selectors
  - Sibling selectors
  - Pseudo Selectors

Attribute Selectors

- The attribute selector allows you to select certain HTML elements based on whether or not the element contains a certain attribute
- The attribute selector also allows you to select elements based on what the value of an attribute is
- 7 main types of attribute selectors
  - [attribute] = matches all elements with have a certain attribute
  - ^ = matches items that start with a value
  - $ = matches items that end with a value
  - * = matches items that contain a certain value
  - ~= matches space separated value of some kind
  - |= matches dash separated value
  - “some text” = matches items that contain exactly the text in the quotes
- You can combine multiple attribute selectors to make a more specific selection

Attribute Selectors

- Examples of attribute selectors:

The selector below will select all “a” elements that have an “href” attribute with a value that ends with “pdf”

```
a[href$="pdf"]{
     background-image: url(images/pdficon_small.png);
     background-repeat: no-repeat;
     background-position: 0px 2px;
}
```

Child Selectors

- Selects “direct” decedents of on an element
- This is different from the related common descendant selector which will select all decedents of an element

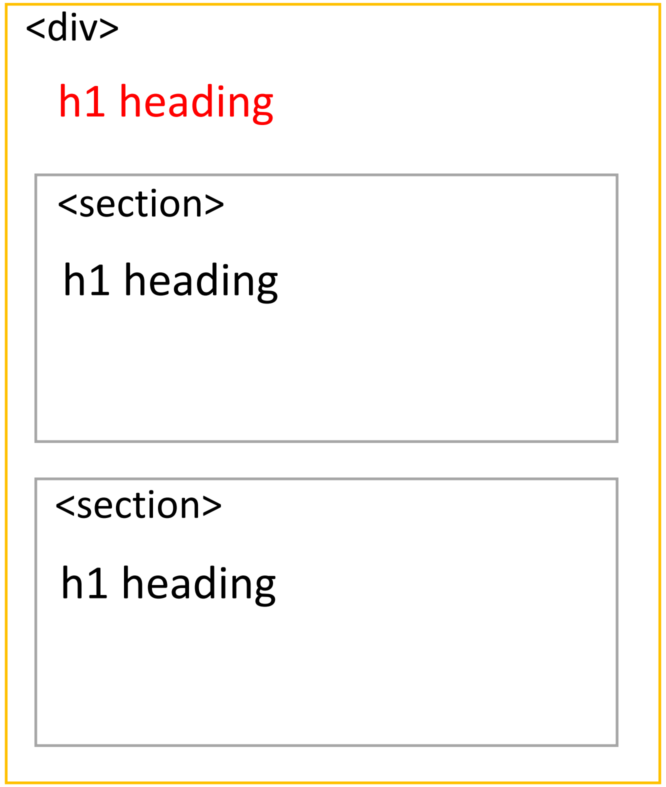
The “>” means this is a child selector

```
div > h1 {
     color: red;
}
```

Child Selectors

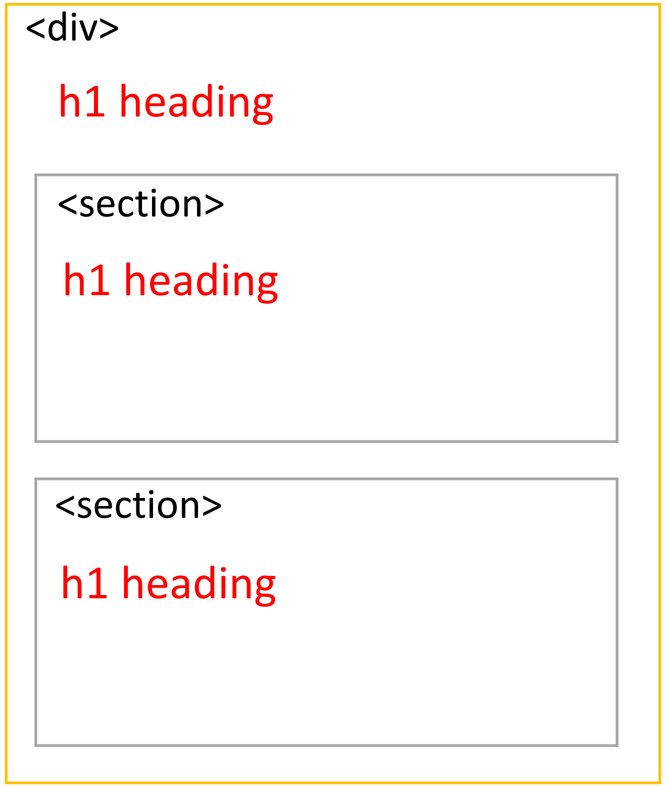
Child Selector

```
div > h1 {
      color: red;
}
```



Descendent Selector

```
div h1 {
      color: red;
}
```



Adjacent Sibling Selector

- Selects elements directly “adjacent to the element in the dom

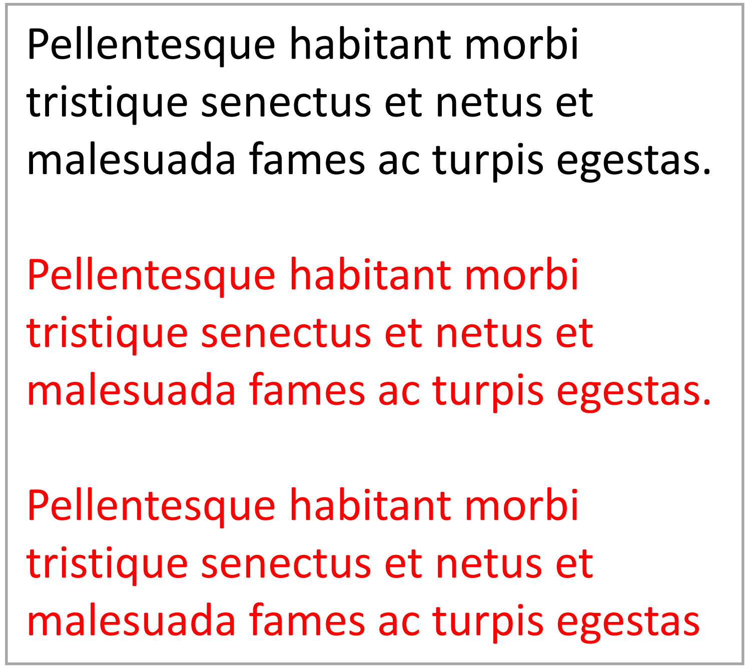
The “+” means this is an adjacent selector

```
p + p {
     color: red;
}
```

Adjacent Sibling Selector

- Example of an Adjacent selector

```
p + p {
      color: red;
}
```



Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas

General Sibling Selector

- Selects any “sibling” element that follows the element
- Different from the adjacent selector in that the element does not have to directly follow the element, it merely has to be after the element
- The element does have to be a sibling element and share the same parent element

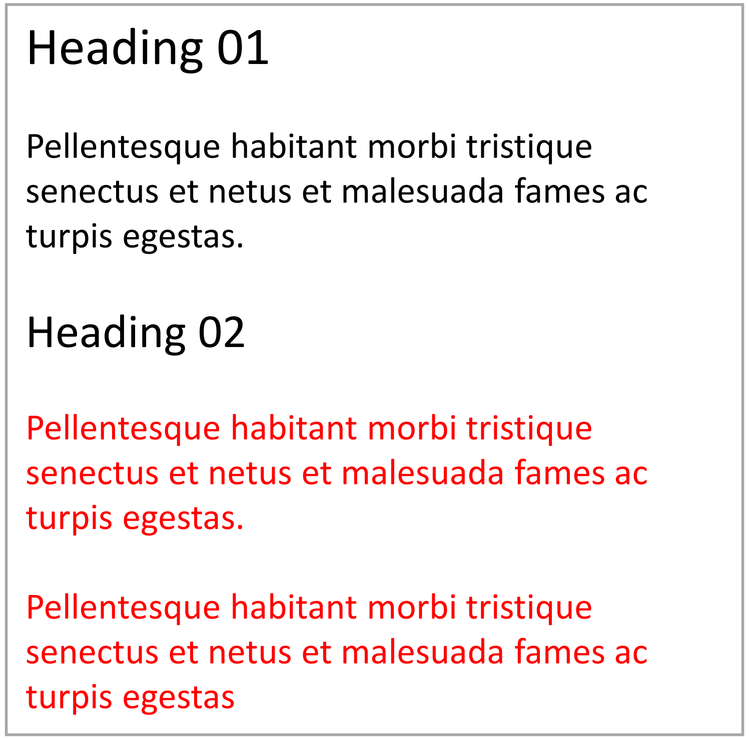
The “~” means this is a general sibling selector

```
p ~ p {
    color: red;
}
```

General Sibling Selector

- Example of a general sibling selector

```
p ~ p {
      color: red;
}
```



Heading 01

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Heading 02

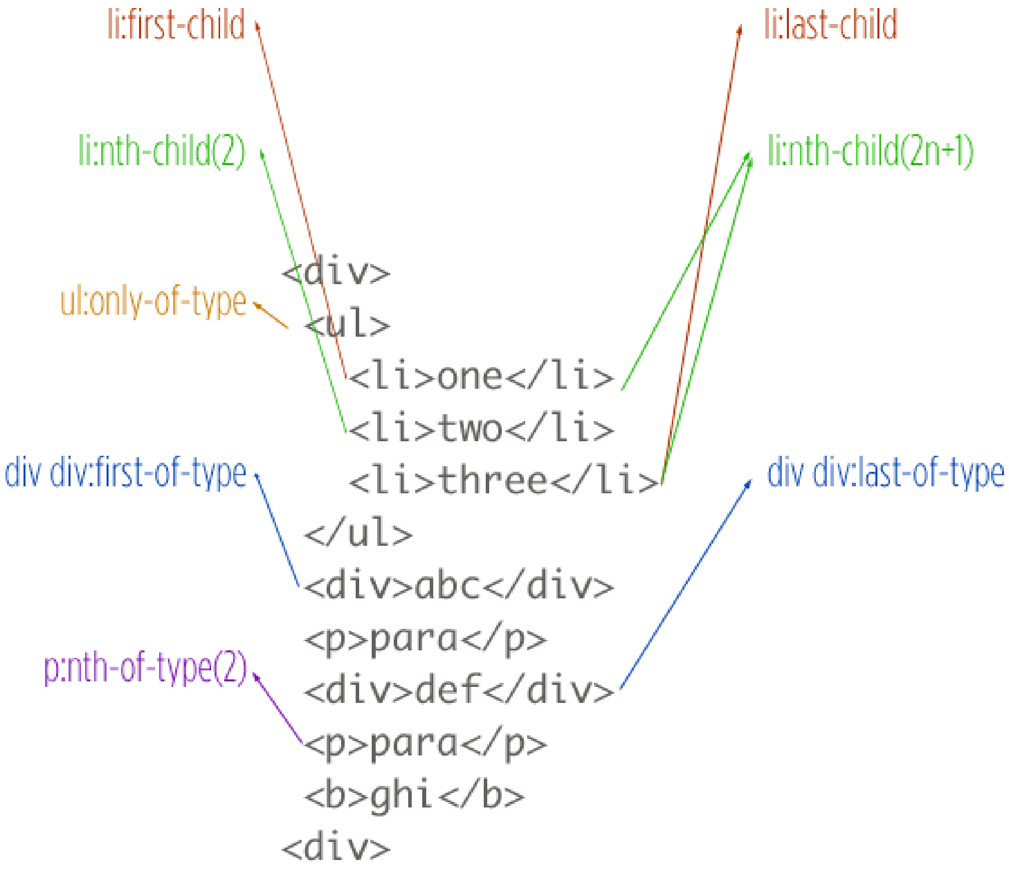
Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas

Pseudo Classes

- CSS3 has several new types pseudo class selectors
  - Some of the more useful are:
    - :first-child (CSS 2.1)
    - :last-child (CSS 2.1)
    - :nth-child(N)
    - :nth-last-child(N)
    - :nth-of-type(N)
    - :nth-last-of-type(N)
  - Go to this web site for more information on Pseudo-classes
    - http://reference.sitepoint.com/css/css3psuedoclasses

Pseudo Classes



```
<div>
 <ul>
   <li>one</li>
   <li>two</li>
   <li>three</li>
 </ul>
 <div>abc</div>
 <p>para</p>
 <div>def</div>
 <p>para</p>
 <b>ghi</b>
<div>
```

Above image from: http://css-tricks.com/pseudo-class-selectors

CSS3 Transforms

- Transform size rotation, skew and position of html elements
- Transforms can be combined with transitions to create interesting effects
- Partial support in IE9

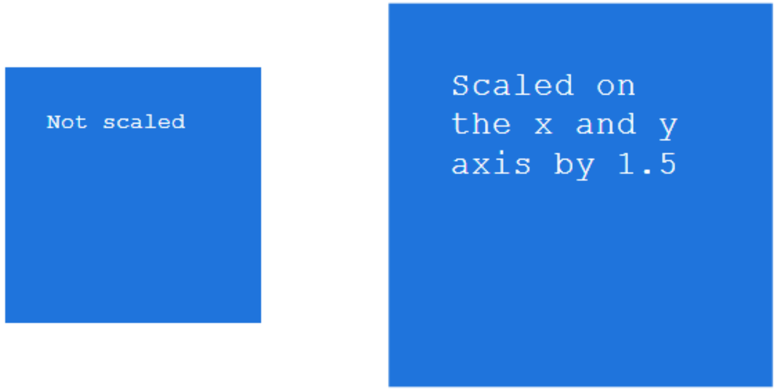
CSS Transforms

- Browser compatibility for CSS Transforms:

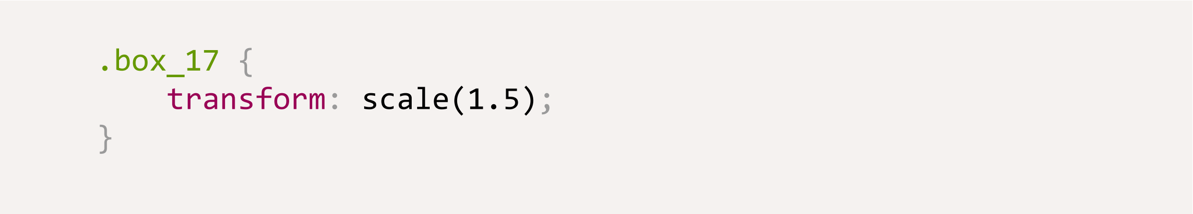
IE9+ Firefox 3.5+ Chrome 4+ Safari 3.1+ Opera 10.5+

CSS3 Transforms - Scale

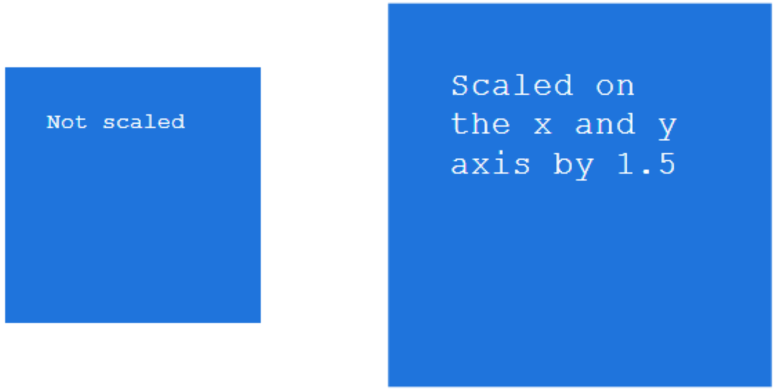
- The scale transform adjusts the size of an element
- A scale of “1” is the equivalent of not scaling the element
- A scale of “2” will make the element twice as big
- A scale of “0.5” will make the element half as big
- The scale can be set to any value
- The scale is a unitless value, it does NOT take units such as px or ems



CSS3 Transforms – Scale



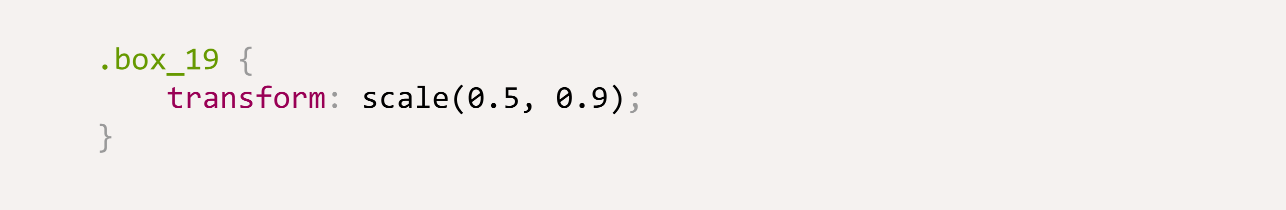
```
.box_17 {
    transform: scale(1.5);
}
```



CSS3 Transforms – Scale – X and Y

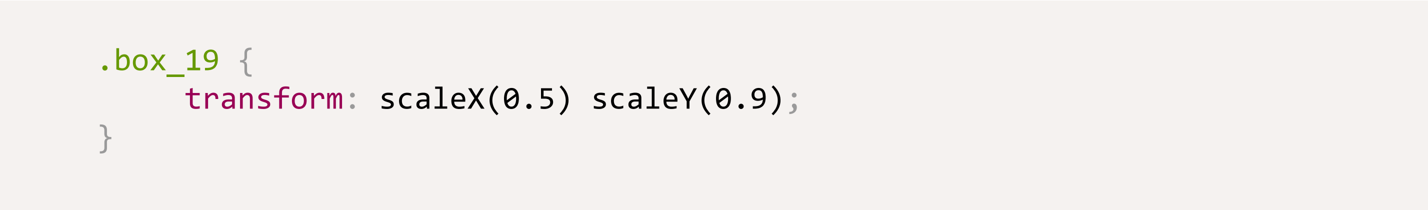
- You can scale the x axis and y axis values independently
- You have two syntax options

Option 01:



```
.box_19 {
    transform: scale(0.5, 0.9);
}
```

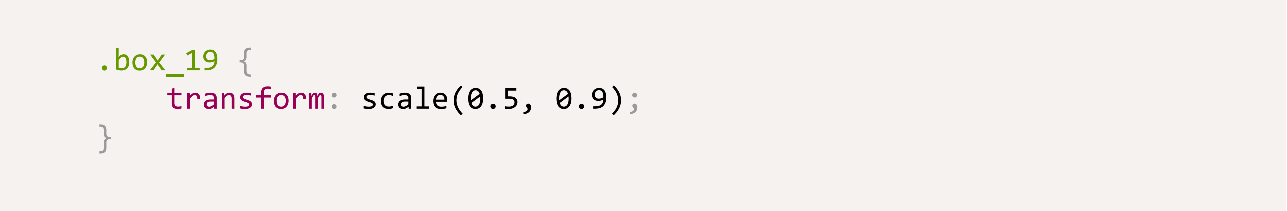
Option 02:



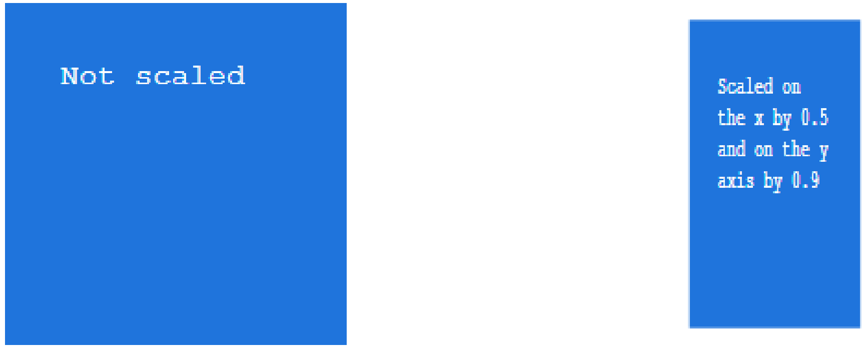
```
.box_19 {
     transform: scaleX(0.5) scaleY(0.9);
}
```

CSS3 Transforms – Scale – X and Y

Option 01:

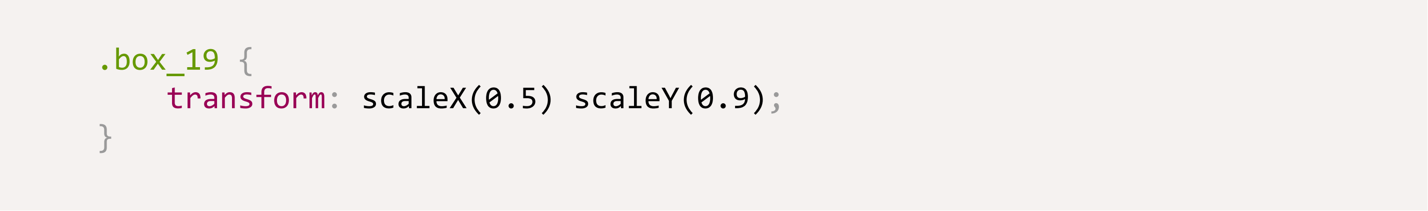


```
.box_19 {
    transform: scale(0.5, 0.9);
}
```

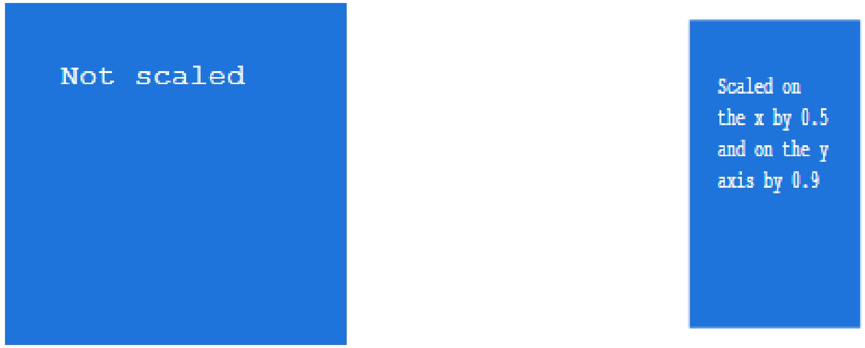


CSS3 Transforms – Scale – X and Y

Option 02:



```
.box_19 {
    transform: scaleX(0.5) scaleY(0.9);
}
```

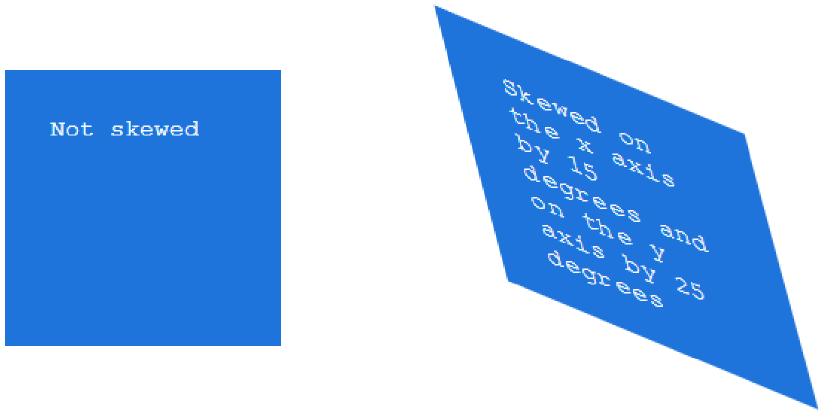


Not scaled

Scaled on
the x by 0.5
and on the y
axis by 0.9

CSS3 Transforms - Skew

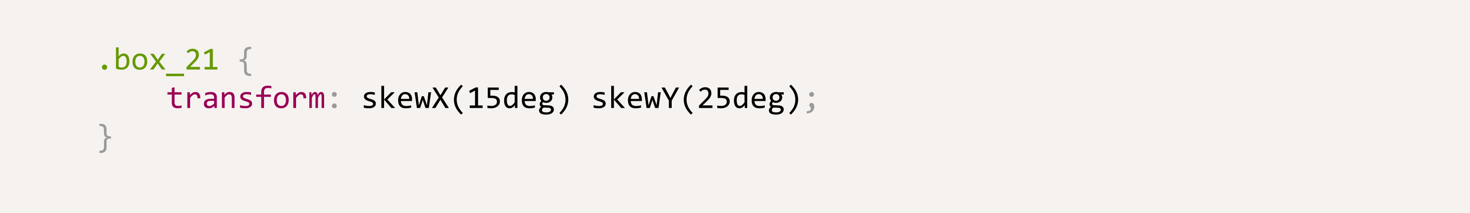
- The skew skews an element along the x and or y axis



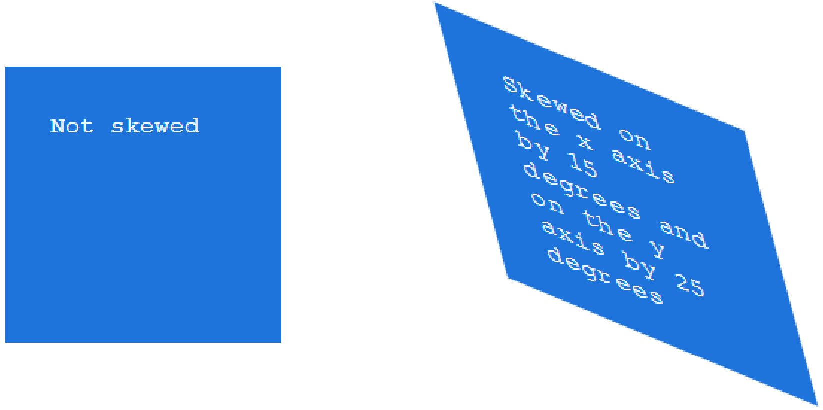
Not skewed

Skewed on the x axis by 15 degrees and on the y axis by 25 degrees

CSS3 Transforms - Skew

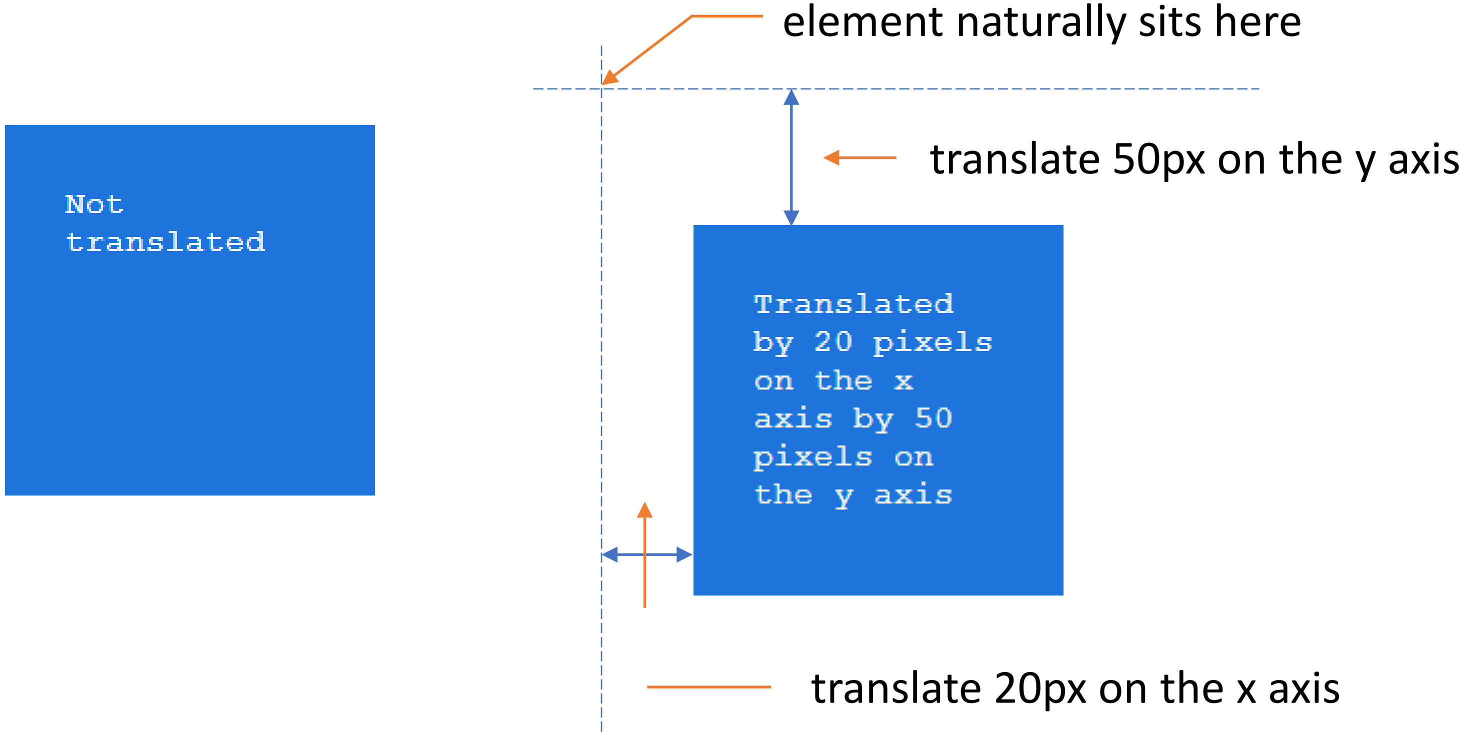


```
.box_21 {
    transform: skewX(15deg) skewY(25deg);
}
```



CSS3 Transforms - Translate

- Translate alters the position of the element relative to where it would naturally sit
- You can move an element along the x and or y axis



element naturally sits here

translate 50px on the y axis

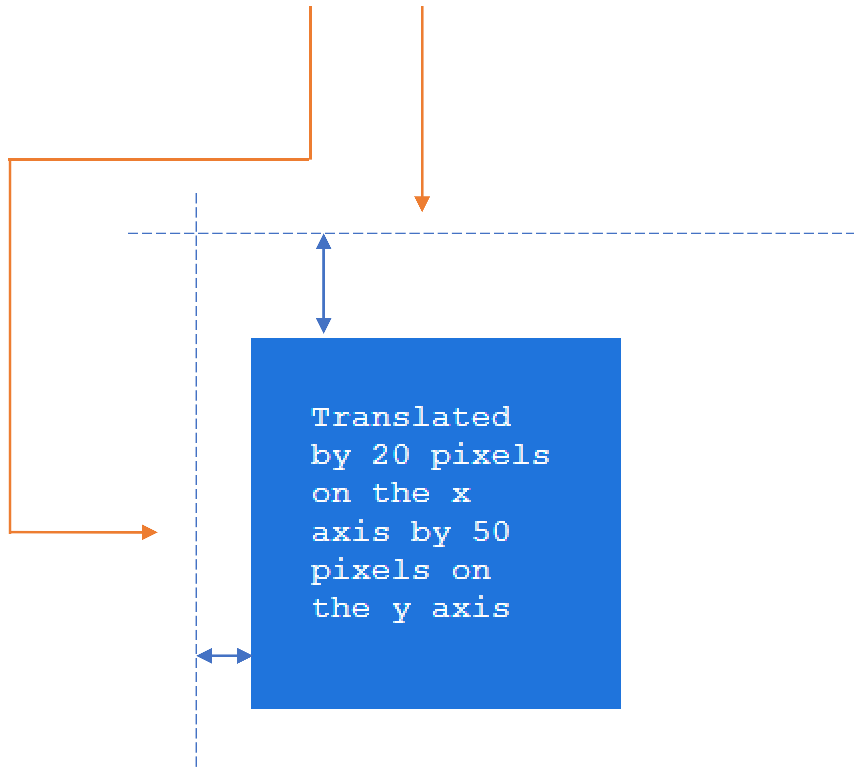
Not translated

Translated by 20 pixels on the x axis by 50 pixels on the y axis

translate 20px on the x axis

CSS3 Transforms - Translate

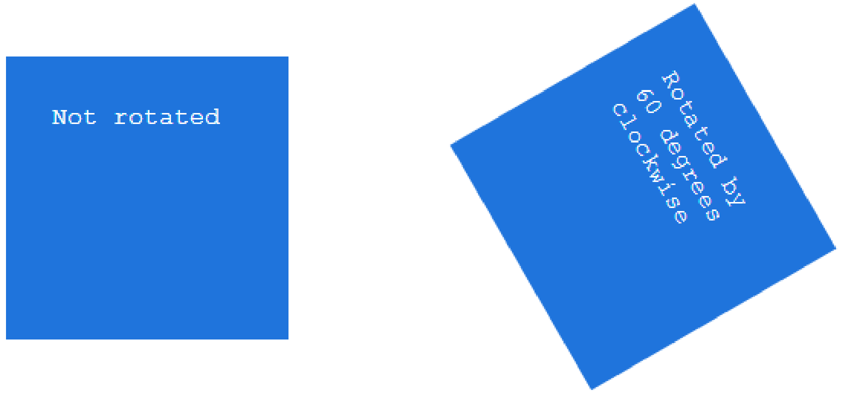
```
.box_23 {
    transform: translate(20px, 50px);
}
```



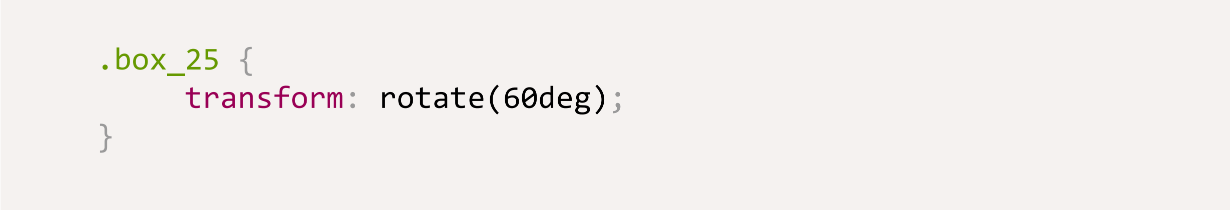
Translated by 20 pixels on the x axis by 50 pixels on the y axis

CSS3 Transforms - Rotate

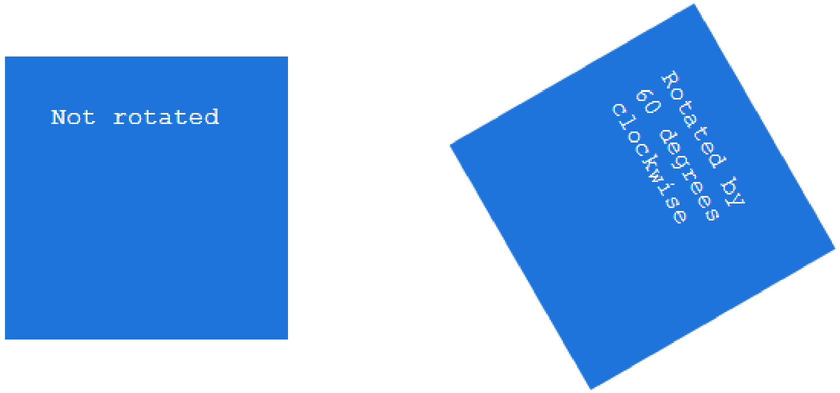
- Rotate transforms the element in a clockwise (positive values) or a counter-clockwise) direction
- Values are set in degrees (deg)



CSS3 Transforms - Rotate

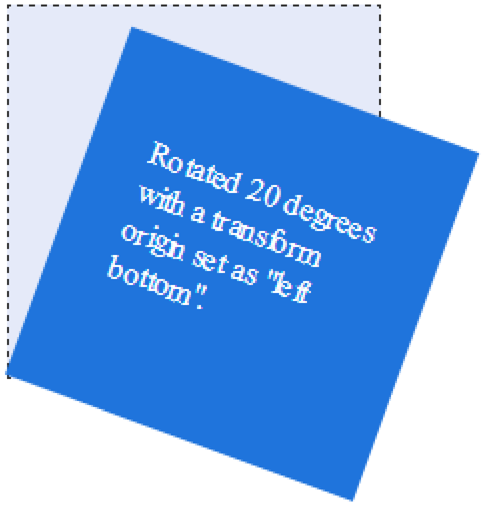


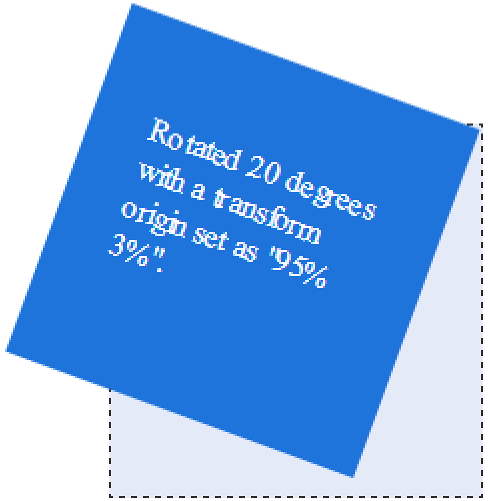
```
.box_25 {
      transform: rotate(60deg);
}
```



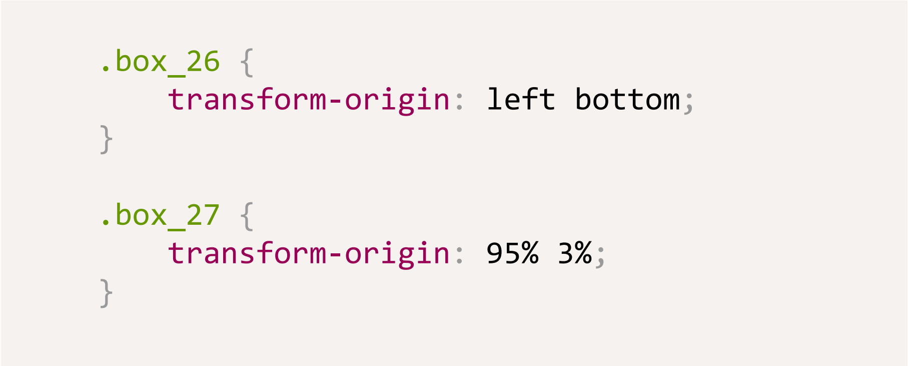
CSS3 Transforms - Origin

- Transform origin sets the transform point on the x and y axis.
- Possible values for the x axis and y axis are
    - left
    - center
    - right
    - Percentage (%).


Rotated 20 degrees with a transform origin set as "left bottom".

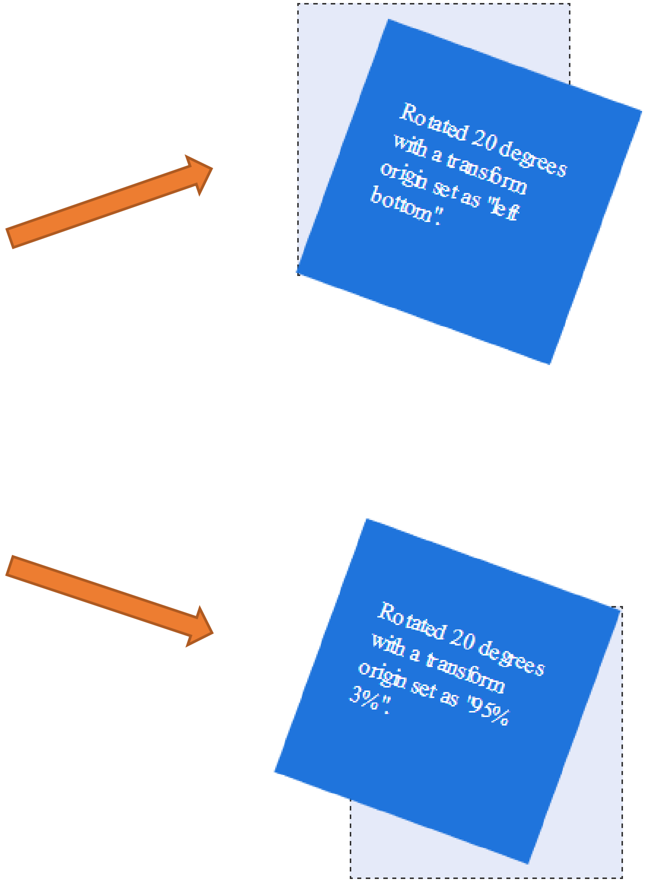


CSS3 Transforms - Origin



```
.box_26 {
    transform-origin: left bottom;
}

.box_27 {
    transform-origin: 95% 3%;
}
```

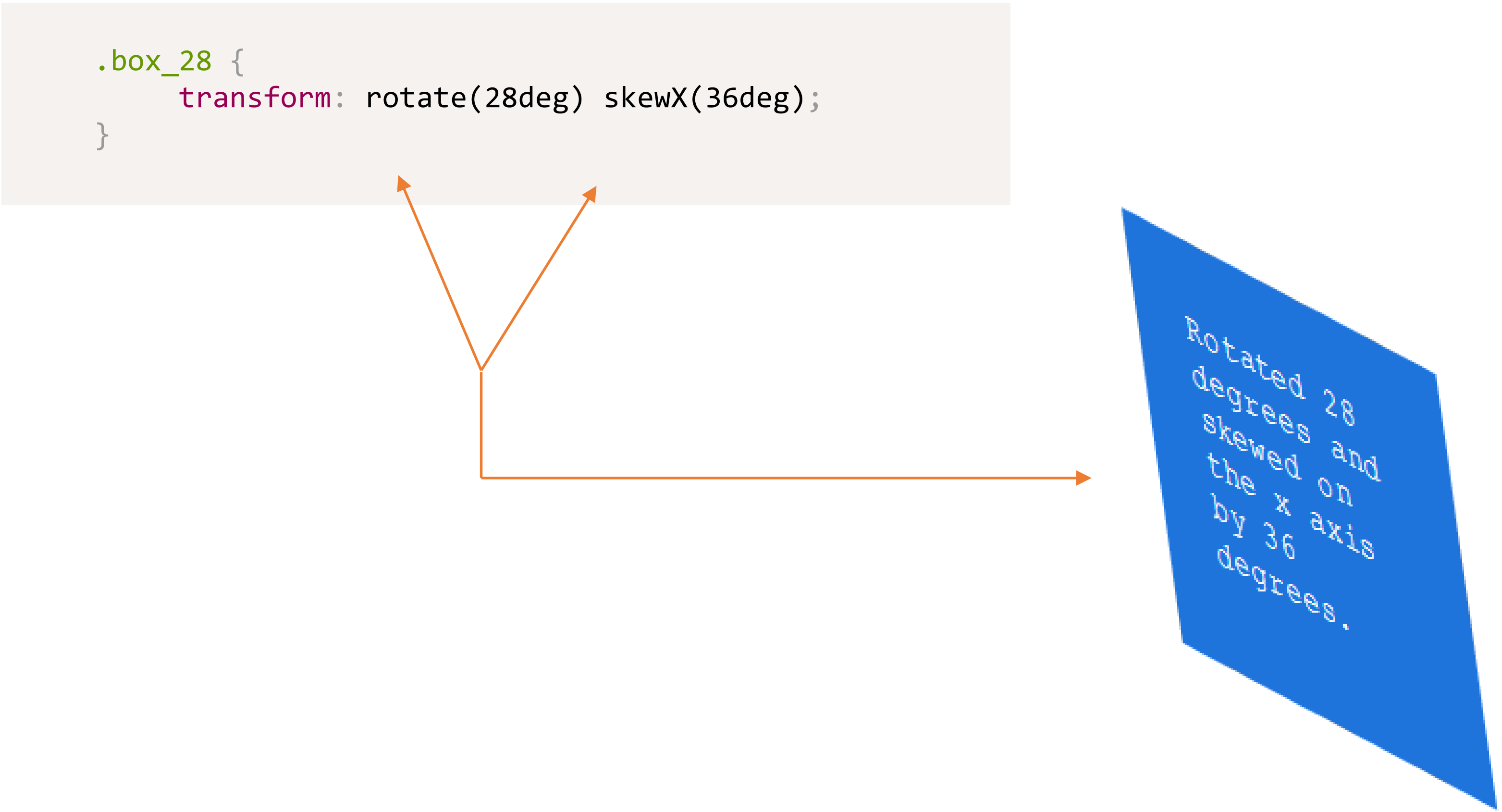


Rotated 20 degrees with a transform origin set as "left bottom".

Rotated 20 degrees with a transform origin set as "95% 3%".

CSS3 Transforms - Multiple

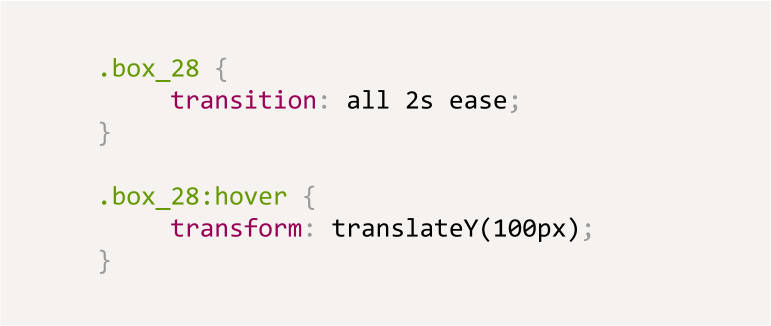
- You can set multiple transform types on a single line by separating the transforms with a space



```
.box_28 {
      transform: rotate(28deg) skewX(36deg);
}
```

CSS 3 – Transitions (introduction)

- Transitions smooth out the “transition” from various states
    - Example: regular state to hover



```
.box_28 {
     transition: all 2s ease;
}

.box_28:hover {
     transform: translateY(100px);
}
```

CSS Transistions

- Browser compatibility for CSS Transforms:

IE10

Firefox 4+

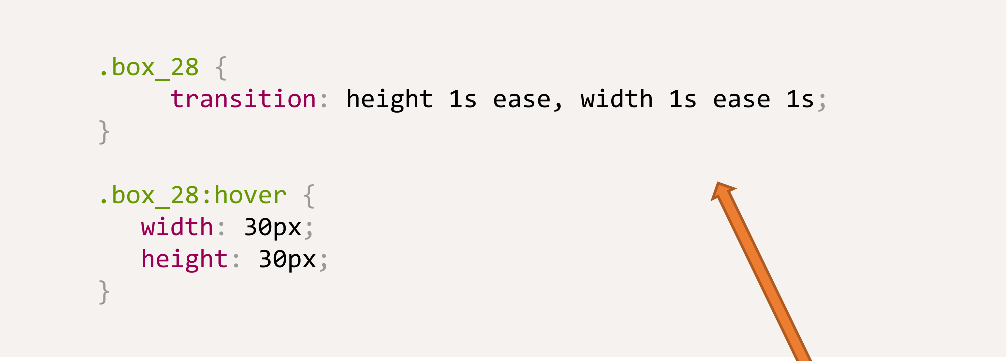
Chrome 4+

Safari 3.1+

Opera 10.5+

CSS 3 – Transitions Multiple

- You can transition a single property, all properties (that change) or multiple properties
  - If you transform multiple specific properties you can have the transitions run one after the other by setting the delay property of the second, third, fourth, etc. transitioned elements to be equal to the total transition time of the elements that ran before it



```
.box_28 {
     transition: height 1s ease, width 1s ease 1s;
}

.box_28:hover {
   width: 30px;
   height: 30px;
}
```

Notice the second time of “1s” after the easing. This is the delay parameter and sets the delay. Here we have set the delay to be equal to the time of the transition of the height. This will cause the width transition to run after the height transition