CSS

Gradients, Transforms, Transitions, Animations

Gradients

- Linear Gradients (goes down/up/left/right/diagonally)

- Radial Gradients (defined by their center)

Linear Gradients

background(-image): linear-gradient(**to** *direction*, *color-stop1*,*color-stop2, ...*);

Ex: background: linear-gradient(to top, red, blue);

Linear Gradient - Diagonal

You can make a gradient diagonally by specifying both the horizontal and vertical starting positions.

```
linear-gradient(to top left, blue, red);
```

An early syntax where the starting corner or side was indicated without the to keyword, and effectively considered as a *from* position.

```
-moz-linear-gradient(bottom right, blue, red);
```

Linear Gradients

A linear gradient that starts at bottom right (and goes to top left)

```
#grad1 {
    height: 200px;
    background: -webkit-linear-gradient(bottom right, red , blue);
    background: -o-linear-gradient(bottom right, red, blue);
    background: -moz-linear-gradient(bottom right, red, blue);
    background: linear-gradient(to top left, red , blue);
    /* Standard syntax (must be last) */
}
```

Linear Gradients

Using Angles

If you want more control over the direction of the gradient, you can define an angle, instead of the predefined directions (to bottom, to top, to right,..)

A linear gradient with a specified angle:

```
#grad {
  background: -webkit-linear-gradient(45deg, red, blue);
  background: -o-linear-gradient(45deg, red, blue);
  background: -moz-linear-gradient(45deg, red, blue);
  background: linear-gradient(45deg, red, blue);
}
```

Linear Gradients

multiple colors:

```
#grad1 {
    height: 55px;
    background: -webkit-linear-gradient(left, red, orange, yellow, green, blue);
    background: -o-linear-gradient(left, red, orange, yellow, green, blue);
    background: -moz-linear-gradient(left, red, orange, yellow, green, blue);
    background: linear-gradient(to right, red, orange, yellow, green, blue);
   }
```

Linear Gradients

Repeating a linear-gradient

The repeating-linear-gradient() function is used to repeat linear gradients:

```
#grad {
  background: -webkit-repeating-linear-gradient(left, red, yellow, green 20%);
  background: -o-repeating-linear-gradient(left, red, yellow, green 20%);
  background: -moz-repeating-linear-gradient(left, red, yellow, green 20%);
  background: repeating-linear-gradient(to right, red, yellow, green 20%);
}
```

Radial Gradients

background(-image): radial-gradient(shape size at position, start-color, ..., last-color);

By default, shape is ellipse, size is farthest-corner, and position is center.

<shape>
The gradient's shape. This is one of circle (meaning that the gradient's shape is a circle with constant radius) or ellipse (meaning that the shape is an axis-aligned ellipse). The default value is ellipse.

Radial Gradients

<size> The size of the gradient. This is one of the Size constants listed below.

- ➢ closest-side: The gradient's ending shape meets the side of the box closest to its center (for circles) or meets both the vertical and horizontal sides closest to the center (for ellipses).
- ➢ closest-corner: The gradient's ending shape is sized so it exactly meets the closest corner of the box from its center.
- ➢ farthest-side: Similar to closest-side, except the ending shape is sized to meet the side of the box farthest from its center (or vertical and horizontal sides).
- ➢ farthest-corner: The gradient's ending shape is sized so it exactly meets the farthest corner of the box from its center.

Radial Gradients

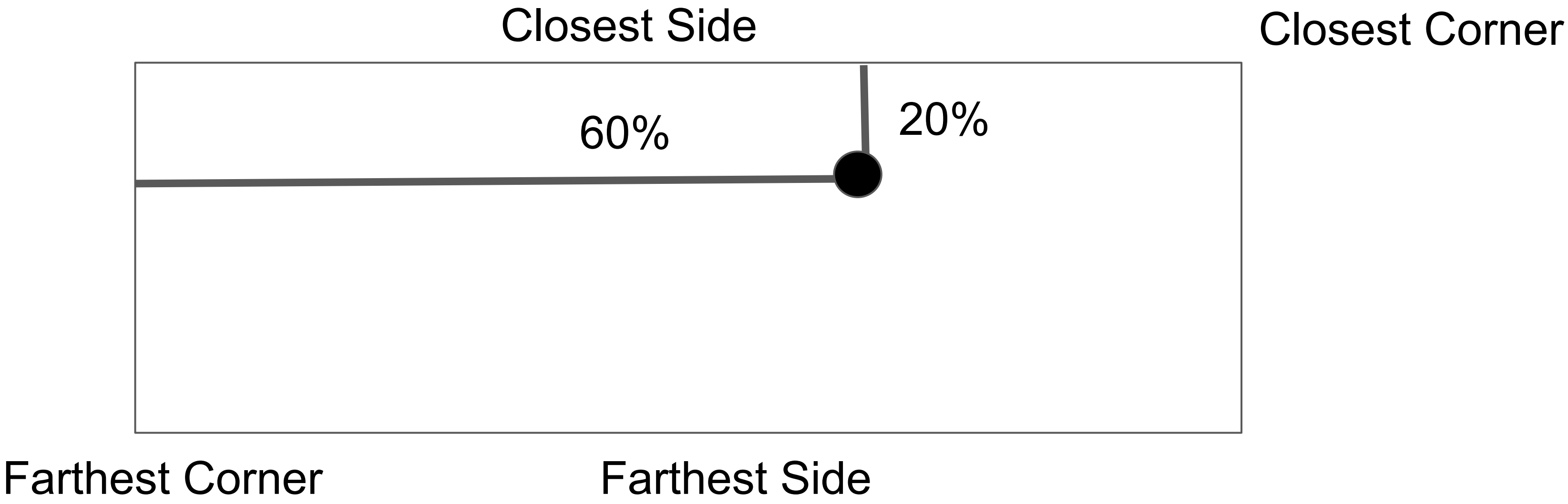
<position>

A <position>, interpreted in the same way as background-position. If omitted, the default is center.

<color-stop>

Representing a fixed color at a precise position, this value is composed by a <color> value, followed by an optional stop position (either a <percentage> or a <length>).

Radial Gradients



Radial Gradients

Radial Gradient - Evenly Spaced Color Stops (this is default)

A radial gradient with evenly spaced color stops:

```
#grad {
  background: -webkit-radial-gradient(red, green, blue);
  background: -o-radial-gradient(red, green, blue);
  background: -moz-radial-gradient(red, green, blue);
  background: radial-gradient(red, green, blue);
}
```

Radial Gradients

Radial Gradient - Differently Spaced Color Stops

A radial gradient with differently spaced color stops:

```
#grad {
  background: -webkit-radial-gradient(red 5%, green 15%, blue 60%);
  background: -o-radial-gradient(red 5%, green 15%, blue 60%);
  background: -moz-radial-gradient(red 5%, green 15%, blue 60%);
  background: radial-gradient(red 5%, green 15%, blue 60%);
}
```

Radial Gradients

Repeating a radial-gradient

The repeating-radial-gradient() function is used to repeat radial gradients:

A repeating radial gradient:

```
#grad {
  background: -webkit-repeating-radial-gradient(red, yellow 10%, green 15%);
  background: -o-repeating-radial-gradient(red, yellow 10%, green 15%);
  background: -moz-repeating-radial-gradient(red, yellow 10%, green 15%);
  background: repeating-radial-gradient(red, yellow 10%, green 15%);
}
```

Transforms

CSS3 transforms allow you to translate, rotate, scale, and skew elements.

A transformation is an effect that lets an element change shape, size and position.

CSS3 supports 2D and 3D transformations.

Transforms

2D Transformations

- ➢ translate(): The translate(x,y) method moves an element from its current position (according to the parameters given for the X-axis and the Y-axis)
- ➢ rotate(): The rotate(xdeg) method rotates an element clockwise or counter-clockwise according to a given degree. Using negative values will rotate the element counter-clockwise.

Transforms

2D Transformations

➢ scale(): The scale(width, height) method increases or decreases the size of an element (according to the parameters given for the width and height).

➢ skew(): The skew(x, y) method skews an element along the X and Y-axis by the given angles. If the second parameter is not specified, it has a zero value.

transform-origin: lets you modify the origin for transformations of an element

Transforms

3D Transformations

perspective: The perspective property defines how many pixels a 3D element is placed from the view. This property allows you to change the perspective on how 3D elements are viewed.

backface-visibility : Defines whether or not an element should be visible when not facing the screen.

Transitions

CSS3 transitions

Allows you to change property values smoothly (from one value to another), given a duration.

To create a transition effect, you must specify two things:

- ➢ the CSS property you want to add an effect to
- ➢ the duration of the effect

The transition effect will start when the specified CSS property (width) changes value.

Transitions

transition:

A shorthand property for setting the four transition properties into a single property (property duration timing-function delay)

OR define each property on it’s own

transition-property:

Specifies the name of the CSS property the transition effect is for

transition-duration

Specifies how many seconds or milliseconds a transition effect takes to complete

Transitions

transition-timing-function

Specifies the speed curve of the transition effect

- ➢ ease - (default)specifies a transition with a slow start, then fast, then end slowly
- ➢ linear - specifies a transition with the same speed from start to end
- ➢ ease-in - specifies a transition with a slow start
- ➢ ease-out - specifies a transition with a slow end
- ➢ ease-in-out - specifies a transition with a slow start and end
- ➢ cubic-bezier(n,n,n,n) - custom values in a cubic-bezier function

Transitions

transition-delay:

Specifies a delay (in seconds) for the transition effect, default is 0.

Transitions can be done on one or more properties in the same rule by separating them with commas.

Ex: transition: width .5s ease-out, color .25s linear, padding 2s ease-in;

transition: all .5s linear; performs transitions on all changed properties.

Transitions can be used with transforms hence transforms are properties.

Animations

An animation lets an element gradually change from one style to another.

You can change as many CSS properties you want, as many times you want.

To use CSS3 animation, you must first specify some keyframes for the animation.

Keyframes hold what styles the element will have at certain times.

Animations

@keyframes Specifies the animation code

animation:
A shorthand property for setting all the animation properties.

animation: name duration timing-function delay iteration-count direction fill-mode play-state;

Each animation property can be defined separate.

Animations

animation-name:

Specifies the name of the @keyframes animation

animation-duration:

Specifies how many seconds or milliseconds an animation takes to complete one cycle

animation-timing-function:

Specifies the speed curve of the animation

ease(default), linear, ease-in, ease-out, ease-in-out, cubic-bezier(n,n,n,n)

Animations

animation-delay

Specifies a delay for the start of an animation

animation-iteration-count

Specifies the number of times an animation should be played

number or infinite

Animations

animation-fill-mode

Specifies a style for the element when the animation is not playing (when it is finished, or when it has a delay)

animation-play-state

Specifies whether the animation is running or paused (paused, running)

Animations

animation-direction

Specifies the animation direction

- normal: Default value. The animation should be played as normal
- reverse: The animation should play in reverse direction
- alternate: The animation will be played as normal every odd time (1,3,5,etc..) and in reverse direction every even time (2,4,6,etc...)
- alternate-reverse: The animation will be played in reverse direction every odd time (1,3,5,etc..) and i in a normal direction every even time (2,4,6,etc...)