kodegod.com

CSS

by Prashant Chaudhari

Index

- Introduction
- Use
- Advantages
- Ways Of Styling
- Syntax
- Selectors
- Margin
- Padding
- Borders
- Color
- Background
- Fonts
- Text
- Links
- Lists
- Tables
- Outline
- Cursors
- Overflow
- Dimension
- Units
- Display
- Visibility
- Position

Index

- Layers
- Float
- Alignment
- Pseudo Classes
- Pseudo Elements
- Media Types
- Opacity
- Gradients
- Drop Shadow
- 2D Transforms
- 3D Transforms
- Transitions
- Animations
- Multi-Column layout
- Box Sizing
- Flexible Box Layout
- Filters
- Media Queries

Introduction

CSS3 by Prashant Chaudhari

Introduction

- Cascading Style Sheets
- Presentational Technology
- CSS introduced in 1996 by the W3C
- Separation of presentation and content
- Simpler HTML markup and better maintainability

Who can learn?

- Anyone
- Students
- Professionals

What is needed?

- Time, patience and dedication
- Completing tutorial

Use of CSS

CSS3 by Prashant Chaudhari

Use of CSS

- Same style rules on multiple elements
- Control presentation of multiple pages with single stylesheet
- Can style dynamic states of elements (hover, focus, etc.)
- Changing position of elements
- Transform elements (scale, rotate, etc.)
- Animation and transition effects

Advantages

CSS3 by Prashant Chaudhari

Advantages

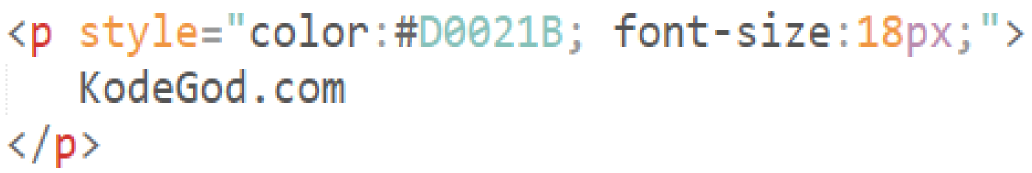
- Pages load faster
- Superior styles to HTML
- Save lots of time
- Easy maintenance
- Multiple device compatibility

Ways of Styling

CSS3 by Prashant Chaudhari

Ways of Styling

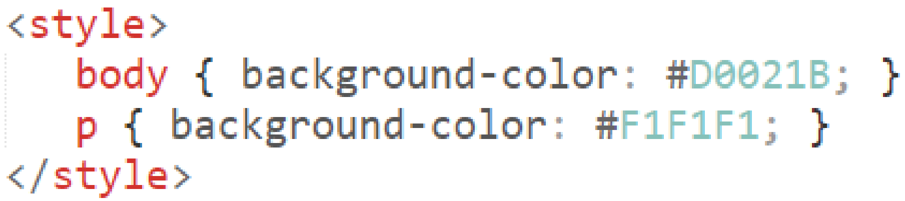
- **Three ways to include css in HTML**
- **Inline Styles**
  - Using **style** attribute in HTML start tag
  - Property and value pairs
  - Each pair separated by ;



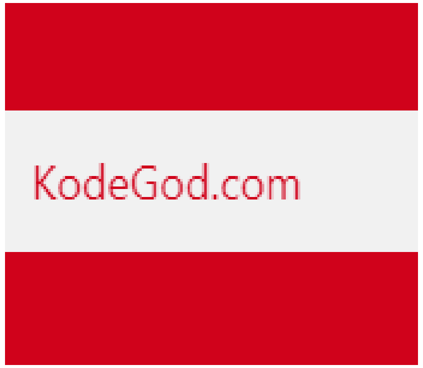
```
<p style="color:#D0021B; font-size:18px;">
   KodeGod.com
</p>
```

Ways of Styling

- **Embedded Style Sheets**
  - Only effect in same document
  - Defined in **<head>** tag using **<style>** element
  - Can define any number of **<style>** elements in **<head>**



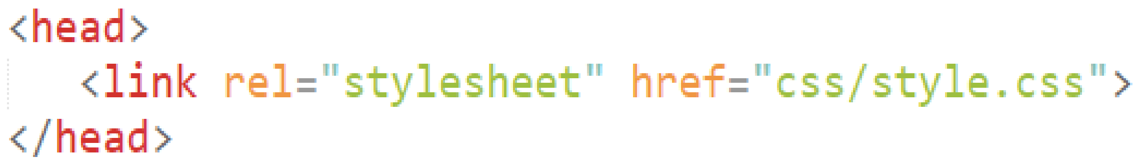
```
<style>
    body { background-color: #D0021B; }
    p { background-color: #F1F1F1; }
</style>
```



KodeGod.com

Ways of Styling

- **External Style Sheets**
  - Applied to many pages
  - Multiple style rules in separate document
  - Two ways to attach **Linking, Importing**
- **Linking**
  - External style sheet file using **<link>** tag in **<head>**



```
<head>
   <link rel="stylesheet" href="css/style.css">
</head>
```

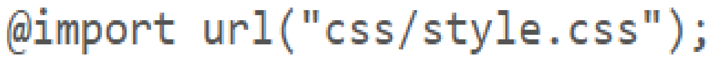
Ways of Styling

- **Importing**

  1. Using **@import** rule inside **<style>** tag inside

     <head> tag

  2. Using **@import** in another style sheets



```
@import url("css/style.css");
```

Syntax

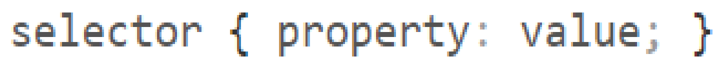
CSS3 by Prashant Chaudhari

Syntax

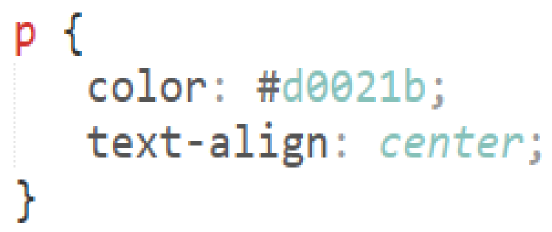
- **Two parts**
  - Selector and one or more declarations

- Use **/* -comment - */** to write comments

  in css file



```
selector { property: value; }
```



```
p {
    color: #d0021b;
    text-align: center;
}
```

Selectors

CSS3 by Prashant Chaudhari

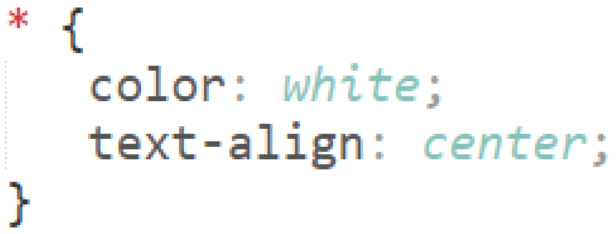
Selectors

- Used to select elements to style
- **Types of selectors**
    - **Universal Selector**

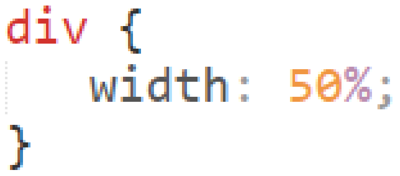
      Uses * to match every element on page

    - **Element Type Selector**

      Matches every instance of element in document



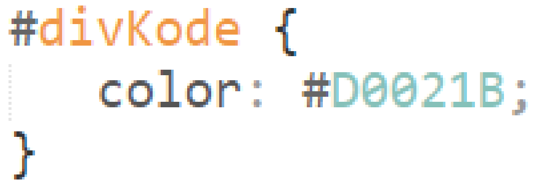
```css
* {
    color: white;
    text-align: center;
}
```



```css
div {
    width: 50%;
}
```

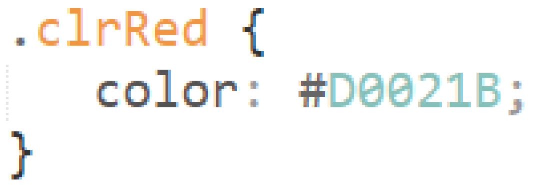
Selectors

- **ID Selector**
  - For single or unique element
  - Defined by **(#)** symbol followed by id value



```
#divKode {
    color: #D0021B;
}
```

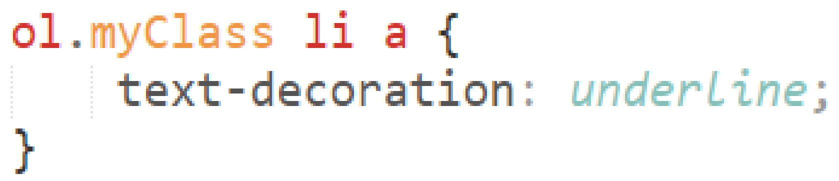
- **Class Selector**
  - Matches every instance of element that has a similar **class** attribute value
  - Defined by **(.)** symbol followed by class value



```
.clrRed {
    color: #D0021B;
}
```

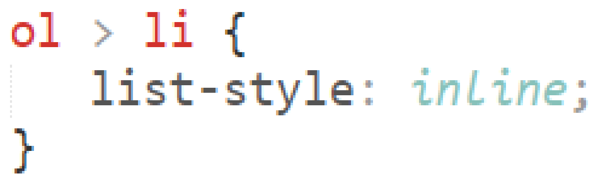
Selectors

- **Descendant Selector**
  - To select element that is descendant of other element



```css
ol.myClass li a {
    text-decoration: underline;
}
```

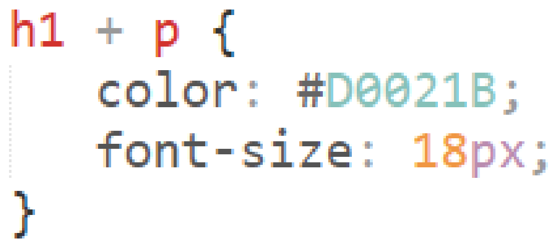
- **Child Selector**
  - Select direct children of some element
  - Defined by **(>)** symbol



```css
ol > li {
   list-style: inline;
}
```

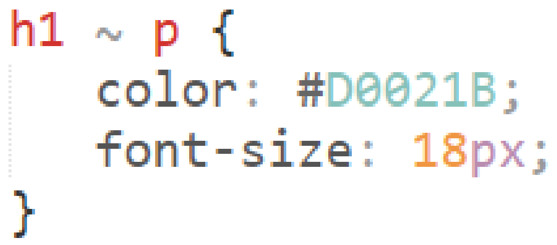
Selectors

- **Adjacent Sibling Selector**
  - To select sibling elements
  - **Syntax** E1 + E2



```css
h1 + p {
    color: #D0021B;
    font-size: 18px;
}
```

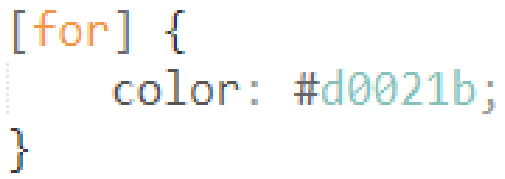
- **General Sibling Selector**
  - Similar to adjacent, but less strict
  - Separated by **(~)** symbol
  - **Syntax** E1 ~ E2



```css
h1 ~ p {
    color: #D0021B;
    font-size: 18px;
}
```

Selectors

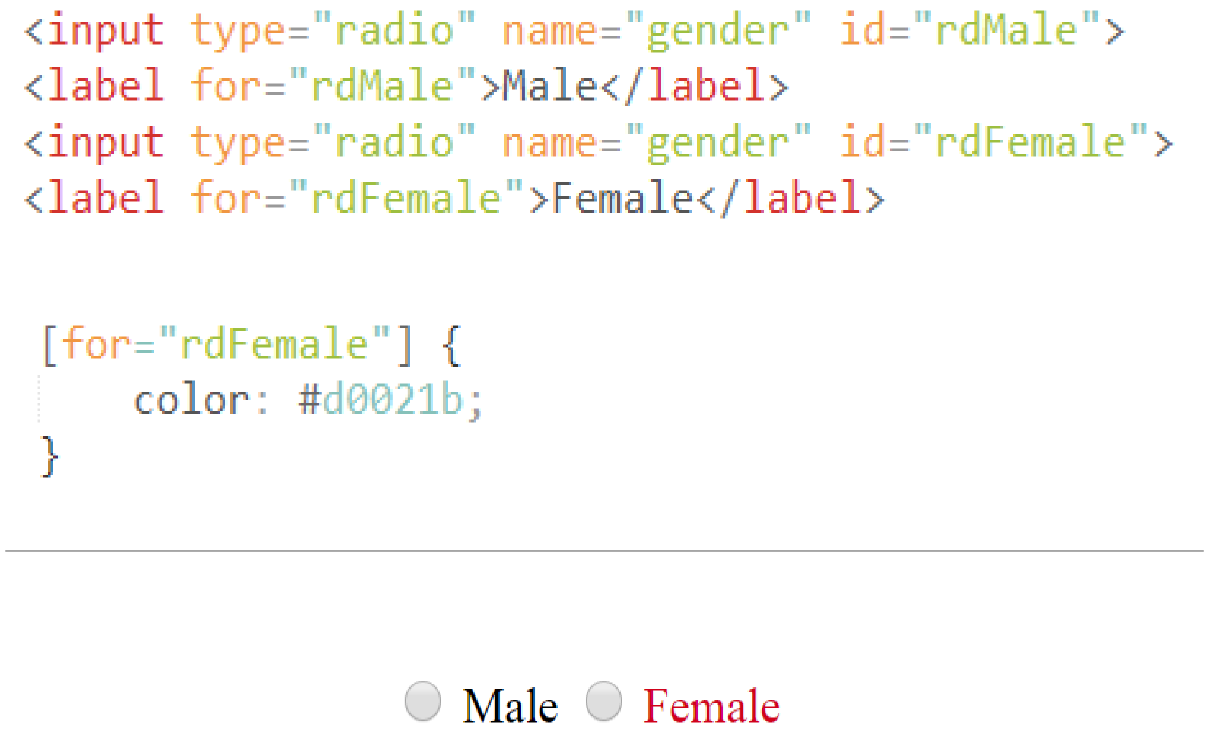
- **Attribute Selector**
  - Selects element with specific attribute or attribute with a specified value
  - Use **square brackets** to create attribute selectors
  - **Common types**
    - **[attribute]**
      - Selects all elements that matches given attribute



```
[for] {
    color: #d0021b;
}
```

Selectors

- **Attribute Selector**
  - **[attribute="value"]**
    - Selects all elements that matches given attribute with exactly same value



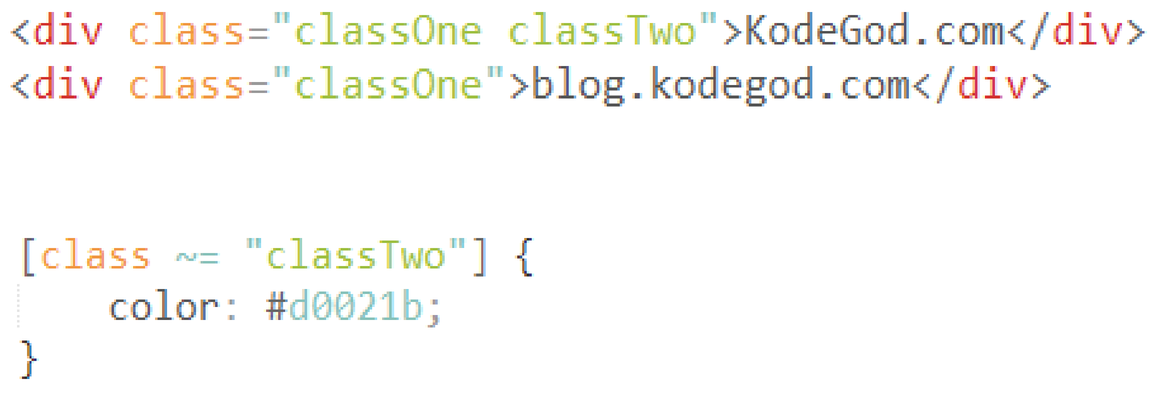
```
<input type="radio" name="gender" id="rdMale">
<label for="rdMale">Male</label>
<input type="radio" name="gender" id="rdFemale">
<label for="rdFemale">Female</label>
```

```
[for="rdFemale"] {
    color: #d0021b;
}
```

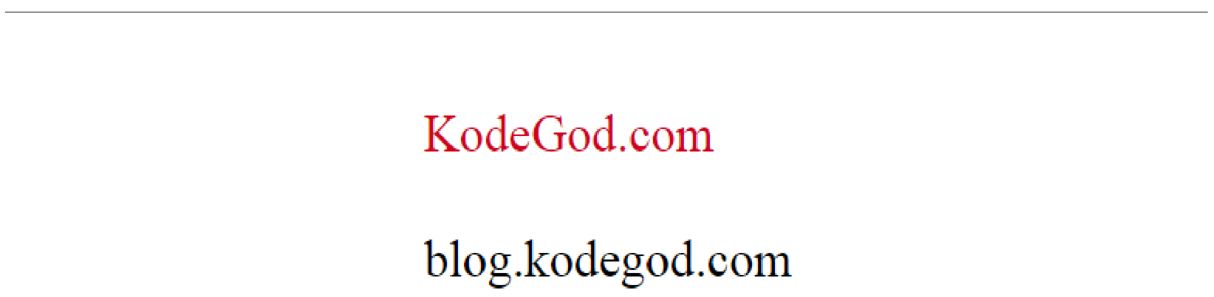
○ Male ○ Female

Selectors

- **Attribute Selector**
    - **[attribute ~="value"]**
        - Selects all elements whose attribute value is list of space separated values one of which is exactly same as specified value



```
<div class="classOne classTwo">KodeGod.com</div>
<div class="classOne">blog.kodegod.com</div>


[class ~= "classTwo"] {
    color: #d0021b;
}
```

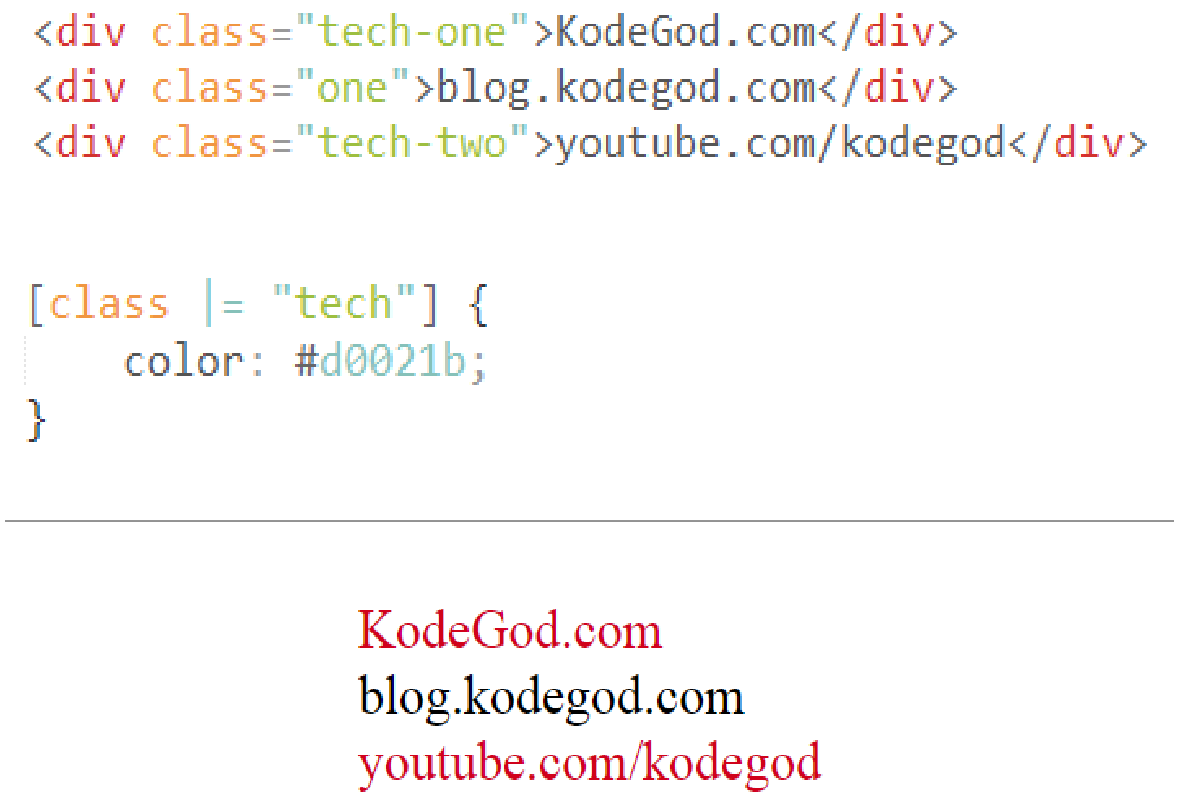


KodeGod.com

blog.kodegod.com

Selectors

- **Attribute Selector**
  - **[attribute |="value"]**
    - Selects all elements whose attribute has hyphen-separated list of values beginning with specified value



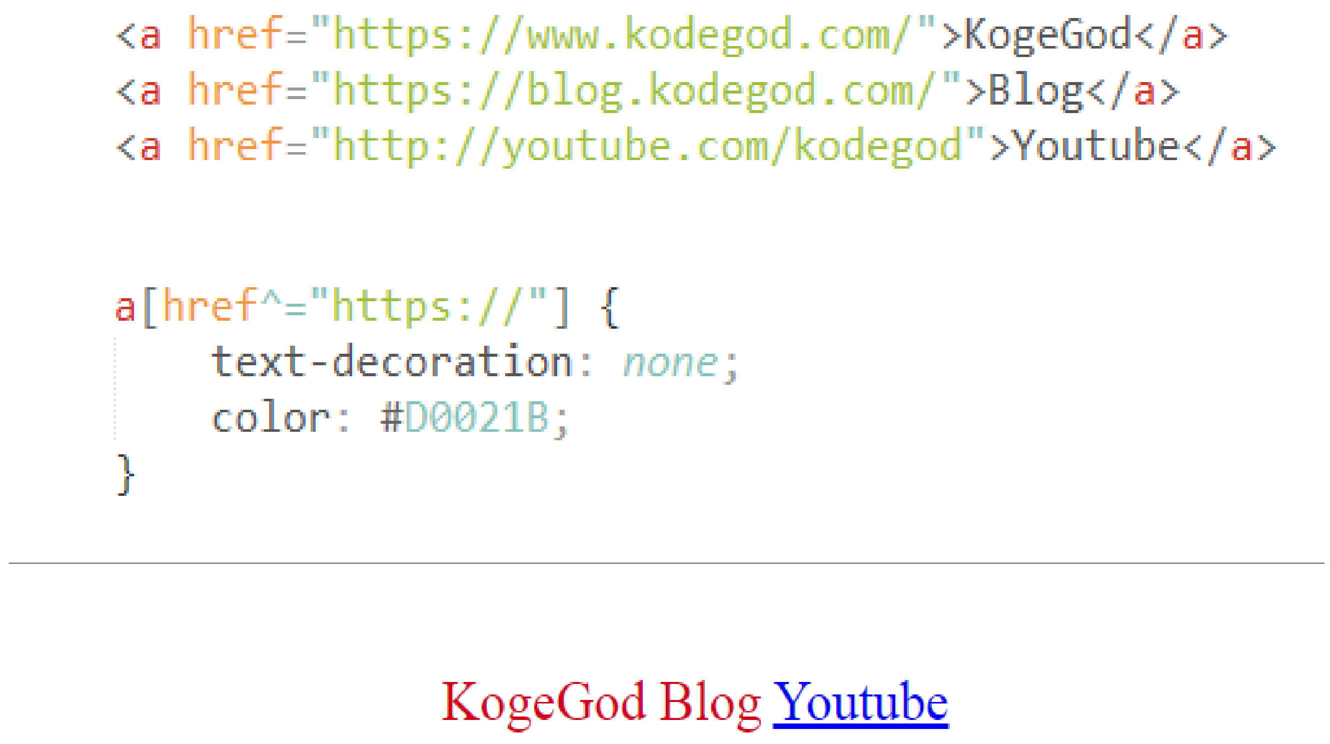
```html
<div class="tech-one">KodeGod.com</div>
<div class="one">blog.kodegod.com</div>
<div class="tech-two">youtube.com/kodegod</div>
```

```css
[class |= "tech"] {
    color: #d0021b;
}
```

KodeGod.com
blog.kodegod.com
youtube.com/kodegod

Selectors

- **Attribute Selector**
  - **[attribute ^="value"]**
    - Selects all elements whose attribute value starts with a specified value

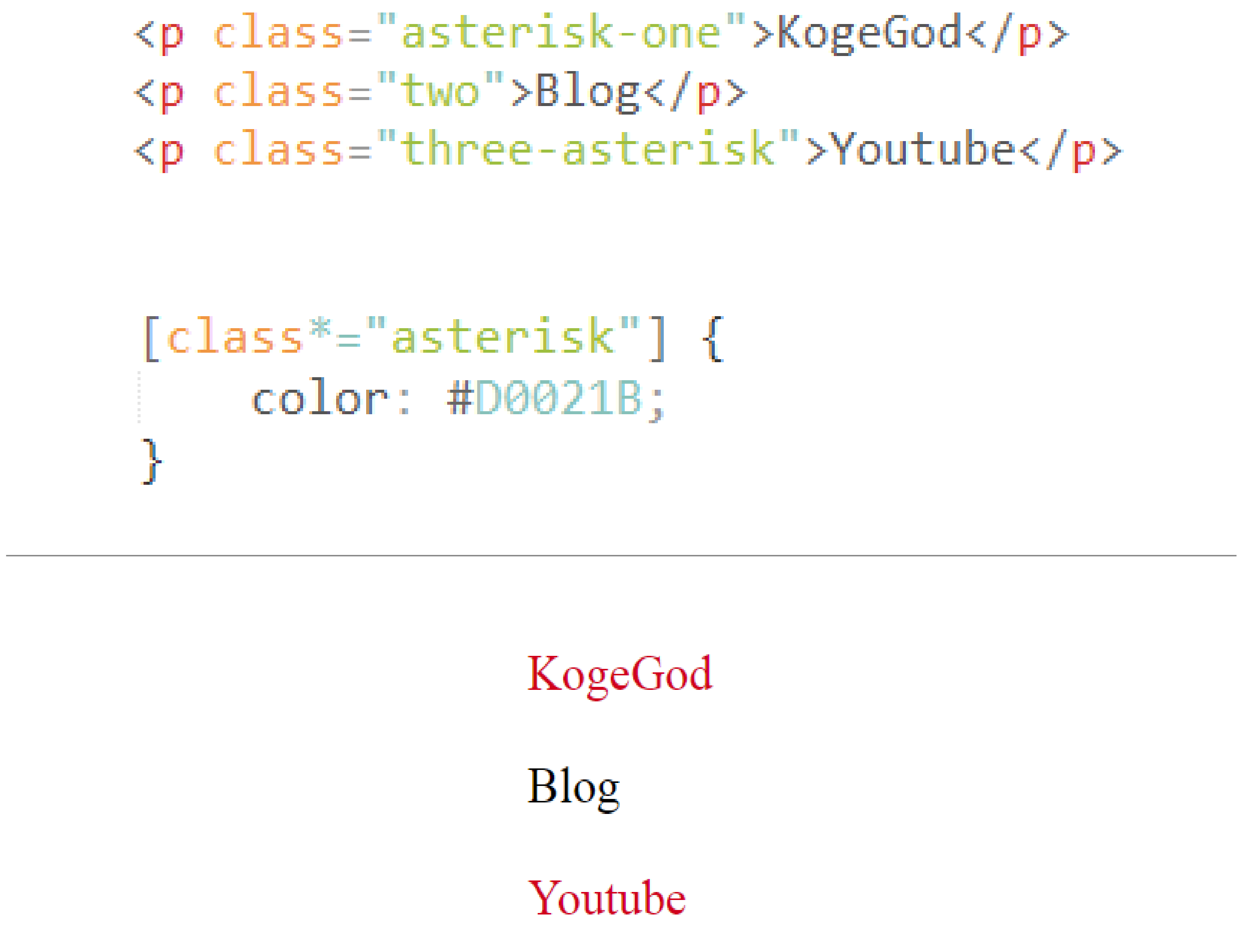


```
<a href="https://www.kodegod.com/">KogeGod</a>
<a href="https://blog.kodegod.com/">Blog</a>
<a href="http://youtube.com/kodegod">Youtube</a>


a[href^="https://"] {
    text-decoration: none;
    color: #D0021B;
}
```

KogeGod Blog Youtube

Selectors

- **Attribute Selector**
  - **[attribute *="value"]**
    - Selects all elements whose attribute value contains a specified value



```
<p class="asterisk-one">KogeGod</p>
<p class="two">Blog</p>
<p class="three-asterisk">Youtube</p>
```

```
[class*="asterisk"] {
    color: #D0021B;
}
```

KogeGod

Blog

Youtube

Margin

CSS3 by Prashant Chaudhari

Margin

- Clears an area around the sides of an element box
- No background-color, completely transparent
- Can set margins to top, right, bottom, left

- Can set margin to all sides using single property named **margin**
- E.g.

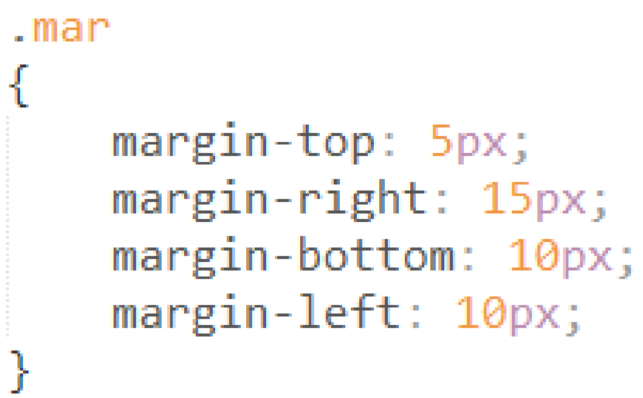
  margin : top right bottom left;

  OR

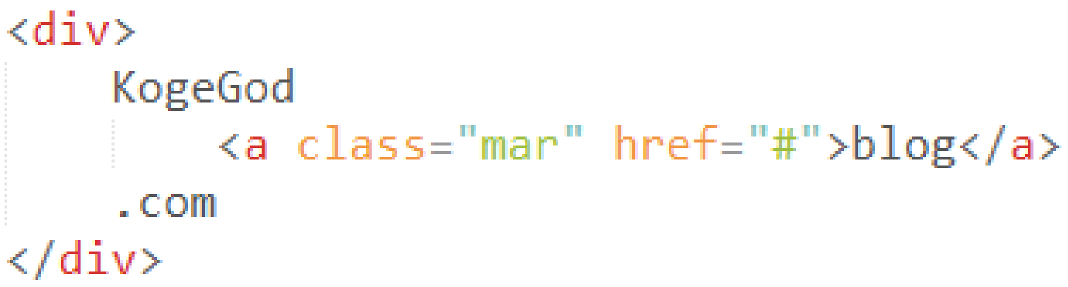
  margin : top|bottom right|left;

Margin

- E.g. (For each side)

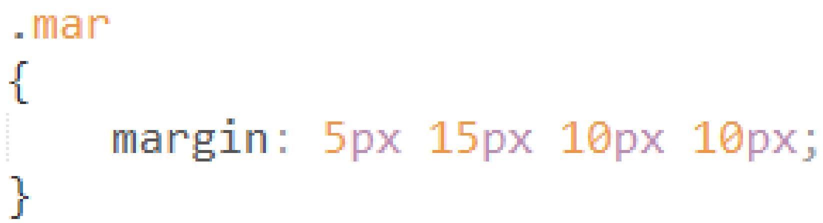


```
.mar
{
    margin-top: 5px;
    margin-right: 15px;
    margin-bottom: 10px;
    margin-left: 10px;
}
```

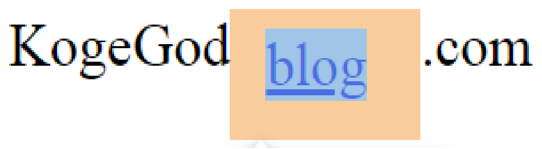


```
<div>
    KogeGod
        <a class="mar" href="#">blog</a>
    .com
</div>
```

- E.g. (All in one)



```
.mar
{
    margin: 5px 15px 10px 10px;
}
```



KogeGod blog .com

Padding

CSS3 by Prashant Chaudhari

Padding

- Separates elements border from its content
- Affected by background-color
- Can set padding to each sides
- Can set padding to all sides using single property named **padding**

- E.g.

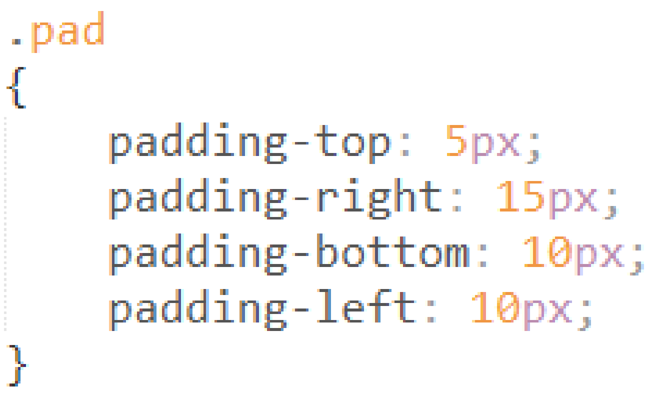
  padding : top right bottom left;

  OR

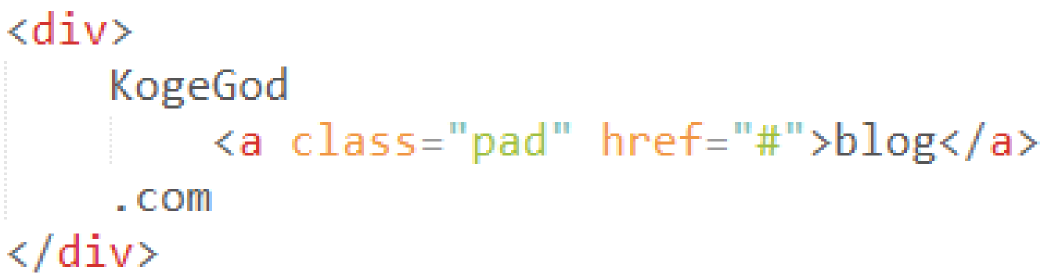
  padding : top|bottom right|left;

Padding

- E.g. (For each side)

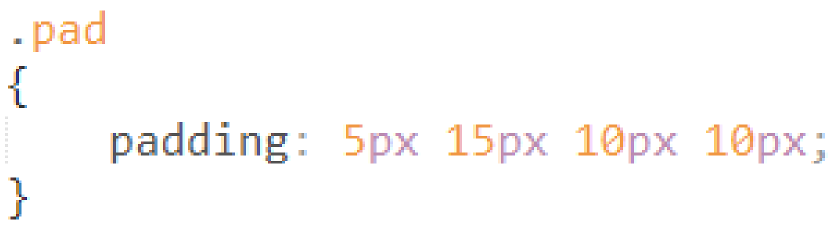


```
.pad
{
    padding-top: 5px;
    padding-right: 15px;
    padding-bottom: 10px;
    padding-left: 10px;
}
```

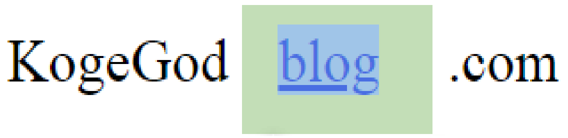


```
<div>
    KogeGod
        <a class="pad" href="#">blog</a>
    .com
</div>
```

- E.g. (All in one)



```
.pad
{
    padding: 5px 15px 10px 10px;
}
```



KogeGod blog .com

Borders

CSS3 by Prashant Chaudhari

Borders

- Defines border area of box
- Goes around padding and content
- **Properties**
  - **border-width**

    sets width of border
  - **border-style**

    sets style of box's border
  - **border-color**

    sets border color

Border – border-styles

- **none**

  No border

- **hidden**

  Hidden border

- **dotted**

  Dotted border

- **dashed**

  Dashed border

- **solid**

  Solid border

- **double**

  Two solid borders

- **groove**

  3D grooved border, depends on border color value

- **ridge**

  3D ridge border, depends on border color value

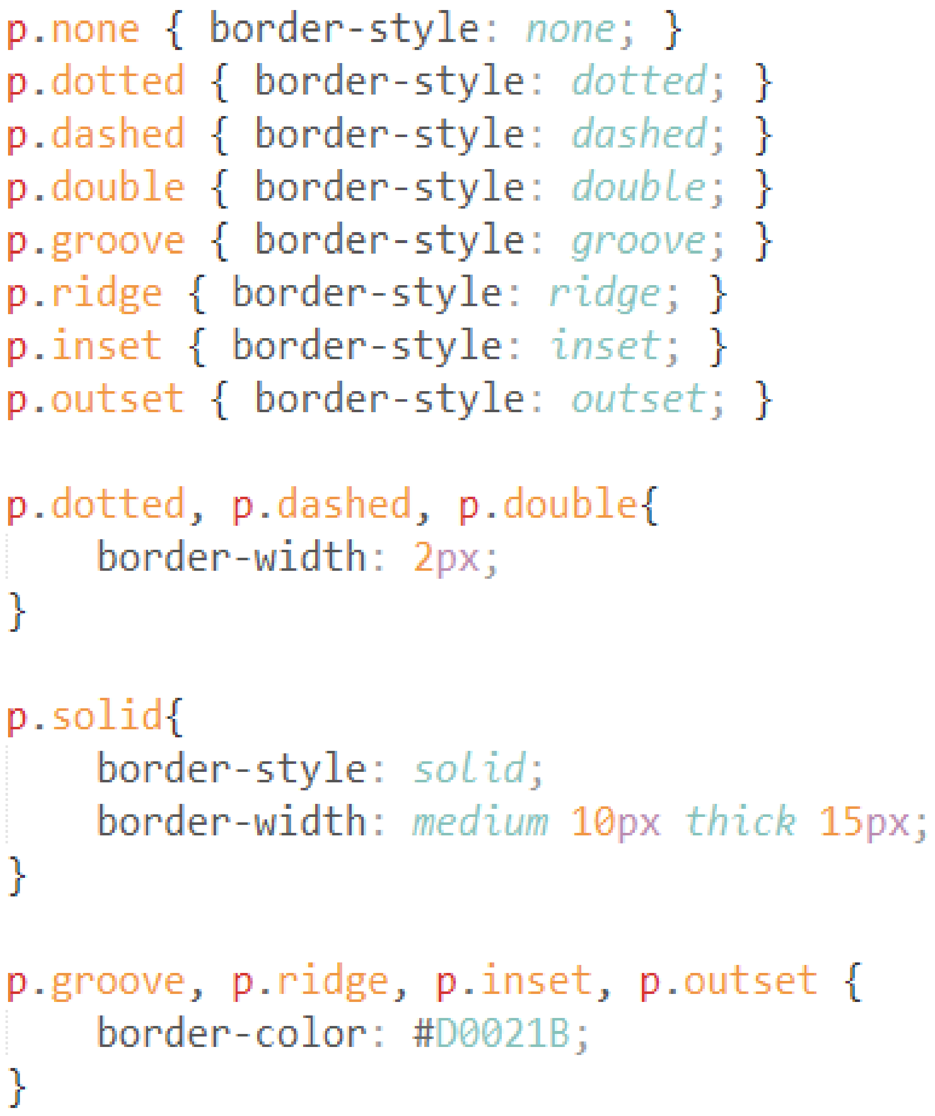
- **inset**

  3D inset border, depends on border color value

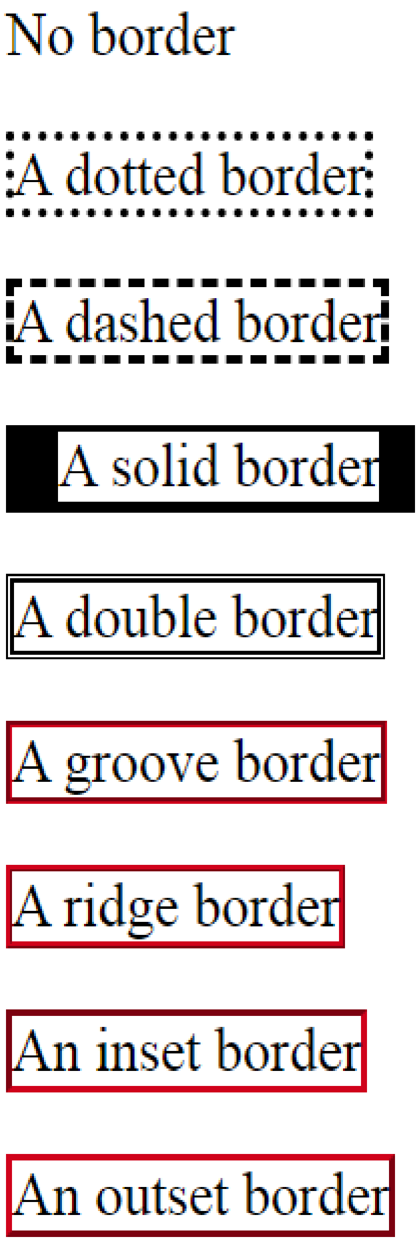
- **outset**

  3D outset border, depends on border color value

Border – Example



```
p.none { border-style: none; }
p.dotted { border-style: dotted; }
p.dashed { border-style: dashed; }
p.double { border-style: double; }
p.groove { border-style: groove; }
p.ridge { border-style: ridge; }
p.inset { border-style: inset; }
p.outset { border-style: outset; }

p.dotted, p.dashed, p.double{
    border-width: 2px;
}

p.solid{
    border-style: solid;
    border-width: medium 10px thick 15px;
}

p.groove, p.ridge, p.inset, p.outset {
    border-color: #D0021B;
}
```



No border

A dotted border

A dashed border

A solid border

A double border

A groove border

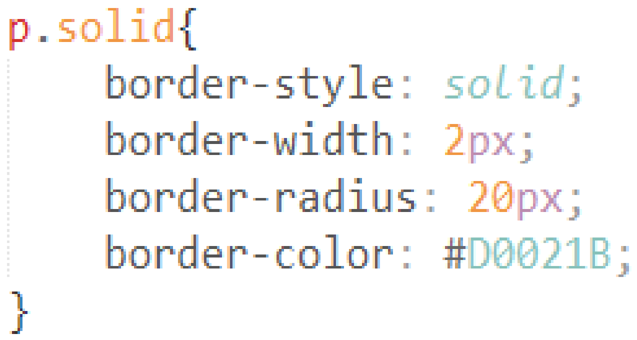
A ridge border

An inset border

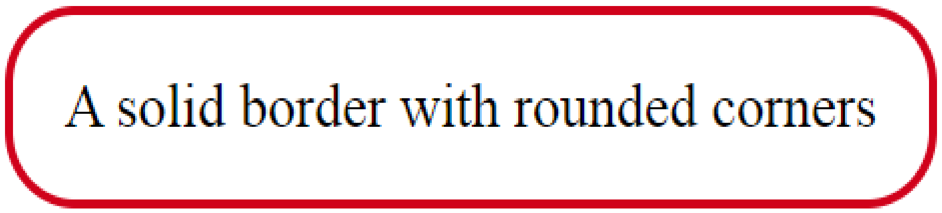
An outset border

Borders

- **Two new properties in CSS3**
  - **border-radius**
    - Making rounded corners



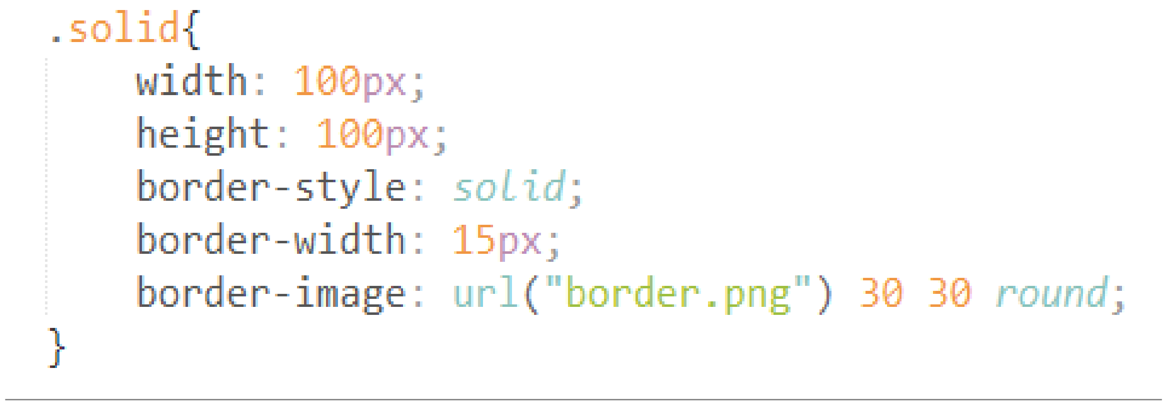
```
p.solid{
    border-style: solid;
    border-width: 2px;
    border-radius: 20px;
    border-color: #D0021B;
}
```



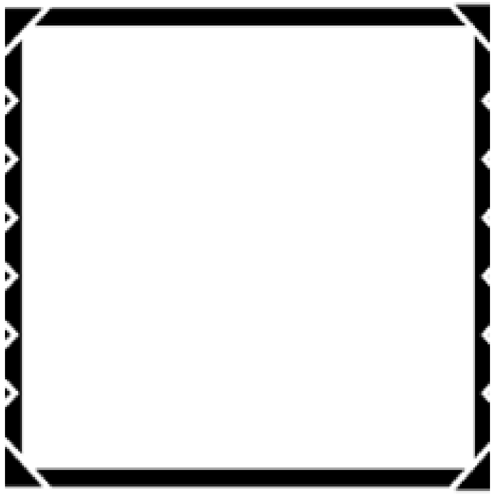
A solid border with rounded corners

Borders

- **border-image**
  - Allows to specify image to act like an element's border
  - Image may be sliced, repeated, scaled, stretched to fit the size of border image area



```
.solid{
    width: 100px;
    height: 100px;
    border-style: solid;
    border-width: 15px;
    border-image: url("border.png") 30 30 round;
}
```



Color

CSS3 by Prashant Chaudhari

Color

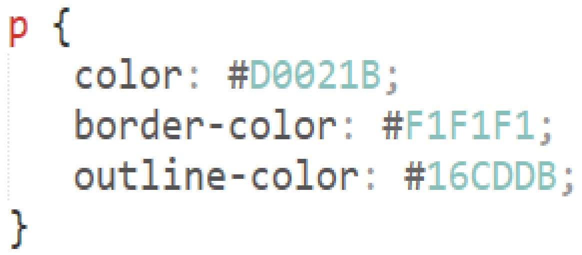
- Defines color of text content of element
- Color values can be in

  Hex value "**#D0021B**"

  RGB value "**rgb(208,2,27)**"

  Color name "**Red**"
- Color can be applied to **borders, outlines, boxes,**

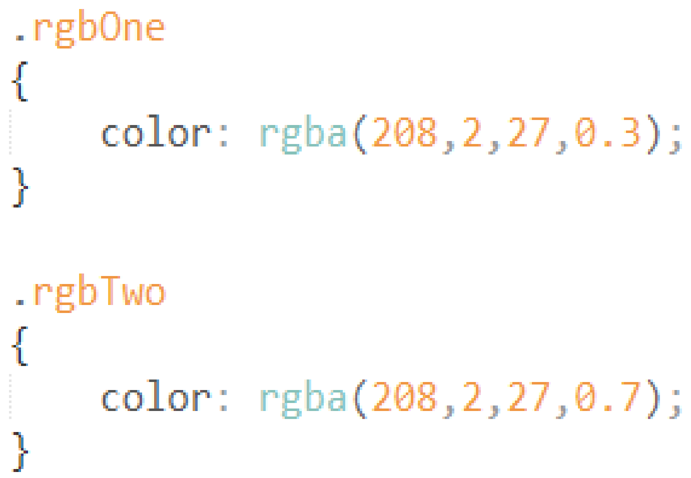
  text, etc.



```
p {
    color: #D0021B;
    border-color: #F1F1F1;
    outline-color: #16CDDB;
}
```

Color

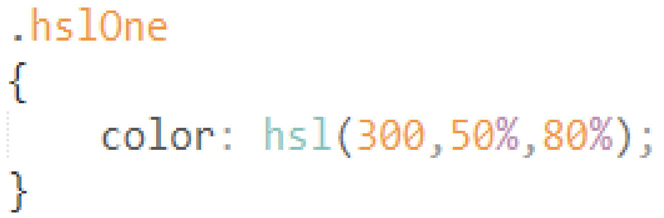
- CSS3 introduced new functional notations to set color values
- **New notations**
  - **RGBA**
    - Red-Green-Blue-Alpha
    - RGB accepts value between 0-255 and alpha accepts value between 0.0-1.0



```
.rgbOne
{
    color: rgba(208,2,27,0.3);
}

.rgbTwo
{
    color: rgba(208,2,27,0.7);
}
```

Color

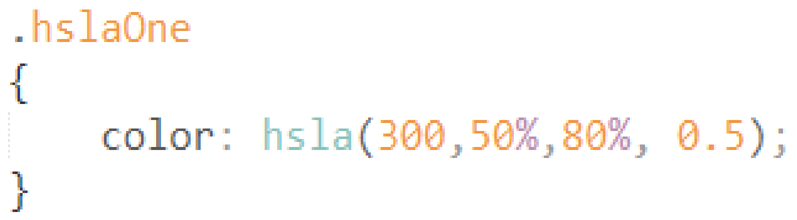
- **HSL**
  - Hue-Saturation-Lightness
  - Hue accepts value between 0-360
  - Saturation and Lightness accepts value in percentage
  - 100% saturation means full color, 0% means shade of gray
  - 100% lightness is white, 0% is black and 50% is normal



```
.hslOne
{
    color: hsl(300,50%,80%);
}
```

Color

- **HSLA**
  - Hue-Saturation-Lightness-Alpha
  - Alpha accept values from 0.0(transparent) to 1.0(fully opaque)



```
.hslaOne
{
    color: hsla(300,50%,80%, 0.5);
}
```

Background

CSS3 by Prashant Chaudhari

Background

- Defines background styles of elements
- **Style properties include**

  background-color,

  background-image,

  background-repeat,

  background-attachment,

  background-position

Background

- **background-color**

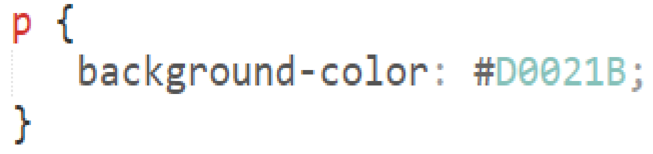
  Sets background color

- **background-image**

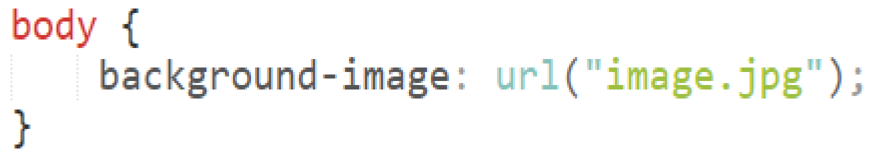
  Sets image as a background

- **background-repeat**
  - Controls how background image is tiled
  - Values can be repeat, round, repeat-x, repeat-y,

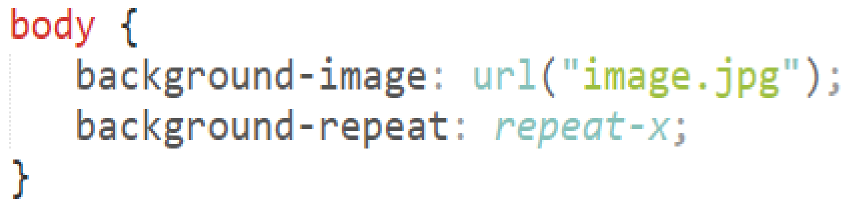
    no-repeat



```
p {
    background-color: #D0021B;
}
```



```
body {
    background-image: url("image.jpg");
}
```

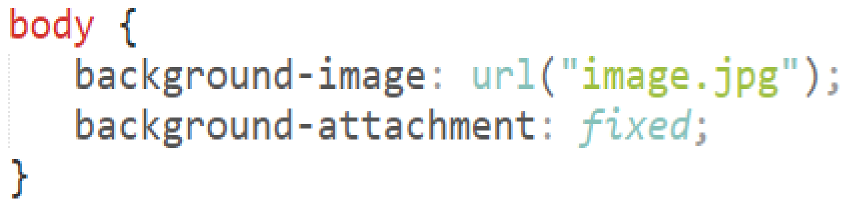


```
body {
    background-image: url("image.jpg");
    background-repeat: repeat-x;
}
```

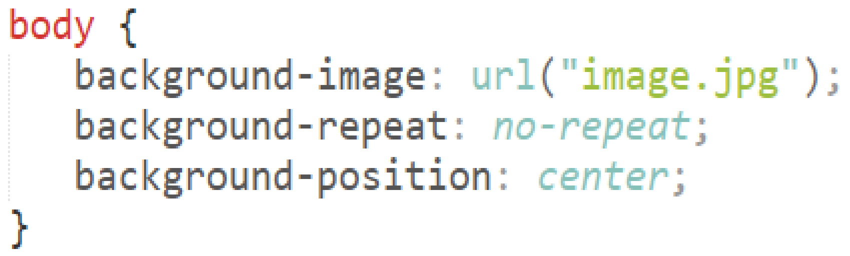
Background

- **background-attachment**
  - Determines whether background image is fixed

    or can be scrolled
  - Values can be fixed, scroll
- **background-position**
  - Controls position of background image
  - Default position (0,0)
  - Values can be bottom, center, left, right, top



```css
body {
    background-image: url("image.jpg");
    background-attachment: fixed;
}
```

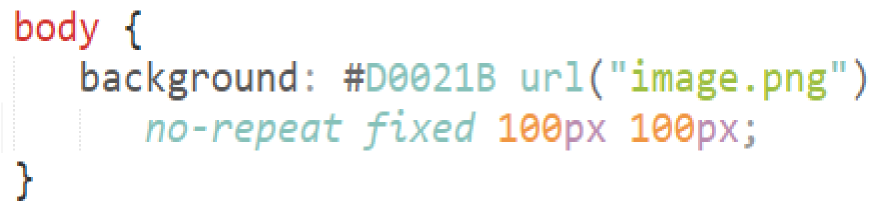


```css
body {
    background-image: url("image.jpg");
    background-repeat: no-repeat;
    background-position: center;
}
```

Background

- **Background shorthand property**
  - Can specify all properties in single rule
  - **background** is a shorthand used to set all

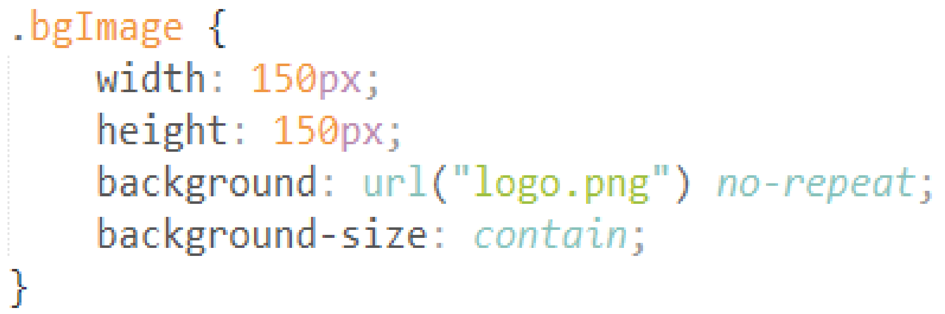
    properties in one



```
body {
    background: #D0021B url("image.png")
        no-repeat fixed 100px 100px;
}
```

Background – New in CSS3

- **background-size**
  - Specifies size of background image
  - Can be specified by pixels, percentage or keywords like auto, contain and cover



```
.bgImage {
    width: 150px;
    height: 150px;
    background: url("logo.png") no-repeat;
    background-size: contain;
}
```

kode
god
.com

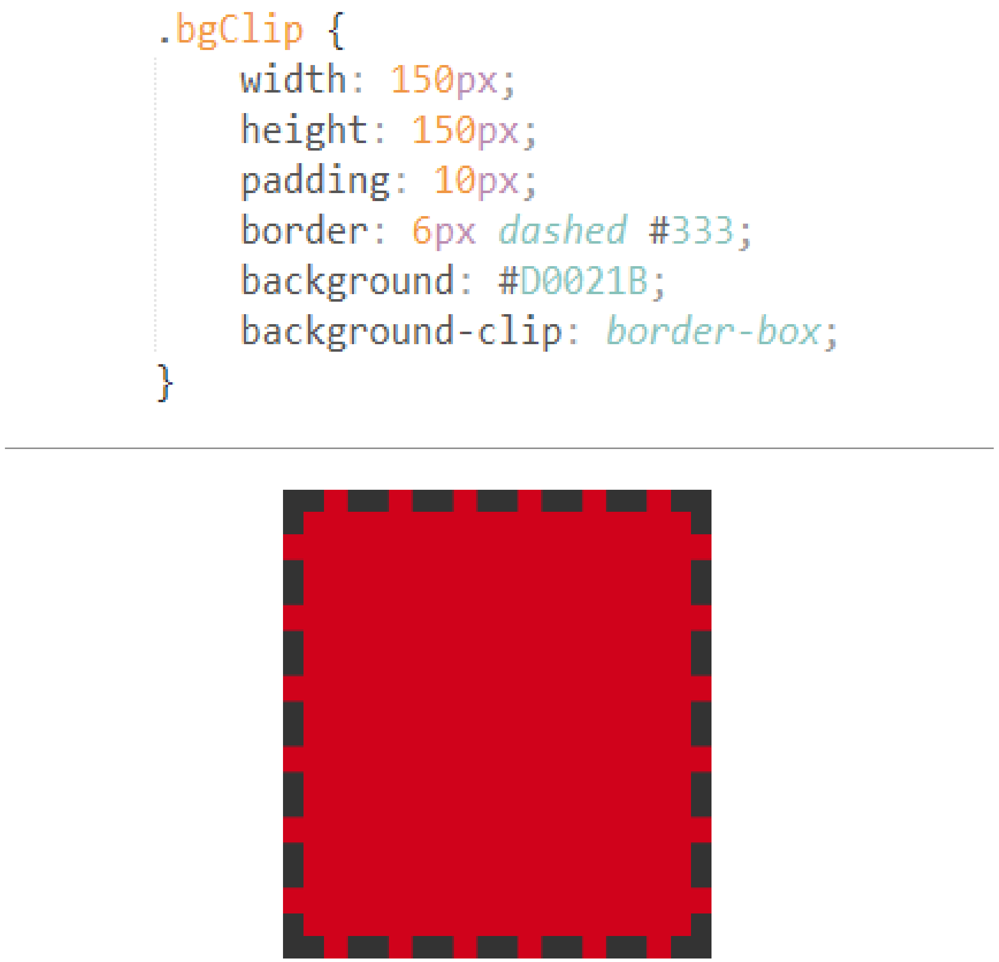
Background – New in CSS3

- **background-clip**
  - Specifies whether background

    extends into the border or not
  - **Takes three values**

    border-box, padding-box, content-box

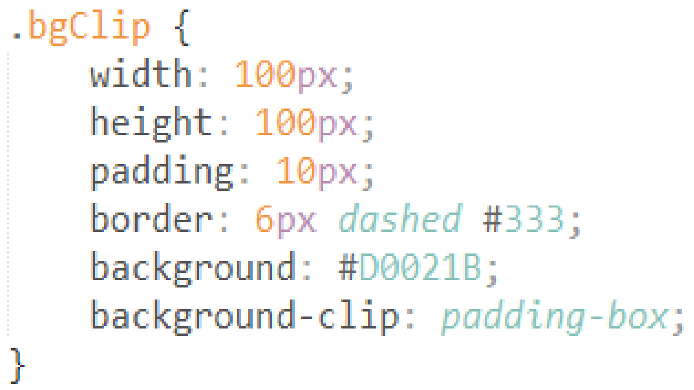
- **border-box**



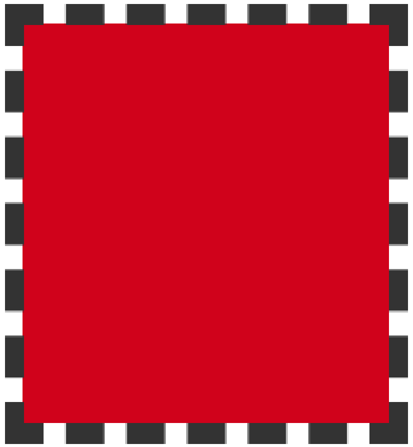
```
.bgClip {
    width: 150px;
    height: 150px;
    padding: 10px;
    border: 6px dashed #333;
    background: #D0021B;
    background-clip: border-box;
}
```

Background – New in CSS3

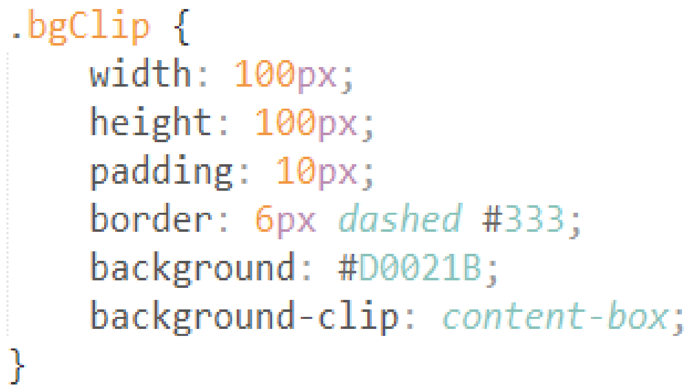
- **padding-box**



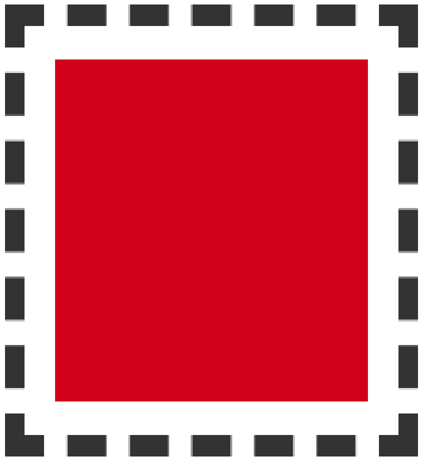
```
.bgClip {
    width: 100px;
    height: 100px;
    padding: 10px;
    border: 6px dashed #333;
    background: #D0021B;
    background-clip: padding-box;
}
```



- **content-box**



```
.bgClip {
    width: 100px;
    height: 100px;
    padding: 10px;
    border: 6px dashed #333;
    background: #D0021B;
    background-clip: content-box;
}
```

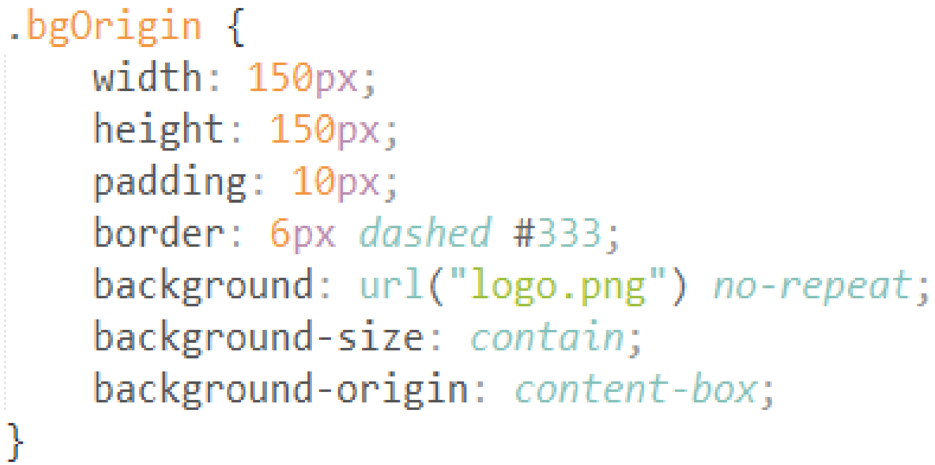


Background – New in CSS3

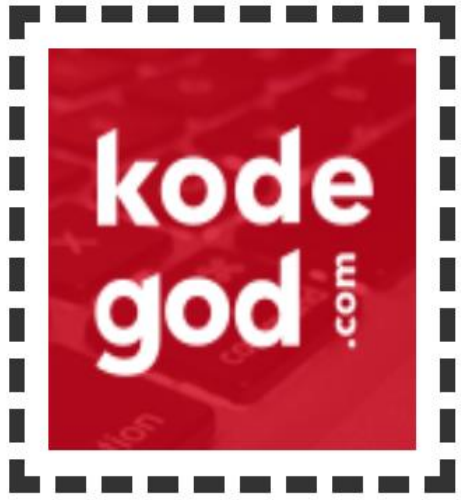
- **background-origin**
  - Specifies positioning area of the

    background image
  - Same as background-clip
  - **Takes three values**

    border-box, padding-box, content-box

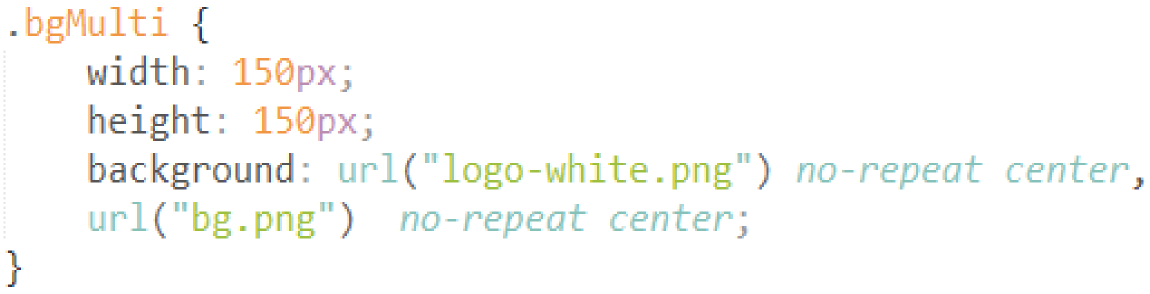


```
.bgOrigin {
    width: 150px;
    height: 150px;
    padding: 10px;
    border: 6px dashed #333;
    background: url("logo.png") no-repeat;
    background-size: contain;
    background-origin: content-box;
}
```

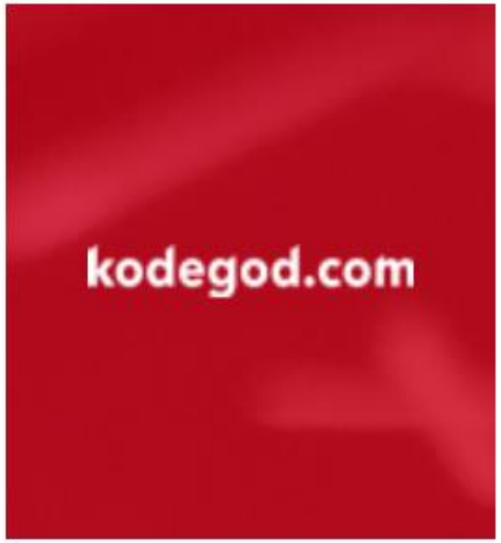


Background

- **Multiple Backgrounds**
  - Add multiple backgrounds to single element
  - Layered top of one another
  - First value in comma-separated list will appear on top and last value at bottom
  - Only last value can contain background-color



```
.bgMulti {
    width: 150px;
    height: 150px;
    background: url("logo-white.png") no-repeat center,
    url("bg.png")  no-repeat center;
}
```



Fonts

CSS3 by Prashant Chaudhari

Fonts

- Styling fonts of text content
- **Font properties include**

  font-family,

  font-style,

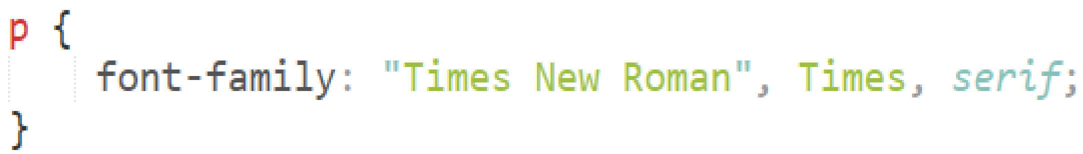
  font-variant,

  font-weight,

  font-size

Fonts

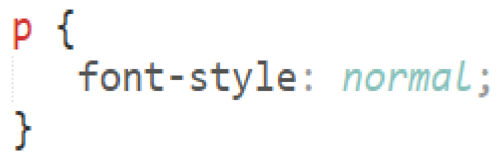
- font-family
  - Font family for text of selected element



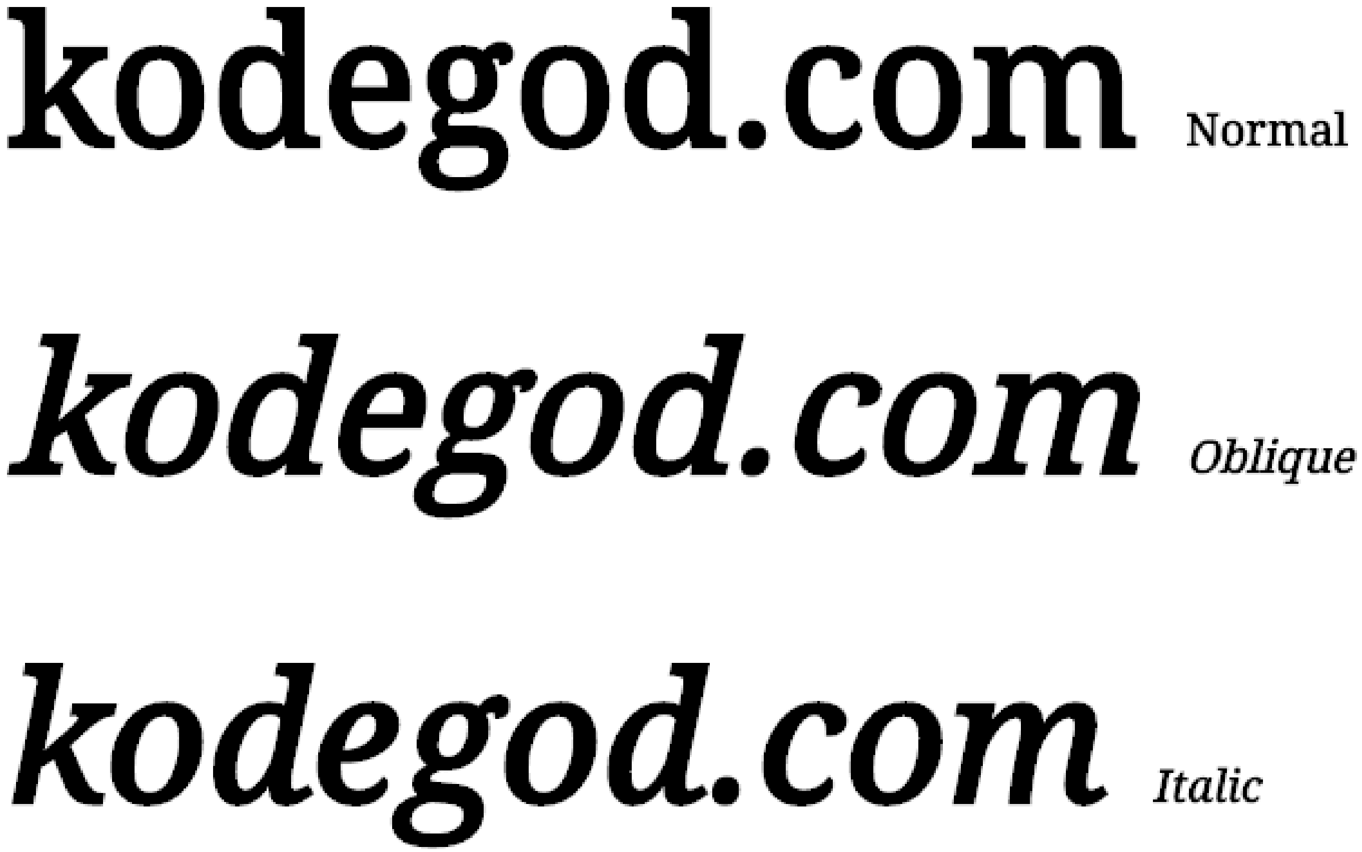
```
p {
    font-family: "Times New Roman", Times, serif;
}
```

Fonts

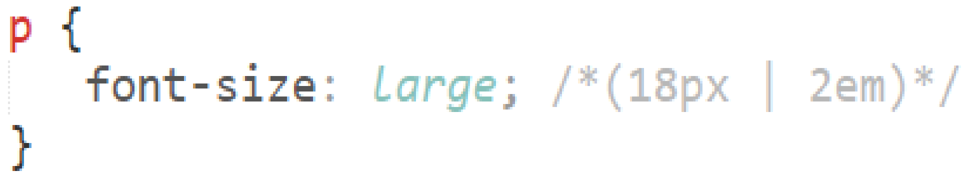
- font-style
  - Font style for text content of an element
  - Property values normal, italic, oblique



```
p {
    font-style: normal;
}
```



Fonts



```
p {
    font-size: large; /*(18px | 2em)*/
}
```

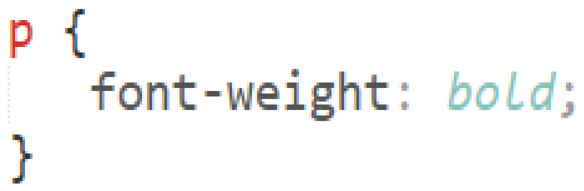
- **font-size**
  - Sets size of font for text content of an element
  - **Property values** with keywords, pixels, ems, %
  - **Keywords** xx-small, x-small, small, medium, large, x-large, xx-large
  - Default 16px
  - If parent is 16px then 1em = 16px
  - If root is 16px then 1rem = 16px

Fonts

- **font-weight**
  - Specifies weight or boldness of font
  - **Property values**

    normal, bold, bolder, lighter, 100, 200, 300,

    400, 500, 600, 700, 800, 900, inherit

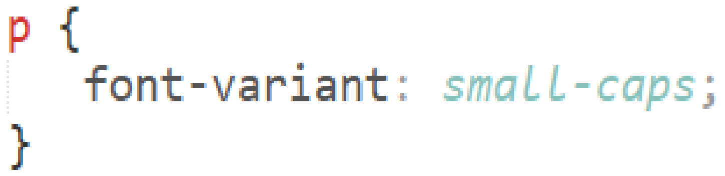


```
p {
    font-weight: bold;
}
```

Fonts

- **font-variant**
  - Text displayed in special small-caps variation
  - **Property values**

    normal, small-caps, inherit



```
p {
    font-variant: small-caps;
}
```

Text

CSS3 by Prashant Chaudhari

Text

- Control over characters, spaces, words,

  paragraphs, etc.

- **Text properties include**

  text-color, text-align,

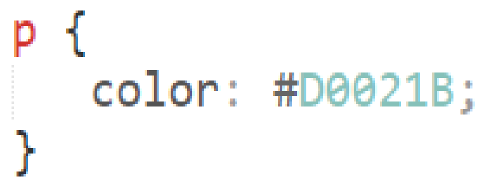
  text-decoration, text-transform,

  text-indent, word-spacing,

  letter-spacing, line-height

Text

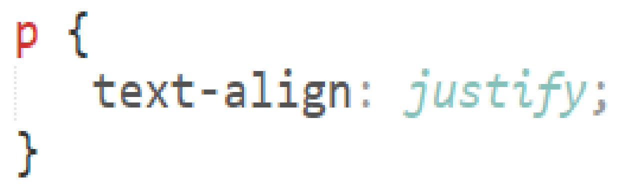
- **text-color**
  - Sets text color using css **color** property



```
p {
    color: #D0021B;
}
```

- **text-align**
  - Sets horizontal alignment of text
  - **Property values**

    left, right, center, justify, inherit



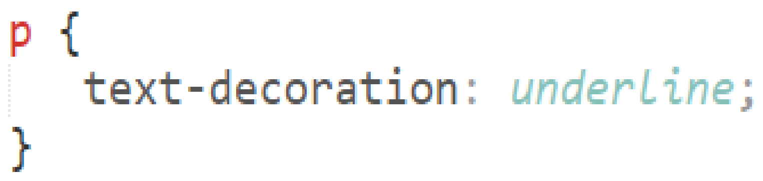
```
p {
    text-align: justify;
}
```

Text

- **text-decoration**
  - Set or remove decoration from text
  - **Property values**

    none, underline, overline, line-through,

    blink, inherit

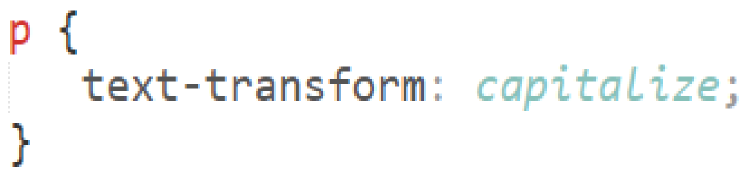


```
p {
    text-decoration: underline;
}
```

- **text-transform**
  - Sets cases for text
  - **Property values**

    none, capitalize, uppercase, lowercase,

    inherit

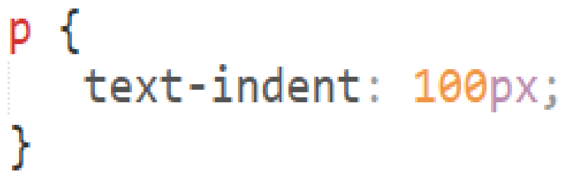


```
p {
    text-transform: capitalize;
}
```

Text

- **text-indent**
  - Set indentation of first line of text
  - **Property values**

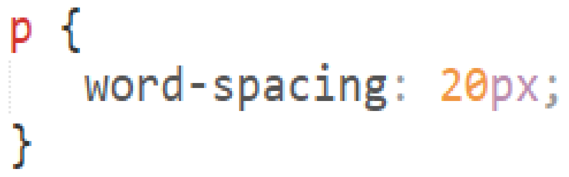
    %, length(px) or inherit



```
p {
    text-indent: 100px;
}
```

- **word-spacing**
  - Sets spacing between words
  - **Property values**

    length(px), normal, inherit

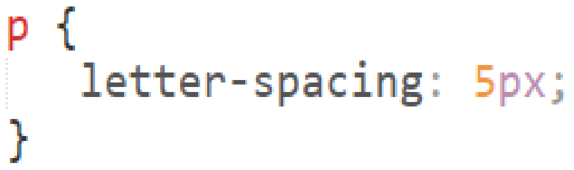


```
p {
    word-spacing: 20px;
}
```

Text

- **letter-spacing**
  - Sets spacing between characters of words
  - **Property values**

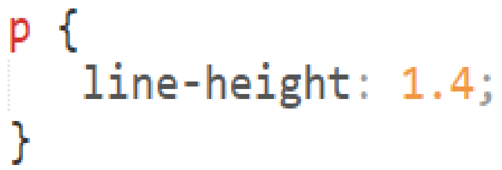
    %, length(px) or inherit



```
p {
    letter-spacing: 5px;
}
```

- **line-height**
  - Sets height of line of text
  - **Property values**

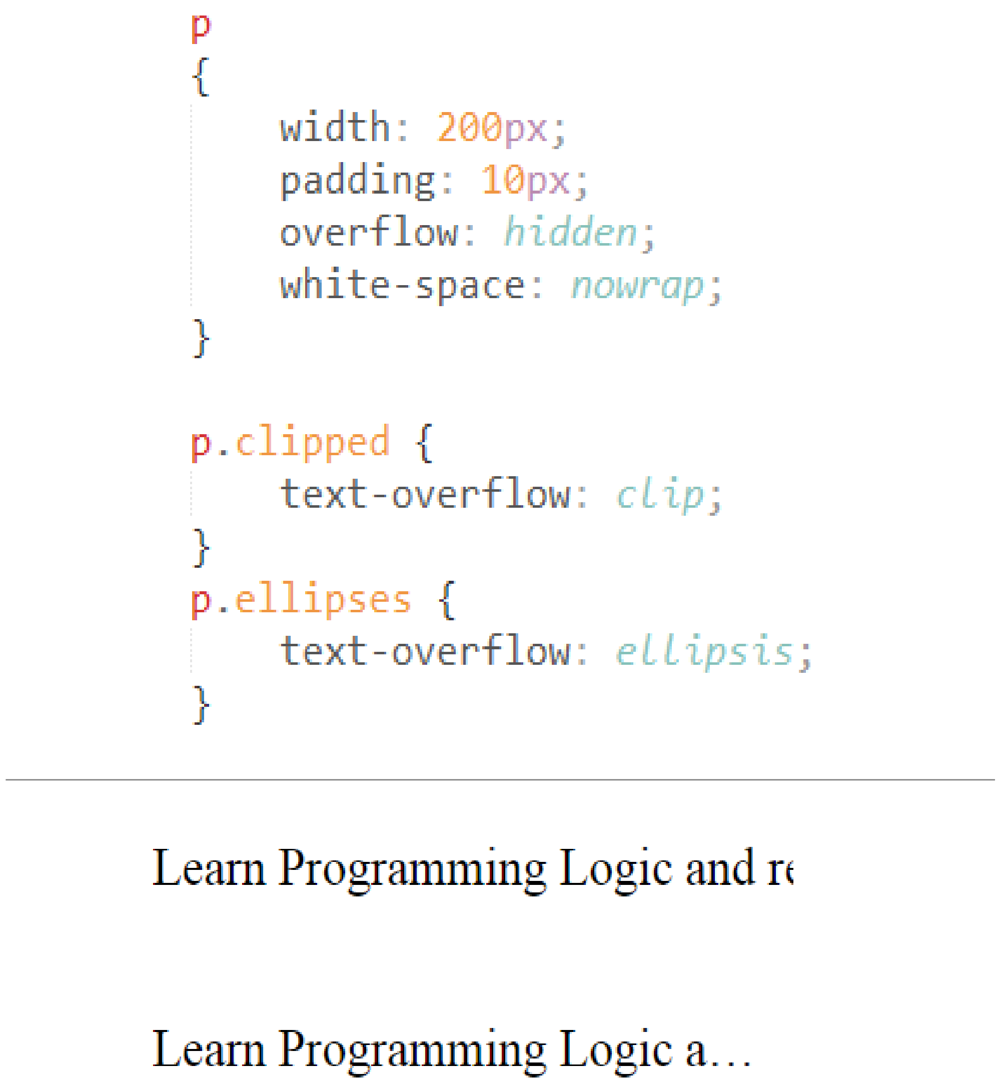
    %, length, normal, inherit



```
p {
    line-height: 1.4;
}
```

Text

- **Hiding overflow text**
  - **text-overflow** property used to hide overflow text using **clip or ellipsis**



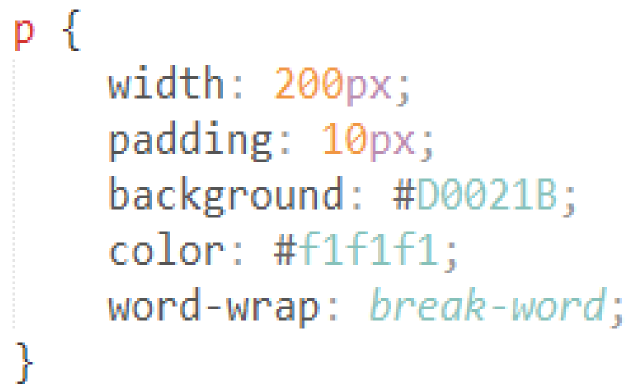
```
p
{
    width: 200px;
    padding: 10px;
    overflow: hidden;
    white-space: nowrap;
}

p.clipped {
    text-overflow: clip;
}
p.ellipses {
    text-overflow: ellipsis;
}
```

Learn Programming Logic and re

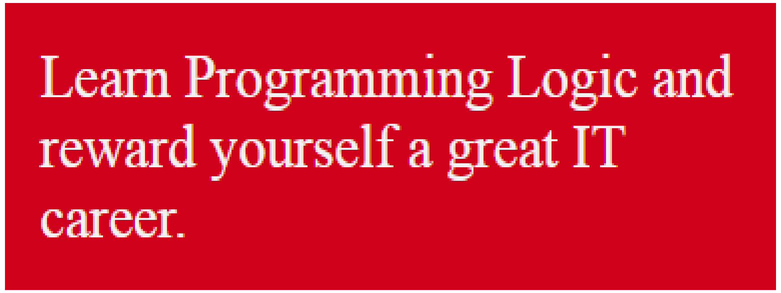
Learn Programming Logic a…

Text

- **Breaking overflow text**
  - **word-wrap** property used to break overflow text to new line using values **normal** and **break-word**



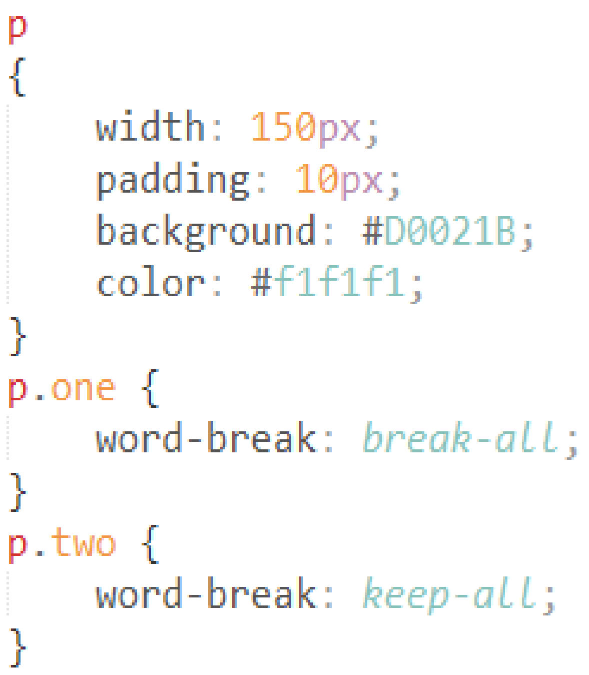
```
p {
    width: 200px;
    padding: 10px;
    background: #D0021B;
    color: #f1f1f1;
    word-wrap: break-word;
}
```



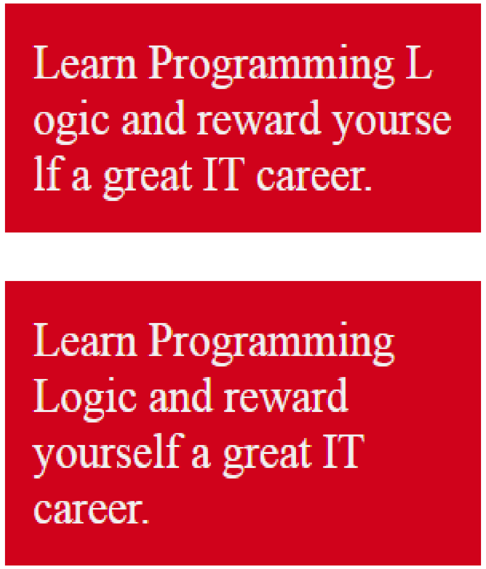
Learn Programming Logic and reward yourself a great IT career.

Text

- **Word breaking rules**
  - **word-break** property used to break lines within words using values **normal, break-all** and **keep-all**



```
p
{
    width: 150px;
    padding: 10px;
    background: #D0021B;
    color: #f1f1f1;
}
p.one {
    word-break: break-all;
}
p.two {
    word-break: keep-all;
}
```



Learn Programming L
ogic and reward yourse
lf a great IT career.

Learn Programming
Logic and reward
yourself a great IT
career.

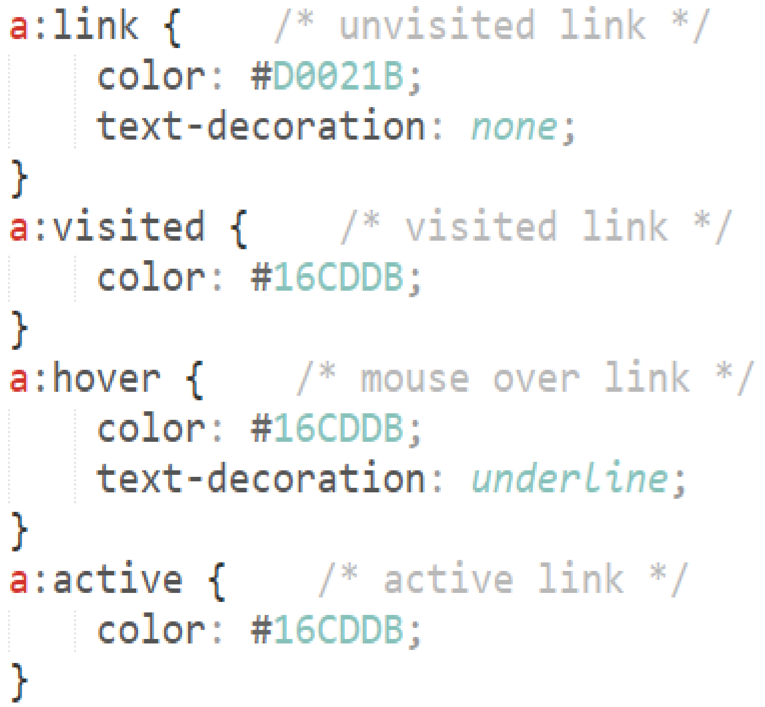
Links

CSS3 by Prashant Chaudhari

Links

- Connection from one web resource to another
- Has four different states

  link, visited, active, hover
- Sets using **selector: state**



```
a:link {     /* unvisited link */
    color: #D0021B;
    text-decoration: none;
}
a:visited {     /* visited link */
    color: #16CDDB;
}
a:hover {     /* mouse over link */
    color: #16CDDB;
    text-decoration: underline;
}
a:active {     /* active link */
    color: #16CDDB;
}
```

Lists

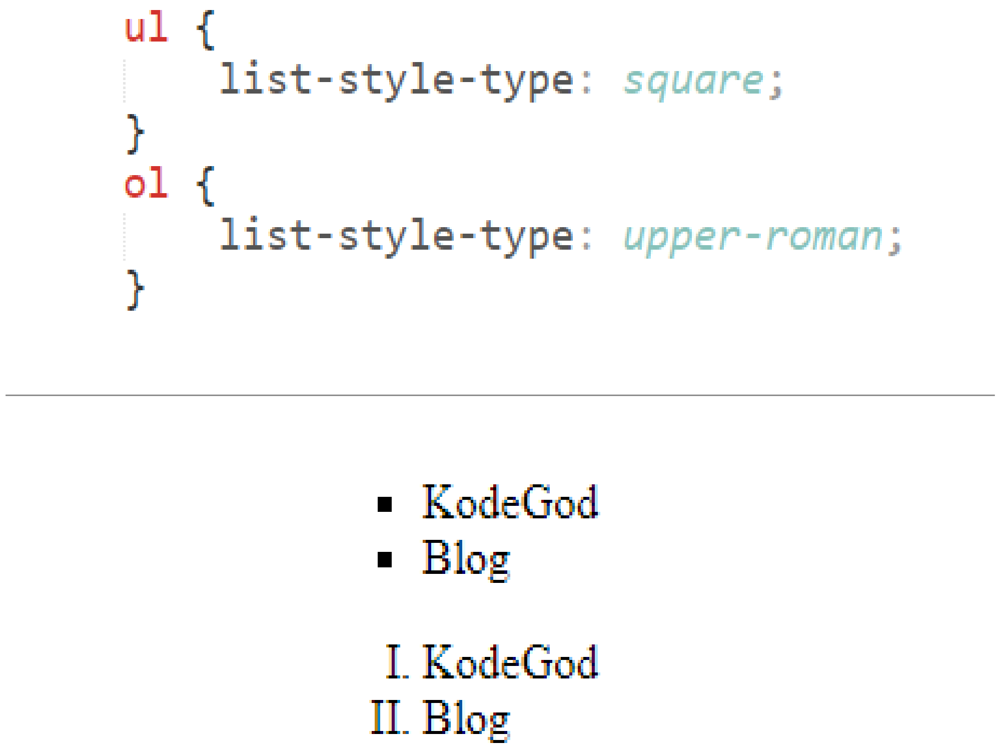
CSS3 by Prashant Chaudhari

Lists

- Controls the presentation of list item markers
- **Properties**
  - **List Style Type**

    Can change marker types to circle, square,

    roman numbers, latin letters, etc.



```
ul {
    list-style-type: square;
}
ol {
    list-style-type: upper-roman;
}
```

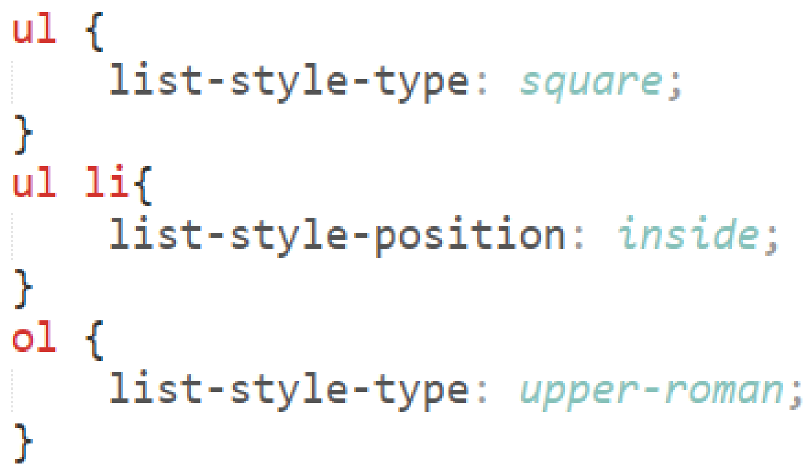
```
▪ KodeGod
▪ Blog

 I. KodeGod
II. Blog
```

Lists - Properties

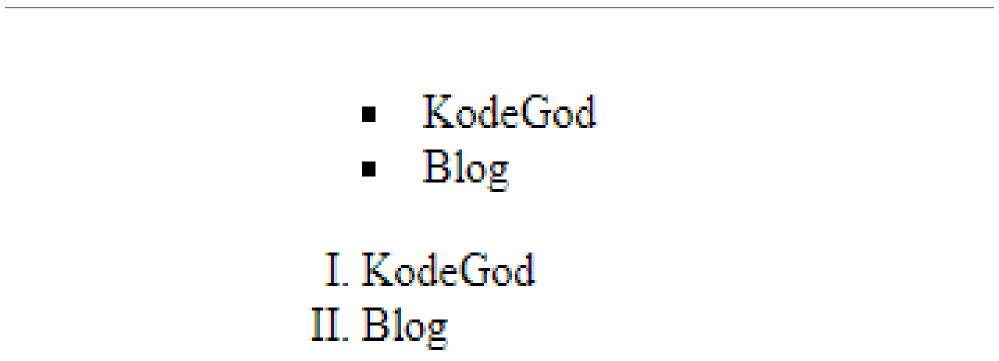
- **List Style Position**

  Specify marker or bullets appear inside or outside

  of list items block boxes



```
ul {
    list-style-type: square;
}
ul li{
    list-style-position: inside;
}
ol {
    list-style-type: upper-roman;
}
```

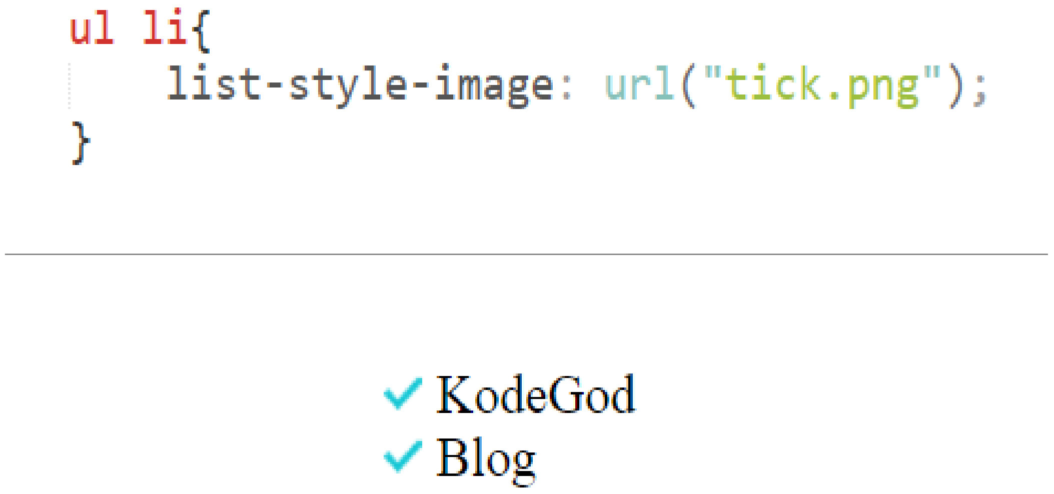


```
   ▪  KodeGod
   ▪  Blog

  I. KodeGod
 II. Blog
```

Lists

- **List Style Image**

  Specify marker or bullets appear as image



```
ul li{
    list-style-image: url("tick.png");
}
```

✓ KodeGod
✓ Blog

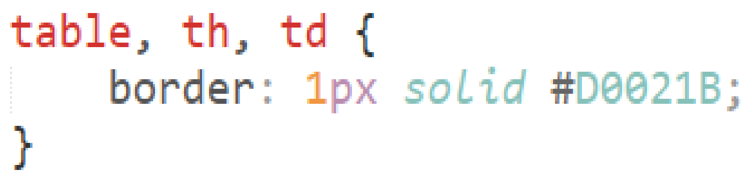
Tables

CSS3 by Prashant Chaudhari

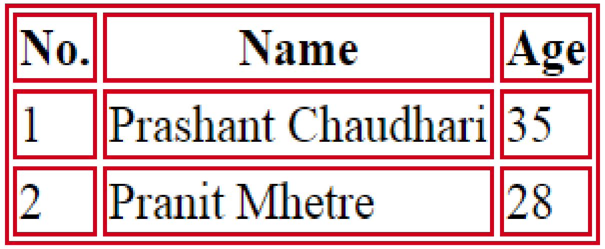
Tables

- Controls the presentation of tables
- **Properties**
  - **border**

    Defines borders of the table



```
table, th, td {
    border: 1px solid #D0021B;
}
```

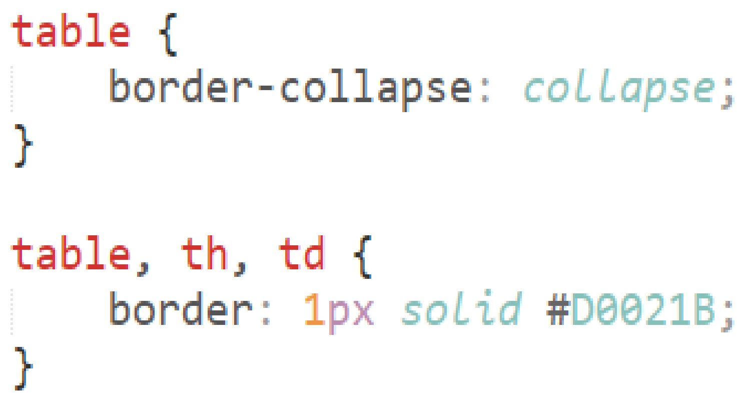


No.	Name	Age
1	Prashant Chaudhari	35
2	Pranit Mhetre	28

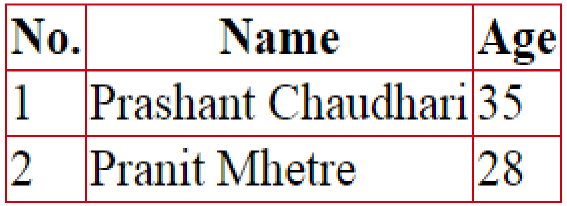
Tables - Properties

- **border-collapse**

  Defines borders of the table without any spacing



```
table {
    border-collapse: collapse;
}

table, th, td {
    border: 1px solid #D0021B;
}
```

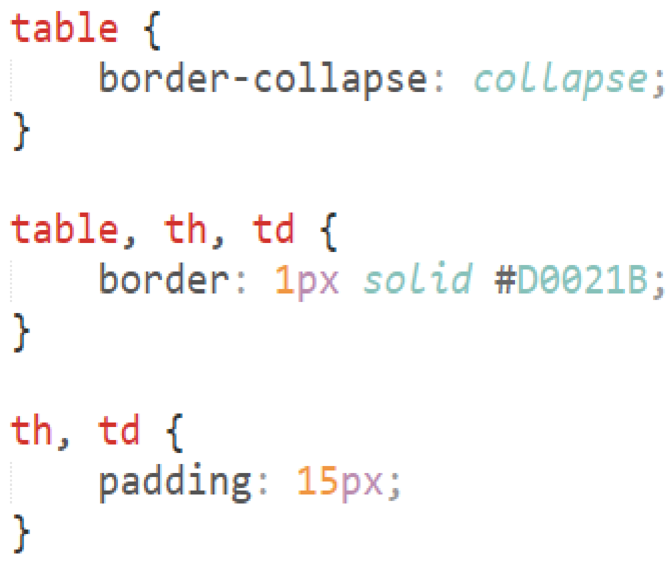


No.	Name	Age
1	Prashant Chaudhari	35
2	Pranit Mhetre	28

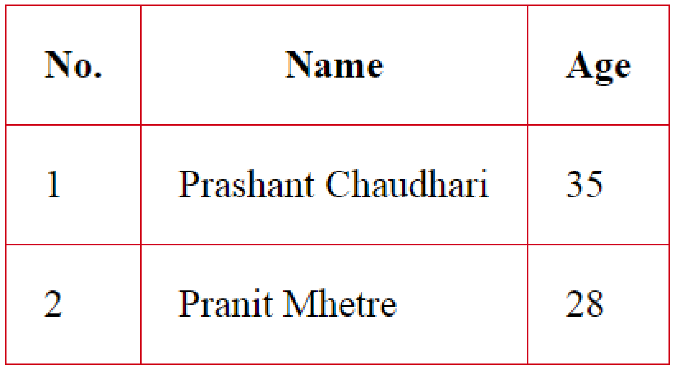
Tables - Properties

- **padding**

  More space around table cells



```
table {
    border-collapse: collapse;
}

table, th, td {
    border: 1px solid #D0021B;
}

th, td {
    padding: 15px;
}
```



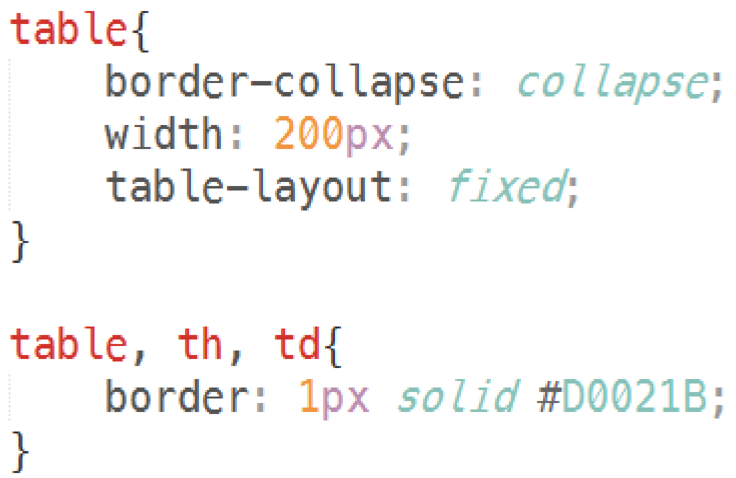
No.	Name	Age
1	Prashant Chaudhari	35
2	Pranit Mhetre	28

Tables - Properties

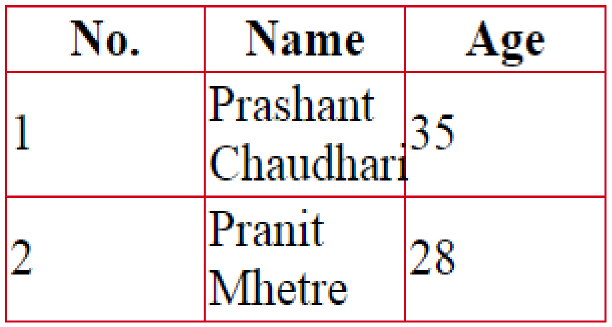
- **table-layout**
  - Control the width of the table
  - **Property values**

    auto Adjusted to fit content

    fixed Depends on table width



```
table{
    border-collapse: collapse;
    width: 200px;
    table-layout: fixed;
}

table, th, td{
    border: 1px solid #D0021B;
}
```

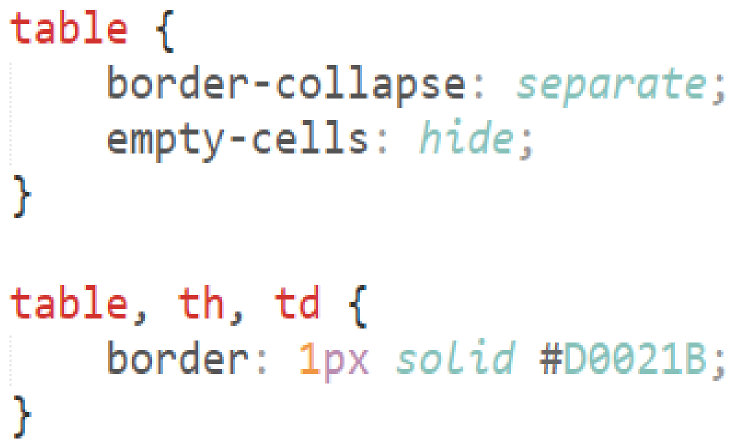


No.	Name	Age
1	Prashant Chaudhari	35
2	Pranit Mhetre	28

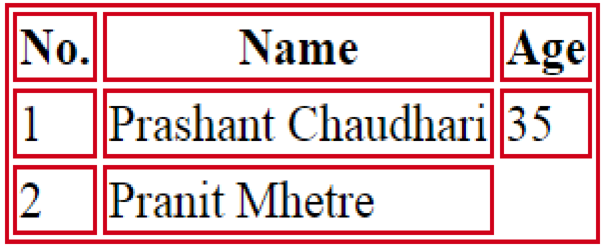
Tables - Properties

- **empty-cells**
  - Rendering of borders and backgrounds of empty cells
  - **Property values**

    show, hide, inherit



```
table {
    border-collapse: separate;
    empty-cells: hide;
}

table, th, td {
    border: 1px solid #D0021B;
}
```



No.	Name	Age
1	Prashant Chaudhari	35
2	Pranit Mhetre	

Outline

CSS3 by Prashant Chaudhari

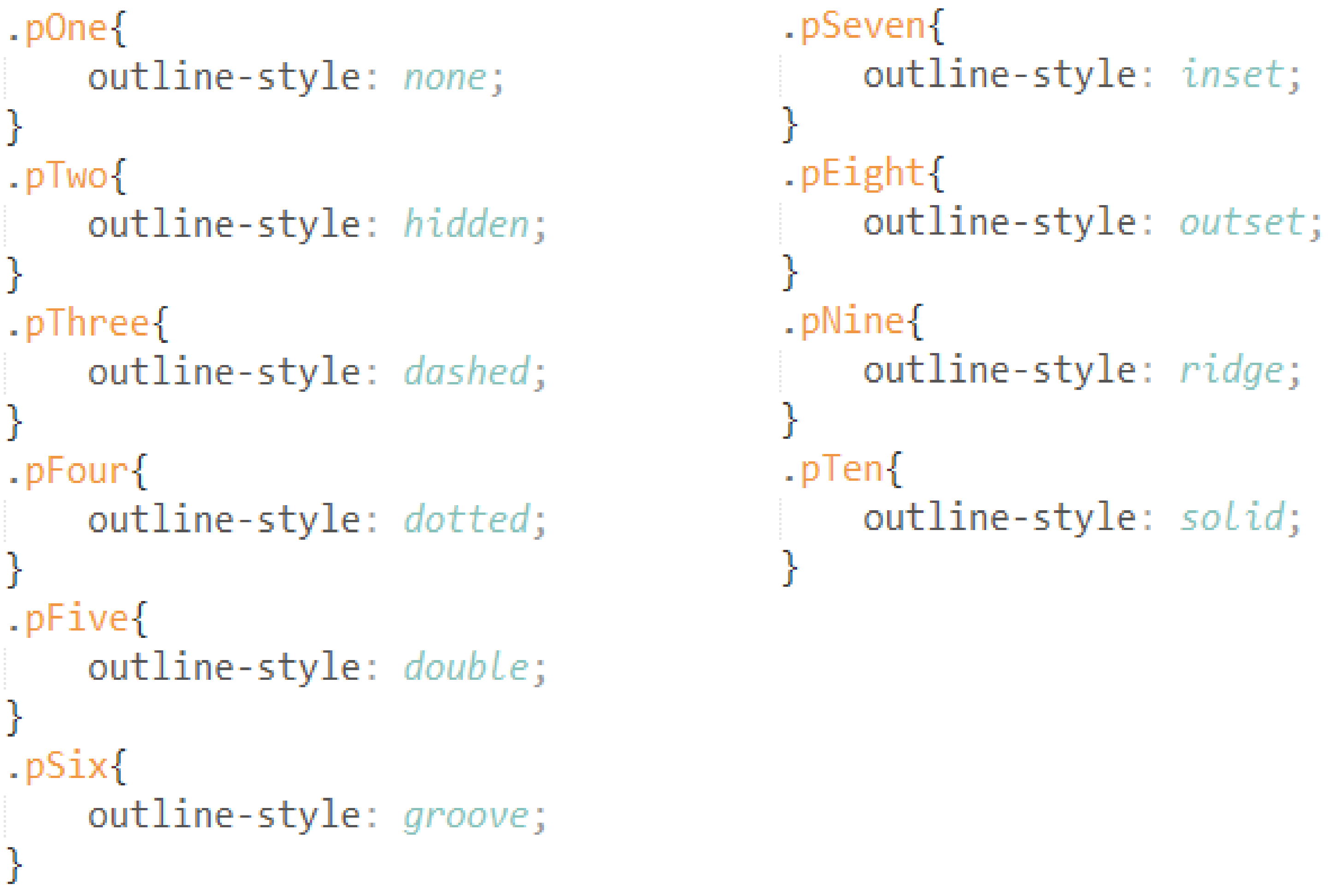
Outline

- Used to highlight elements
- **Properties**
  - **outline-style**
    - Set style for outline
    - **Property Values**

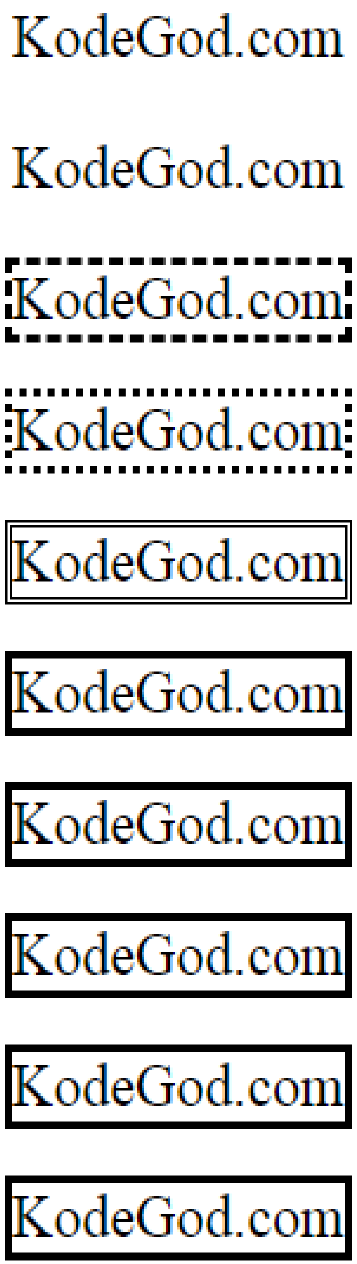
      none, hidden, dashed, dotted, double, groove, inset,

      outset, ridge and solid

Outline (outline-style) - Example



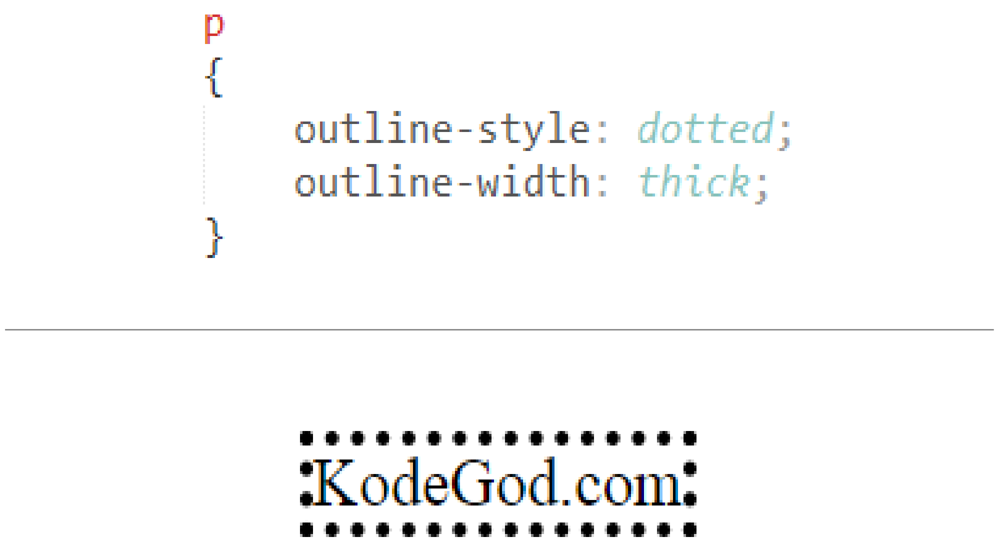
```
.pOne{
    outline-style: none;
}
.pTwo{
    outline-style: hidden;
}
.pThree{
    outline-style: dashed;
}
.pFour{
    outline-style: dotted;
}
.pFive{
    outline-style: double;
}
.pSix{
    outline-style: groove;
}
.pSeven{
    outline-style: inset;
}
.pEight{
    outline-style: outset;
}
.pNine{
    outline-style: ridge;
}
.pTen{
    outline-style: solid;
}
```



```
KodeGod.com
KodeGod.com
KodeGod.com
KodeGod.com
KodeGod.com
KodeGod.com
KodeGod.com
KodeGod.com
KodeGod.com
KodeGod.com
```

Outline - Properties

- **outline-width**
  - Specifies width of outline
  - Values should be in length (px, em, pt and so on) or allowed keywords (medium, thick, thin, inherit)
  - Values cannot be in % or negative

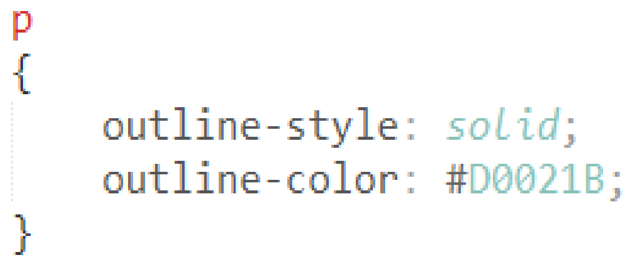


```
p
{
     outline-style: dotted;
     outline-width: thick;
}
```

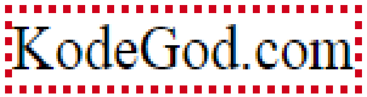
KodeGod.com

Outline - Properties

- **outline-color**
  - Sets the color of an outline



```
p
{
    outline-style: solid;
    outline-color: #D0021B;
}
```

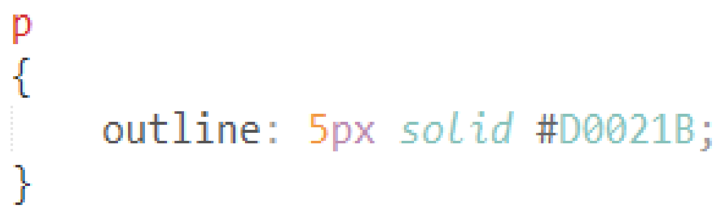


KodeGod.com

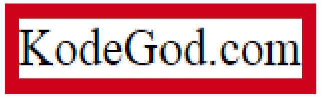
Outline

- **Shorthand Property**
  - Setting multiple properties in single rule
  - **outline** keyword is used for shorthand
  - **Syntax**

    outline: outline-width outline-style outline-color;



```
p
{
    outline: 5px solid #D0021B;
}
```



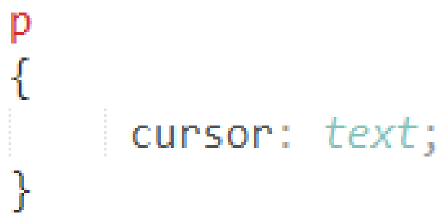
KodeGod.com

Cursors

CSS3 by Prashant Chaudhari

Cursors

- **cursor** property used to define cursor type
- Can redefine default cursor properties
- **Cursors** auto, default, none, context-menu, help, pointer, progress, wait, cell, crosshair, text, vertical-text, alias, copy, move, no-drop, not-allowed, grab, grabbing, all-scroll, zoom-in, zoom out, initial, inherit, etc.

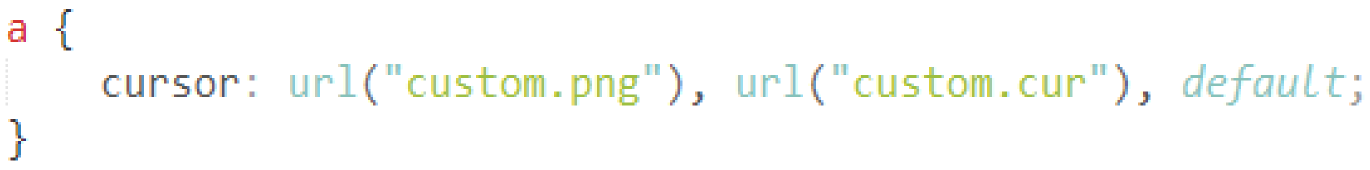


```
p
{
    cursor: text;
}
```

Cursors

- We can have customized cursors
- **cursor** property can have comma-separated list of user-defined

  cursors followed by generic cursor
- Standard format for cursor is **(.cur)**



```css
a {
    cursor: url("custom.png"), url("custom.cur"), default;
}
```

Overflow

CSS3 by Prashant Chaudhari

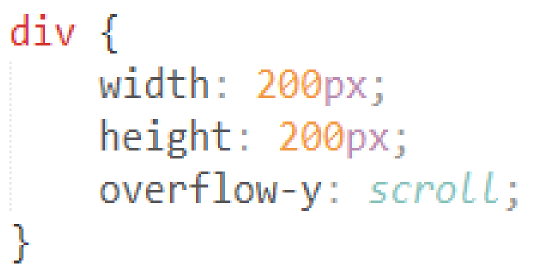
Overflow

- Occurs when element's content overflows element's box
- **overflow** property allows to specify whether to clip or render scroll

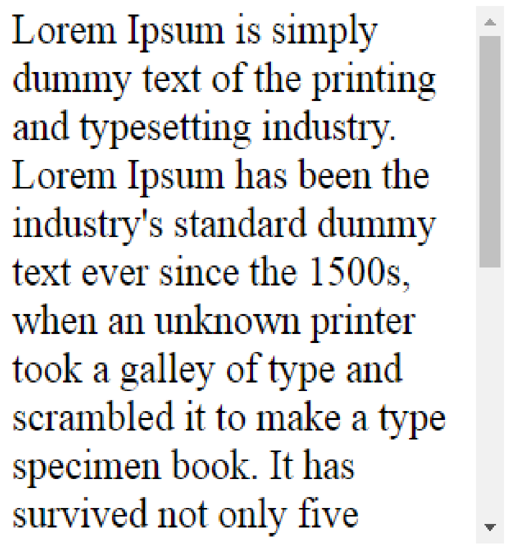
  bar of block level element
- **Property Values**

  visible, hidden, scroll, auto
- **Two More Properties**

  overflow-x and overflow-y for vertical and horizontal clipping



```css
div {
    width: 200px;
    height: 200px;
    overflow-y: scroll;
}
```



Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five

Dimension

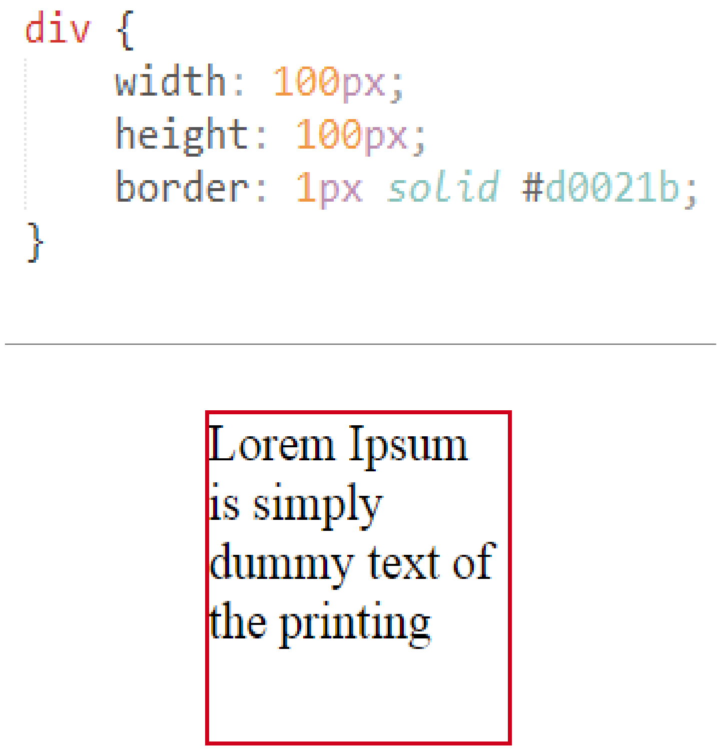
CSS3 by Prashant Chaudhari

Dimension

- Controls height and width of element
- **Properties**

  height, width, max-height, max-width, min-height, min-width
- **Height and Width**
  - Does not include padding, border, margins
  - Takes values in length (px, pt, em, etc.), percentage or keyword

    auto

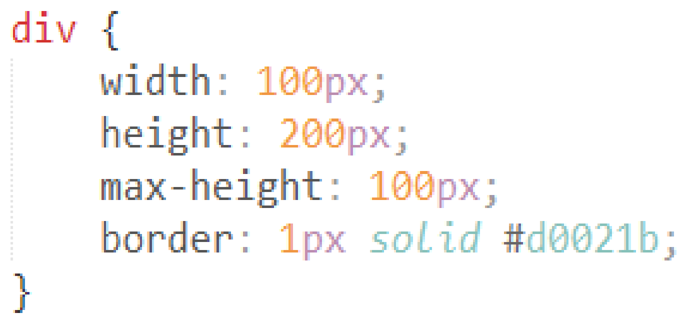


```css
div {
    width: 100px;
    height: 100px;
    border: 1px solid #d0021b;
}
```

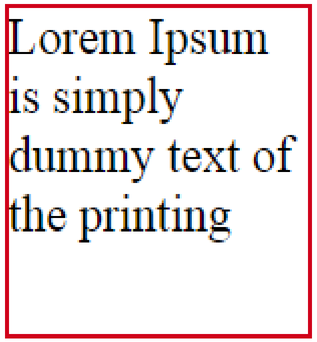
```
Lorem Ipsum
is simply
dummy text of
the printing
```

Dimension - Properties

- **max-height**
  - Maximum height of the box
  - If height set to 200px and max-height set to 100px, then actual height of element is 100px



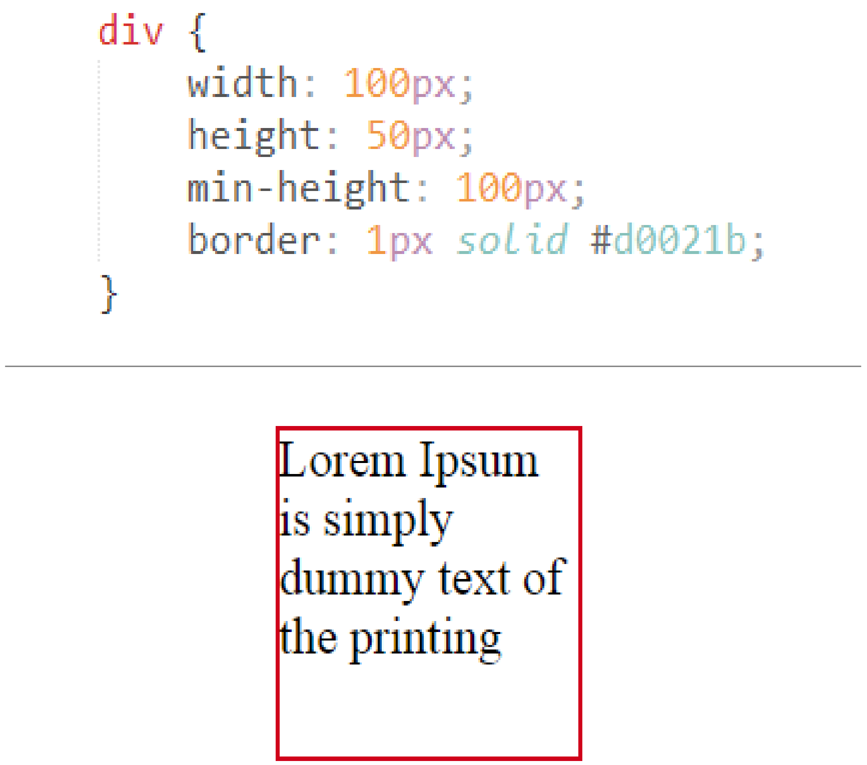
```
div {
    width: 100px;
    height: 200px;
    max-height: 100px;
    border: 1px solid #d0021b;
}
```



Lorem Ipsum is simply dummy text of the printing

Dimension - Properties

- **min-height**
  - Minimum height of the box
  - If height set to 50px and min-height set to 100px, then actual height of element is 100px

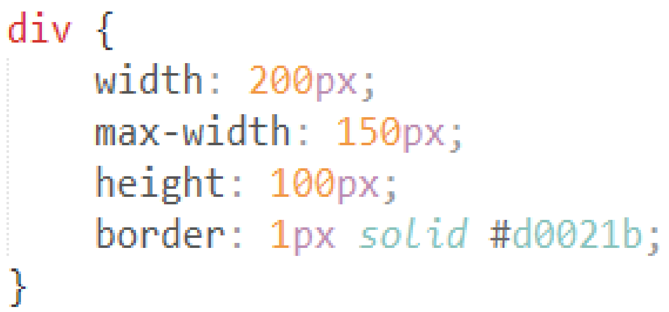


```
div {
    width: 100px;
    height: 50px;
    min-height: 100px;
    border: 1px solid #d0021b;
}
```

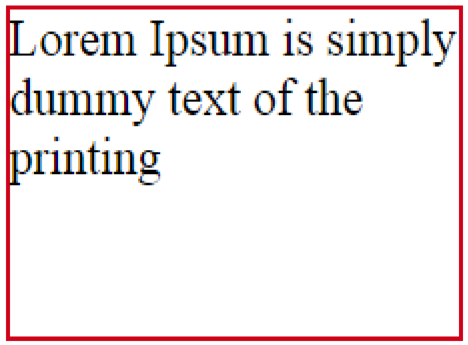
Lorem Ipsum is simply dummy text of the printing

Dimension - Properties

- **max-width**
  - Maximum width of the box
  - If width set to 200px and max-width set to 150px, then actual height of element is 150px



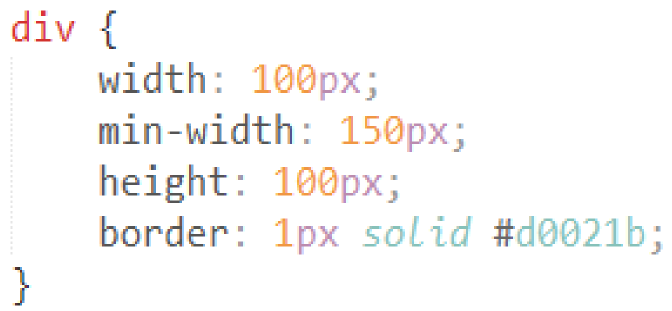
```
div {
    width: 200px;
    max-width: 150px;
    height: 100px;
    border: 1px solid #d0021b;
}
```



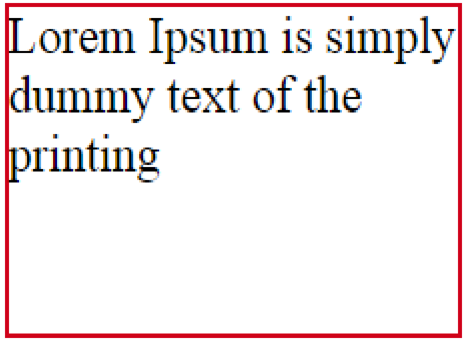
Lorem Ipsum is simply dummy text of the printing

Dimension - Properties

- **min-width**
  - Minimum width of the box
  - If width set to 100px and min-width set to 150px, then actual height of element is 150px



```
div {
    width: 100px;
    min-width: 150px;
    height: 100px;
    border: 1px solid #d0021b;
}
```



Lorem Ipsum is simply dummy text of the printing

Units

CSS3 by Prashant Chaudhari

Units

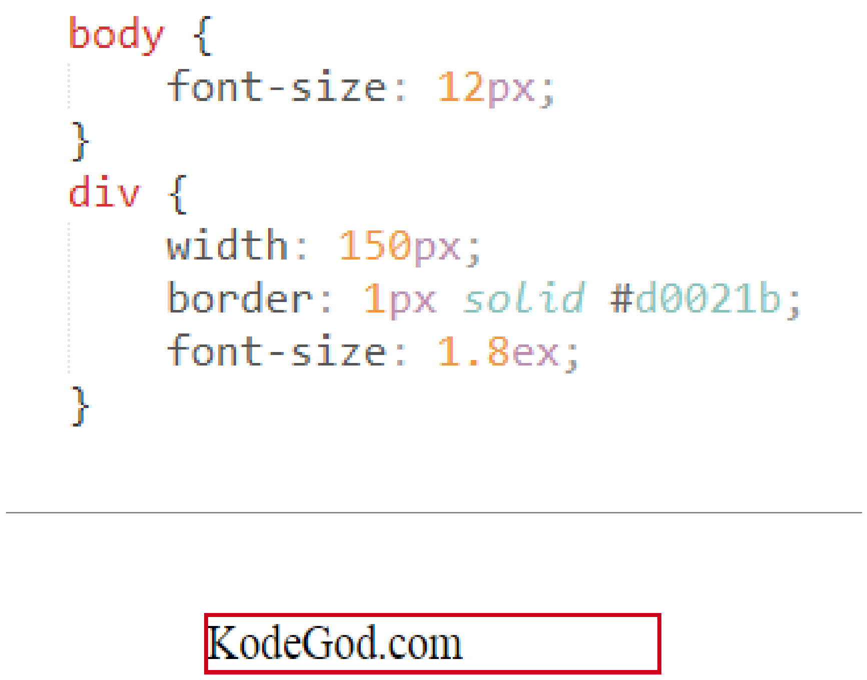
- Units in which length is measured
- **Two types**
  - **Relative length units**
  - **Absolute length units**

Relative Length Units

- Specifies length relative to another length property
- Scales better between different rendering mediums
- em, ex, cm, rem, vw, vh, vmin, vmax, %

Relative Length Units

- **ex**
  - ex units equals to x-height of current font
  - x-height means height of lowercase **'x'**
  - If 'x' is not present, then font size is set to default



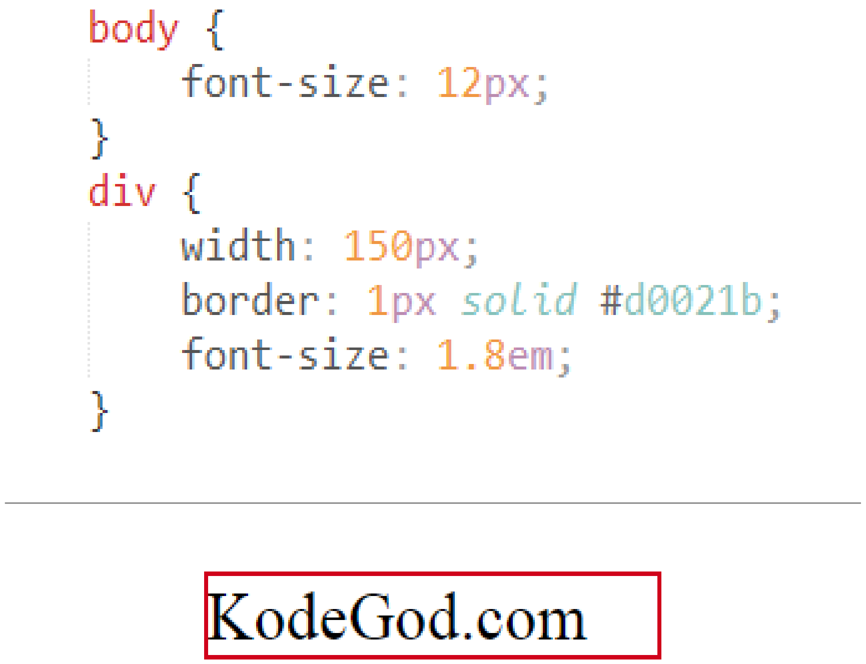
```
body {
    font-size: 12px;
}
div {
    width: 150px;
    border: 1px solid #d0021b;
    font-size: 1.8ex;
}
```

KodeGod.com

Relative Length Units

- **em**
  - Used for vertical and horizontal measurement
  - E.g. If font-size set to 18px and line-height set to 1.8em, then line-height in pixels is

    1.8 * 18 = 32.4px
  - If specify font size in em like 1.8em, makes the text 1.8 times larger than parent element's text



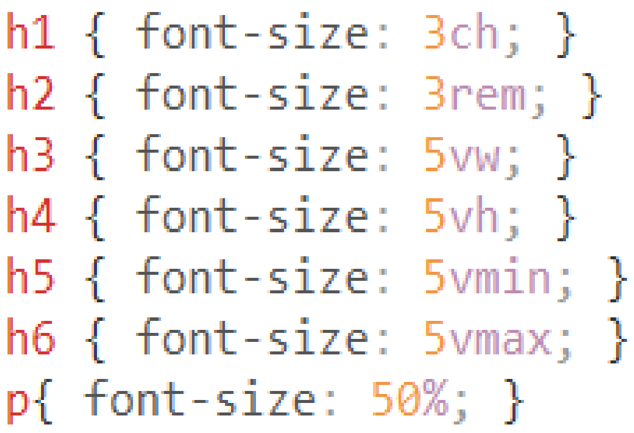
```
body {
    font-size: 12px;
}
div {
    width: 150px;
    border: 1px solid #d0021b;
    font-size: 1.8em;
}
```

KodeGod.com

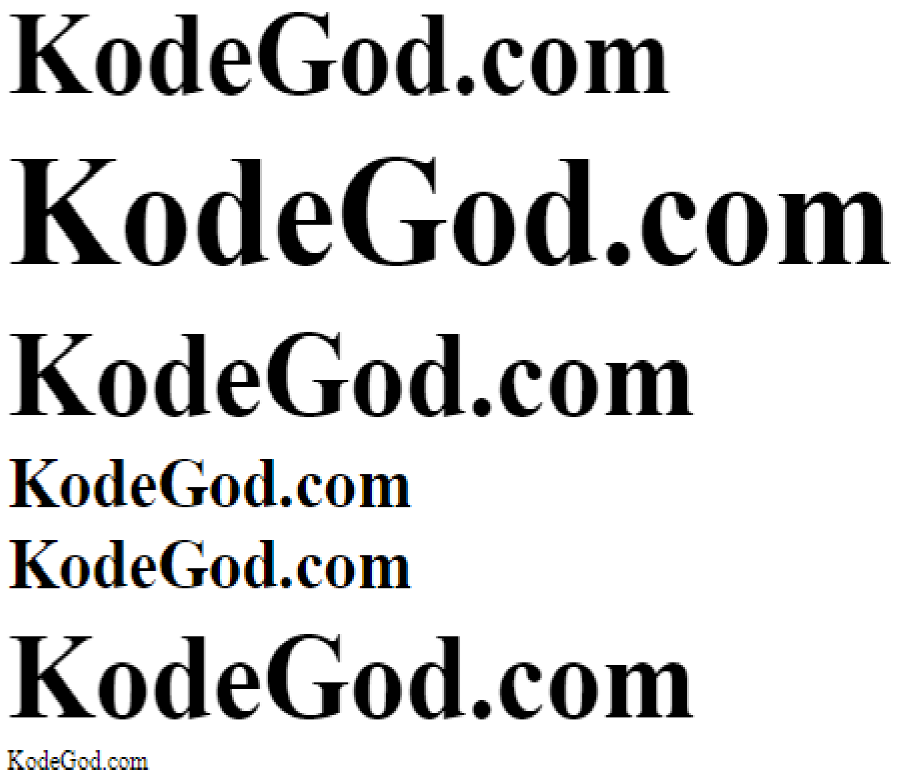
Relative Length Units

- **ch**
  - Relative to the width of character zero
- **rem**
  - Relative to font-size of root element
- **vw**
  - Relative to 1% of width of viewport
- **%**
  - Relative to parent element
- **vh**
  - Relative to 1% of height of viewport
- **vmin**
  - Relative to 1% of viewports smaller dimension
- **vmax**
  - Relative to 1% of viewports larger dimension

Relative Length Units - Example



```
h1 { font-size: 3ch; }
h2 { font-size: 3rem; }
h3 { font-size: 5vw; }
h4 { font-size: 5vh; }
h5 { font-size: 5vmin; }
h6 { font-size: 5vmax; }
p{ font-size: 50%; }
```



KodeGod.com

KodeGod.com

KodeGod.com

KodeGod.com

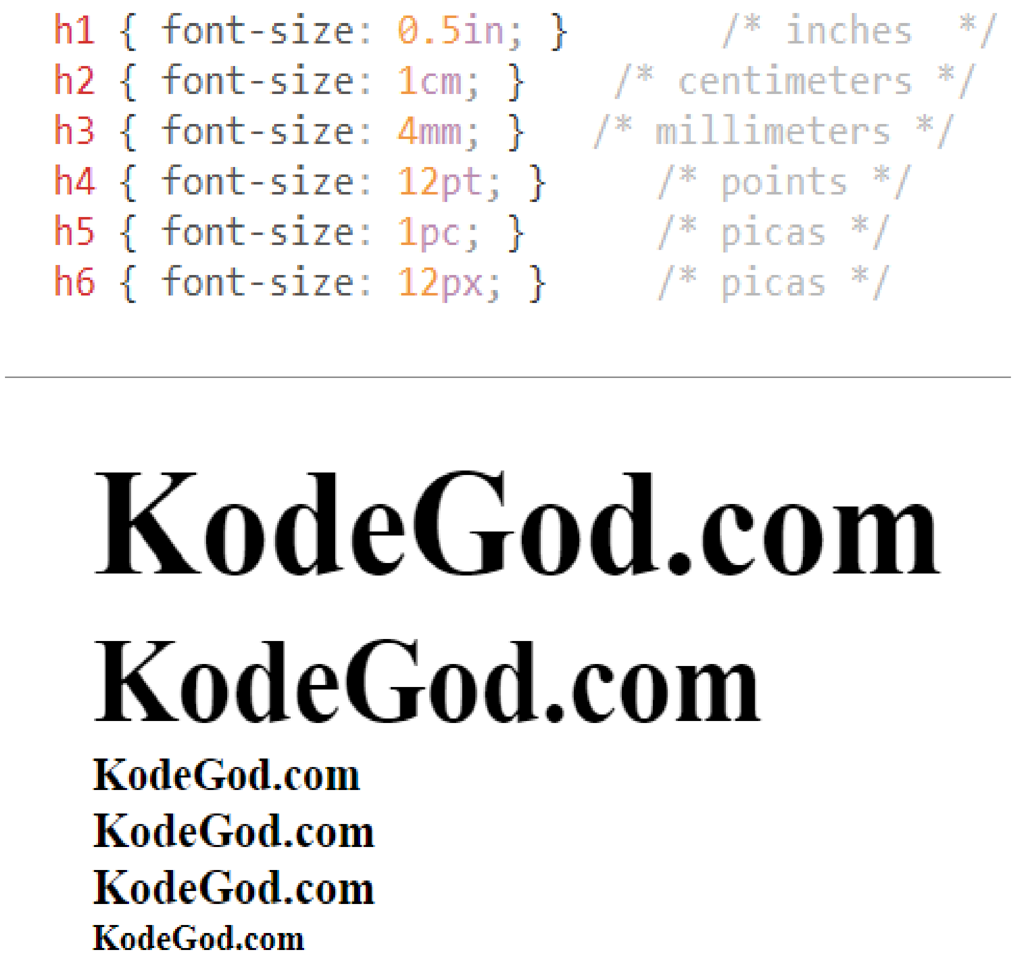
KodeGod.com

KodeGod.com

KodeGod.com

Absolute Length Units

- This are fixed and appear as specified
- Used when output medium is known
- in, cm, mm used for print media and high-resolution devices
- px, pt, pc used for desktop and lower resolution devices



```
h1 { font-size: 0.5in; }       /* inches  */
h2 { font-size: 1cm; }     /* centimeters */
h3 { font-size: 4mm; }    /* millimeters */
h4 { font-size: 12pt; }      /* points */
h5 { font-size: 1pc; }       /* picas */
h6 { font-size: 12px; }      /* picas */
```

KodeGod.com

KodeGod.com

KodeGod.com

KodeGod.com

KodeGod.com

KodeGod.com

Display

CSS3 by Prashant Chaudhari

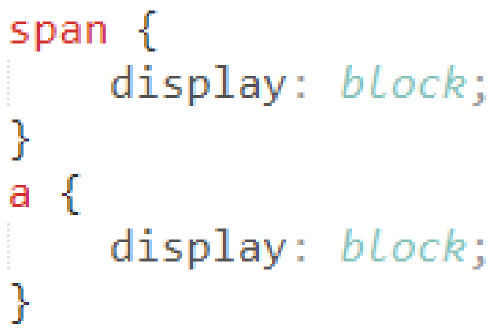
Display

- Controls the box’s type generated by element
- Every element has default display value
- E.g. div element has display value block and span has inline
- To change default display value **display** property is used

Display - Properties

- **Display Block**
  - Forces element to behave like

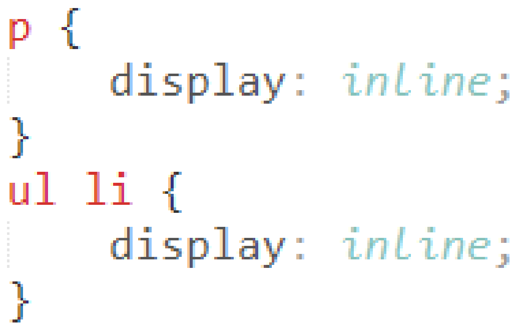
    block-level element like **div** and **p**



```
span {
    display: block;
}
a {
    display: block;
}
```

- **Display Inline**
  - Forces element to behave like

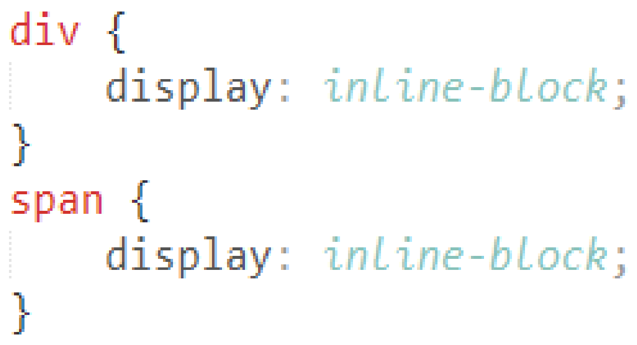
    inline-level element like **span** and **a**



```
p {
    display: inline;
}
ul li {
    display: inline;
}
```

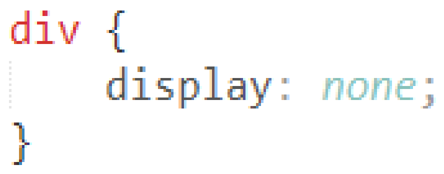
Display - Properties

- **Display Inline-Block**
  - Forces element to generate block box with content in the same line



```
div {
    display: inline-block;
}
span {
    display: inline-block;
}
```

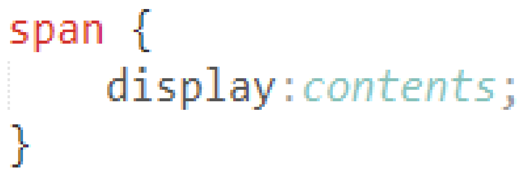
- **Display None**
  - Simply hides the element and its child elements



```
div {
    display: none;
}
```

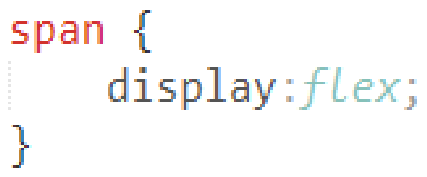
Display - Properties

- **Display Contents**
  - Makes container disappear
  - Makes child elements children of the element next level up in the DOM



```
span {
    display:contents;
}
```

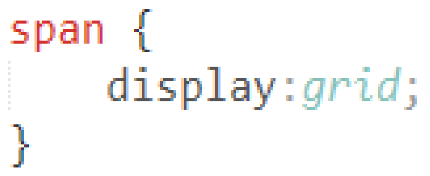
- **Display Flex**
  - Displays element as block-level flex container



```
span {
    display:flex;
}
```

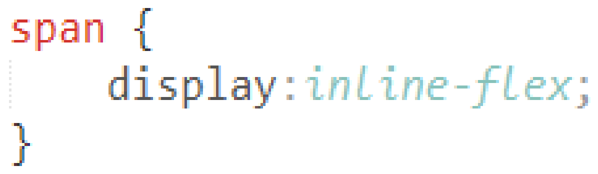
Display - Properties

- **Display Grid**
  - Displays element as block-level grid container



```
span {
    display:grid;
}
```

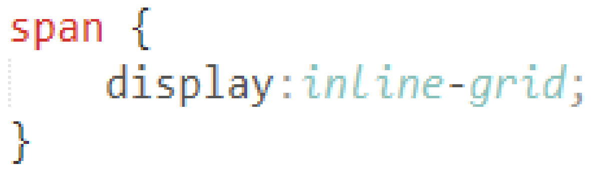
- **Display Inline-Flex**
  - Displays element as inline-level flex container



```
span {
    display:inline-flex;
}
```

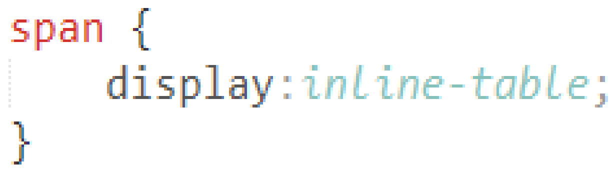
Display - Properties

- **Display Inline-Grid**
  - Displays element as inline-level grid container



```
span {
    display:inline-grid;
}
```

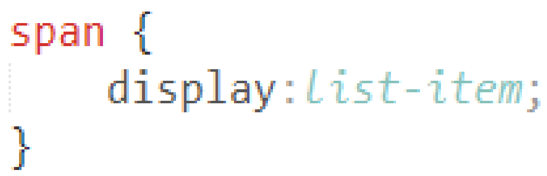
- **Display Inline-Table**
  - Displays element as inline-level table



```
span {
    display:inline-table;
}
```

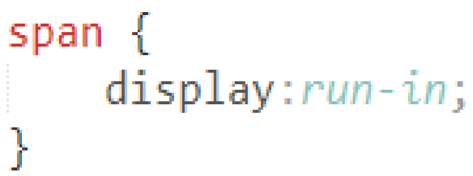
Display - Properties

- **Display List-Item**
  - Let element behave like <li> element



```
span {
    display:list-item;
}
```

- **Display Run-In**
  - Displays element as either block or inline, depending on context

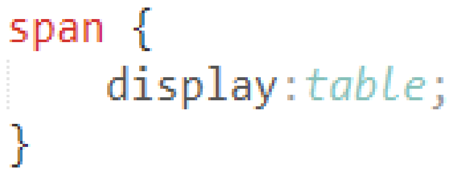


```
span {
    display:run-in;
}
```

Display - Properties

- **Display Table**
  - Let the element behave like

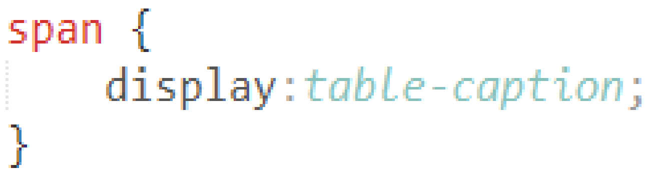
    <table> element



```
span {
    display:table;
}
```

- **Display Table-Caption**
  - Let the element behave like

    <caption> element

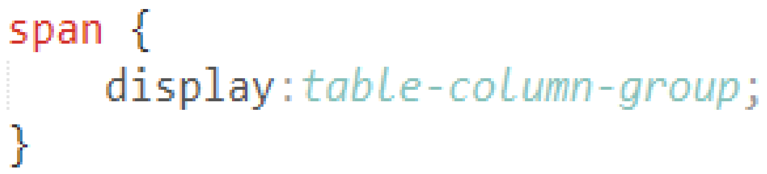


```
span {
    display:table-caption;
}
```

Display - Properties

- **Display Table-Column-Group**
  - Let the element behave like

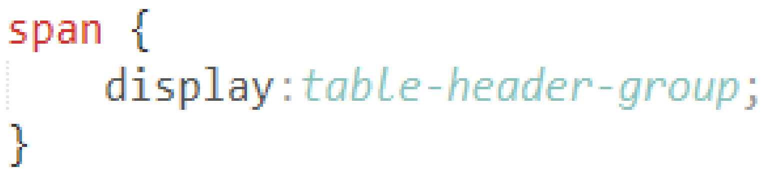
    <colgroup> element



```
span {
    display:table-column-group;
}
```

- **Display Table-Header-Group**
  - Let the element behave like

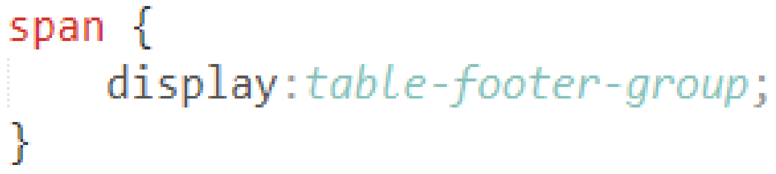
    <thead> element



```
span {
    display:table-header-group;
}
```

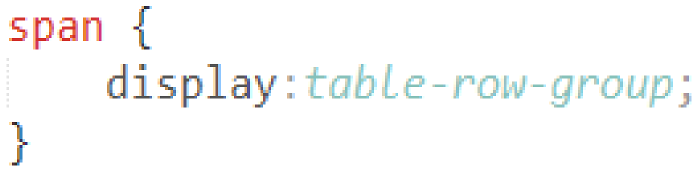
Display - Properties

- **Display Table-Footer-Group**
  - Let the element behave like <tfoot> element



```
span {
    display:table-footer-group;
}
```

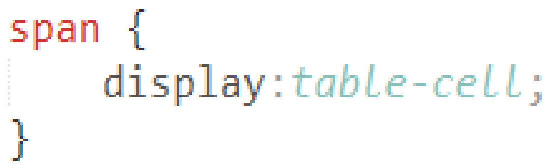
- **Display Table-Row-Group**
  - Let the element behave like <tbody> element



```
span {
    display:table-row-group;
}
```

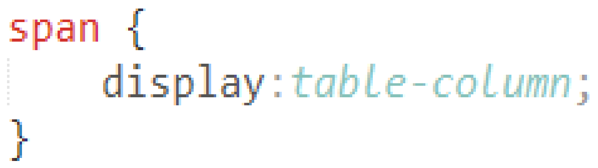
Display - Properties

- **Display Table-Cell**
  - Let the element behave like <td> element



```
span {
    display:table-cell;
}
```

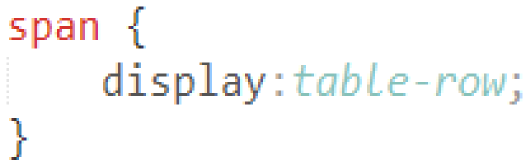
- **Display Table-Column**
  - Let the element behave like <col> element



```
span {
    display:table-column;
}
```

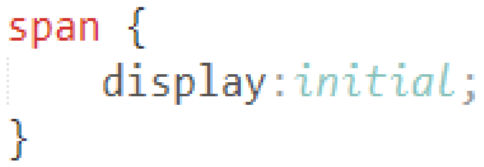
Display - Properties

- **Display Table-Row**
  - Let the element behave like <tr> element



```
span {
    display:table-row;
}
```

- **Display Initial**
  - Sets to default value



```
span {
    display:initial;
}
```

Visibility

CSS3 by Prashant Chaudhari

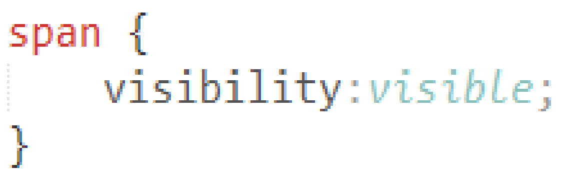
Visibility

- Determines whether element is visible or hidden
- **visibility** property is used
- **Property values**

  visible, hidden, collapse, inherit

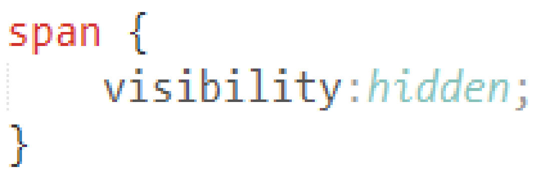
Visibility - Properties

- **visible**
  - Element block and its contents are visible



```
span {
    visibility:visible;
}
```

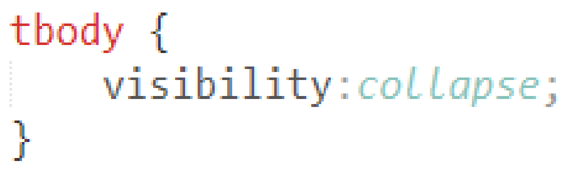
- **hidden**
  - Element block and its contents are invisible but still can affect the layout of page



```
span {
    visibility:hidden;
}
```

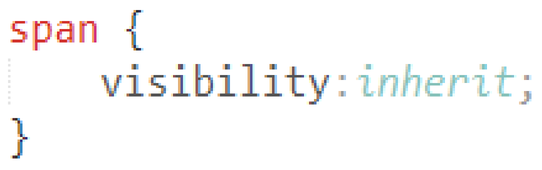
Visibility - Properties

- **collapse**
  - Causes entire row and column to be removed from display.
  - Used for row, row-group, column, column-group elements



```css
tbody {
    visibility:collapse;
}
```

- **inherit**
  - Property value inherited from parent element
  - i.e. if parent elements visibility value is collapse then inherited value is collapse



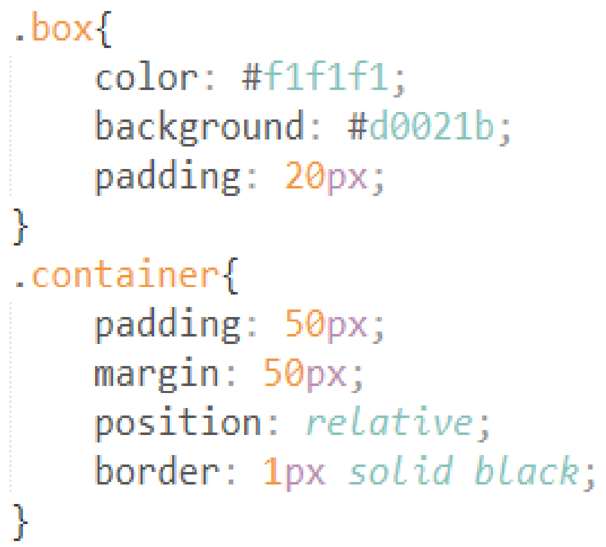
```css
span {
    visibility:inherit;
}
```

Position

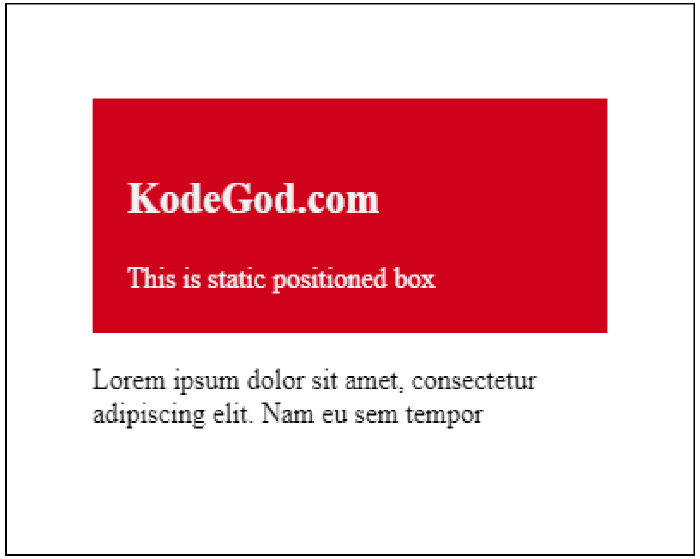
CSS3 by Prashant Chaudhari

Position

- Defines how an element will be positioned on a page
- **Positioning methods**
- **Static positioning**
  - Positioned according to the normal flow of page
  - Elements positioned static by default
  - Static positioned elements not affected by top, bottom, left, right, z-index properties



```
.box{
    color: #f1f1f1;
    background: #d0021b;
    padding: 20px;
}
.container{
    padding: 50px;
    margin: 50px;
    position: relative;
    border: 1px solid black;
}
```



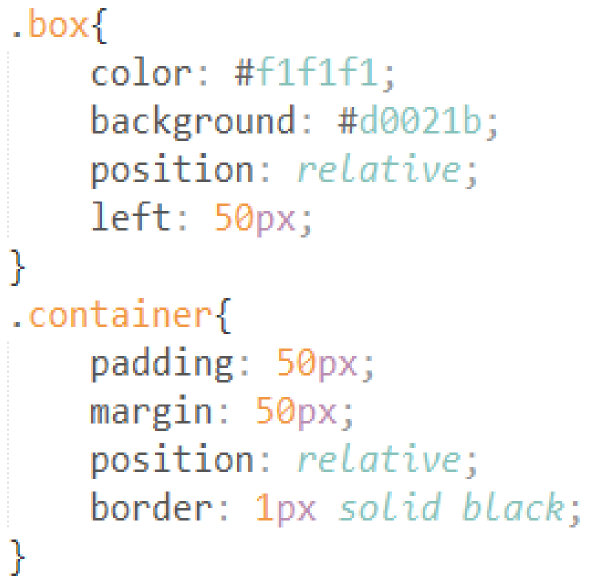
KodeGod.com

This is static positioned box

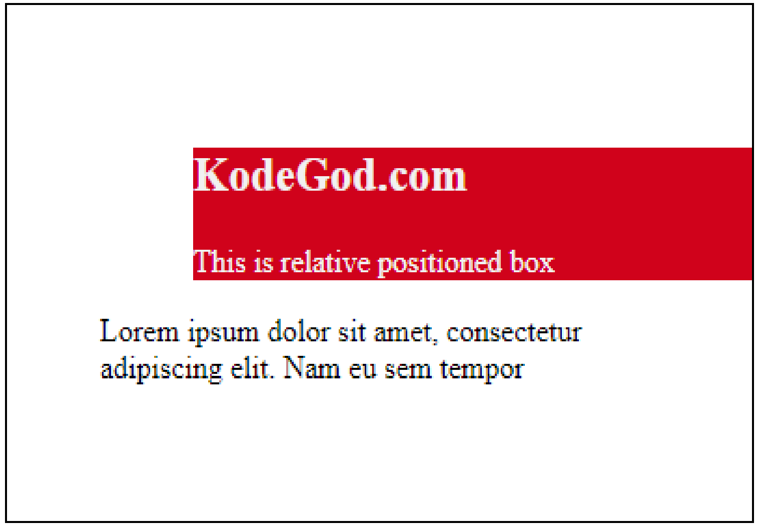
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem tempor

Position

- **Relative positioning**
  - Positioned relative to its normal position
  - Elements shifted from its normal position using properties top, bottom, left, right



```
.box{
    color: #f1f1f1;
    background: #d0021b;
    position: relative;
    left: 50px;
}
.container{
    padding: 50px;
    margin: 50px;
    position: relative;
    border: 1px solid black;
}
```



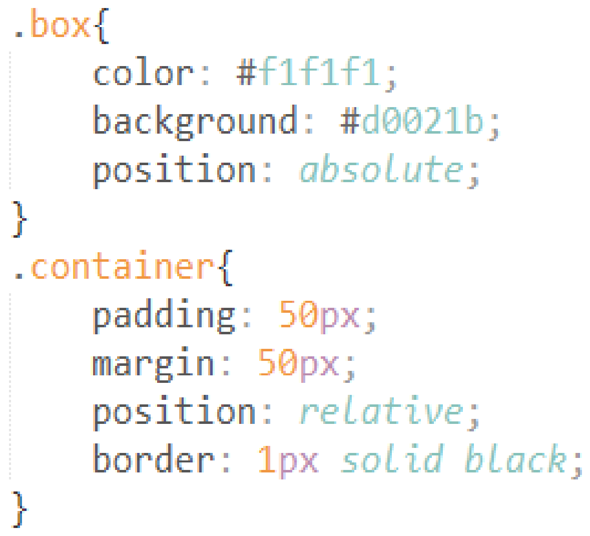
KodeGod.com

This is relative positioned box

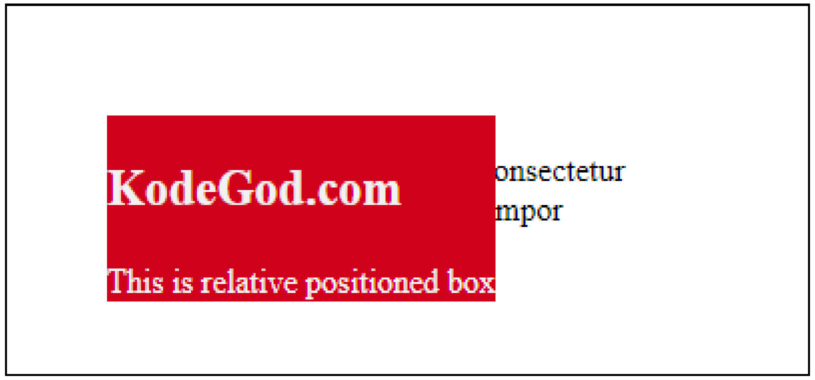
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem tempor

Position

- **Absolute positioning**
  - Positioned relative to first parent element that has position other than static
  - It can overlap other elements depending z-index property value



```
.box{
    color: #f1f1f1;
    background: #d0021b;
    position: absolute;
}
.container{
    padding: 50px;
    margin: 50px;
    position: relative;
    border: 1px solid black;
}
```



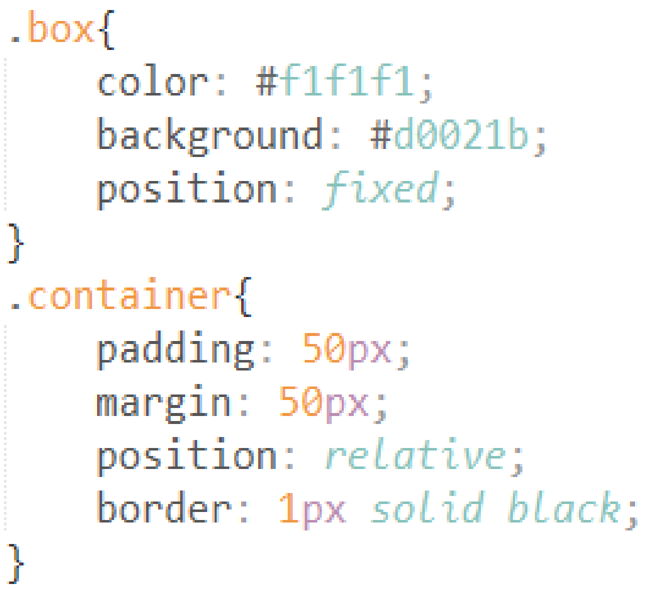
KodeGod.com

This is relative positioned box

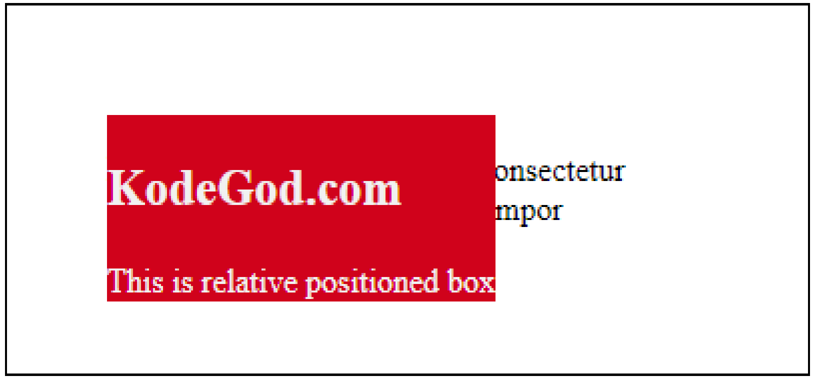
onsectetur
mpor

Position

- **Fixed positioning**
  - Subcategory of absolute positioning
  - Fixed with respect to browser's viewport
  - Does not move when scroll



```
.box{
    color: #f1f1f1;
    background: #d0021b;
    position: fixed;
}
.container{
    padding: 50px;
    margin: 50px;
    position: relative;
    border: 1px solid black;
}
```



KodeGod.com

This is relative positioned box

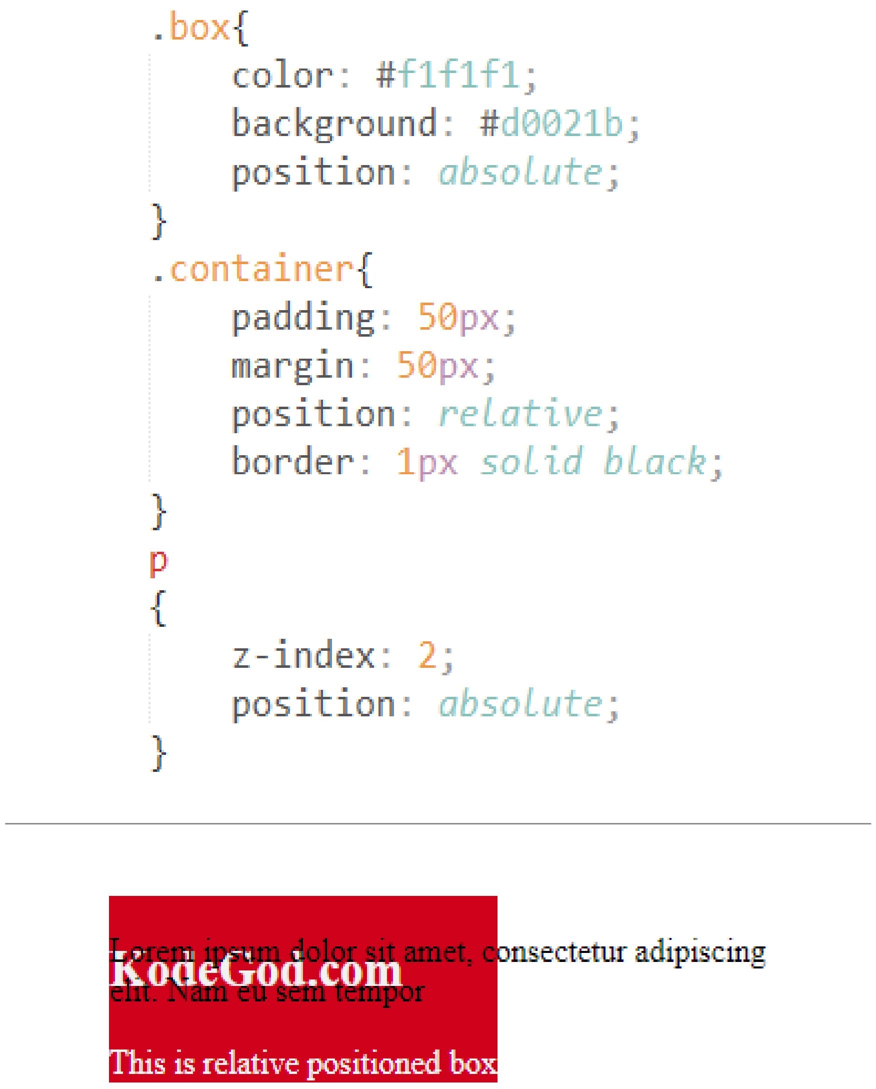
onsectetur
mpor

Layers

CSS3 by Prashant Chaudhari

Layers

- **z-index** property can be used with **position** property to create layers
- Elements can be stacked along z-axis one on top of other using z-index
- Values is in integer represents stacking order of rendering
- Larger z-index overlaps lower one



```
.box{
    color: #f1f1f1;
    background: #d0021b;
    position: absolute;
}
.container{
    padding: 50px;
    margin: 50px;
    position: relative;
    border: 1px solid black;
}
p
{
    z-index: 2;
    position: absolute;
}
```

KodeGod.com

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem tempor

This is relative positioned box

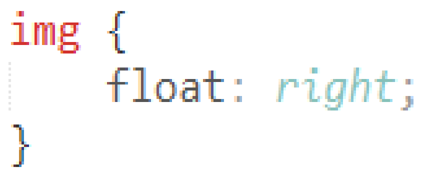
Float

CSS3 by Prashant Chaudhari

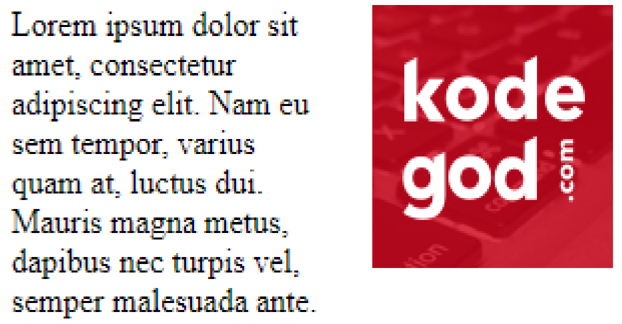
Float

- **float** property specifies whether element should float or not
- Applies only to box-level elements
- Elements which are not absolutely positioned
- **Property values**

  left, right, none



```
img {
    float: right;
}
```



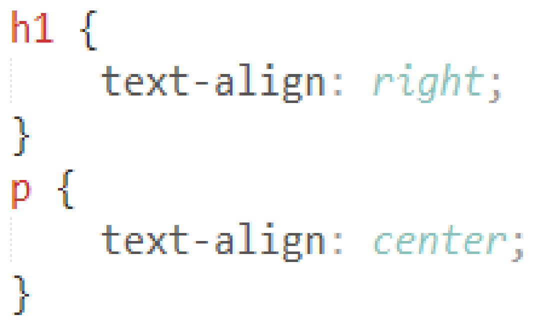
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem tempor, varius quam at, luctus dui. Mauris magna metus, dapibus nec turpis vel, semper malesuada ante.

Alignment

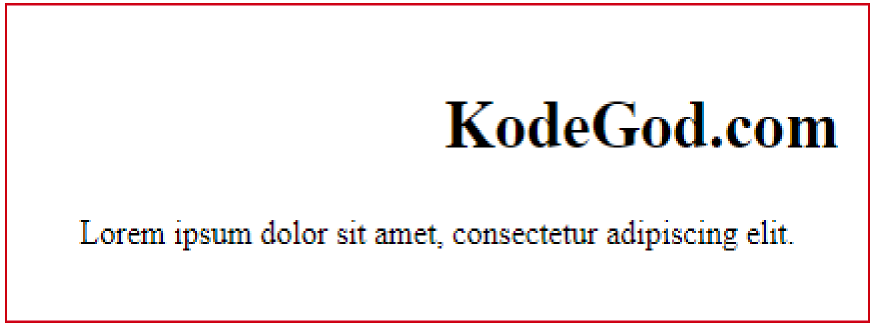
CSS3 by Prashant Chaudhari

Alignment

- Text inside box-level elements align using **text-align** property
- **Property values**
  - left align to left
  - right align to right
  - center align to center
  - initial sets default value
  - inherit takes value of parent element
- start align to start of element
- end align to end of element



```css
h1 {
    text-align: right;
}
p {
    text-align: center;
}
```



KodeGod.com

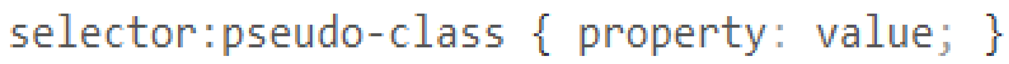
Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Pseudo Classes

CSS3 by Prashant Chaudhari

Pseudo Classes

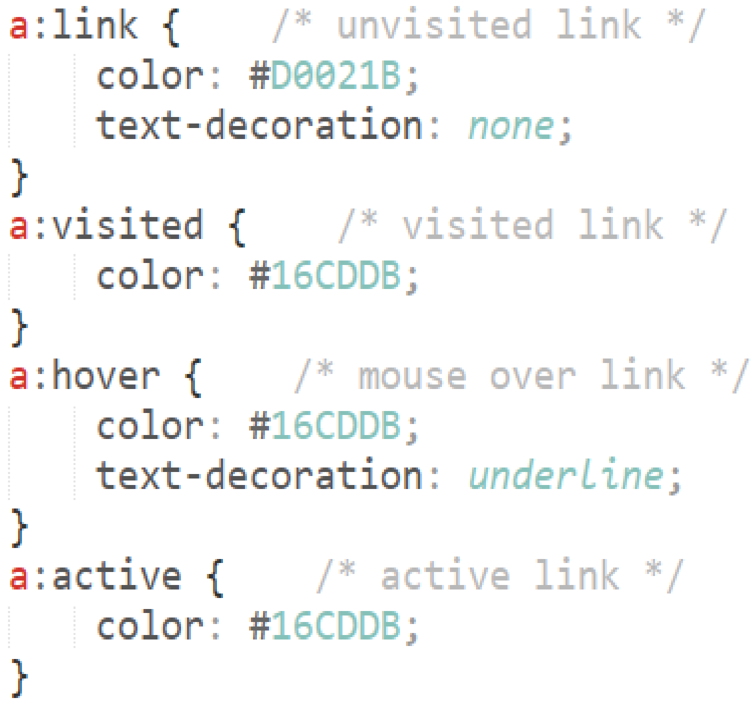
- Matches elements based on additional condition
- Allows to style dynamic states of elements (hover, active, focus)
- Starts with (:)
- **Syntax**



```
selector:pseudo-class { property: value; }
```

Pseudo Classes

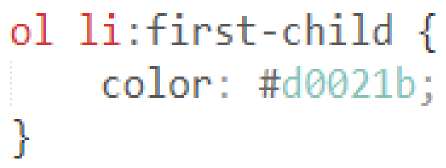
- **Anchor Pseudo-Classes**
    - Links can displayed in different ways



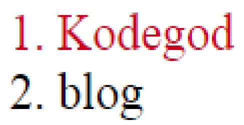
```
a:link {     /* unvisited link */
    color: #D0021B;
    text-decoration: none;
}
a:visited {     /* visited link */
    color: #16CDDB;
}
a:hover {     /* mouse over link */
    color: #16CDDB;
    text-decoration: underline;
}
a:active {     /* active link */
    color: #16CDDB;
}
```

Pseudo Classes

- **:first-child** pseudo-class
  - Matches element that is first child element of other element



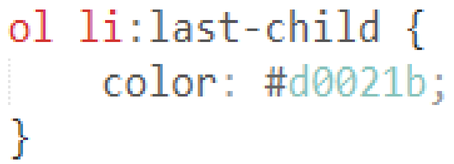
```
ol li:first-child {
    color: #d0021b;
}
```



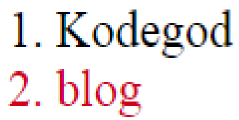
1. Kodegod
2. blog

Pseudo Classes

- **:last-child** pseudo-class
  - Matches element that is last child element of other element



```
ol li:last-child {
    color: #d0021b;
}
```



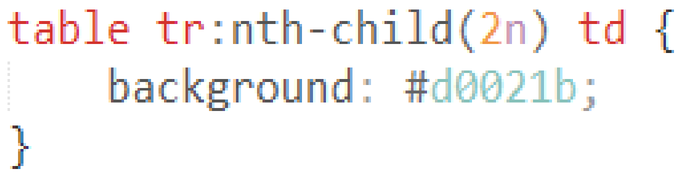
1. Kodegod
2. blog

Pseudo Classes

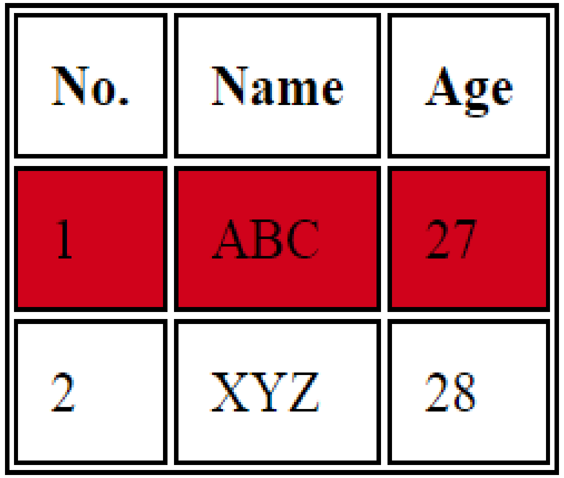
- **:nth-child** pseudo-class
  - Targets one or more element of parent element
  - **Syntax**

    :nth-child(N), where (**N**) can be a number,

    keyword (odd, even) or expression (xn +y)



```css
table tr:nth-child(2n) td {
    background: #d0021b;
}
```

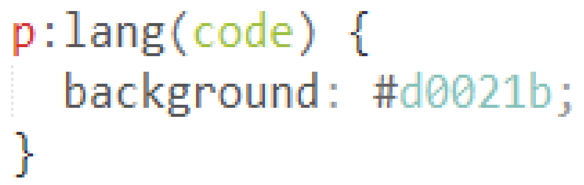


No.	Name	Age
1	ABC	27
2	XYZ	28

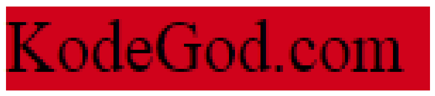
Pseudo Classes

- **:lang** pseudo-class
  - Used to select elements with **lang** attribute with specified value
  - **Syntax**

    :lang(languagecode)



```
p:lang(code) {
  background: #d0021b;
}
```



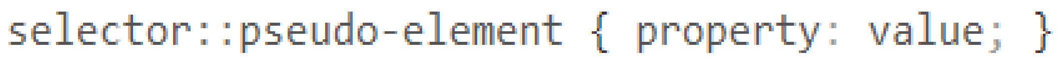
KodeGod.com

Pseudo Elements

CSS3 by Prashant Chaudhari

Pseudo Elements

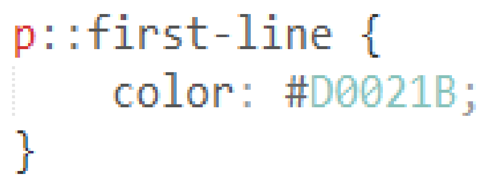
- Way of styling elements that are explicitly defined by a position in document tree
- Can style elements without id or class
- **Syntax**



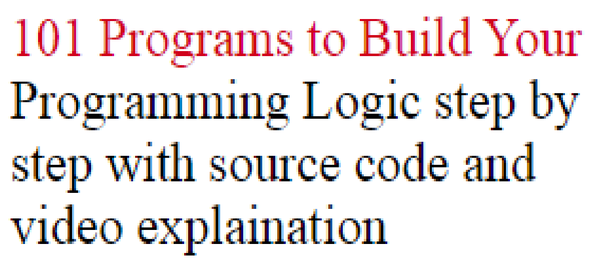
```
selector::pseudo-element { property: value; }
```

Pseudo Elements

- **::first-line** pseudo element
  - Applies to first line of text



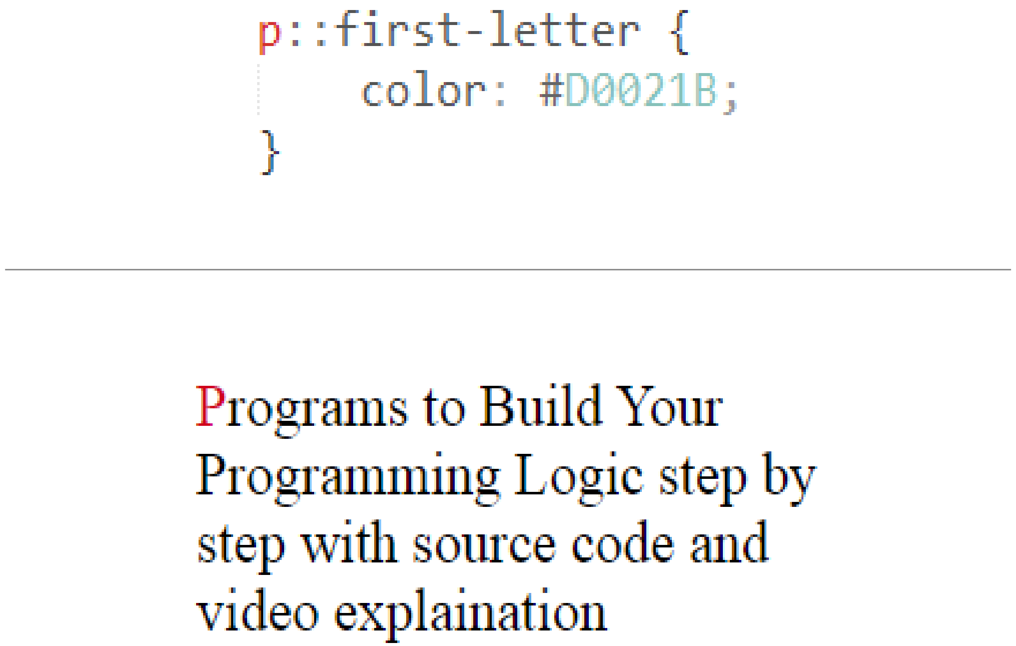
```
p::first-line {
    color: #D0021B;
}
```



101 Programs to Build Your Programming Logic step by step with source code and video explaination

Pseudo Elements

- **::first-letter** pseudo element
  - Applies style to first letter of first line of text

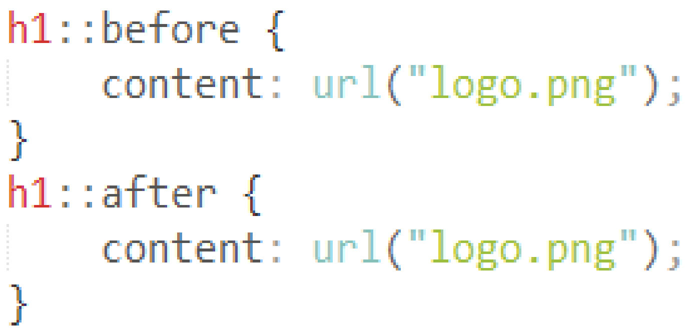


```
p::first-letter {
    color: #D0021B;
}
```

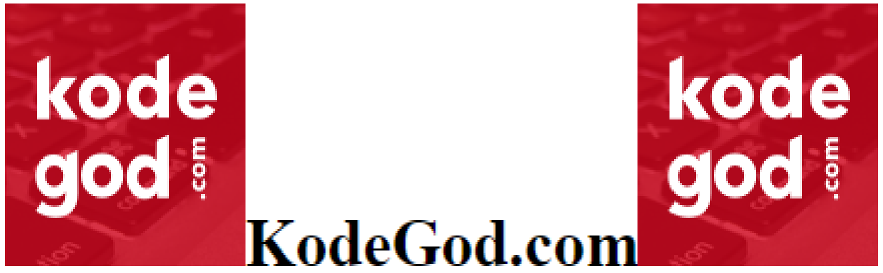
Programs to Build Your Programming Logic step by step with source code and video explaination

Pseudo Elements

- **::before and ::after** pseudo element
  - Used to insert content before or after element’s content
  - **content** property is used to add content



```css
h1::before {
    content: url("logo.png");
}
h1::after {
    content: url("logo.png");
}
```



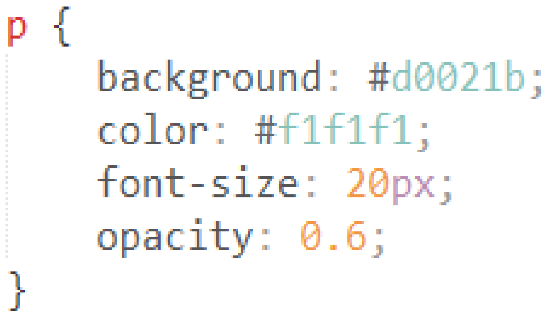
KodeGod.com

Opacity

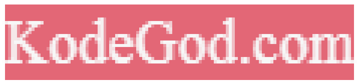
CSS3 by Prashant Chaudhari

Opacity

- Specifies transparency of element
- Cross browser opacity
  - Firefox, safari, chrome, opera, IE9



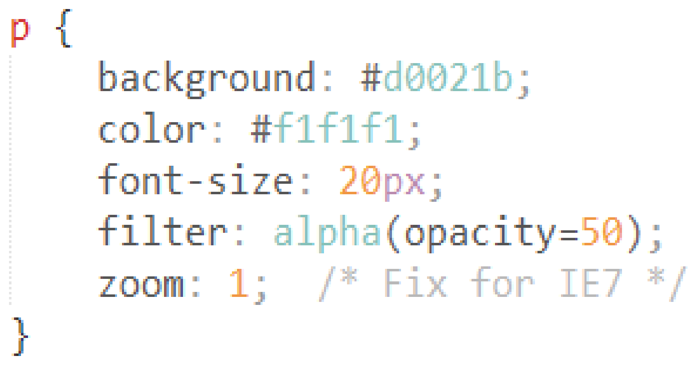
```css
p {
    background: #d0021b;
    color: #f1f1f1;
    font-size: 20px;
    opacity: 0.6;
}
```



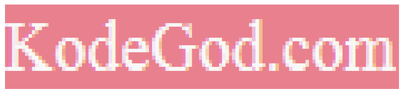
KodeGod.com

Opacity

- Cross browser opacity
  - IE8 and lower
    - Supports a Microsoft-only property "alpha filter"
  - **zoom** property is used to scale the content

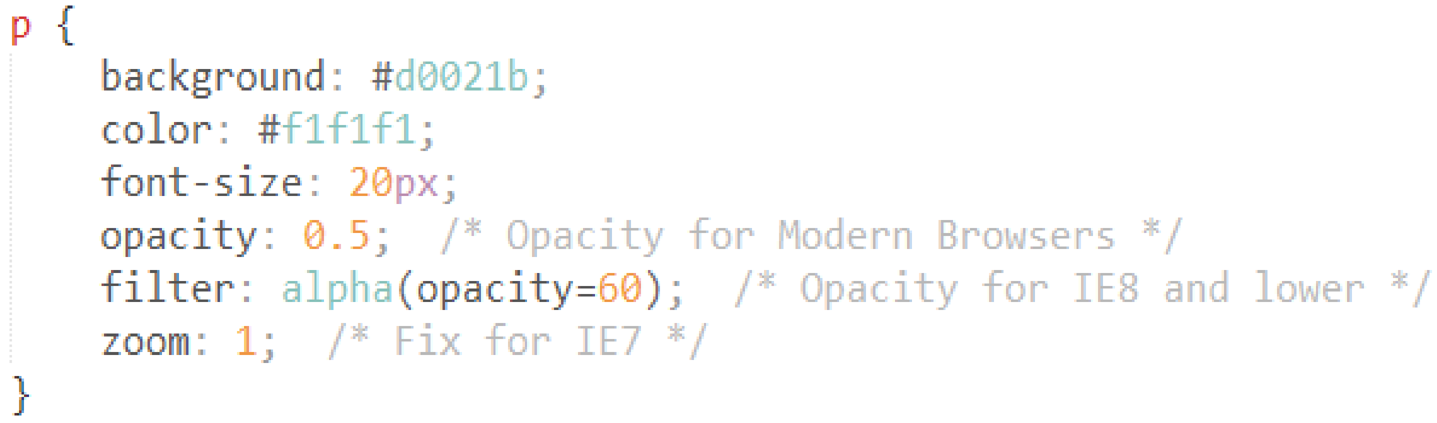


```
p {
    background: #d0021b;
    color: #f1f1f1;
    font-size: 20px;
    filter: alpha(opacity=50);
    zoom: 1;  /* Fix for IE7 */
}
```



Opacity

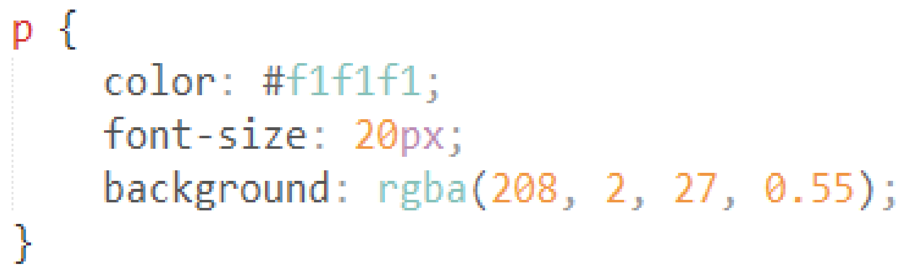
- All browsers



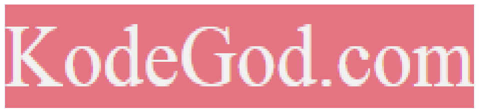
```
p {
    background: #d0021b;
    color: #f1f1f1;
    font-size: 20px;
    opacity: 0.5;  /* Opacity for Modern Browsers */
    filter: alpha(opacity=60);  /* Opacity for IE8 and lower */
    zoom: 1;  /* Fix for IE7 */
}
```

Opacity

- Using **RGBA**
  - rgba value includes alpha transparency as part of color value.
  - RGB value ranges between (0-255) for each red, green and blue color and alpha value ranges between 0 to 1



```
p {
    color: #f1f1f1;
    font-size: 20px;
    background: rgba(208, 2, 27, 0.55);
}
```



KodeGod.com

Gradients

CSS3 by Prashant Chaudhari

Gradients

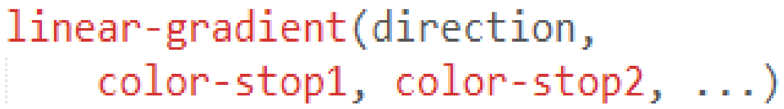
- Allows to create smooth transitions between two

  or more colors
- **Available in two styles**
  - **Linear**
  - **Radial**

Linear Gradient

- Must define at least two color stops
- Can also set starting point and a direction along

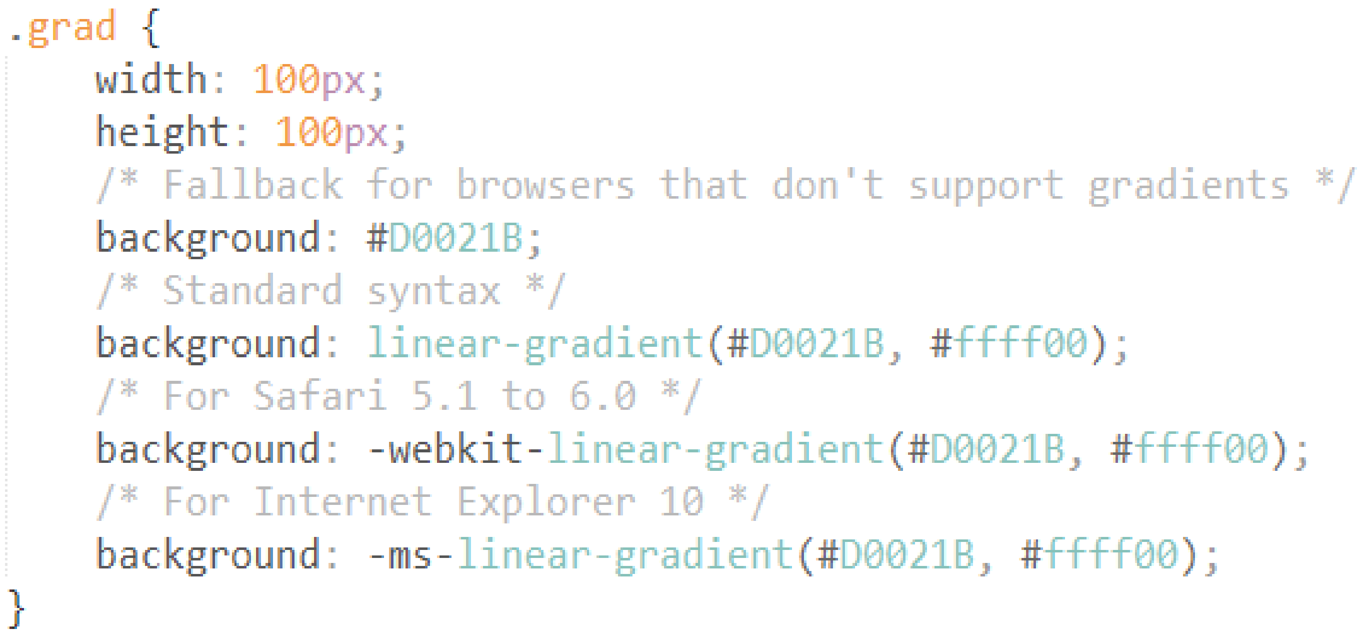
  which gradient effect is applied
- **Syntax**



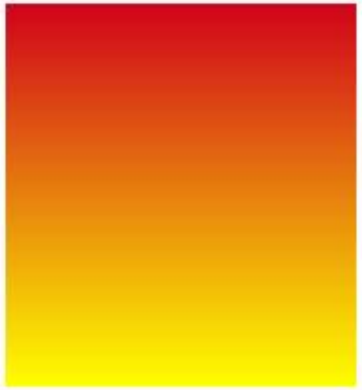
```
linear-gradient(direction,
    color-stop1, color-stop2, ...)
```

Linear Gradient

- **Linear gradient (Top to bottom)**

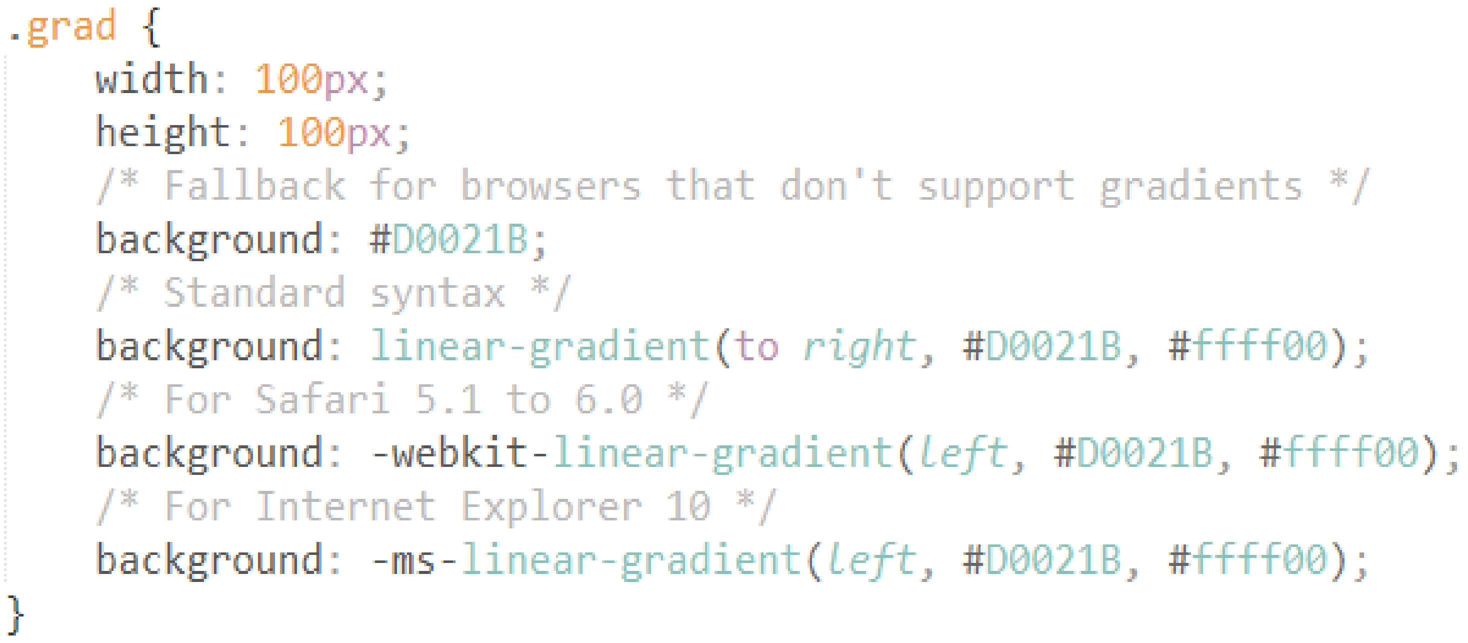


```css
.grad {
    width: 100px;
    height: 100px;
    /* Fallback for browsers that don't support gradients */
    background: #D0021B;
    /* Standard syntax */
    background: linear-gradient(#D0021B, #ffff00);
    /* For Safari 5.1 to 6.0 */
    background: -webkit-linear-gradient(#D0021B, #ffff00);
    /* For Internet Explorer 10 */
    background: -ms-linear-gradient(#D0021B, #ffff00);
}
```

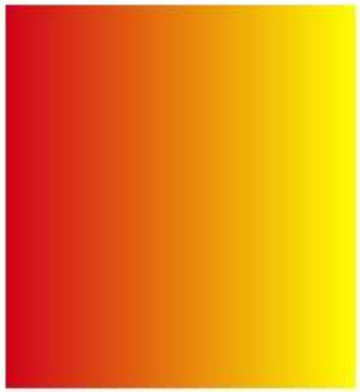


Linear Gradient

- **Linear gradient (Left to Right)**

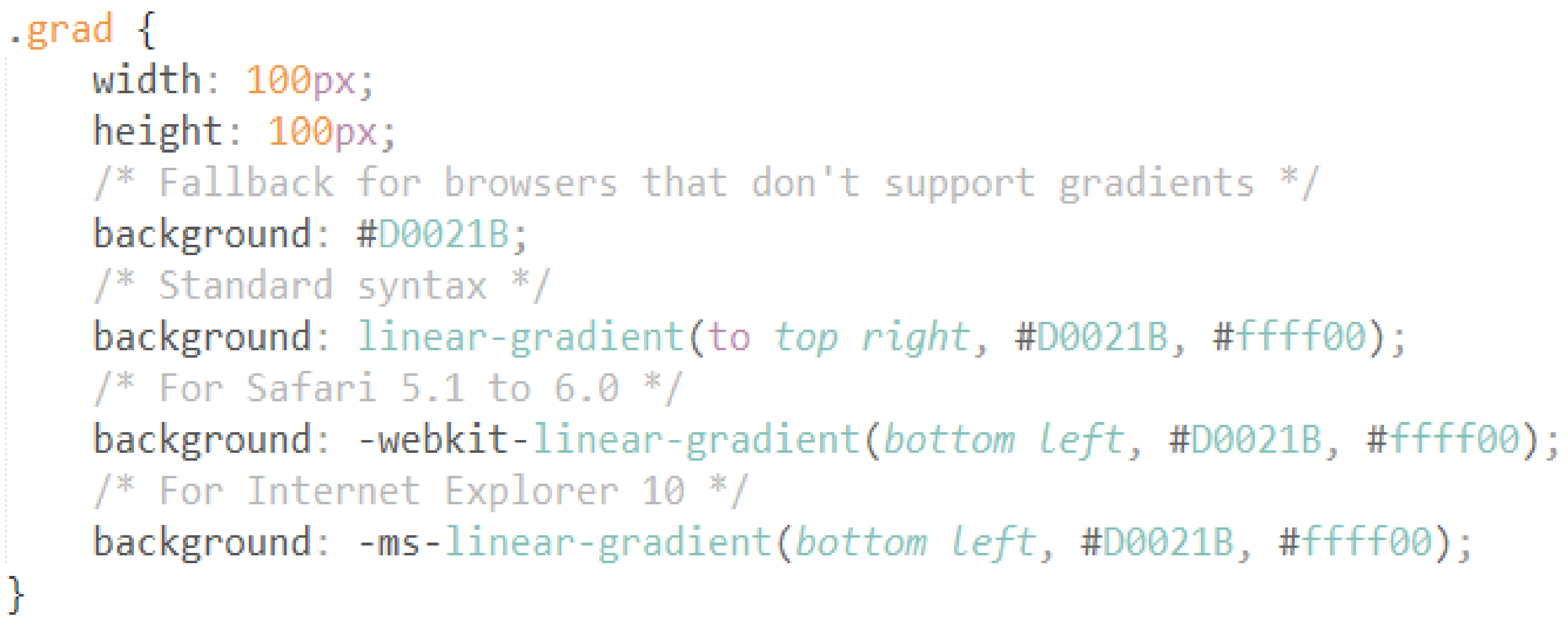


```css
.grad {
    width: 100px;
    height: 100px;
    /* Fallback for browsers that don't support gradients */
    background: #D0021B;
    /* Standard syntax */
    background: linear-gradient(to right, #D0021B, #ffff00);
    /* For Safari 5.1 to 6.0 */
    background: -webkit-linear-gradient(left, #D0021B, #ffff00);
    /* For Internet Explorer 10 */
    background: -ms-linear-gradient(left, #D0021B, #ffff00);
}
```

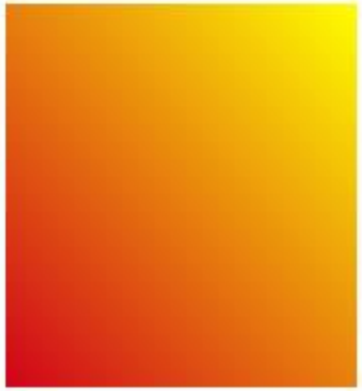


Linear Gradient

- **Linear gradient (Diagonal)**

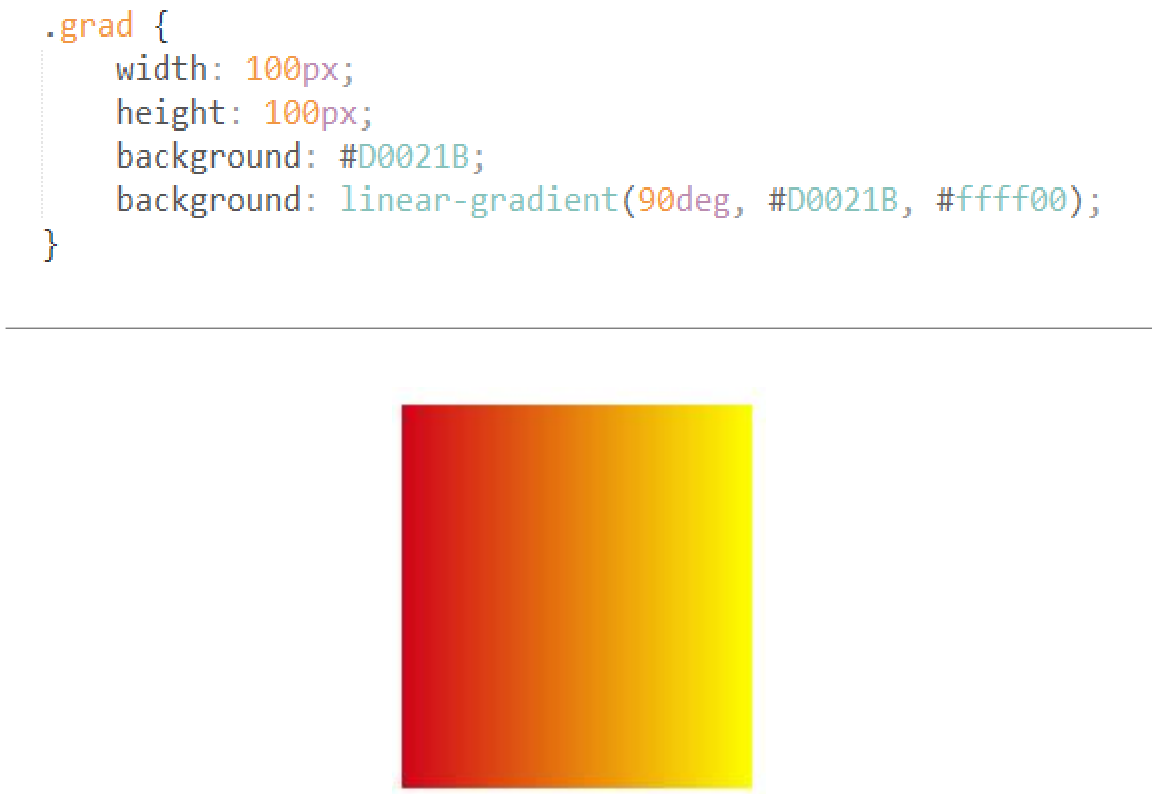


```
.grad {
    width: 100px;
    height: 100px;
    /* Fallback for browsers that don't support gradients */
    background: #D0021B;
    /* Standard syntax */
    background: linear-gradient(to top right, #D0021B, #ffff00);
    /* For Safari 5.1 to 6.0 */
    background: -webkit-linear-gradient(bottom left, #D0021B, #ffff00);
    /* For Internet Explorer 10 */
    background: -ms-linear-gradient(bottom left, #D0021B, #ffff00);
}
```



Linear Gradient

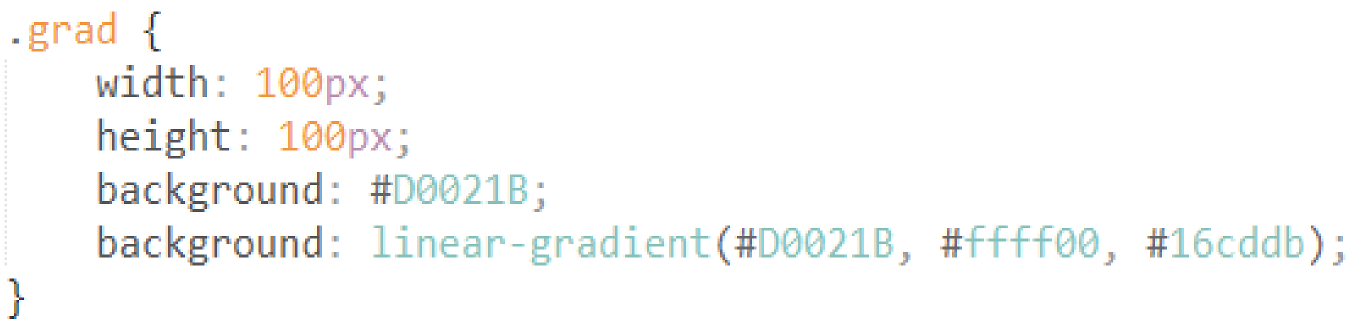
- **Linear gradient (Using angles)**
  - Can control directions using angles instead of predefined keywords
  - Angle 0deg creates bottom to top gradient, and positive angle represents clockwise rotation



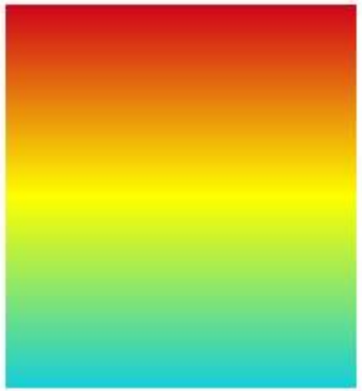
```
.grad {
    width: 100px;
    height: 100px;
    background: #D0021B;
    background: linear-gradient(90deg, #D0021B, #ffff00);
}
```

Linear Gradient

- **Linear gradient (Multiple colors)**

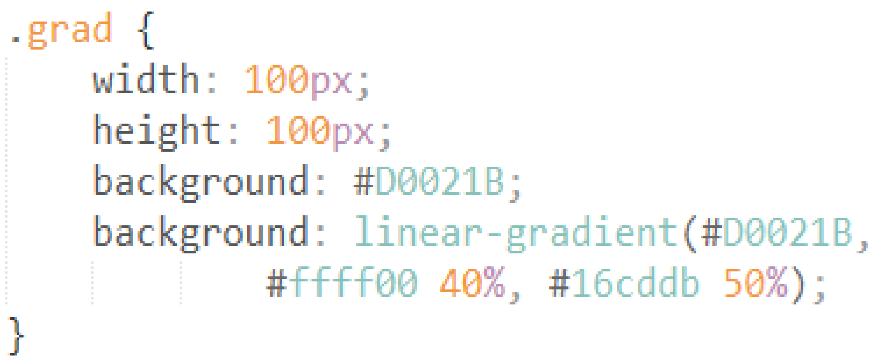


```css
.grad {
    width: 100px;
    height: 100px;
    background: #D0021B;
    background: linear-gradient(#D0021B, #ffff00, #16cddb);
}
```

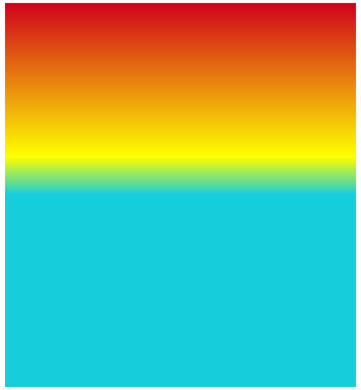


Linear Gradient

- **Linear gradient (Location color stops)**
  - Location color stop can be specified either as percentage or an absolute length

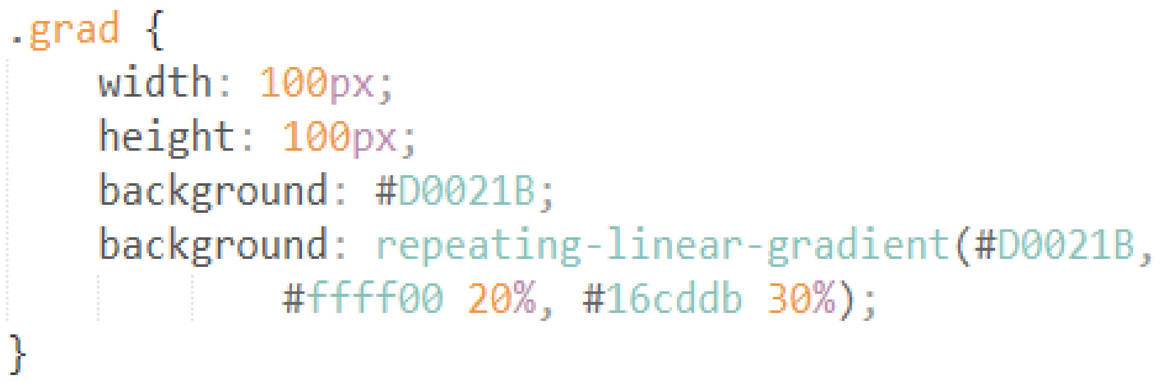


```
.grad {
    width: 100px;
    height: 100px;
    background: #D0021B;
    background: linear-gradient(#D0021B,
            #ffff00 40%, #16cddb 50%);
}
```

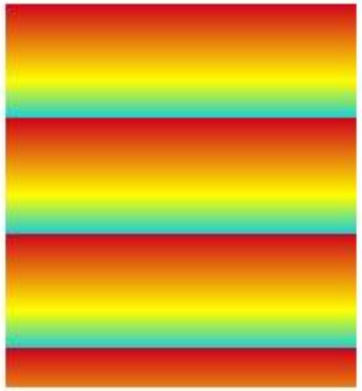


Linear Gradient

- **Repeating Linear gradient**

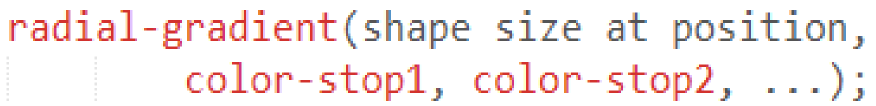


```
.grad {
    width: 100px;
    height: 100px;
    background: #D0021B;
    background: repeating-linear-gradient(#D0021B,
                #ffff00 20%, #16cddb 30%);
}
```



Radial Gradient

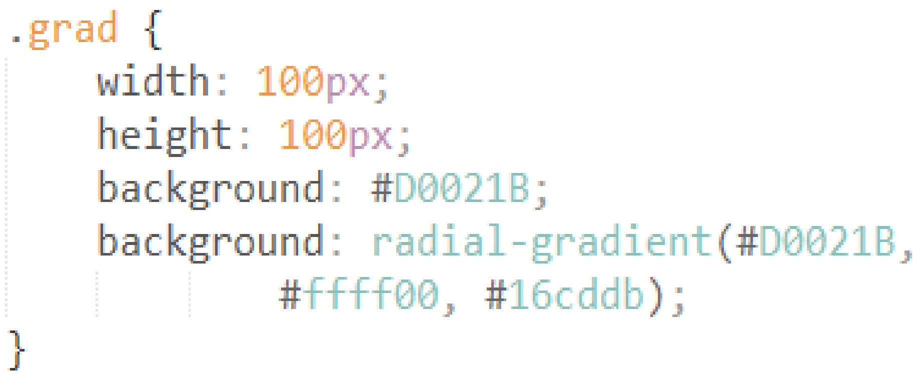
- Color emerge from single point and smoothly spread in circular or elliptical shape
- **Syntax**



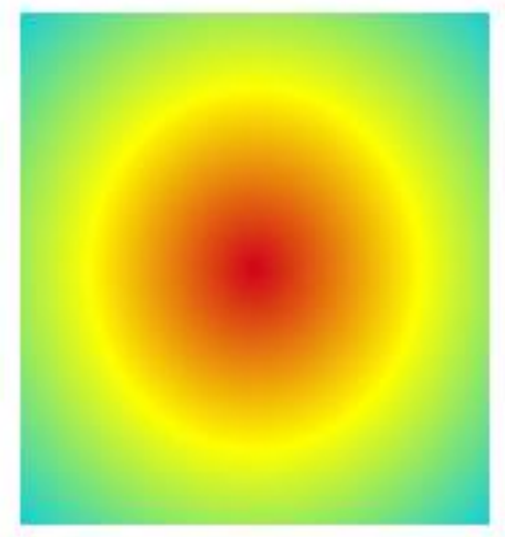
```
radial-gradient(shape size at position,
          color-stop1, color-stop2, ...);
```

- **Shape** Gradients ending shape, can be circle or ellipse
- **Size** Size of the gradients ending shape, default is farthest-side
- **Position** Starting point of the gradient, can be in units (px, em or %) or keywords (left, right, top, bottom)

Radial Gradient

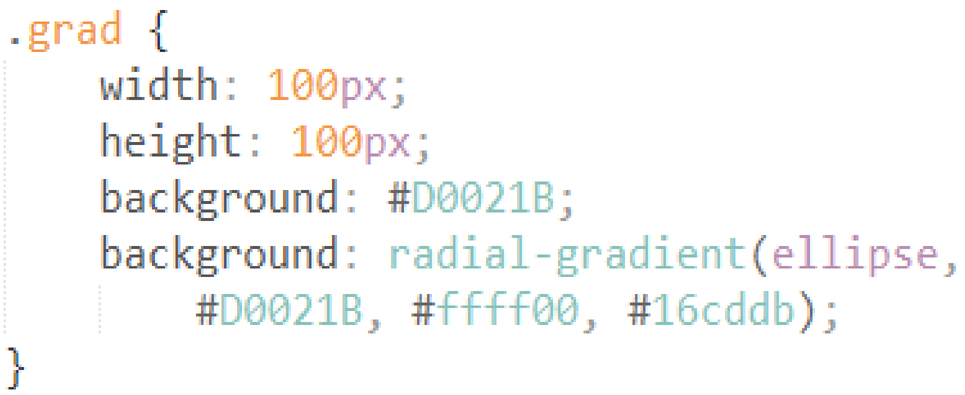


```
.grad {
    width: 100px;
    height: 100px;
    background: #D0021B;
    background: radial-gradient(#D0021B,
            #ffff00, #16cddb);
}
```

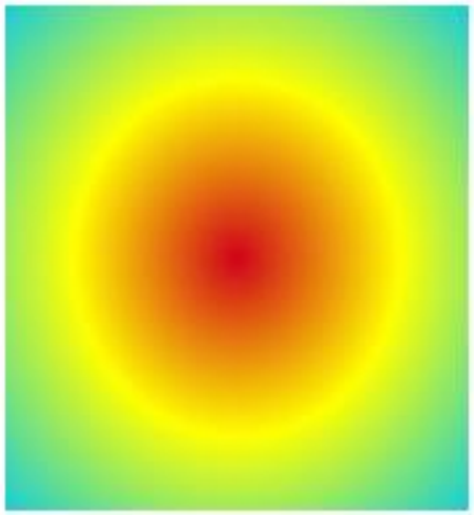


Radial Gradient

- **Radial (Using Shape)**

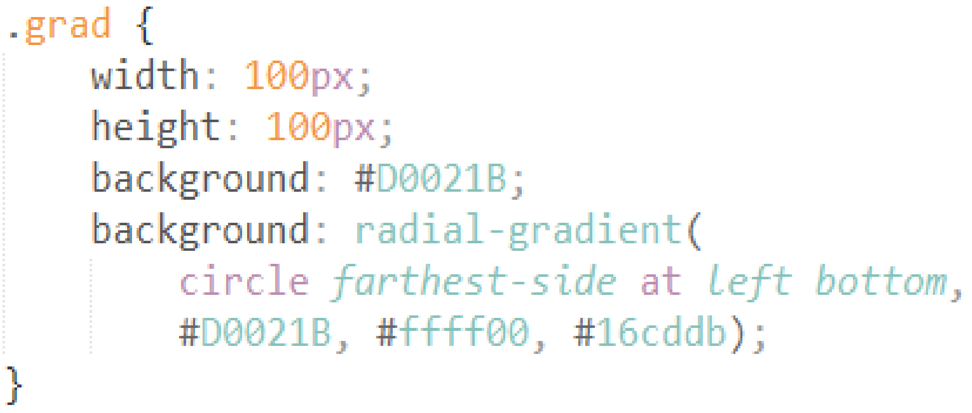


```
.grad {
    width: 100px;
    height: 100px;
    background: #D0021B;
    background: radial-gradient(ellipse,
        #D0021B, #ffff00, #16cddb);
}
```

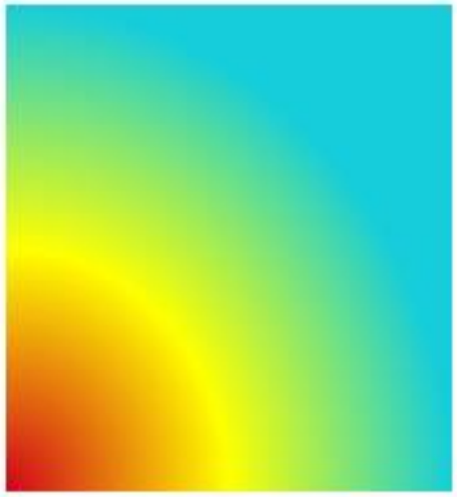


Radial Gradient

- **Radial (Using Size and Position)**
  - Can be set using units or keywords (closest-side, farthest-side, closest-corner, farthest-corner)

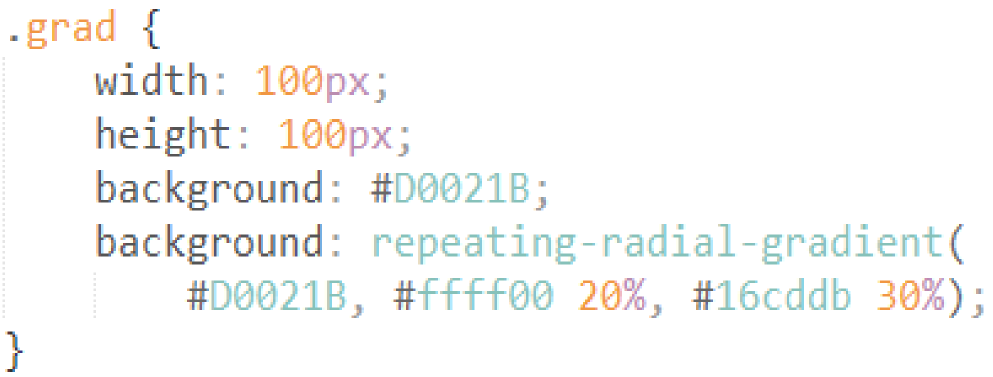


```
.grad {
    width: 100px;
    height: 100px;
    background: #D0021B;
    background: radial-gradient(
        circle farthest-side at left bottom,
        #D0021B, #ffff00, #16cddb);
}
```

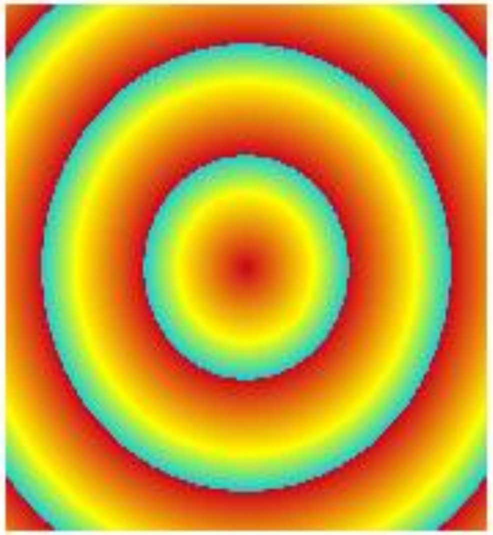


Radial Gradient

- **Repeating Radial Gradient**

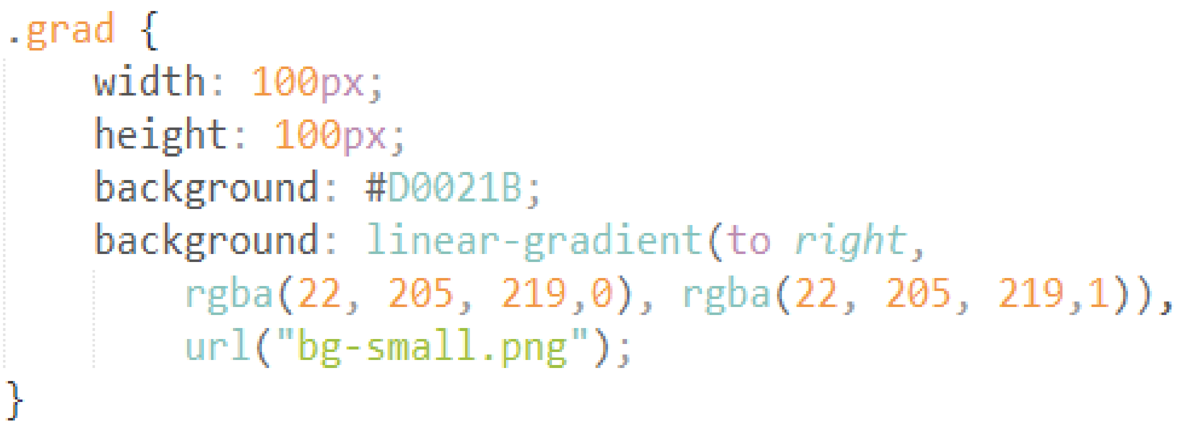


```
.grad {
    width: 100px;
    height: 100px;
    background: #D0021B;
    background: repeating-radial-gradient(
        #D0021B, #ffff00 20%, #16cddb 30%);
}
```

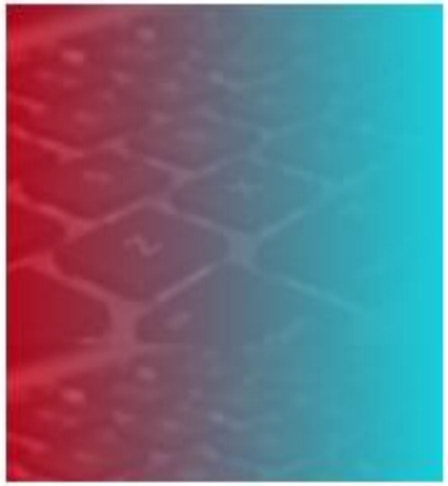


Gradients

- **Transparency with Gradient**



```
.grad {
    width: 100px;
    height: 100px;
    background: #D0021B;
    background: linear-gradient(to right,
        rgba(22, 205, 219,0), rgba(22, 205, 219,1)),
        url("bg-small.png");
}
```

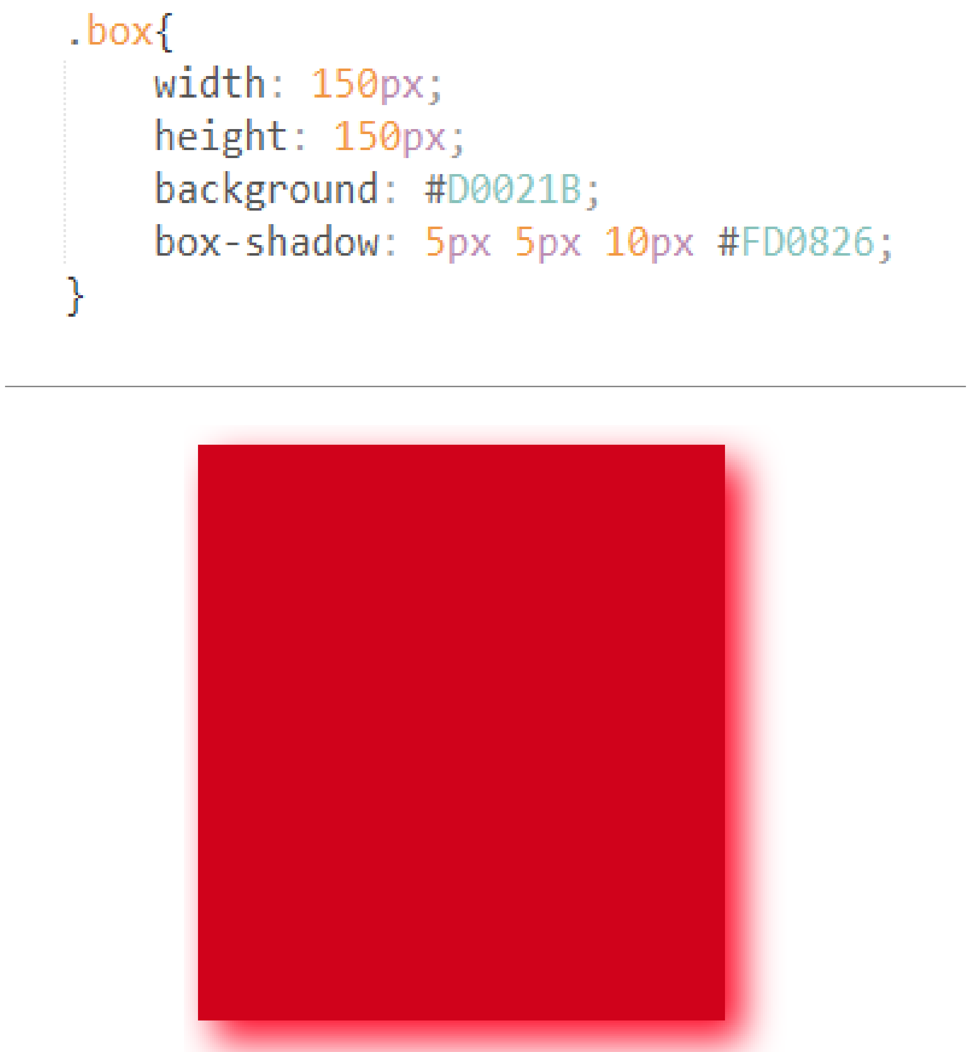


Drop Shadow

CSS3 by Prashant Chaudhari

Drop Shadow

- Allows to apply shadow to elements
- **Properties**
  - **box-shadow**
    - Adds shadow to the element’s boxes
    - Can add one or more shadow effects using comma-separated list of shadows
    - **Syntax**

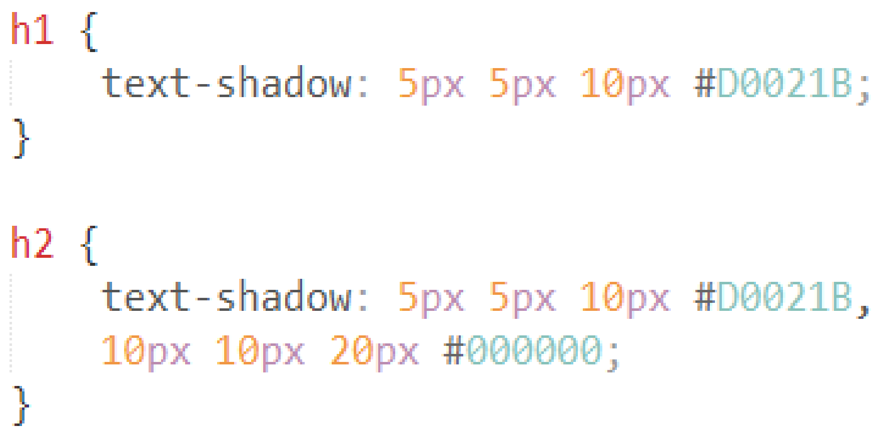


```
.box{
    width: 150px;
    height: 150px;
    background: #D0021B;
    box-shadow: 5px 5px 10px #FD0826;
}
```

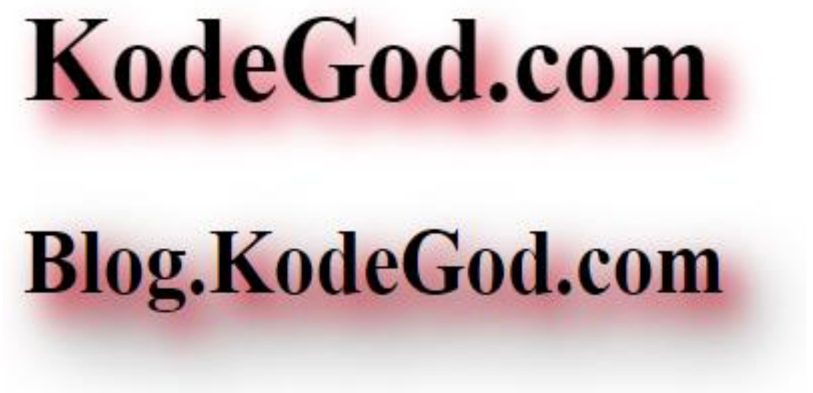
```
box-shadow: offset-x offset-y blur-radius color;
```

Drop Shadow - Properties

- **text-shadow**
  - Apply shadow effects on text



```
h1 {
    text-shadow: 5px 5px 10px #D0021B;
}

h2 {
    text-shadow: 5px 5px 10px #D0021B,
    10px 10px 20px #000000;
}
```



KodeGod.com

Blog.KodeGod.com

2D Transforms

CSS3 by Prashant Chaudhari

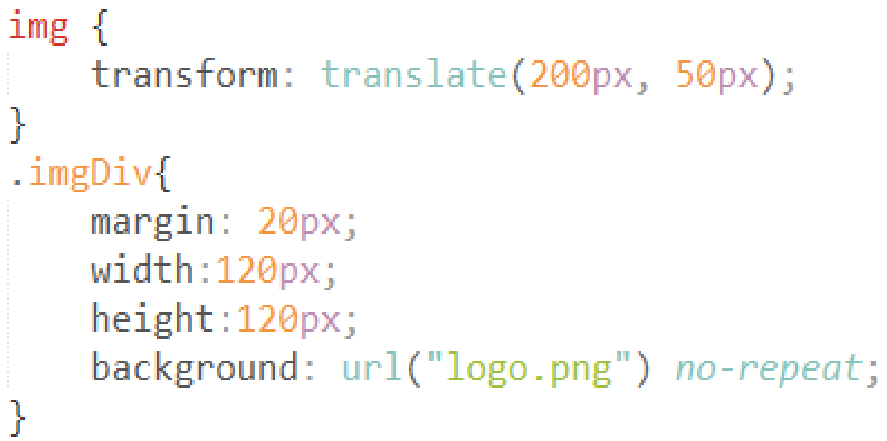
2D Transforms

- Allows elements to be transformed in 2D space
- Can move , rotate, scale, skew elements
- Not affects on surrounding elements but can overlap them
- Manipulates coordinate system used by an element

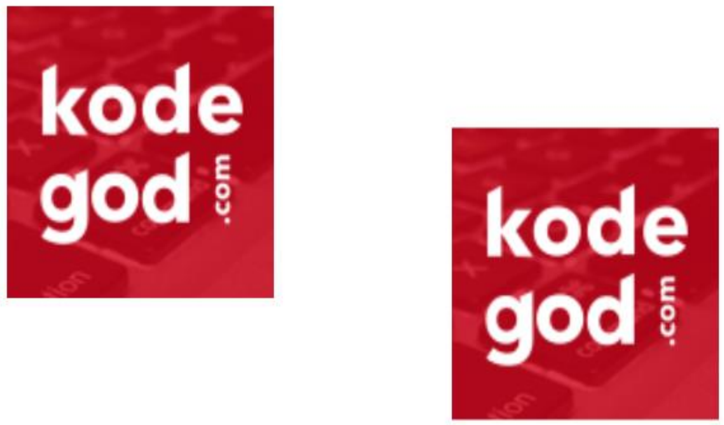
2D Transforms - Functions

- **translate**
  - moves element along x and y axis
  - **Syntax**

    translate(tx, ty)



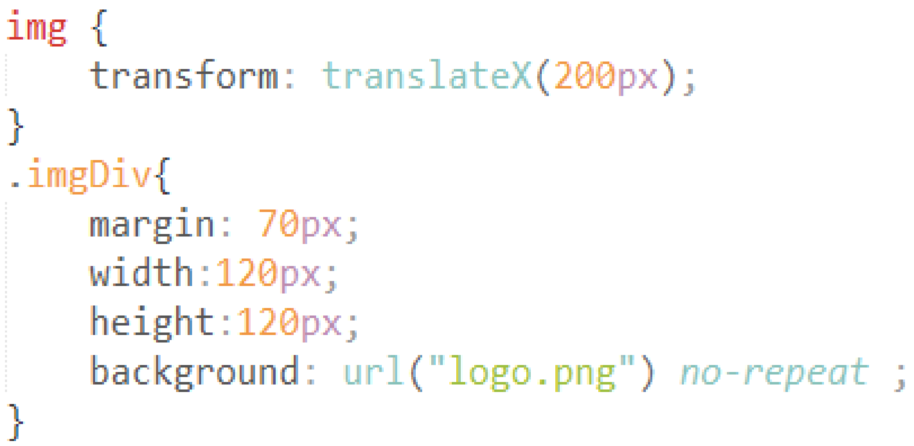
```
img {
    transform: translate(200px, 50px);
}
.imgDiv{
    margin: 20px;
    width:120px;
    height:120px;
    background: url("logo.png") no-repeat;
}
```



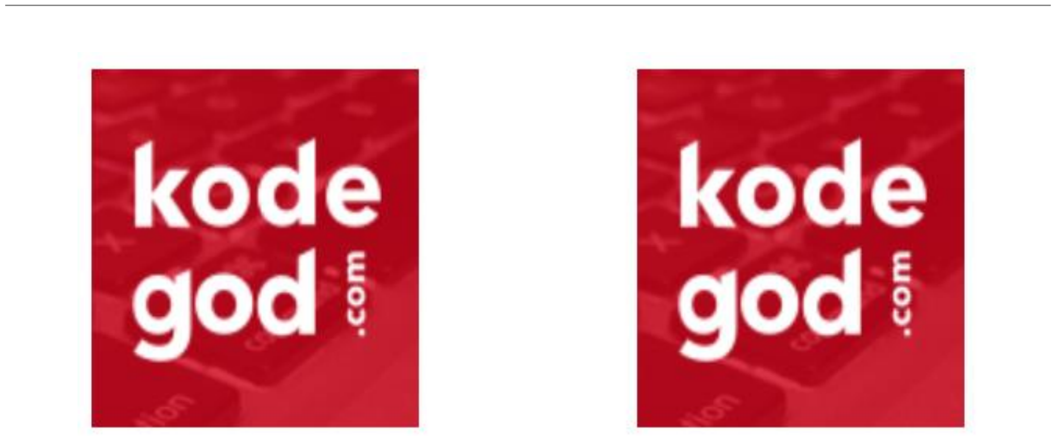
2D Transforms - Functions

- **translate** (x-axis)
  - moves element along x axis
  - **Syntax**

    translateX(tx)



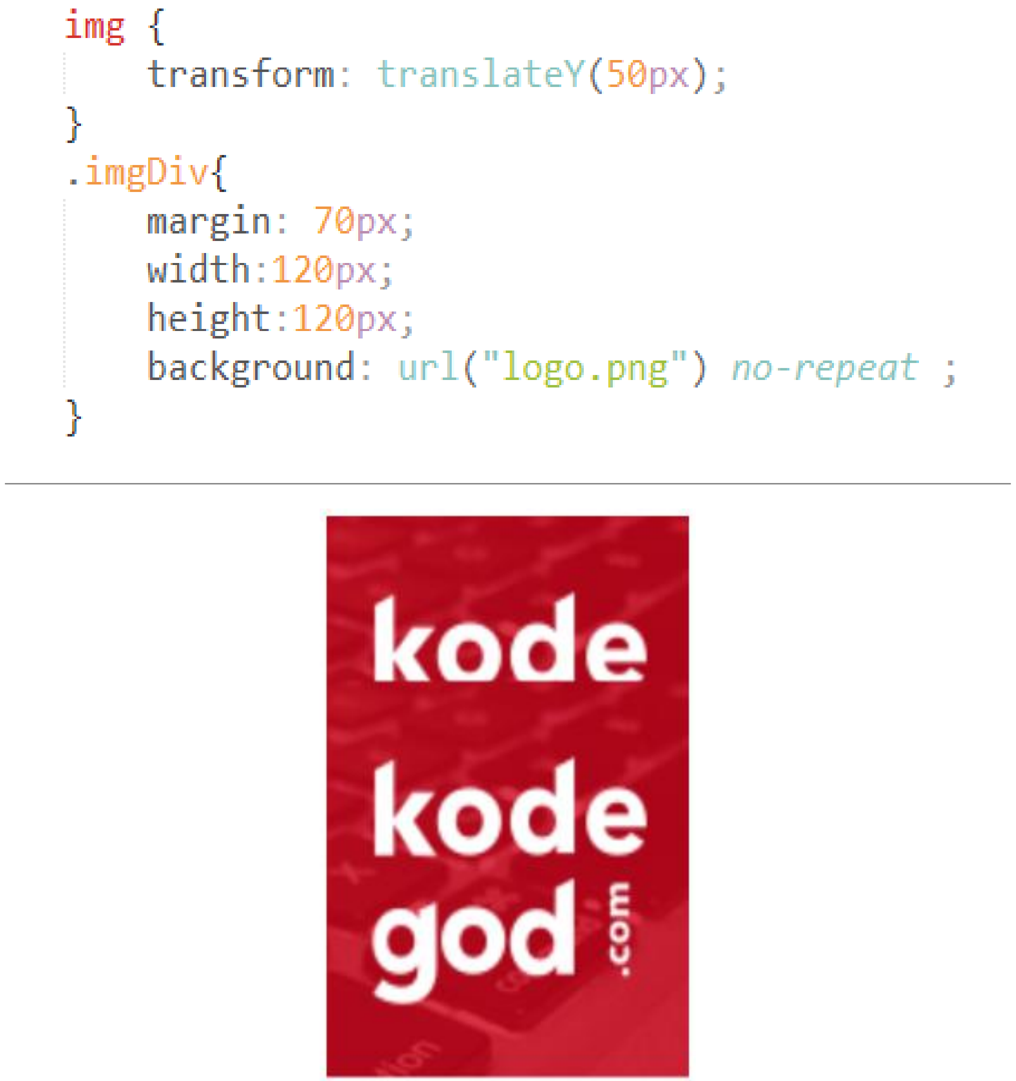
```
img {
    transform: translateX(200px);
}
.imgDiv{
    margin: 70px;
    width:120px;
    height:120px;
    background: url("logo.png") no-repeat ;
}
```



2D Transforms - Functions

- **translate** (y-axis)
  - moves element along y axis
  - **Syntax**

    translateY(ty)

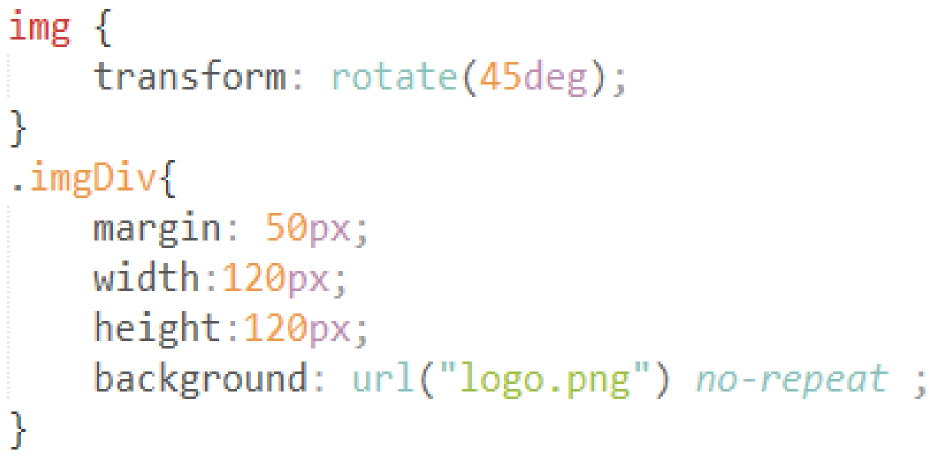


```
img {
    transform: translateY(50px);
}
.imgDiv{
    margin: 70px;
    width:120px;
    height:120px;
    background: url("logo.png") no-repeat ;
}
```

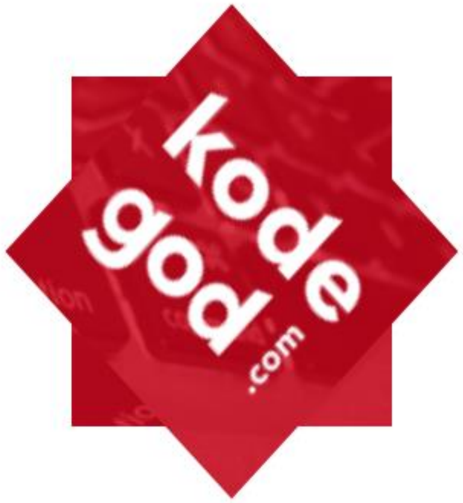
2D Transforms - Functions

- **rotate**
  - rotates element around its origin by specified angle
  - **Syntax**

    rotate(a)



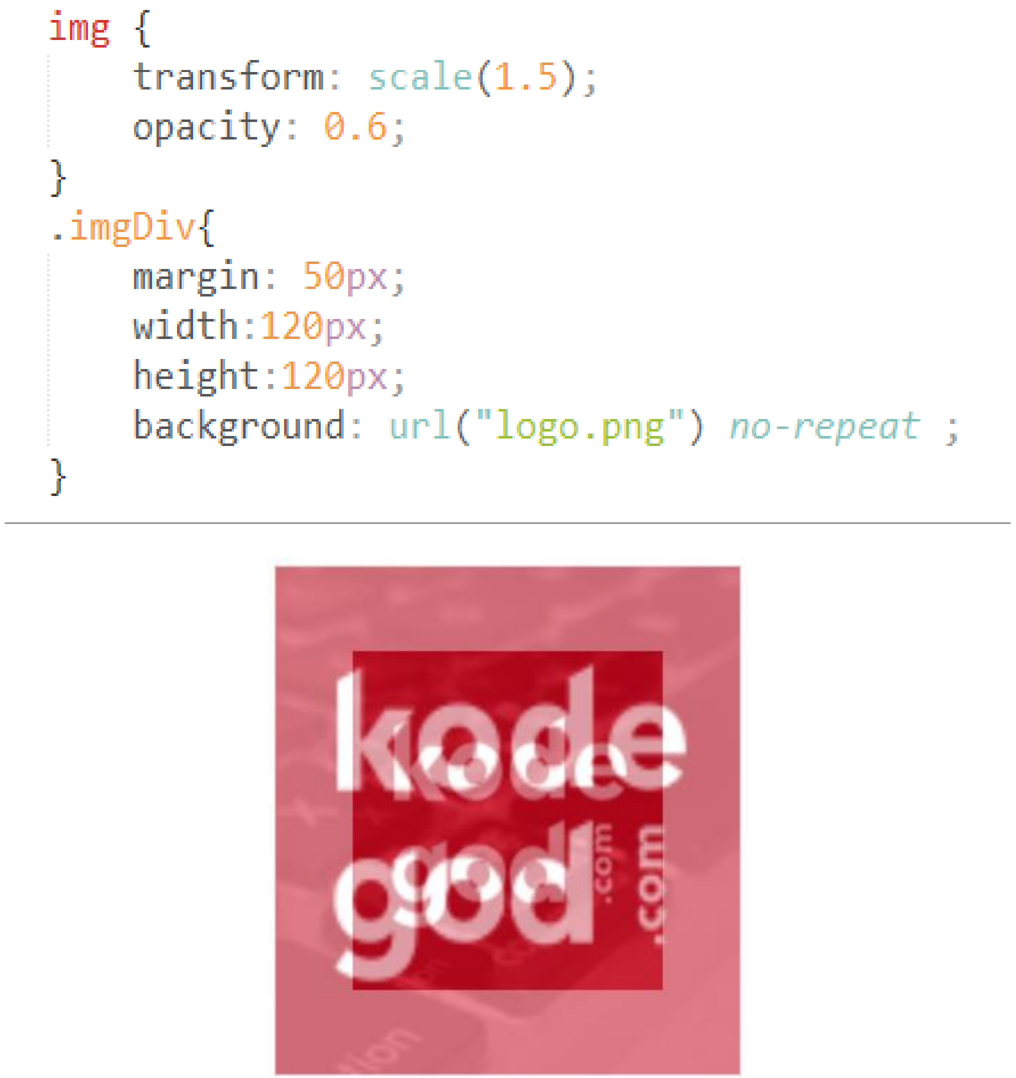
```
img {
    transform: rotate(45deg);
}
.imgDiv{
    margin: 50px;
    width:120px;
    height:120px;
    background: url("logo.png") no-repeat ;
}
```



2D Transforms - Functions

- **scale**
  - Increases or decreases size of an element
  - **Syntax**

    scale(size)

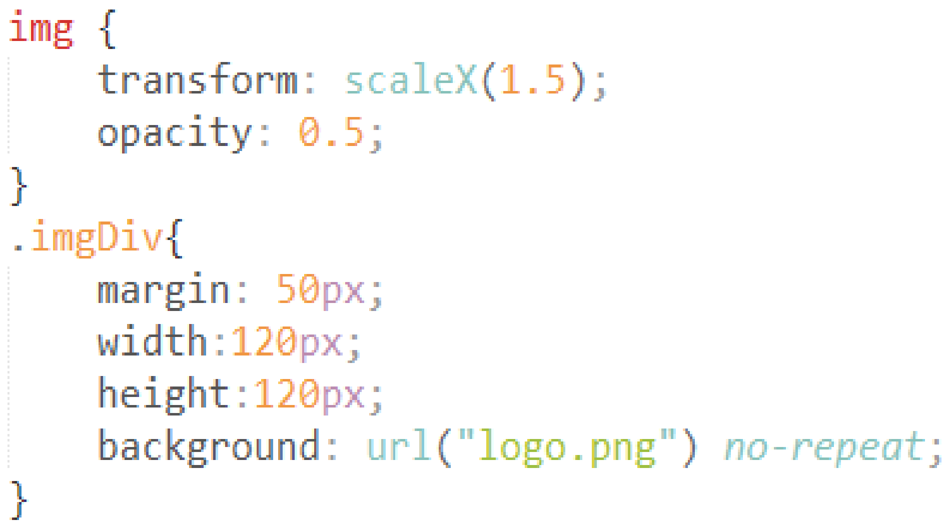


```
img {
    transform: scale(1.5);
    opacity: 0.6;
}
.imgDiv{
    margin: 50px;
    width:120px;
    height:120px;
    background: url("logo.png") no-repeat ;
}
```

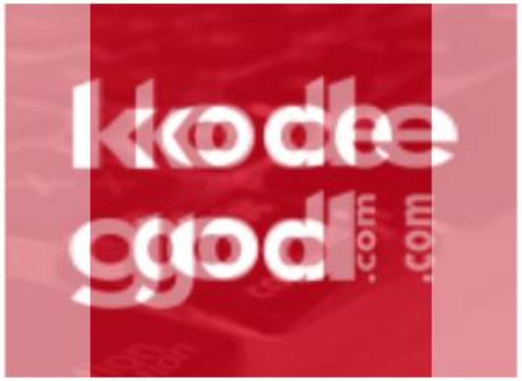
2D Transforms - Functions

- **scale** (X-axis)
    - Scales width of an element by given amount
    - **Syntax**

      scaleX(sx)



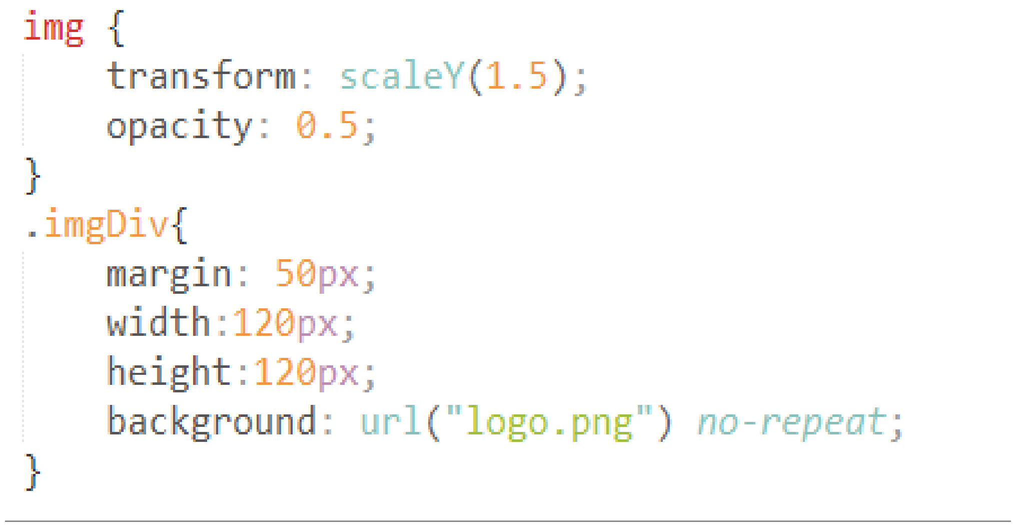
```
img {
    transform: scaleX(1.5);
    opacity: 0.5;
}
.imgDiv{
    margin: 50px;
    width:120px;
    height:120px;
    background: url("logo.png") no-repeat;
}
```



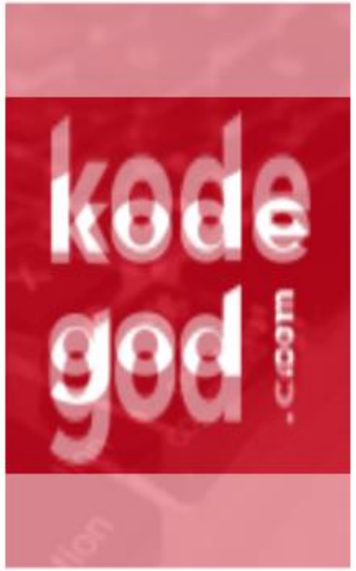
2D Transforms - Functions

- **scale** (Y-axis)
    - Scales height of an element by given amount
    - **Syntax**

      scaleY(sy)



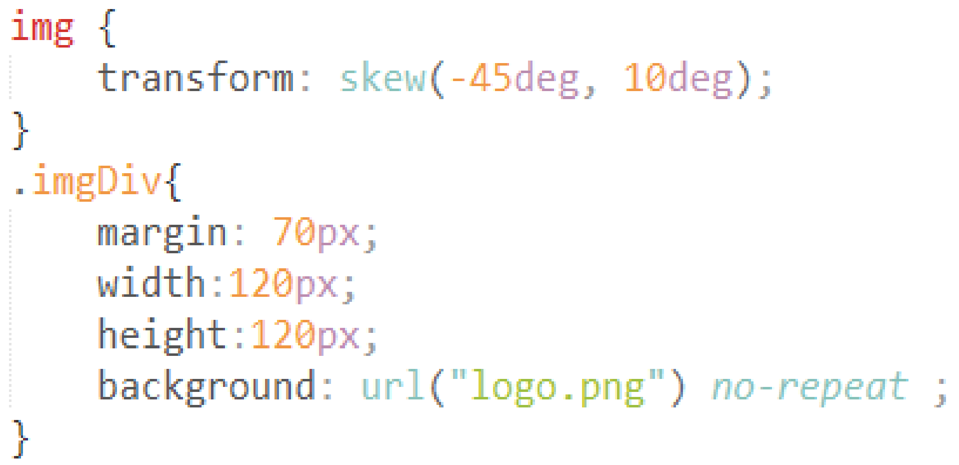
```
img {
    transform: scaleY(1.5);
    opacity: 0.5;
}
.imgDiv{
    margin: 50px;
    width:120px;
    height:120px;
    background: url("logo.png") no-repeat;
}
```



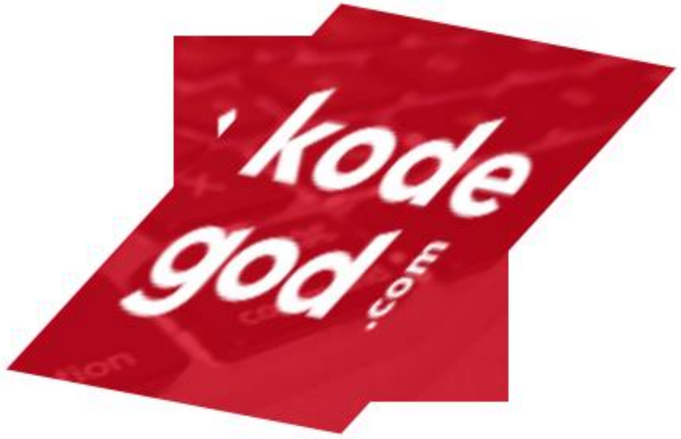
2D Transforms - Functions

- **skew**
  - Skews element along x and y axes by specified angle
  - **Syntax**

    skew(ax, ay)



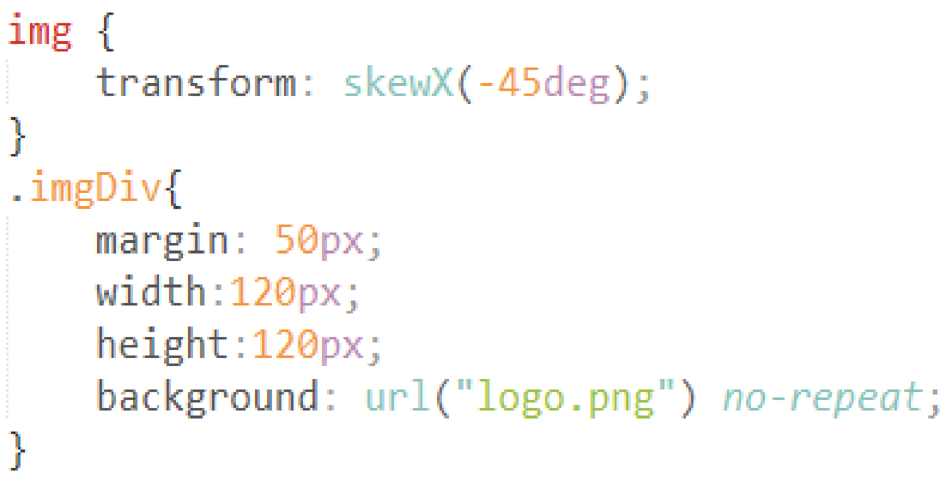
```
img {
    transform: skew(-45deg, 10deg);
}
.imgDiv{
    margin: 70px;
    width:120px;
    height:120px;
    background: url("logo.png") no-repeat ;
}
```



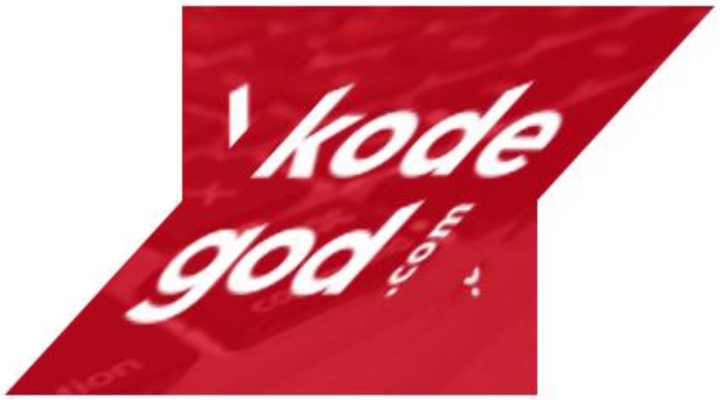
2D Transforms - Functions

- **skew** (X-axis)
  - Skews element along x axis by specified angle
  - **Syntax**

    skewX(ax)



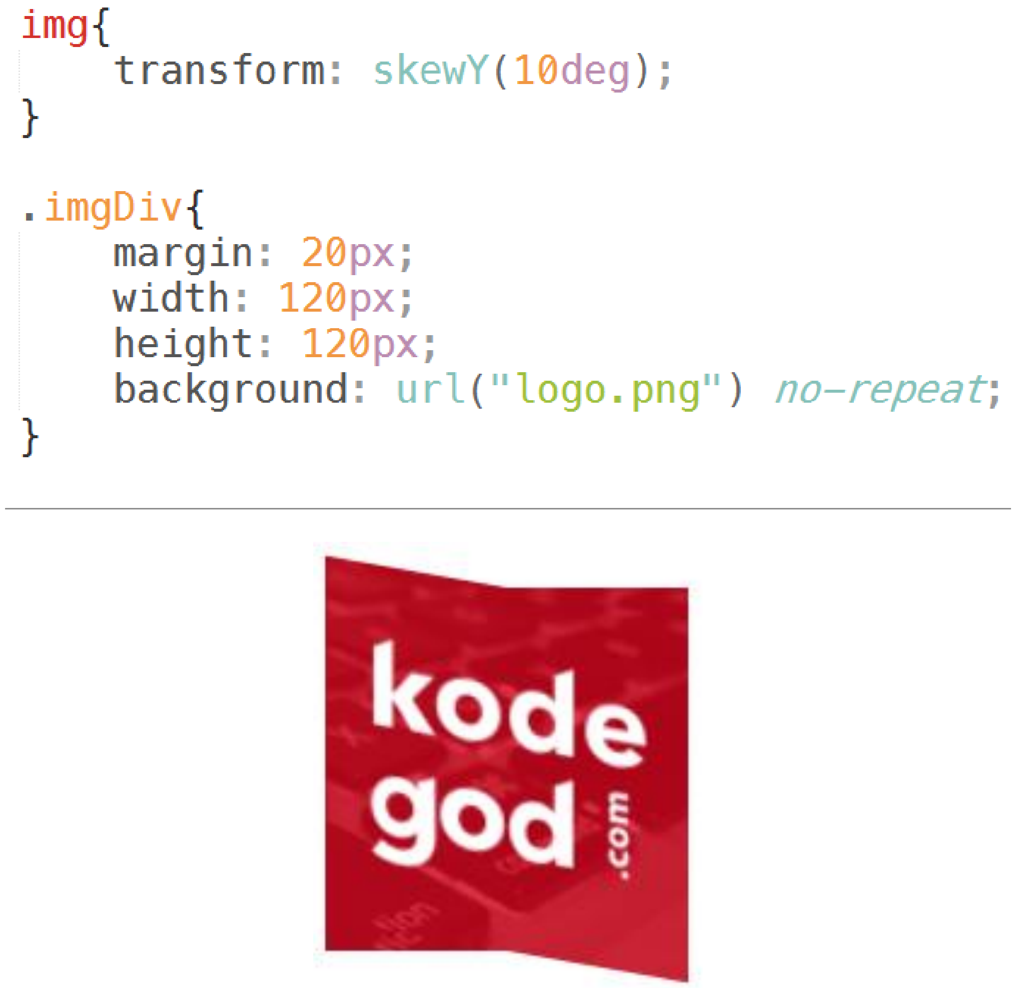
```
img {
    transform: skewX(-45deg);
}
.imgDiv{
    margin: 50px;
    width:120px;
    height:120px;
    background: url("logo.png") no-repeat;
}
```



2D Transforms - Functions

- **skew** (Y-axis)
  - Skews element along y axis by specified angle
  - **Syntax**

    skewY(ay)



```
img{
    transform: skewY(10deg);
}

.imgDiv{
    margin: 20px;
    width: 120px;
    height: 120px;
    background: url("logo.png") no-repeat;
}
```

2D Transforms - Functions

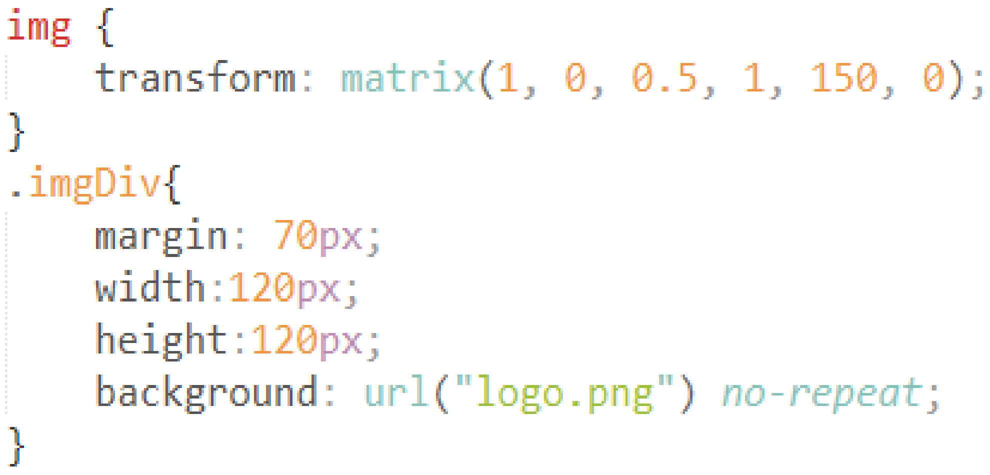
- **matrix**
  - Performs all 2D transformations at once
  - Takes six parameters in form of matrix
  - **Syntax**

    matrix(a, b, c, d, e, f)

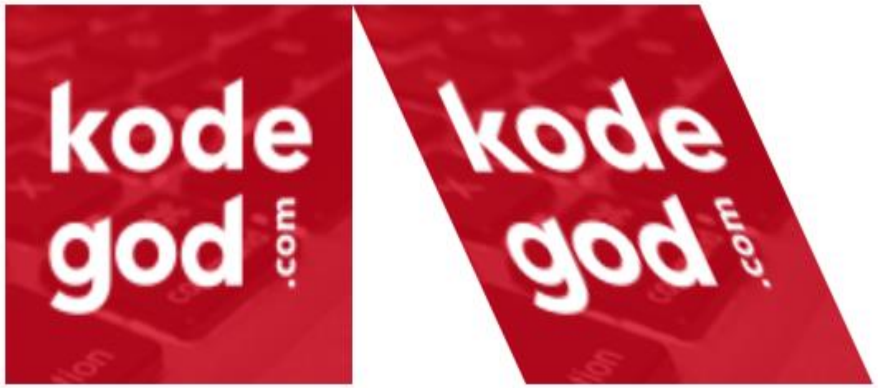
- **translate(tx, ty) = matrix(1, 0, 0, 1, tx, ty)**, where tx and ty are horizontal and vertical translation values
- **rotate(a) = matrix(cos(v), sin(v), -sin(v), cos(v), 0, 0),** where **v** is a value in degree. Swap sin(v) and –sin(v) values to reverse the rotation

2D Transforms - Functions

- **scale(sx, sy) = matrix(sx, 0, 0, sy, 0, 0)**, where sx and sy are horizontal and vertical scaling values
- **skew(ax, ay) = matrix(1, tan(ax), tan(ay), 1, 0, 0),** where **ax** and **ay** are horizontal and vertical values in degree

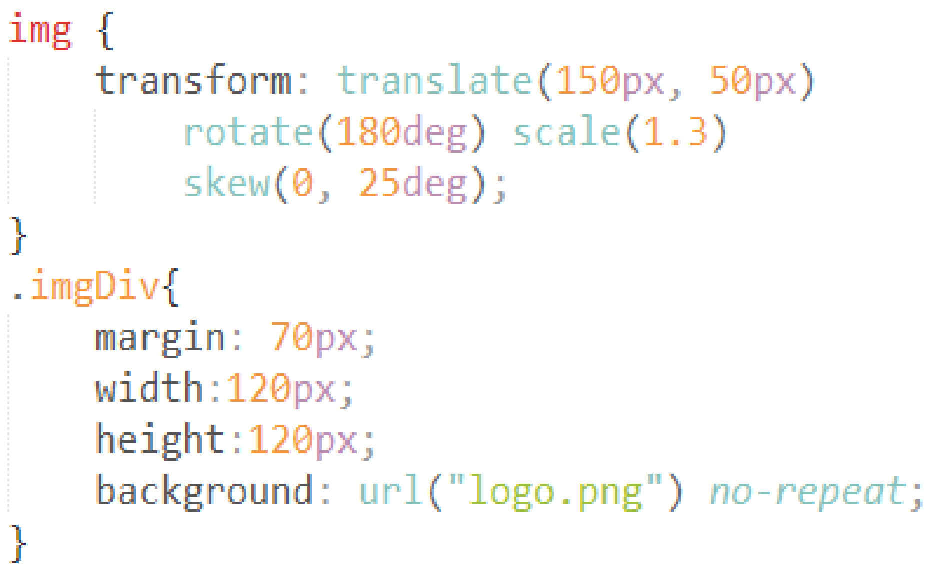


```
img {
    transform: matrix(1, 0, 0.5, 1, 150, 0);
}
.imgDiv{
    margin: 70px;
    width:120px;
    height:120px;
    background: url("logo.png") no-repeat;
}
```

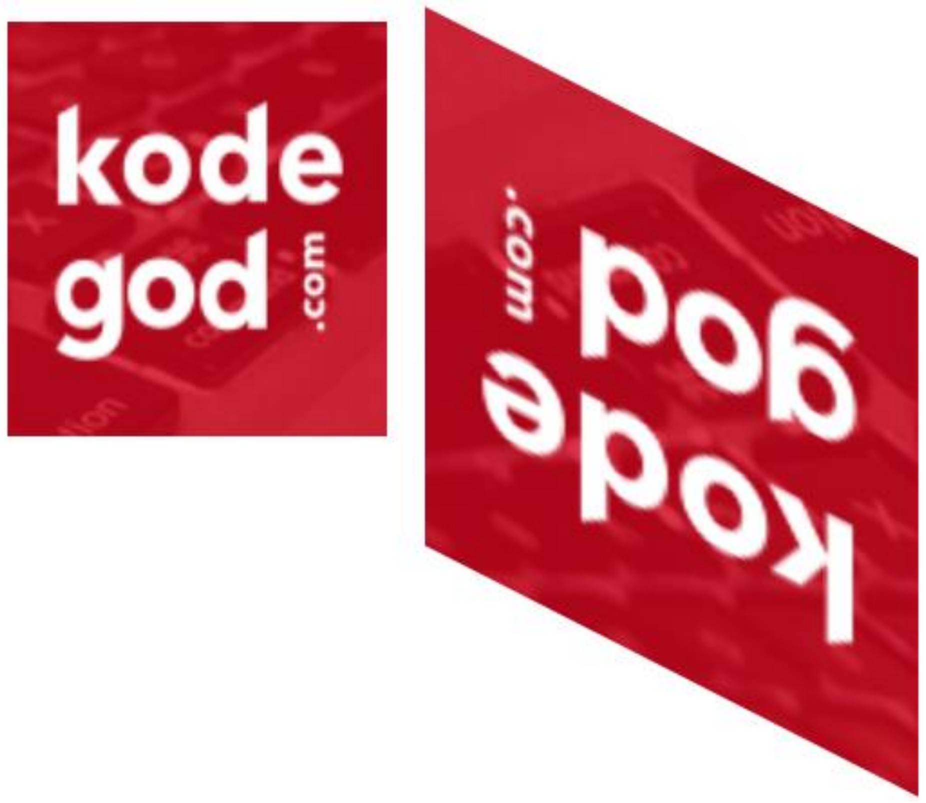


2D Transforms - Functions

- Performing more than one transformation at once



```
img {
    transform: translate(150px, 50px)
        rotate(180deg) scale(1.3)
        skew(0, 25deg);
}
.imgDiv{
    margin: 70px;
    width:120px;
    height:120px;
    background: url("logo.png") no-repeat;
}
```



3D Transforms

CSS3 by Prashant Chaudhari

3D Transforms

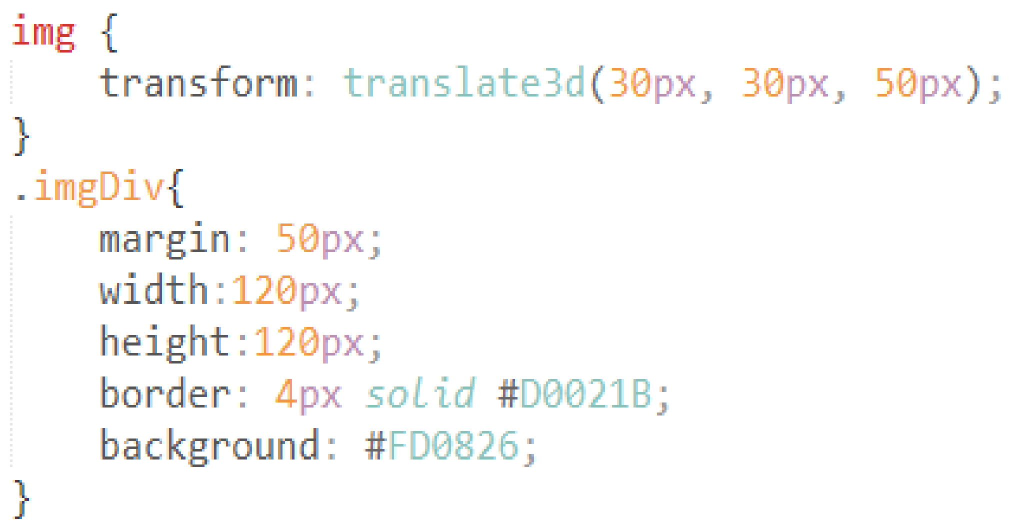
- Allows elements to be transformed in 3D space
- Can move, rotate, scale, skew elements
- Not affects on surrounding elements but can overlap them
- Manipulates coordinate system used by an element

- **Imp**
  - **perspective(length)**
    - Defines perspective view for 3D transformed element
    - Value of this function increases, element will appear further away from viewer

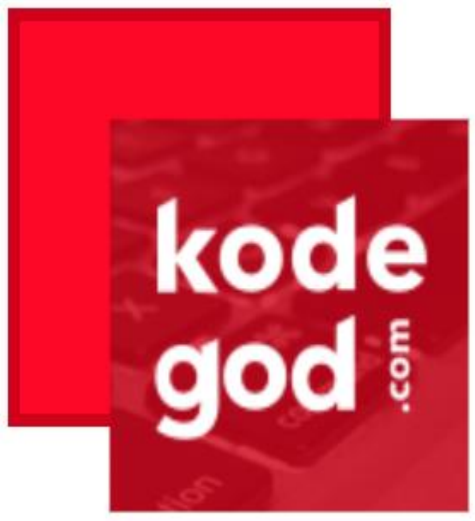
3D Transforms - Functions

- **translate3d**
  - moves element along x, y and z axis
  - **Syntax**

    translate3d(tx, ty, tz)



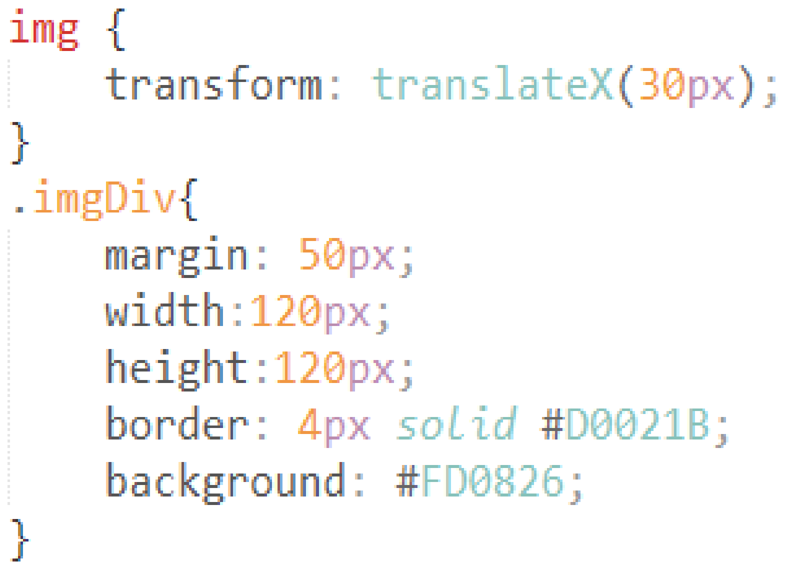
```
img {
    transform: translate3d(30px, 30px, 50px);
}
.imgDiv{
    margin: 50px;
    width:120px;
    height:120px;
    border: 4px solid #D0021B;
    background: #FD0826;
}
```



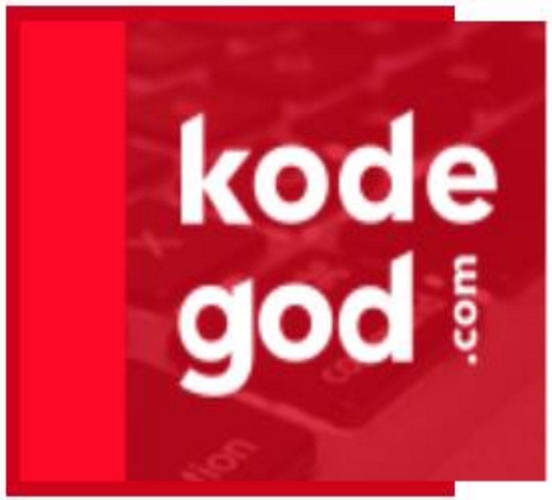
3D Transforms - Functions

- **translateX** (x-axis)
    - moves element along x axis
    - **Syntax**

      translateX(tx)



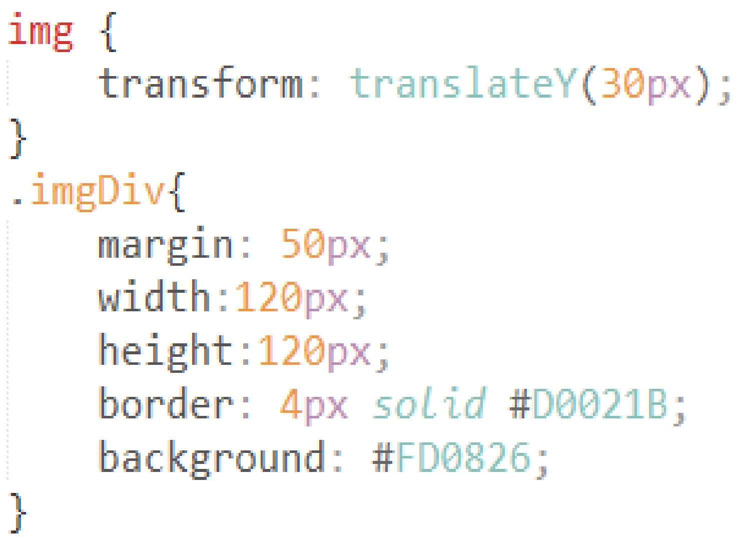
```
img {
    transform: translateX(30px);
}
.imgDiv{
    margin: 50px;
    width:120px;
    height:120px;
    border: 4px solid #D0021B;
    background: #FD0826;
}
```



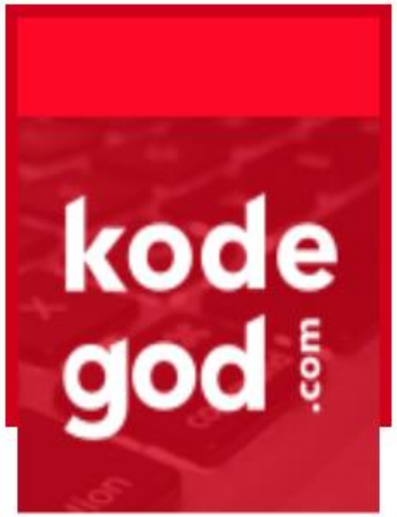
3D Transforms - Functions

- **translateY** (y-axis)
  - moves element along y axis
  - **Syntax**

    translateY(ty)



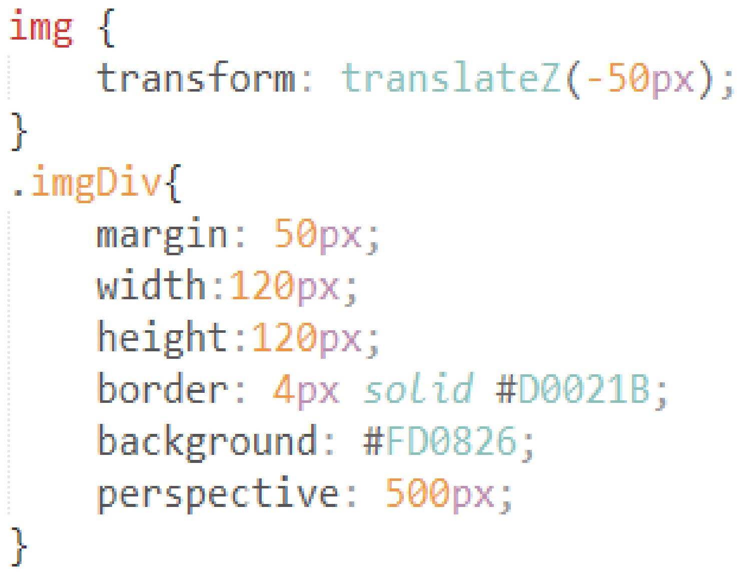
```
img {
    transform: translateY(30px);
}
.imgDiv{
    margin: 50px;
    width:120px;
    height:120px;
    border: 4px solid #D0021B;
    background: #FD0826;
}
```



3D Transforms - Functions

- **translateZ** (z-axis)
  - moves element along z axis
  - **Syntax**

    translateZ(tz)



```
img {
    transform: translateZ(-50px);
}
.imgDiv{
    margin: 50px;
    width:120px;
    height:120px;
    border: 4px solid #D0021B;
    background: #FD0826;
    perspective: 500px;
}
```

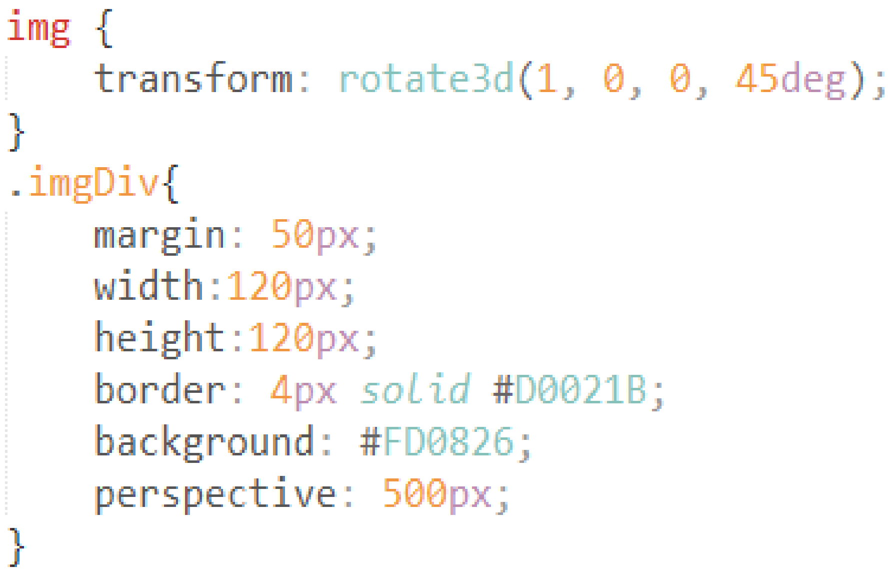
kode
god
.com

3D Transforms - Functions

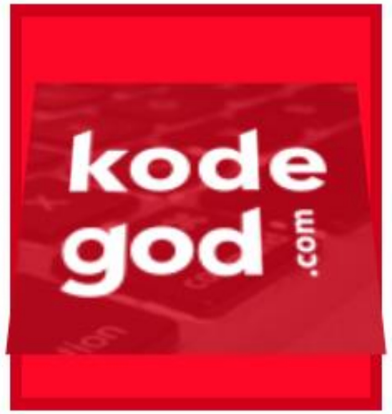
- **rotate3d**
  - rotates element in 3D space by specified

    angle around [x, y, z] direction vector
  - **Syntax**

    rotate3d(x, y, z, a)



```css
img {
    transform: rotate3d(1, 0, 0, 45deg);
}
.imgDiv{
    margin: 50px;
    width:120px;
    height:120px;
    border: 4px solid #D0021B;
    background: #FD0826;
    perspective: 500px;
}
```

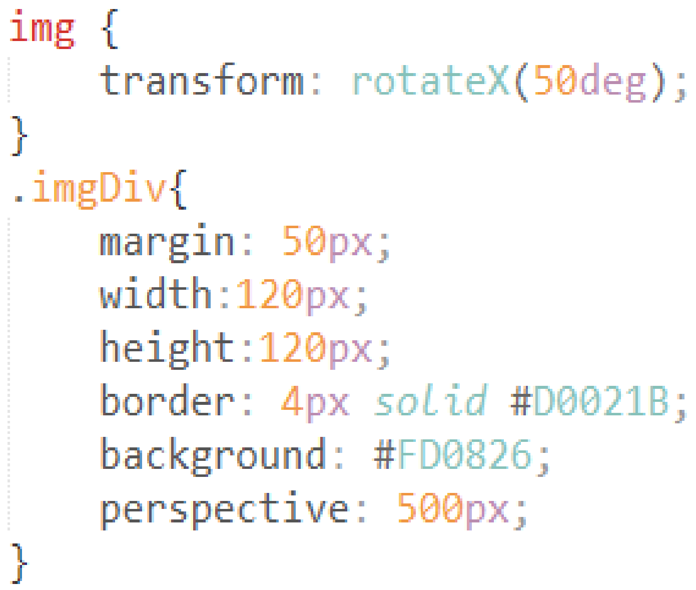


3D Transforms - Functions

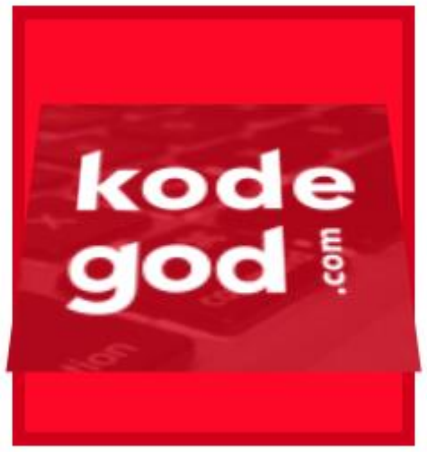
- **rotateX (X-axis)**
  - rotates element by specified angle

    around x-axis
  - **Syntax**

    rotateX(a)



```
img {
    transform: rotateX(50deg);
}
.imgDiv{
    margin: 50px;
    width:120px;
    height:120px;
    border: 4px solid #D0021B;
    background: #FD0826;
    perspective: 500px;
}
```

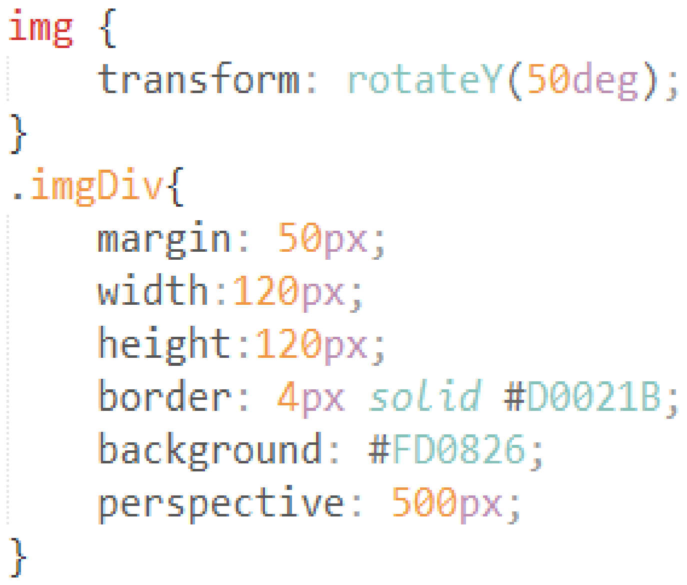


3D Transforms - Functions

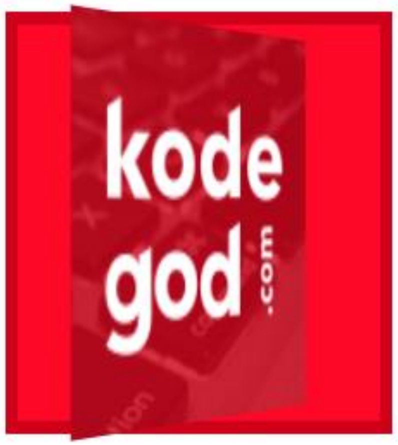
- **rotateY (Y-axis)**
  - rotates element by specified angle

    around y-axis
  - **Syntax**

    rotateY(a)



```
img {
    transform: rotateY(50deg);
}
.imgDiv{
    margin: 50px;
    width:120px;
    height:120px;
    border: 4px solid #D0021B;
    background: #FD0826;
    perspective: 500px;
}
```

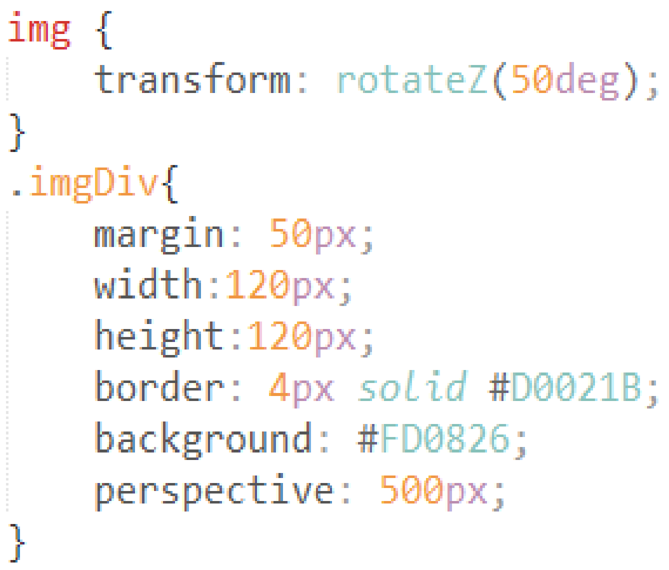


3D Transforms - Functions

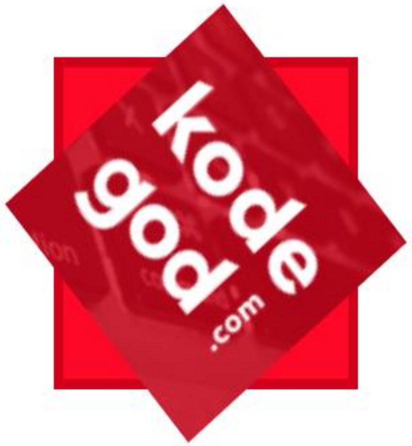
- **rotateZ (Z-axis)**
  - rotates element by specified angle

    around z-axis
  - **Syntax**

    rotateZ(a)



```
img {
    transform: rotateZ(50deg);
}
.imgDiv{
    margin: 50px;
    width:120px;
    height:120px;
    border: 4px solid #D0021B;
    background: #FD0826;
    perspective: 500px;
}
```

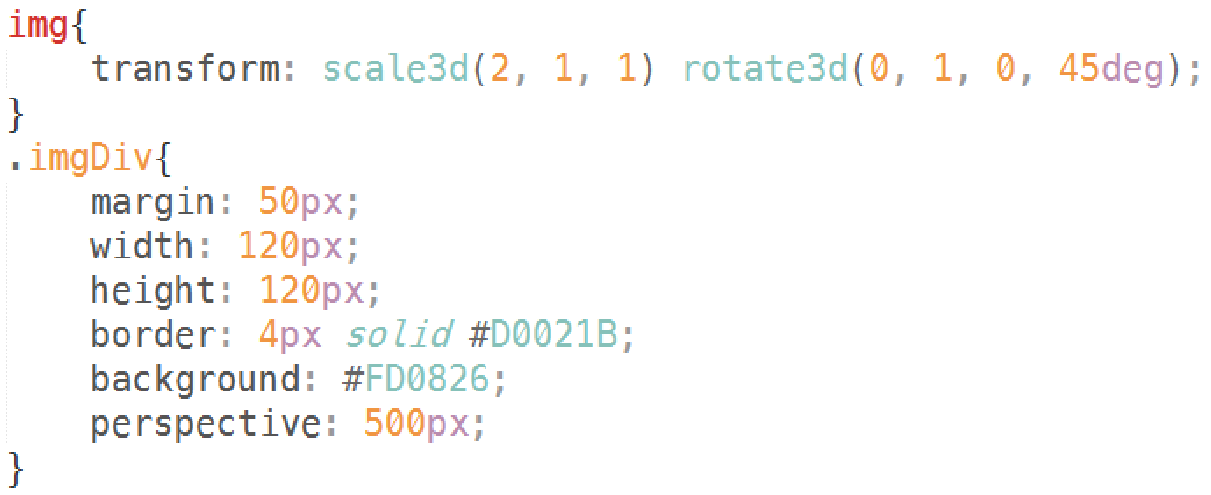


3D Transforms - Functions

- **scale3d**
  - Changes size of an element
  - **Syntax**

    scale3d(sx, sy, sz)
  - Must used with combination of rotate

    and perspective function



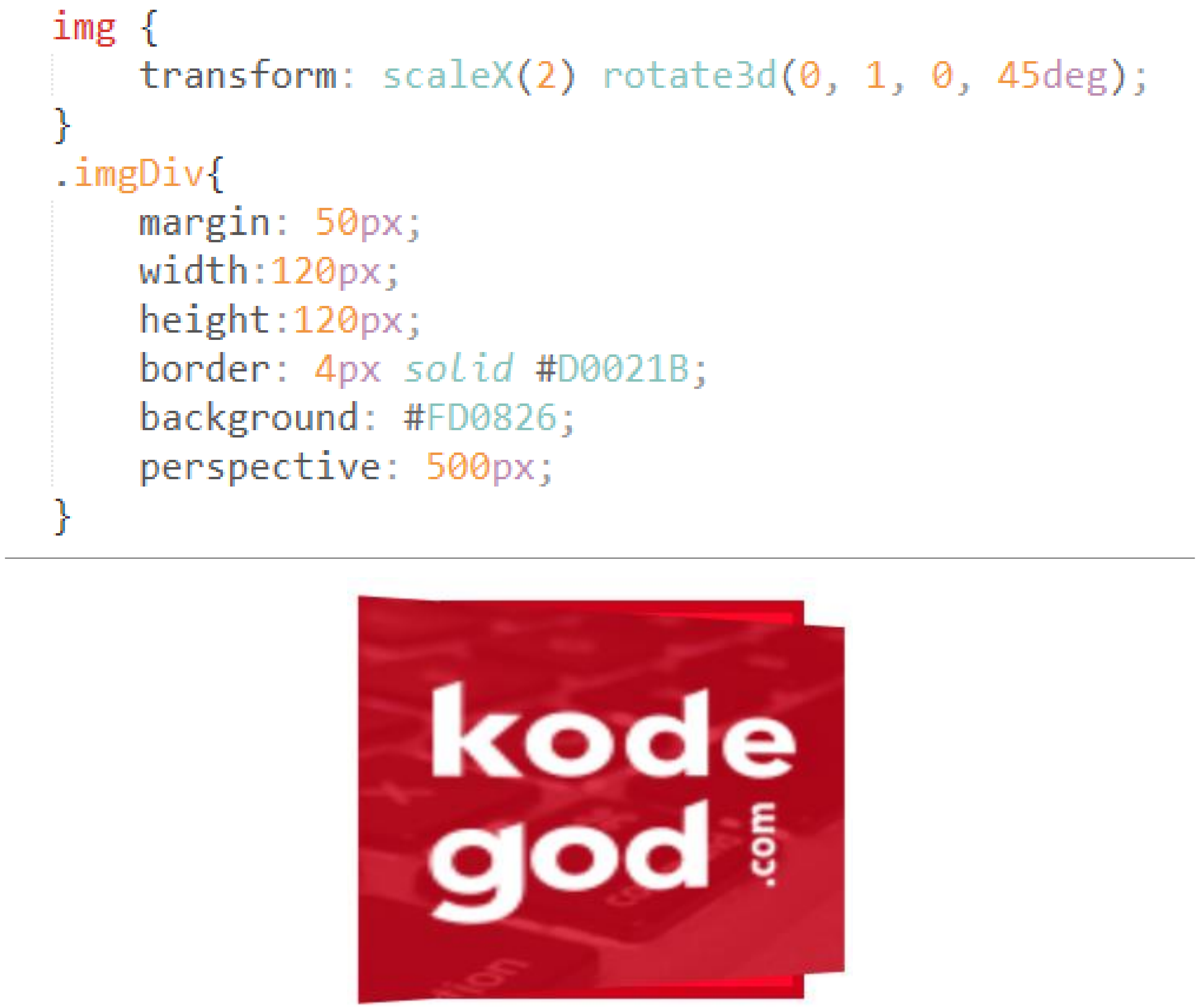
```
img{
    transform: scale3d(2, 1, 1) rotate3d(0, 1, 0, 45deg);
}
.imgDiv{
    margin: 50px;
    width: 120px;
    height: 120px;
    border: 4px solid #D0021B;
    background: #FD0826;
    perspective: 500px;
}
```

kode
god
.com

3D Transforms - Functions

- **scaleX** (X-axis)
  - Scales element along x-axis
  - **Syntax**

    scaleX(sx)

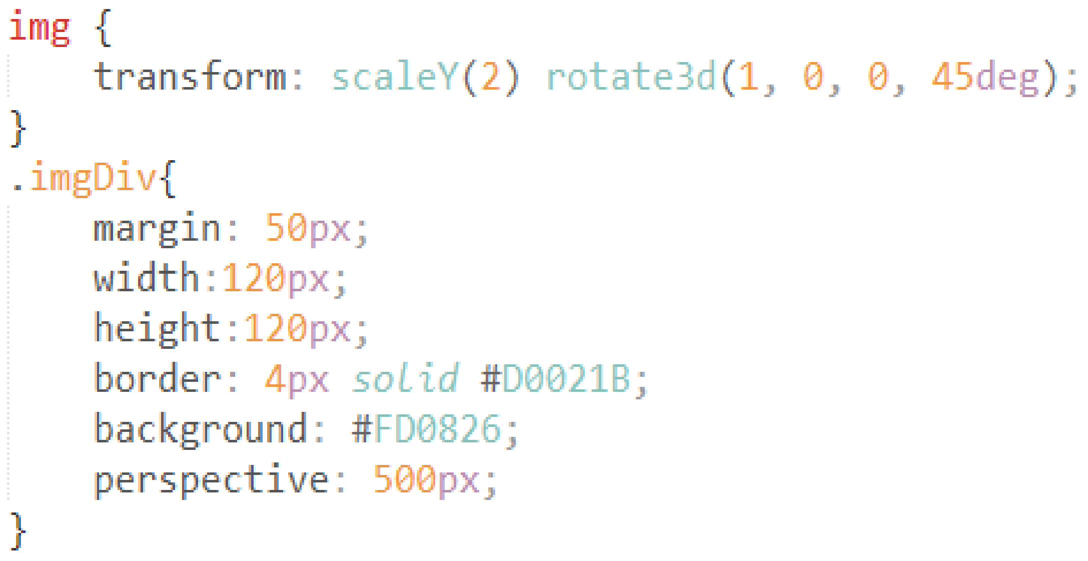


```
img {
    transform: scaleX(2) rotate3d(0, 1, 0, 45deg);
}
.imgDiv{
    margin: 50px;
    width:120px;
    height:120px;
    border: 4px solid #D0021B;
    background: #FD0826;
    perspective: 500px;
}
```

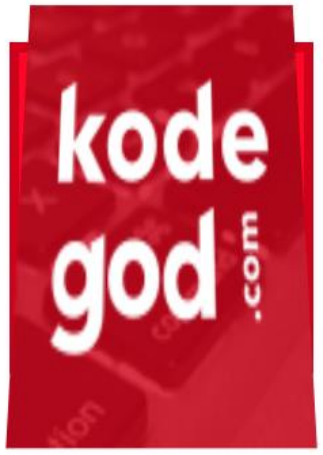
3D Transforms - Functions

- **scaleY** (Y-axis)
  - Scales element along y-axis
  - **Syntax**

    scaleY(sy)



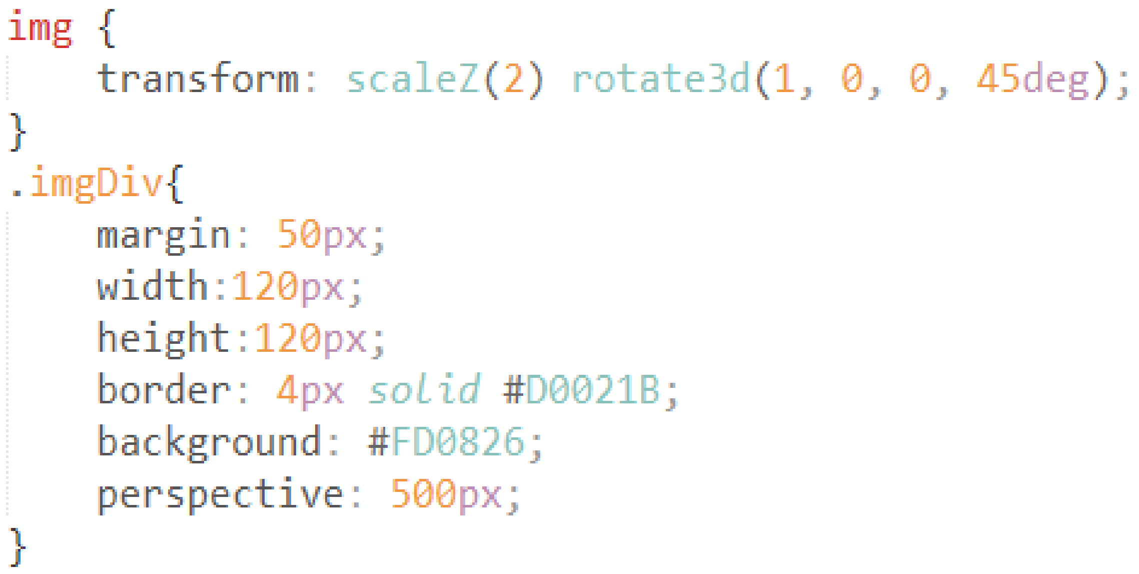
```
img {
    transform: scaleY(2) rotate3d(1, 0, 0, 45deg);
}
.imgDiv{
    margin: 50px;
    width:120px;
    height:120px;
    border: 4px solid #D0021B;
    background: #FD0826;
    perspective: 500px;
}
```



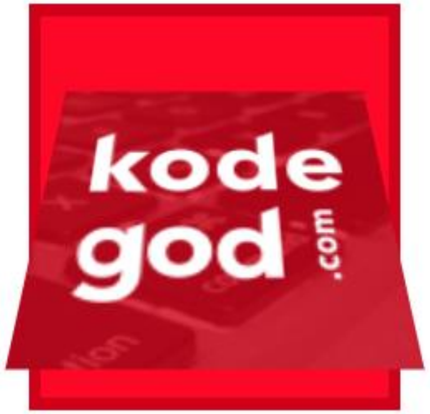
3D Transforms - Functions

- **scaleZ** (Z-axis)
  - Scales element along z-axis
  - **Syntax**

    scaleZ(sz)

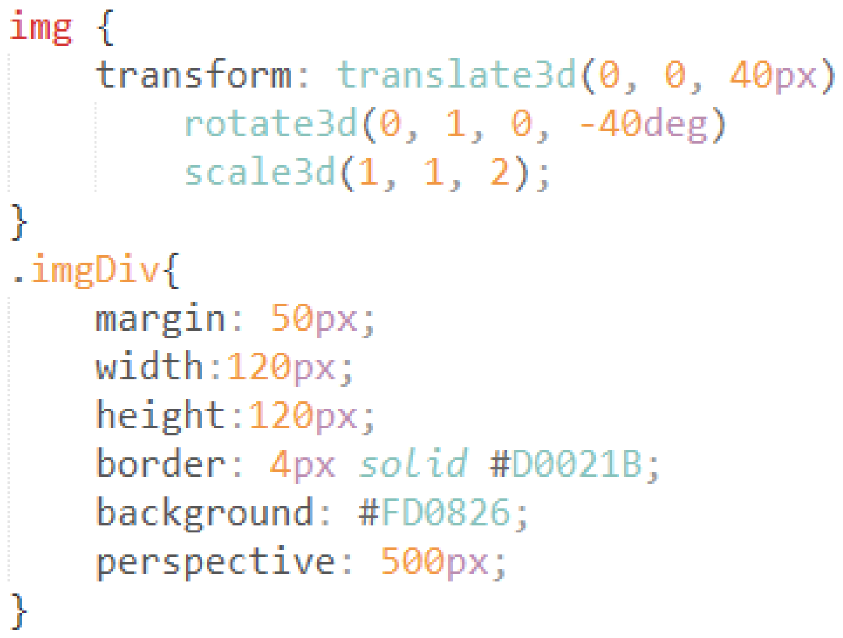


```
img {
    transform: scaleZ(2) rotate3d(1, 0, 0, 45deg);
}
.imgDiv{
    margin: 50px;
    width:120px;
    height:120px;
    border: 4px solid #D0021B;
    background: #FD0826;
    perspective: 500px;
}
```

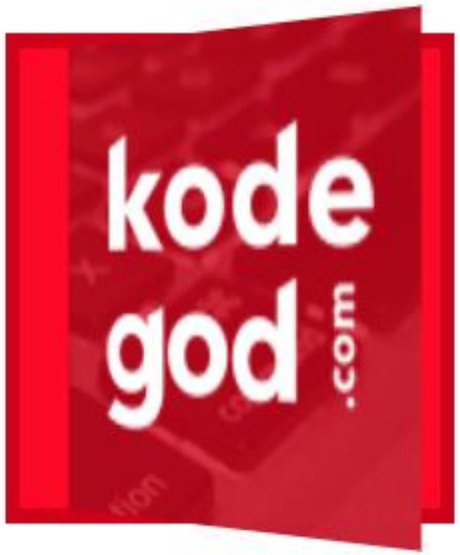


3D Transforms - Functions

- To perform more than one transformation at once, use individual transformation and list them in order



```
img {
    transform: translate3d(0, 0, 40px)
        rotate3d(0, 1, 0, -40deg)
        scale3d(1, 1, 2);
}
.imgDiv{
    margin: 50px;
    width:120px;
    height:120px;
    border: 4px solid #D0021B;
    background: #FD0826;
    perspective: 500px;
}
```



Transitions

CSS3 by Prashant Chaudhari

Transitions

- Allows to animate property from old value to new value smoothly over time
- **Properties**

  transition, transition-delay, transition-property, transition-duration, transition-timing-function

- **transition-property**
  - Specifies name of css properties to which transition effect applied
  - **Values** border, background, etc.

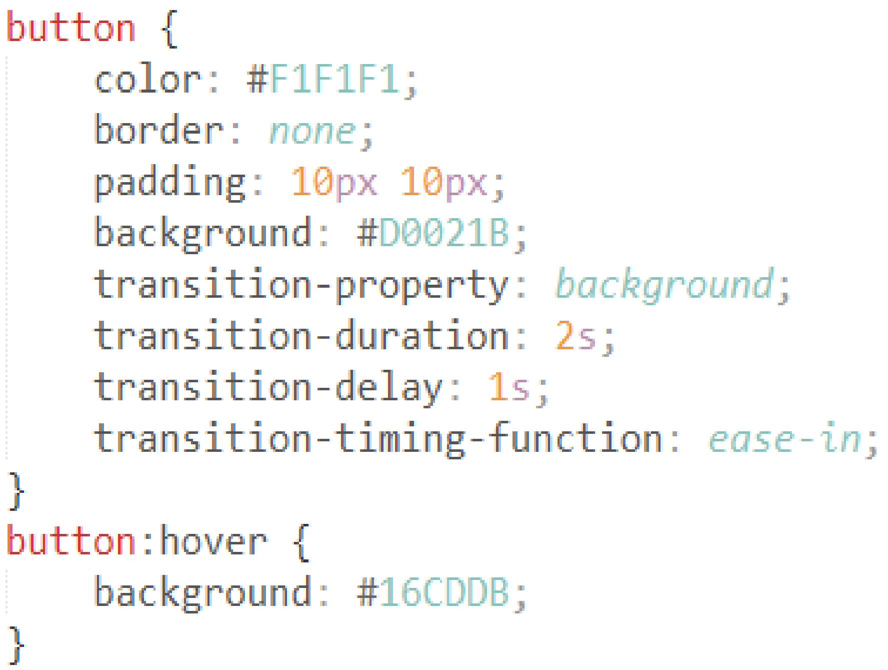
Transitions

- **transition-duration**
  - Specifies number of seconds or milliseconds transition operation takes to complete
- **transition-delay**
  - Specifies when transition will start

- **transition-timing-function**
  - Specifies how values of css properties affected by transition
  - **Possible values** cubic-bezier, ease, ease-in, ease-in-out, ease-out, frames, linear, steps, step-end, step-middle, step-start

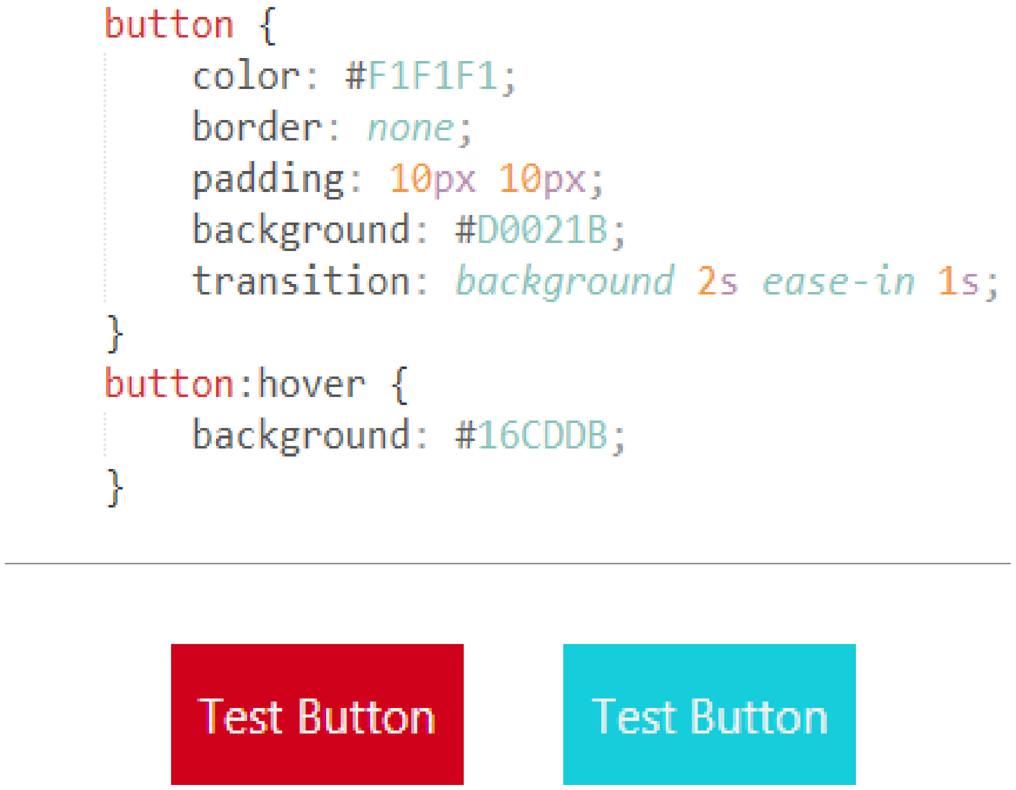
Transitions

- E.g. (Using individual property)



```
button {
    color: #F1F1F1;
    border: none;
    padding: 10px 10px;
    background: #D0021B;
    transition-property: background;
    transition-duration: 2s;
    transition-delay: 1s;
    transition-timing-function: ease-in;
}
button:hover {
    background: #16CDDB;
}
```

- E.g. (Shorthand property)



```
button {
    color: #F1F1F1;
    border: none;
    padding: 10px 10px;
    background: #D0021B;
    transition: background 2s ease-in 1s;
}
button:hover {
    background: #16CDDB;
}
```

Animations

CSS3 by Prashant Chaudhari

Animations

- Allows to specify changes in css properties over time as a set of keyframes
- **Creating animation is two step process**
  1. Defining individual keyframes and naming an animation with keyframe declaration
  2. Referencing keyframe by name using **animation-name** property

Animation - Properties

- **animation-name**

  specifies name of keyframes defined animations

- **animation-duration**

  time to complete one animation cycle (in s or ms)

- **animation-timing-function**

  specifies how animation will progress over each cycle

- **animation-iteration-count**

  number of animation cycles

- **animation-direction**

  specifies whether or not animation should play in reverse

- **animation-fill-mode**

  specifies how css animation should apply styles to its target before and after execution

Animation - Properties

- **animation-play-state**

  animation running or paused

- **@keyframes**

  value of animating properties at various points

  during animation

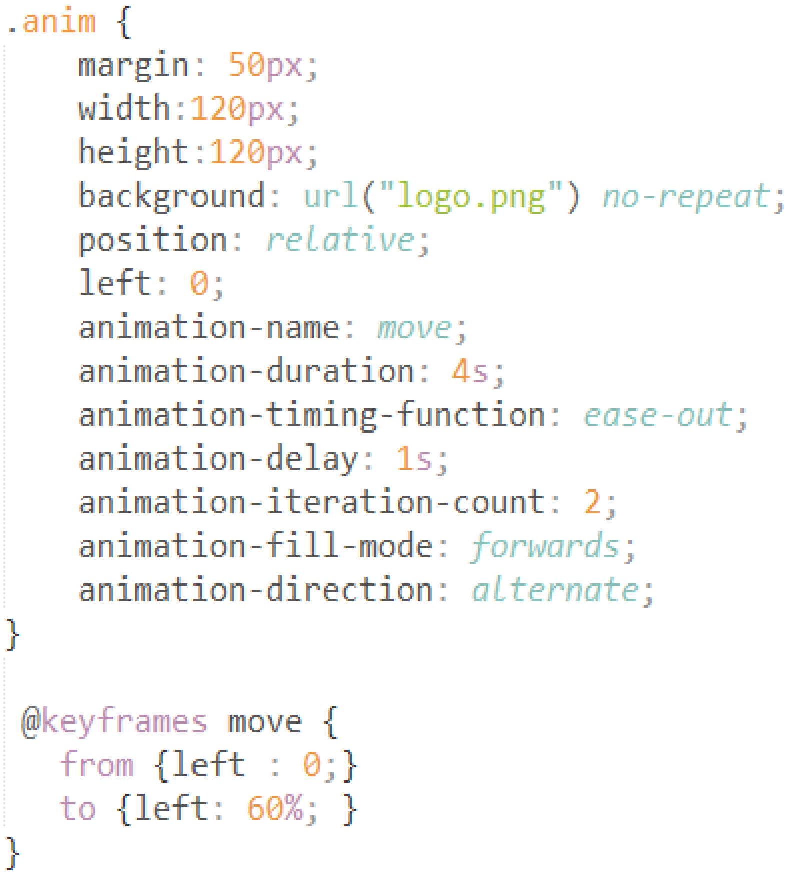
- **animation**

  shorthand property to specify all properties in

  single declaration

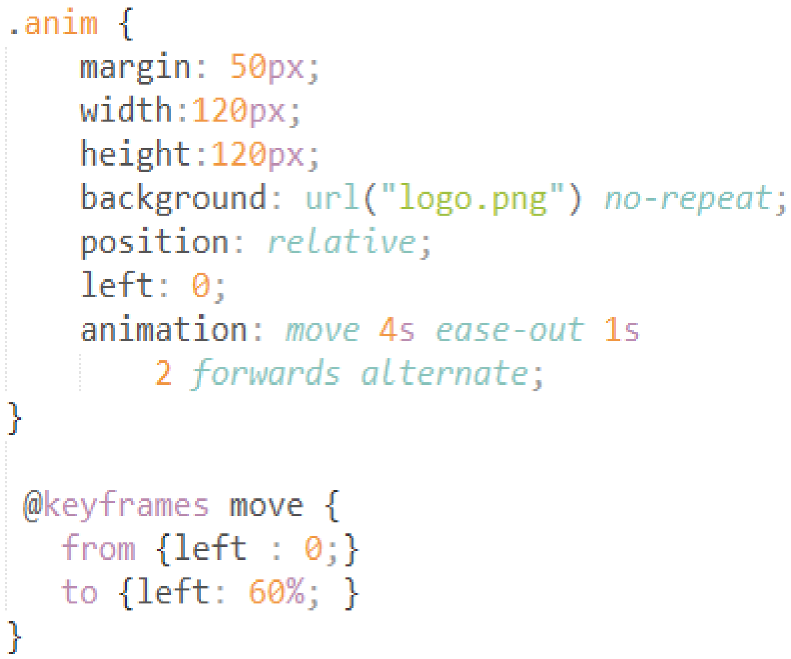
Animation - Example

- E.g. (Each property)



```css
.anim {
    margin: 50px;
    width:120px;
    height:120px;
    background: url("logo.png") no-repeat;
    position: relative;
    left: 0;
    animation-name: move;
    animation-duration: 4s;
    animation-timing-function: ease-out;
    animation-delay: 1s;
    animation-iteration-count: 2;
    animation-fill-mode: forwards;
    animation-direction: alternate;
}

 @keyframes move {
   from {left : 0;}
   to {left: 60%; }
}
```

- E.g. (Shorthand)



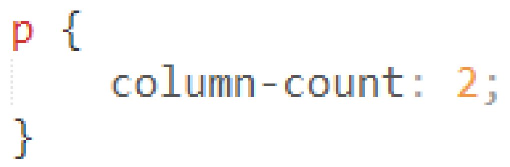
```css
.anim {
    margin: 50px;
    width:120px;
    height:120px;
    background: url("logo.png") no-repeat;
    position: relative;
    left: 0;
    animation: move 4s ease-out 1s
        2 forwards alternate;
}

 @keyframes move {
   from {left : 0;}
   to {left: 60%; }
}
```

Multi-Column Layout

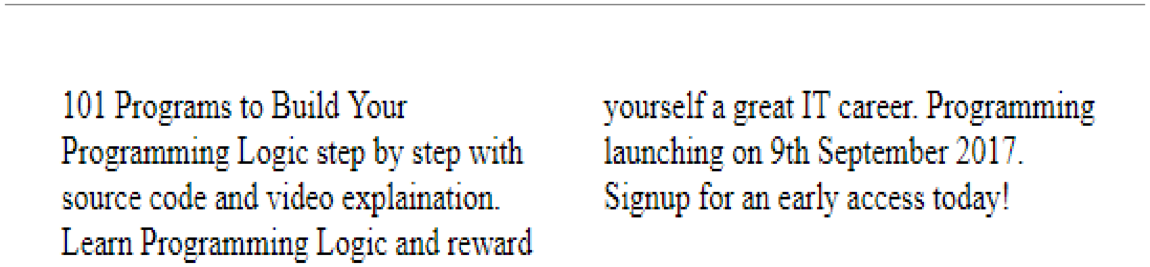
CSS3 by Prashant Chaudhari

Multi-Column Layout

- Allows to create multiple columns without using floating boxes
- **Properties**
    - **column-count**
        - Specifies how many columns will appear



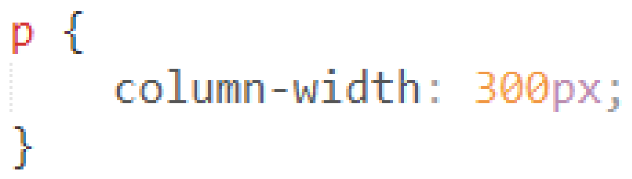
```
p {
    column-count: 2;
}
```



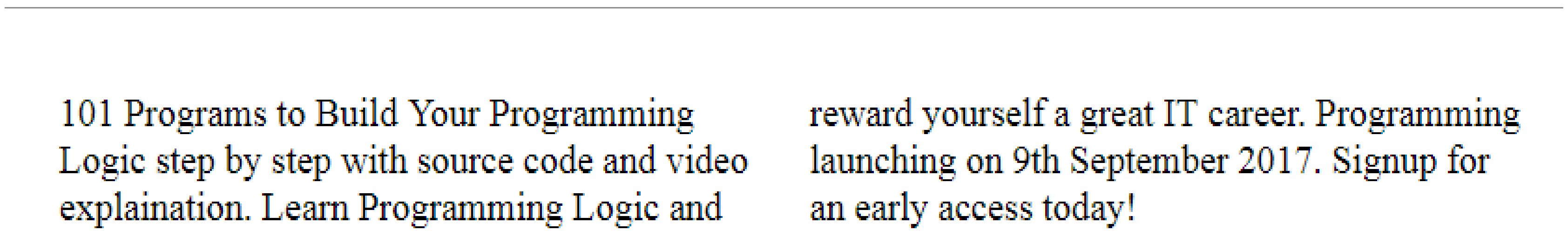
101 Programs to Build Your Programming Logic step by step with source code and video explaination. Learn Programming Logic and reward yourself a great IT career. Programming launching on 9th September 2017. Signup for an early access today!

Multi-Column Layout - Properties

- **column-width**
  - Sets desired column width



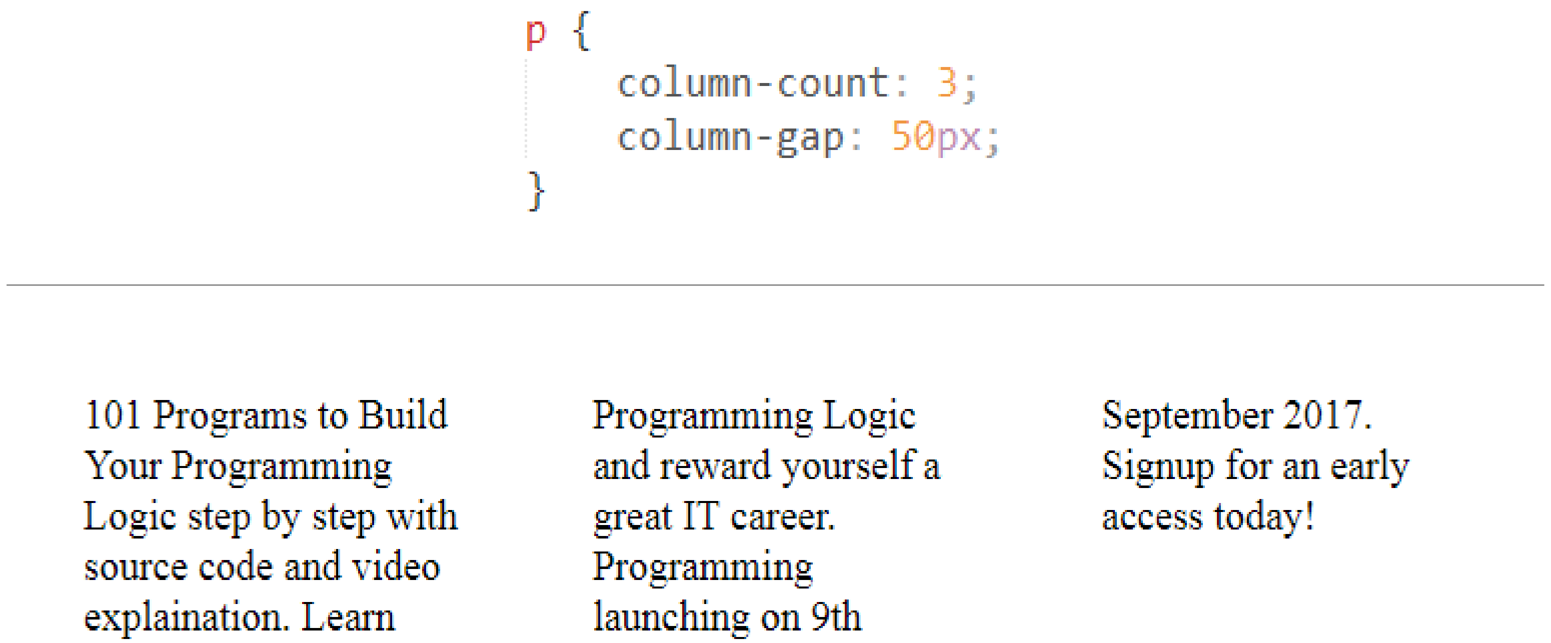
```
p {
    column-width: 300px;
}
```



101 Programs to Build Your Programming Logic step by step with source code and video explaination. Learn Programming Logic and reward yourself a great IT career. Programming launching on 9th September 2017. Signup for an early access today!

Multi-Column Layout - Properties

- **column-gap**
  - Controls gaps between columns

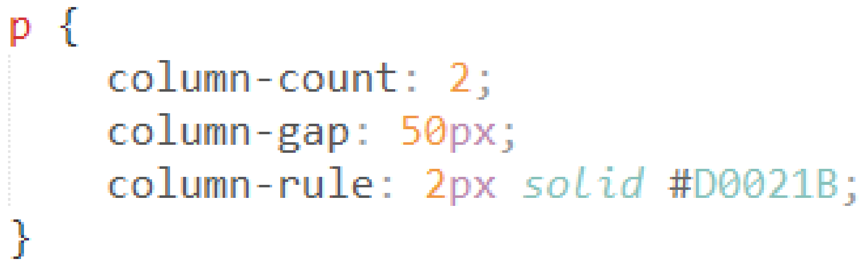


```
p {
    column-count: 3;
    column-gap: 50px;
}
```

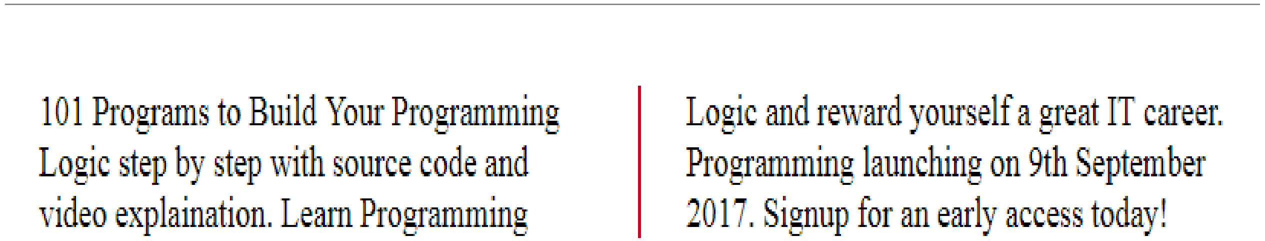
101 Programs to Build Your Programming Logic step by step with source code and video explaination. Learn Programming Logic and reward yourself a great IT career. Programming launching on 9th September 2017. Signup for an early access today!

Multi-Column Layout - Properties

- **column-rule**
  - Adds line between columns with width, style and color



```
p {
    column-count: 2;
    column-gap: 50px;
    column-rule: 2px solid #D0021B;
}
```

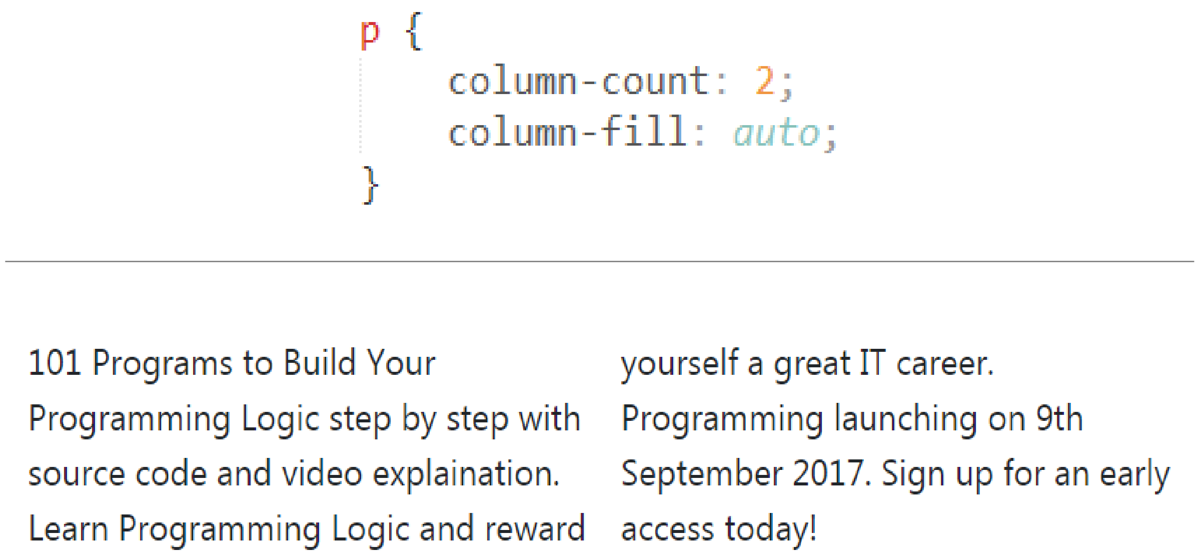


101 Programs to Build Your Programming Logic step by step with source code and video explaination. Learn Programming Logic and reward yourself a great IT career. Programming launching on 9th September 2017. Signup for an early access today!

Multi-Column Layout - Properties

- **column-fill**
  - Specifies how the column lengths in multi-column element are affected by content flow
  - **Property values**

    Balance, auto, initial, inherit
  - **Default value** balance



```
p {
    column-count: 2;
    column-fill: auto;
}
```

101 Programs to Build Your Programming Logic step by step with source code and video explaination. Learn Programming Logic and reward yourself a great IT career. Programming launching on 9th September 2017. Sign up for an early access today!

Box Sizing

CSS3 by Prashant Chaudhari

Box Sizing

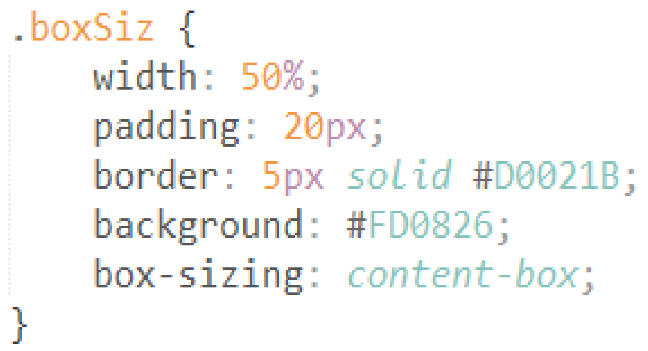
- **CSS box model** adds padding or border outside of content area which affects elements width and height
- **box-sizing** property alters css box model and any padding or border specified on element is laid out and drawn inside of content area
- Using this rendered width and height of element is same as specified CSS width and height properties
- **Property values**

  border-box, content-box

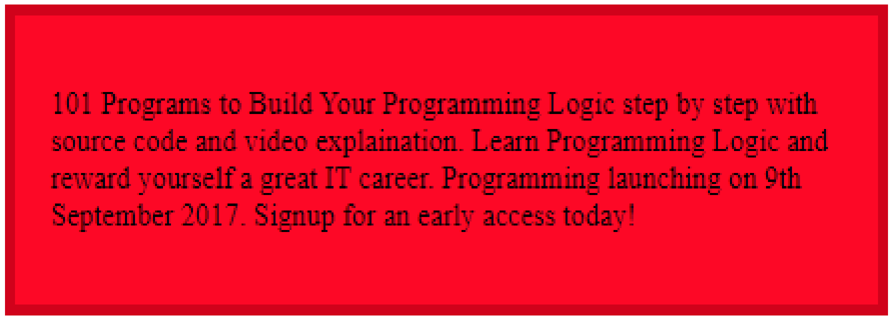
Box Sizing

- **content-box**
  - Default box-sizing behavior
  - Adds elements padding and border width

    in elements width making element wider



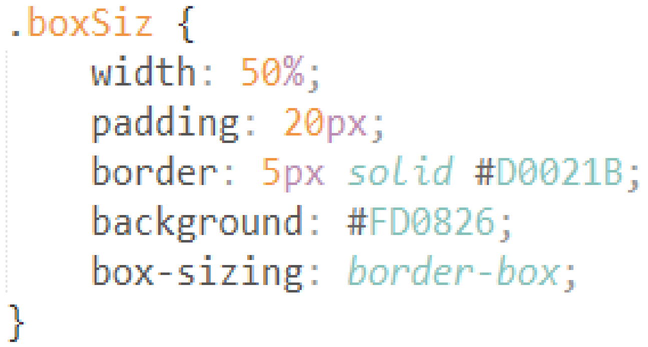
```
.boxSiz {
    width: 50%;
    padding: 20px;
    border: 5px solid #D0021B;
    background: #FD0826;
    box-sizing: content-box;
}
```



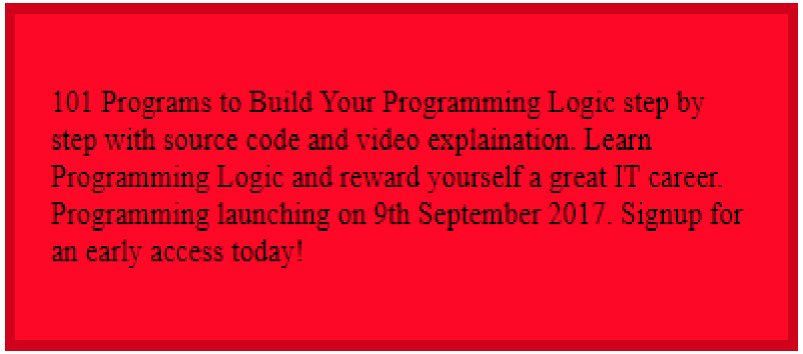
101 Programs to Build Your Programming Logic step by step with source code and video explaination. Learn Programming Logic and reward yourself a great IT career. Programming launching on 9th September 2017. Signup for an early access today!

Box Sizing

- **border-box**
  - Absorbs elements padding and border width, i.e. not increasing elements width



```
.boxSiz {
    width: 50%;
    padding: 20px;
    border: 5px solid #D0021B;
    background: #FD0826;
    box-sizing: border-box;
}
```



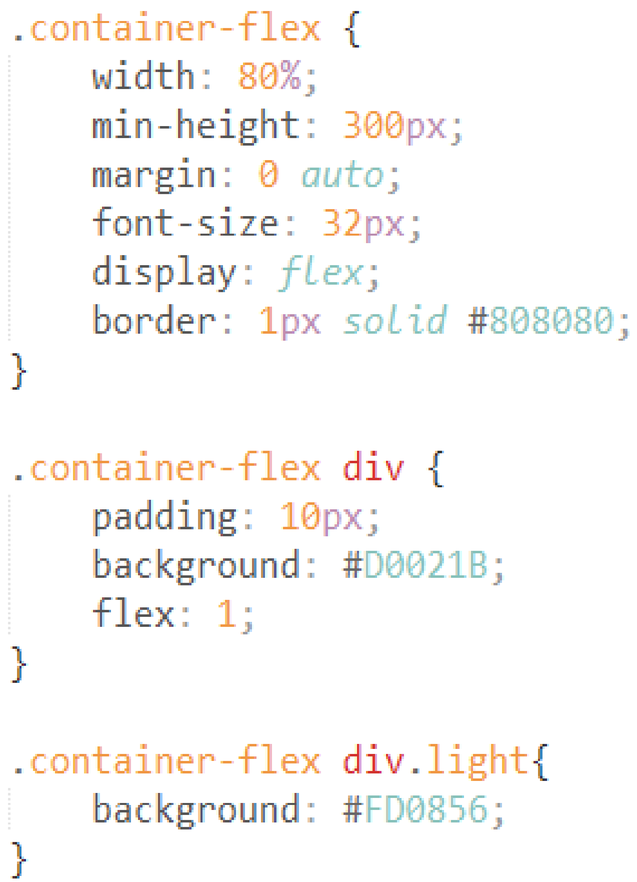
101 Programs to Build Your Programming Logic step by step with source code and video explaination. Learn Programming Logic and reward yourself a great IT career. Programming launching on 9th September 2017. Signup for an early access today!

Flexible Box Layout

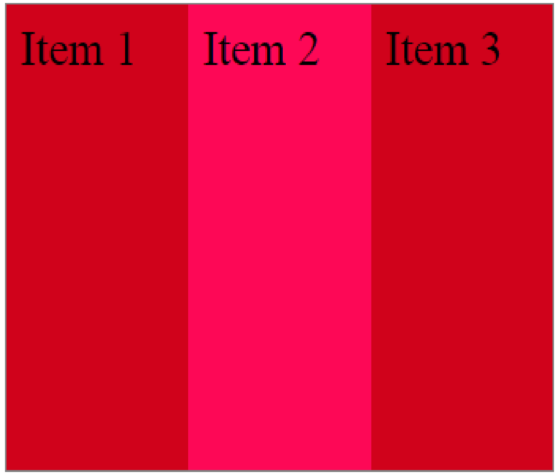
CSS3 by Prashant Chaudhari

Flexible Box Layout

- **Flexbox**
  - flexible user interface with rows and column without percentage or fixed length values

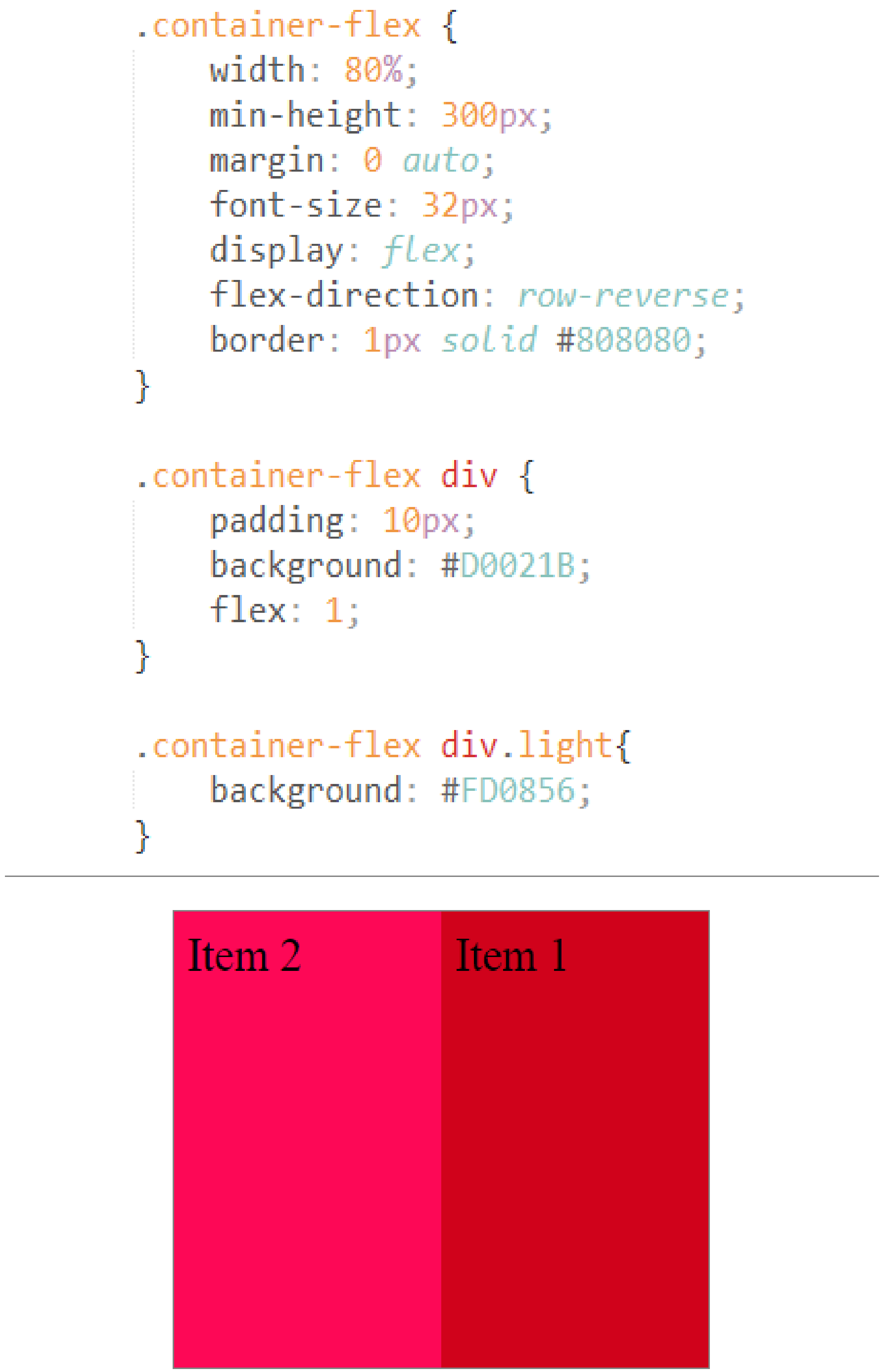


```
.container-flex {
    width: 80%;
    min-height: 300px;
    margin: 0 auto;
    font-size: 32px;
    display: flex;
    border: 1px solid #808080;
}

.container-flex div {
    padding: 10px;
    background: #D0021B;
    flex: 1;
}

.container-flex div.light{
    background: #FD0856;
}
```



Flexible Box Layout

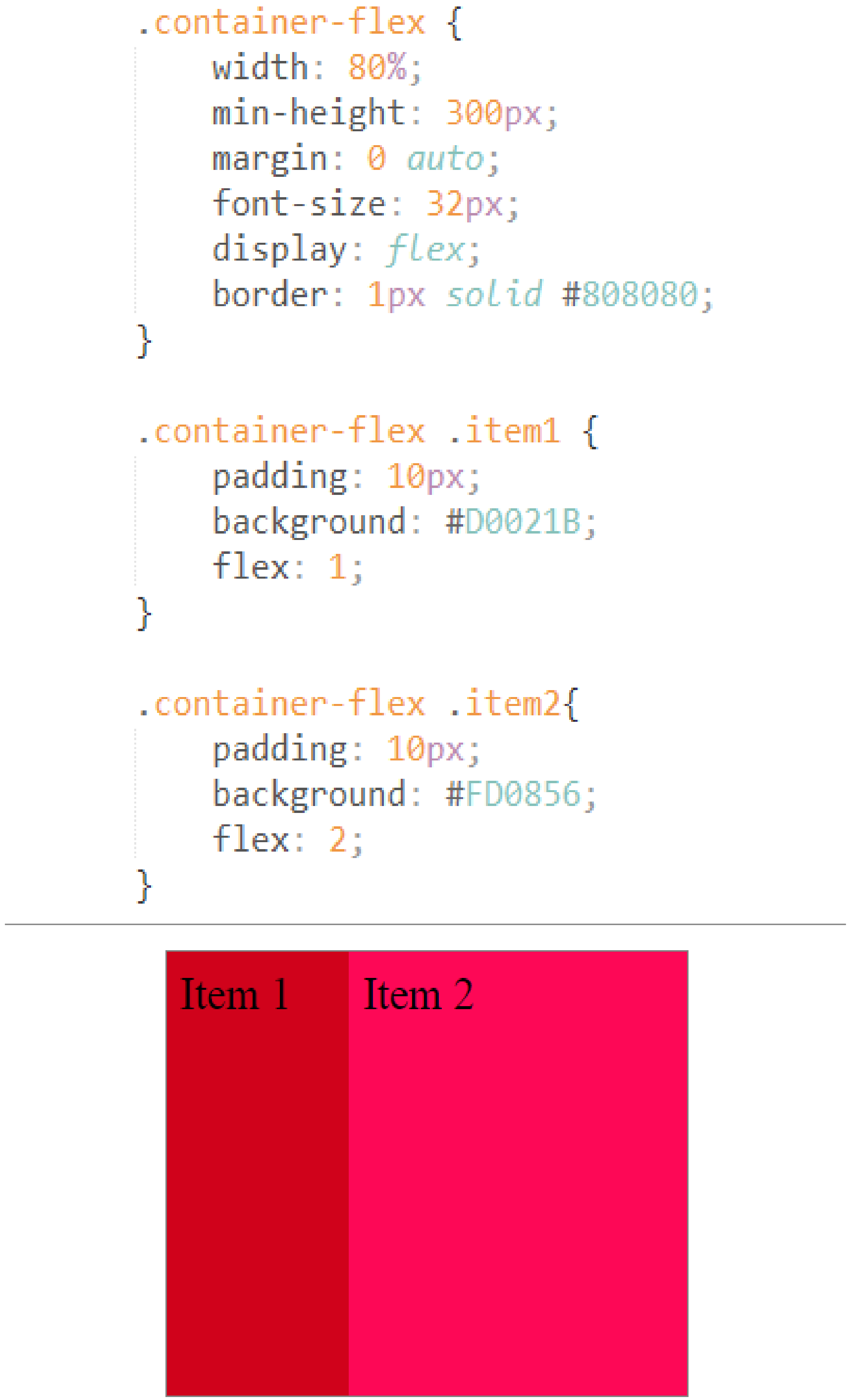
- **flex-direction**
  - Controls flow of flex items inside flex container
  - Default value is **row**
  - **Possible values**
    - row, row-reverse, column, column-reverse



```
.container-flex {
    width: 80%;
    min-height: 300px;
    margin: 0 auto;
    font-size: 32px;
    display: flex;
    flex-direction: row-reverse;
    border: 1px solid #808080;
}

.container-flex div {
    padding: 10px;
    background: #D0021B;
    flex: 1;
}

.container-flex div.light{
    background: #FD0856;
}
```

Flexible Box Layout

- **flex**
  - Alter width or height to fill available space



```css
.container-flex {
    width: 80%;
    min-height: 300px;
    margin: 0 auto;
    font-size: 32px;
    display: flex;
    border: 1px solid #808080;
}

.container-flex .item1 {
    padding: 10px;
    background: #D0021B;
    flex: 1;
}

.container-flex .item2{
    padding: 10px;
    background: #FD0856;
    flex: 2;
}
```

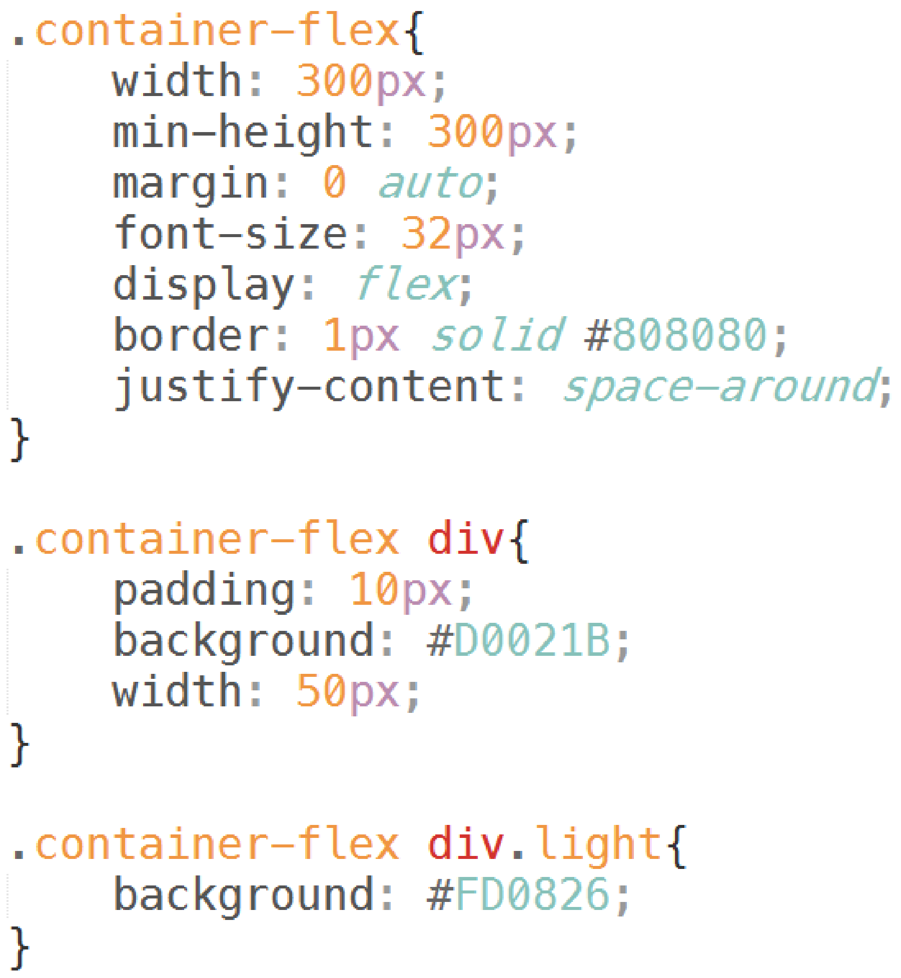
Flexible Box Layout

- **Aligning flex items**
    - **Properties**
        - justify-content
        - align-content
        - align-items
        - align-self
    - First three for flex-container and last one for flex-items

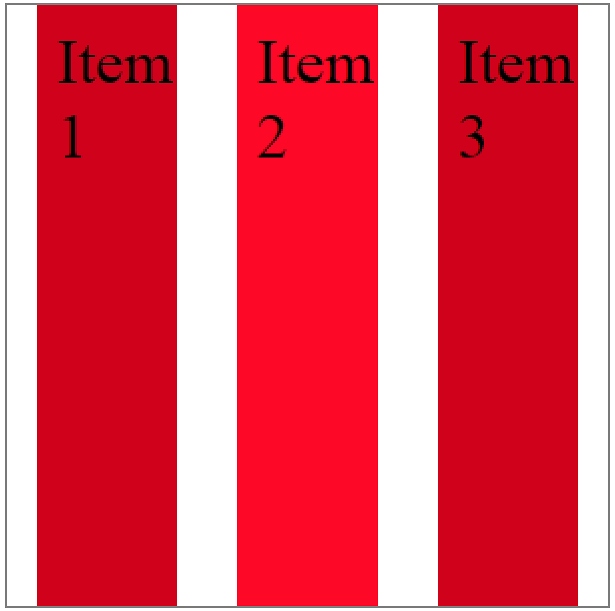
Flexible Box Layout - Properties

- **justify-content**
  - align items in horizontal direction of container
  - **Values**
    - flex-start
    - flex-end
    - center
    - space-between
    - space-around

Flexible Box Layout – justify-content



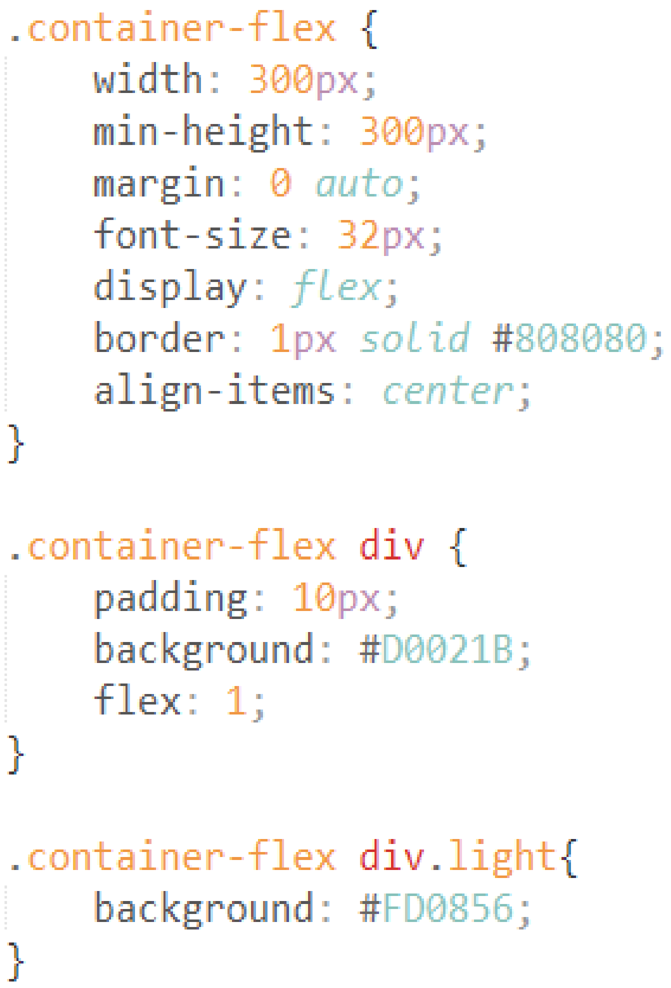
```
.container-flex{
    width: 300px;
    min-height: 300px;
    margin: 0 auto;
    font-size: 32px;
    display: flex;
    border: 1px solid #808080;
    justify-content: space-around;
}

.container-flex div{
    padding: 10px;
    background: #D0021B;
    width: 50px;
}

.container-flex div.light{
    background: #FD0826;
}
```



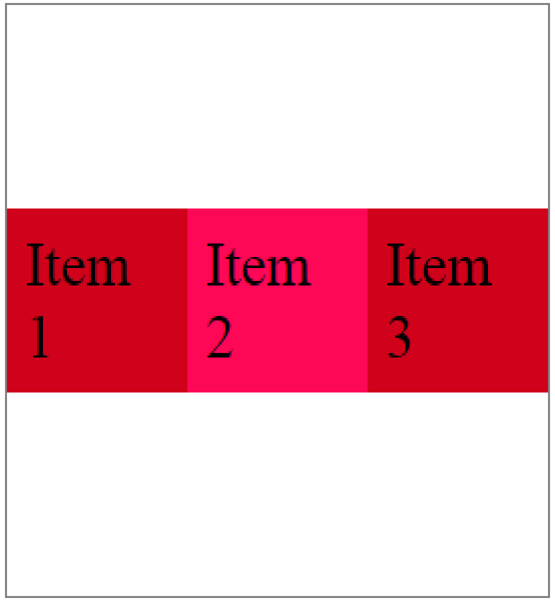
Flexible Box Layout - Properties

- **align-items**
  - align items in perpendicular direction of container
  - **Values**
    - flex-start
    - flex-end
    - Center
    - baseline
    - stretch

Flexible Box Layout – align-items



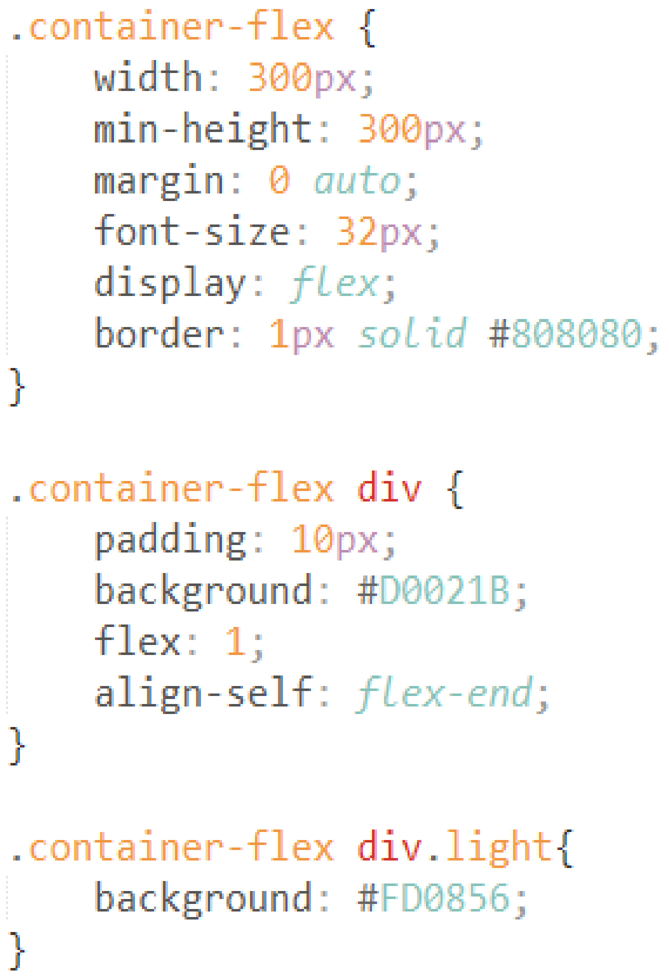
```
.container-flex {
    width: 300px;
    min-height: 300px;
    margin: 0 auto;
    font-size: 32px;
    display: flex;
    border: 1px solid #808080;
    align-items: center;
}

.container-flex div {
    padding: 10px;
    background: #D0021B;
    flex: 1;
}

.container-flex div.light{
    background: #FD0856;
}
```



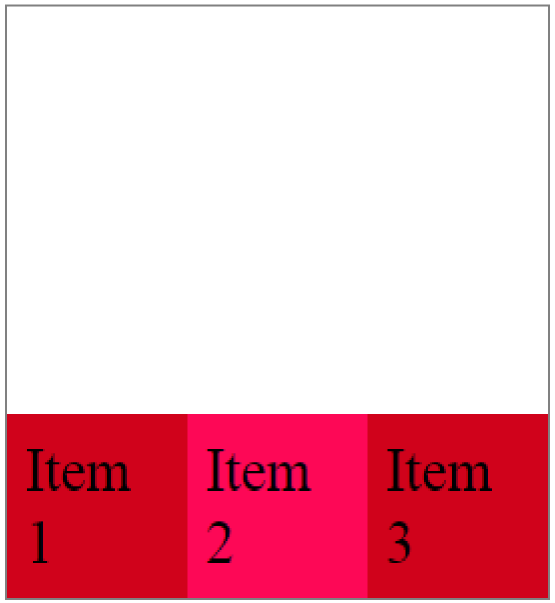
Flexible Box Layout - Properties

- **align-self**
  - specifies the alignment for the selected item inside the flexible container
  - **Values**
    - flex-start
    - flex-end
    - center
    - baseline
    - stretch

Flexible Box Layout – align-self



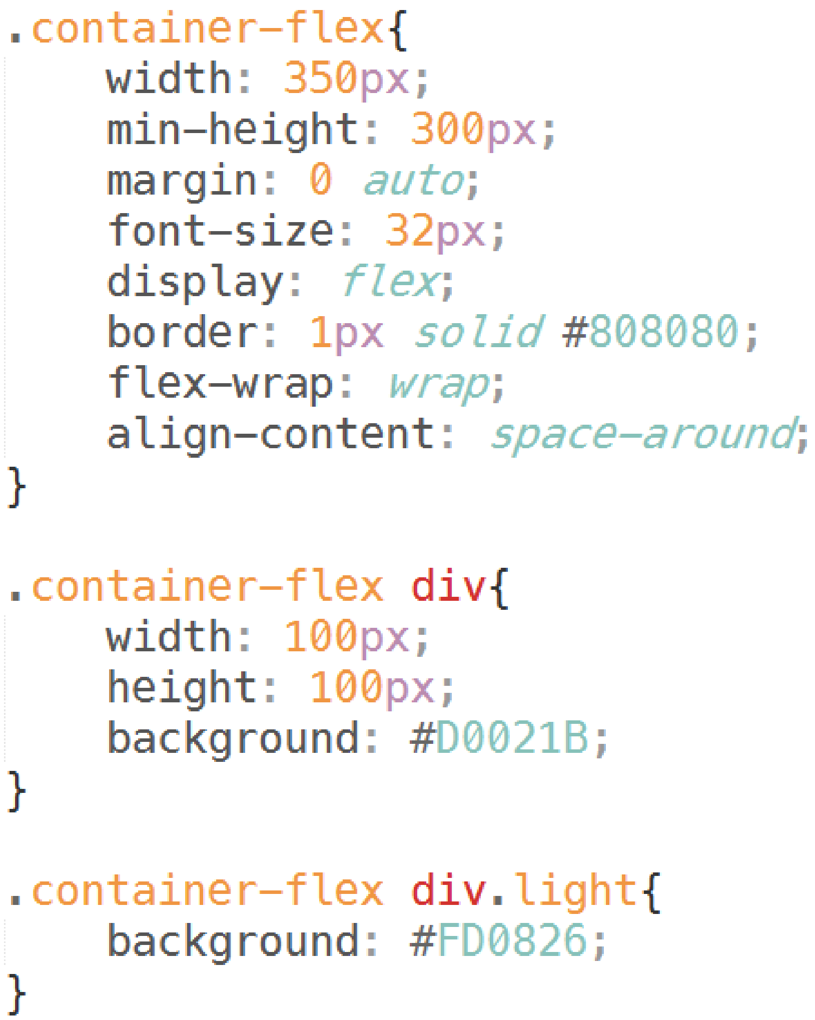
```css
.container-flex {
    width: 300px;
    min-height: 300px;
    margin: 0 auto;
    font-size: 32px;
    display: flex;
    border: 1px solid #808080;
}

.container-flex div {
    padding: 10px;
    background: #D0021B;
    flex: 1;
    align-self: flex-end;
}

.container-flex div.light{
    background: #FD0856;
}
```



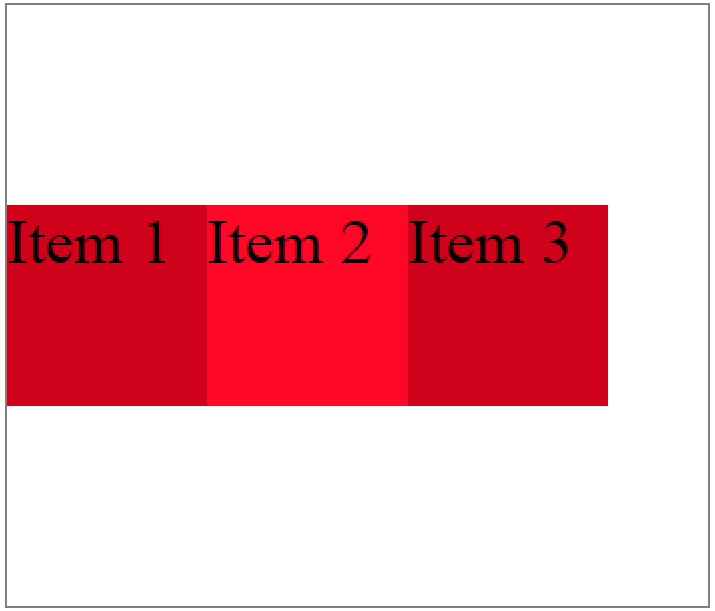
Flexible Box Layout - Properties

- **align-content**
  - Helps to align flex-containers lines within it when there is an extra space in cross-axis
  - Takes similar values as justify-content, but applies them to cross-axis rather than main-axis
  - **Accepts one more value**
    - stretch

Flexible Box Layout – align-content

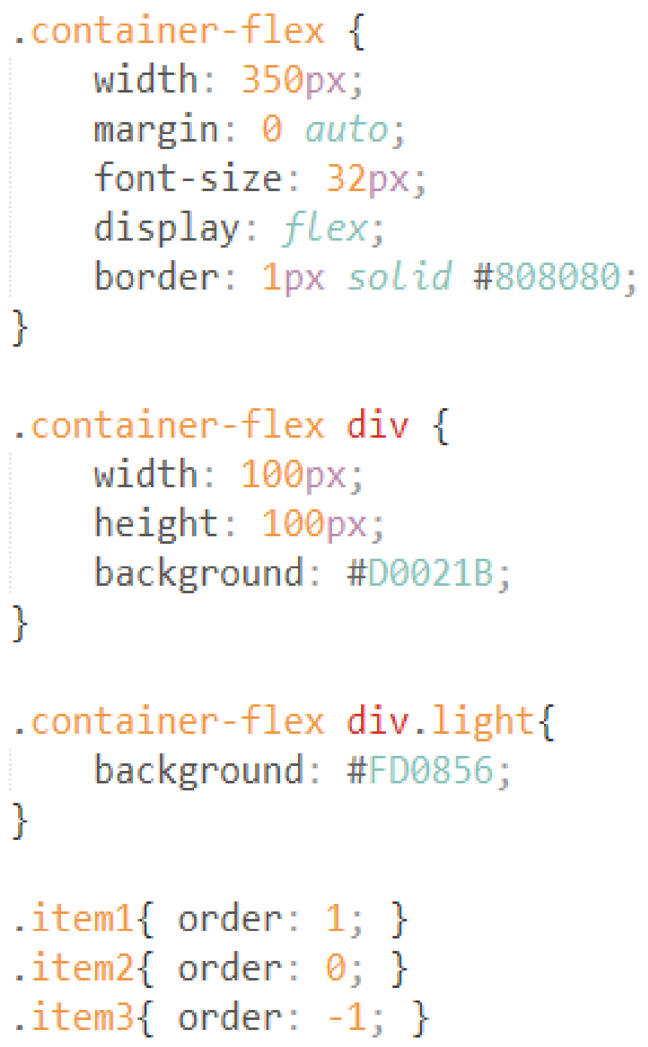


```css
.container-flex{
    width: 350px;
    min-height: 300px;
    margin: 0 auto;
    font-size: 32px;
    display: flex;
    border: 1px solid #808080;
    flex-wrap: wrap;
    align-content: space-around;
}

.container-flex div{
    width: 100px;
    height: 100px;
    background: #D0021B;
}

.container-flex div.light{
    background: #FD0826;
}
```

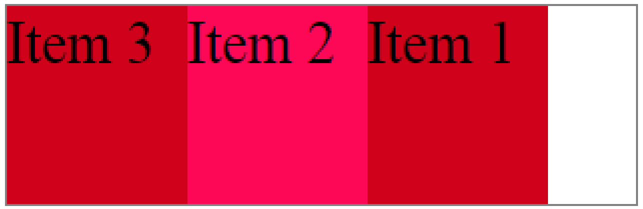


Flexible Box Layout - Properties

- **order**
  - Can change order of flex item
  - Accepts positive and negative values
  - Default value is 0

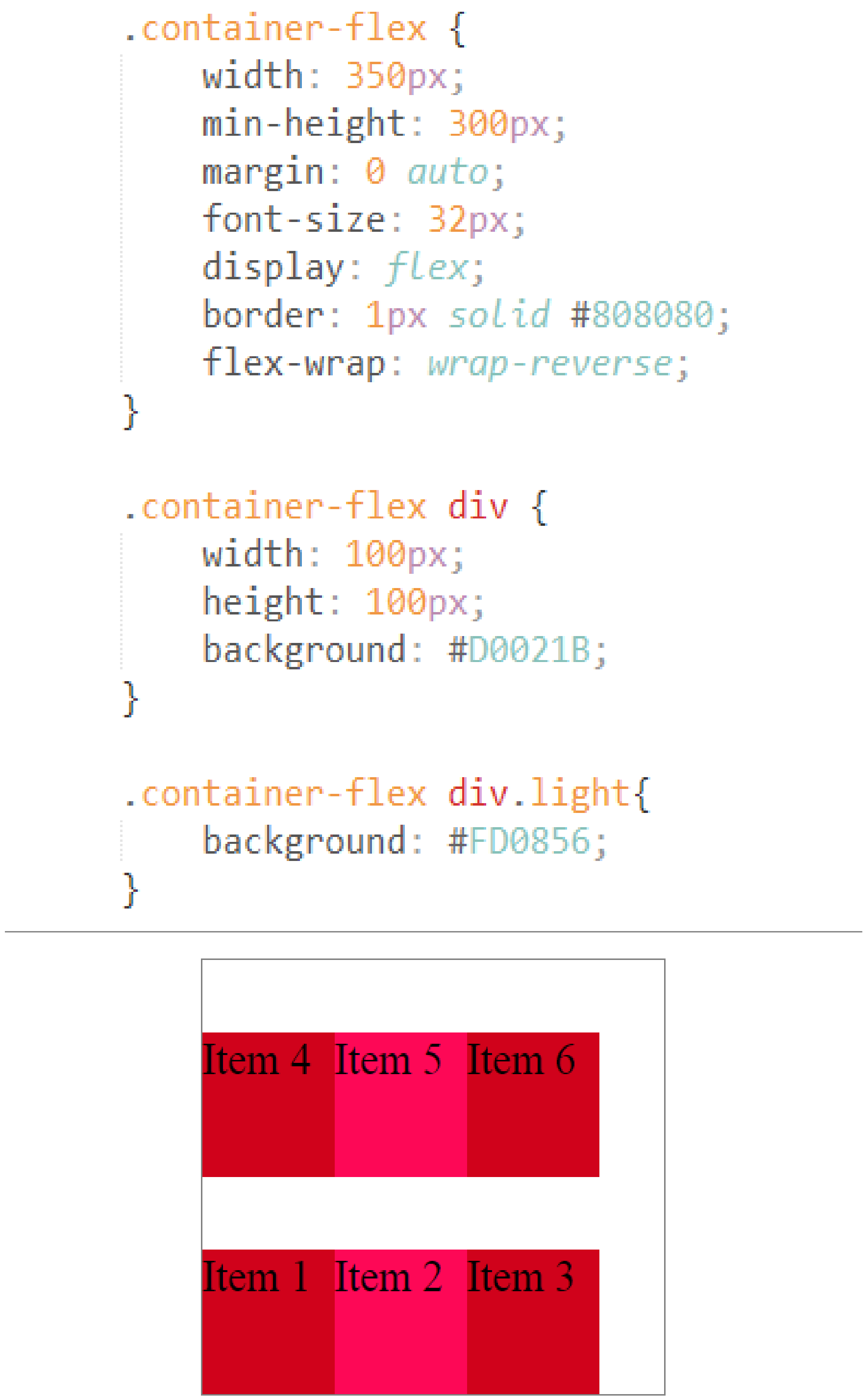


```
.container-flex {
    width: 350px;
    margin: 0 auto;
    font-size: 32px;
    display: flex;
    border: 1px solid #808080;
}

.container-flex div {
    width: 100px;
    height: 100px;
    background: #D0021B;
}

.container-flex div.light{
    background: #FD0856;
}

.item1{ order: 1; }
.item2{ order: 0; }
.item3{ order: -1; }
```



Flexible Box Layout - Properties

- **flex-wrap**
  - Wraps flex items into multiple lines if no space in flex line
  - **Values**
    - nowrap
    - wrap
    - wrap-reverse

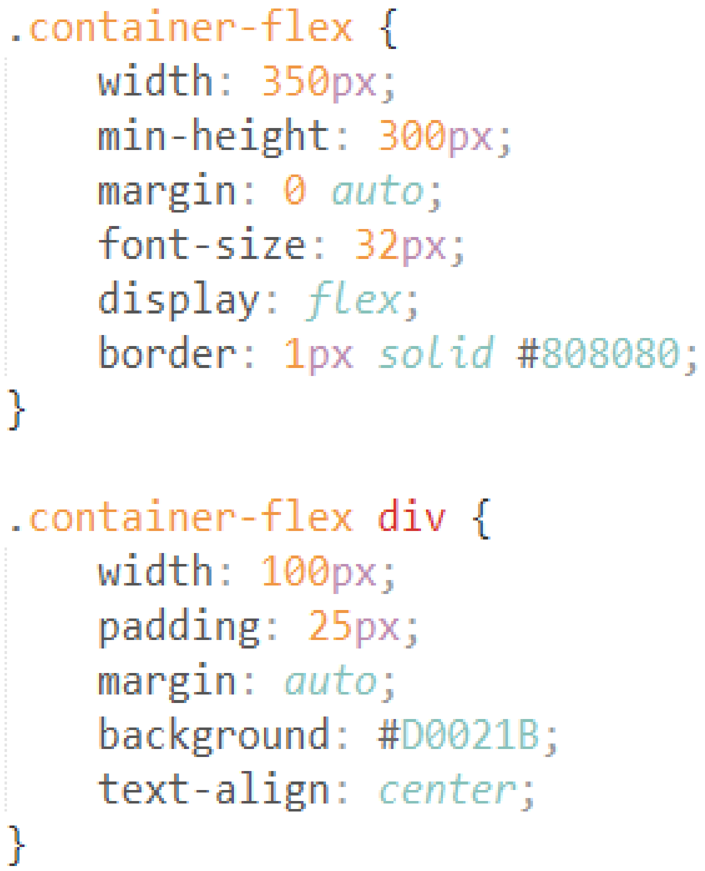


```
.container-flex {
    width: 350px;
    min-height: 300px;
    margin: 0 auto;
    font-size: 32px;
    display: flex;
    border: 1px solid #808080;
    flex-wrap: wrap-reverse;
}

.container-flex div {
    width: 100px;
    height: 100px;
    background: #D0021B;
}

.container-flex div.light{
    background: #FD0856;
}
```

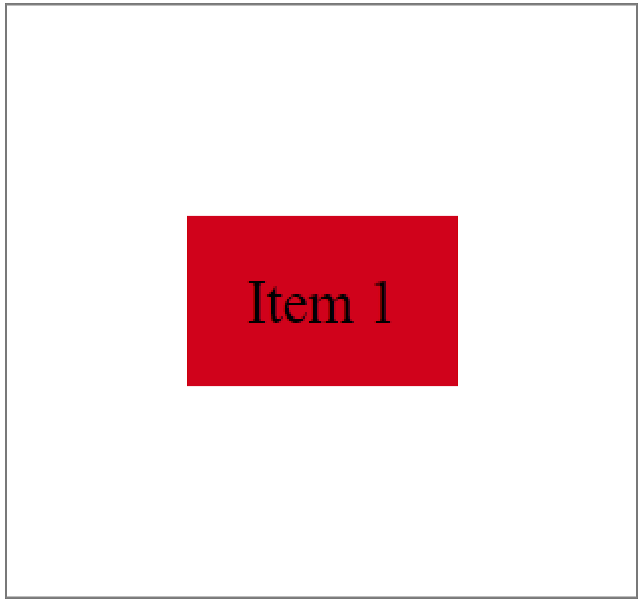
```
Item 4  Item 5  Item 6

Item 1  Item 2  Item 3
```

Flexible Box Layout - Properties

- **Horizontal and Vertical Center**



```
.container-flex {
    width: 350px;
    min-height: 300px;
    margin: 0 auto;
    font-size: 32px;
    display: flex;
    border: 1px solid #808080;
}

.container-flex div {
    width: 100px;
    padding: 25px;
    margin: auto;
    background: #D0021B;
    text-align: center;
}
```



Filters

CSS3 by Prashant Chaudhari

Filters

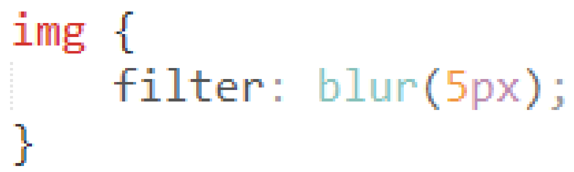
- Applies visual effects on graphical elements
- Filter effects applied to element using **filter** property
- **Filter functions**

  blur(), brightness(), contrast(), drop-shadow(), grayscale(), hue-rotate(), invert(),

  opacity(), sepia(), saturate(), url()

Filter - Functions

- **blur()**
  - Adds blur effect to element
  - Takes length value which defines blur radius
  - Larger value will create more blur

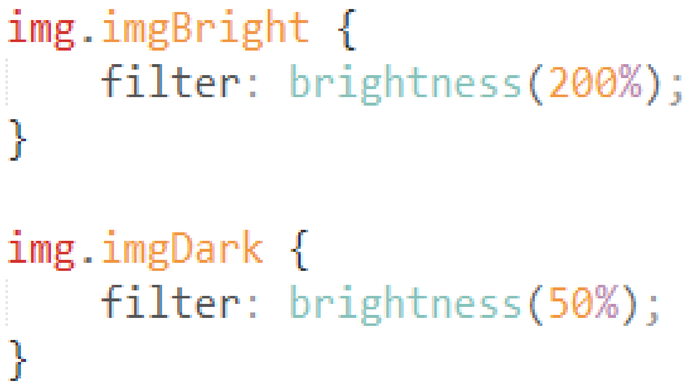


```css
img {
    filter: blur(5px);
}
```

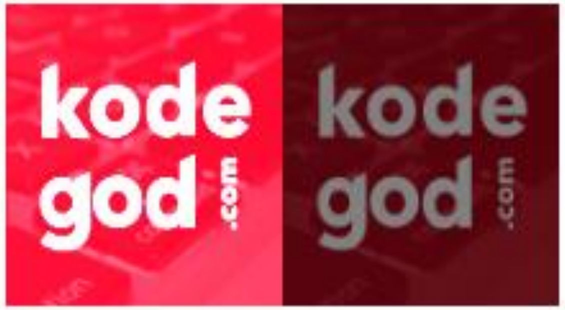
Filter - Functions

- **brightness()**
  - Sets brightness of an image
  - Value 0% makes image completely black and value

    100% or 1 leaves image unchanged
  - Value grater than 100% makes image more brighter

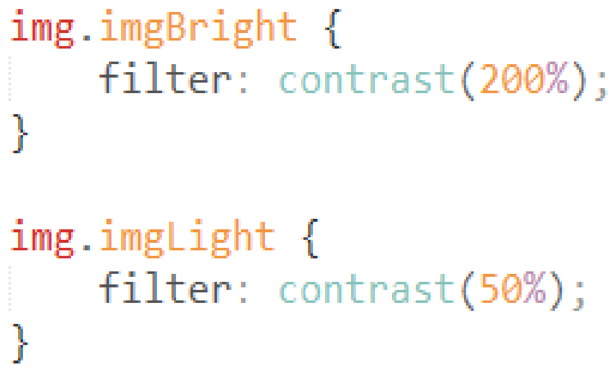


```
img.imgBright {
    filter: brightness(200%);
}

img.imgDark {
    filter: brightness(50%);
}
```

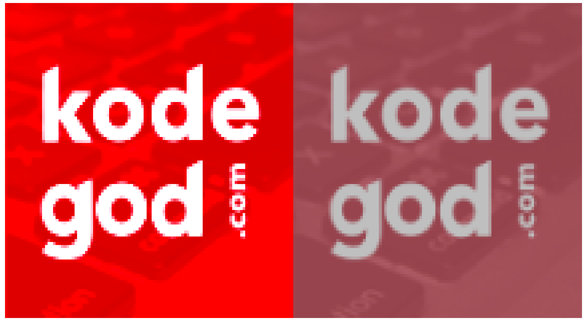


Filter - Functions

- **contrast()**
  - Adjusts contrast of an image
  - Value 0% makes image completely black and value 100% or 1 leaves image unchanged
  - Value grater than 100% provides result with less contrast

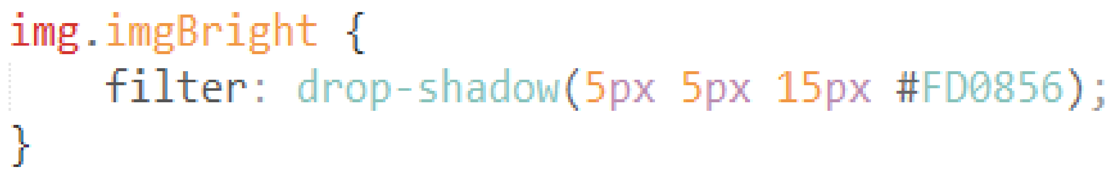


```
img.imgBright {
    filter: contrast(200%);
}

img.imgLight {
    filter: contrast(50%);
}
```

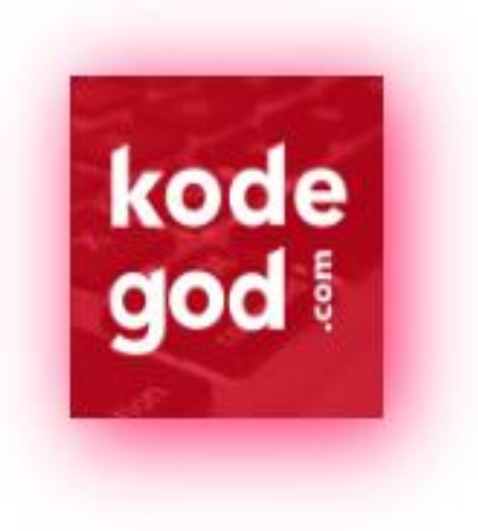


Filter - Functions

- **drop-shadow()**
  - Applies drop shadow effect to image
  - Similar to box-shadow property

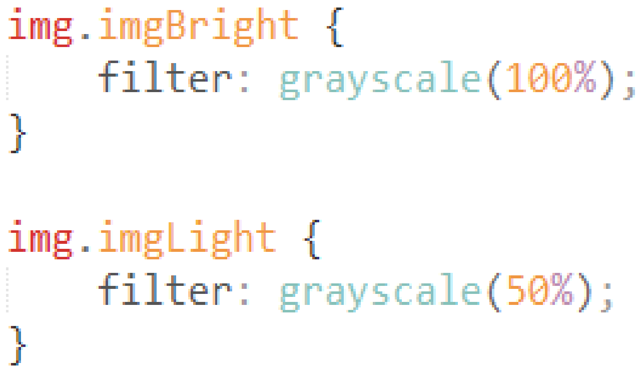


```css
img.imgBright {
    filter: drop-shadow(5px 5px 15px #FD0856);
}
```

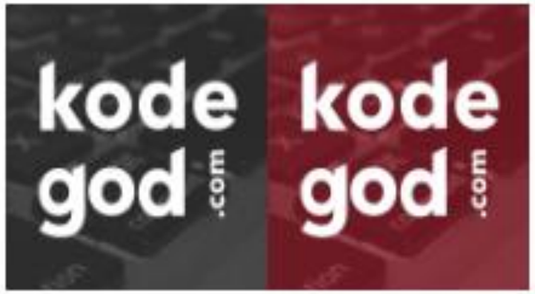


Filter - Functions

- **grayscale()**
  - Images converted to grayscale
  - Value 100% makes image completely grayscale and value 0% leaves image unchanged

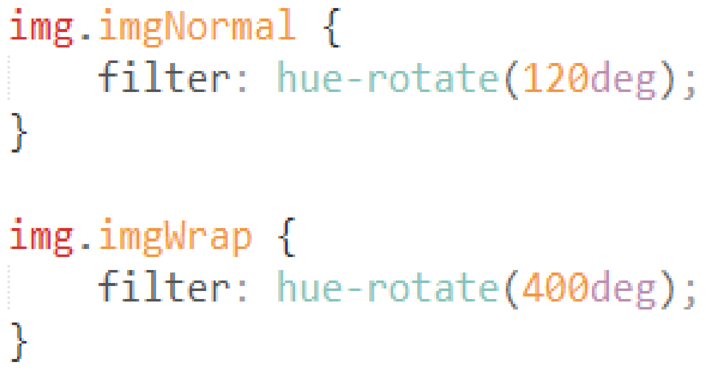


```
img.imgBright {
    filter: grayscale(100%);
}

img.imgLight {
    filter: grayscale(50%);
}
```

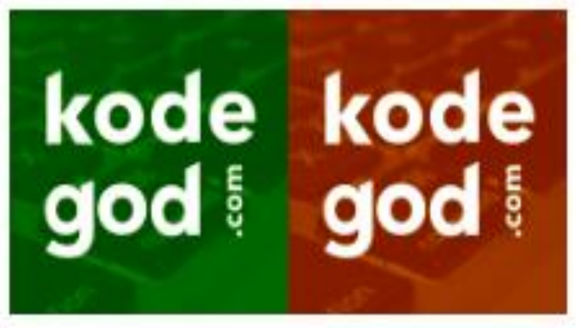


Filter - Functions

- **hue-rotate()**
  - Applies hue rotation on image
  - Value 0deg leaves image unchanged
  - No maximum value, the effect of values above 360deg wraps around



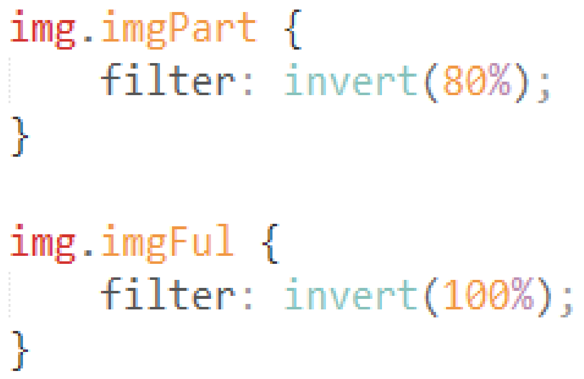
```
img.imgNormal {
    filter: hue-rotate(120deg);
}

img.imgWrap {
    filter: hue-rotate(400deg);
}
```



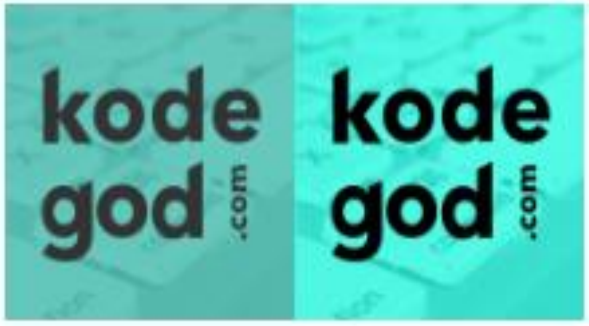
Filter - Functions

- **invert()**
  - Applies invert effect on image
  - Value 100% or 1completely inverted, value 0% leaves

    image unchanged
  - Negative values are not allowed

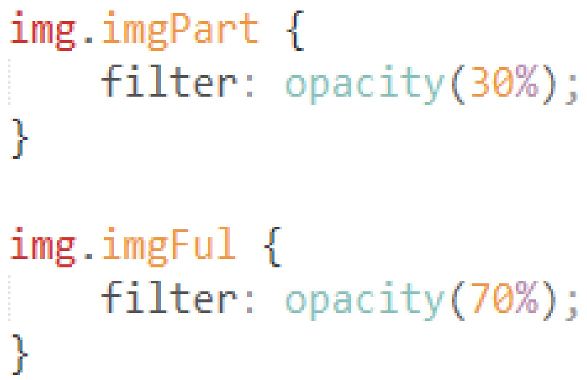


```
img.imgPart {
    filter: invert(80%);
}

img.imgFul {
    filter: invert(100%);
}
```

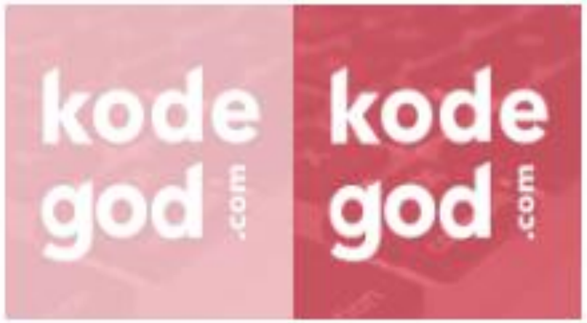


Filter - Functions

- **opacity()**
  - Applies transparency to image
  - Value 0% is completely transparent, value 100% or 1 leaves image unchanged

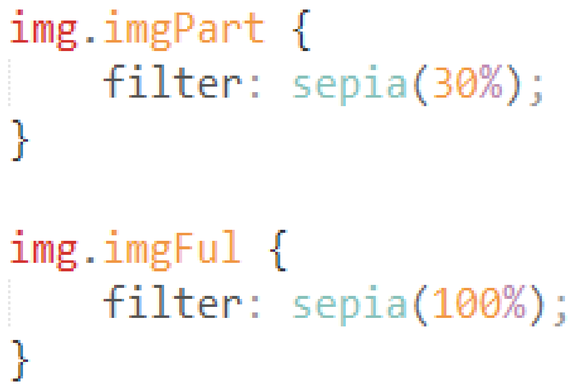


```
img.imgPart {
    filter: opacity(30%);
}

img.imgFul {
    filter: opacity(70%);
}
```

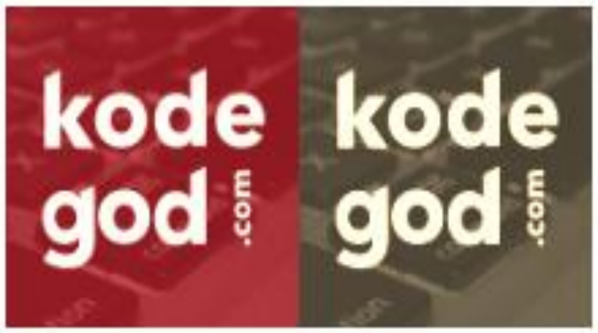


Filter - Functions

- **sepia()**
  - Converts image to sepia
  - Value 100% or 1 is completely sepia, value 0% leaves image unchanged



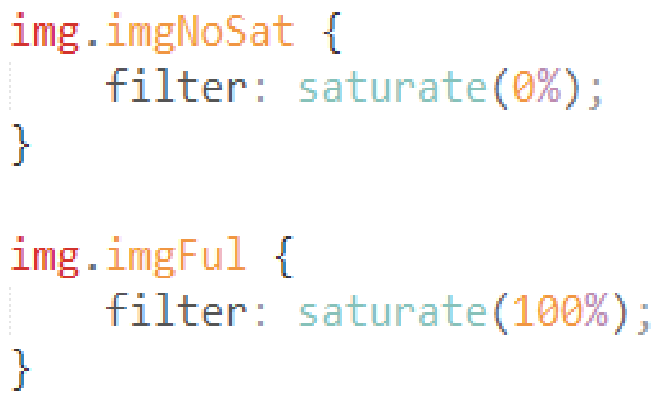
```css
img.imgPart {
    filter: sepia(30%);
}

img.imgFul {
    filter: sepia(100%);
}
```



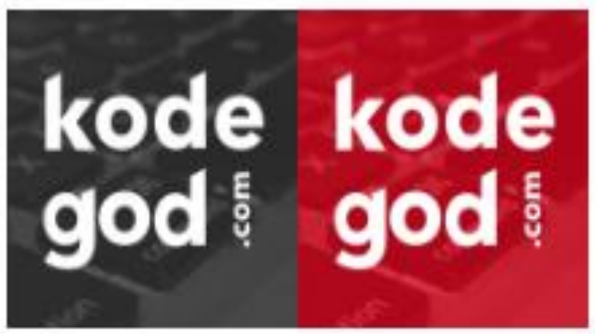
Filter - Functions

- **saturation()**
  - Adjusts saturation of the image
  - Value 0% is completely un-saturated, value 100% or 1

    leaves image unchanged



```
img.imgNoSat {
    filter: saturate(0%);
}

img.imgFul {
    filter: saturate(100%);
}
```



Media Types

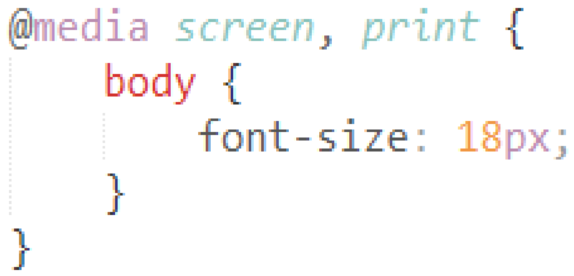
CSS3 by Prashant Chaudhari

Media Types

- Allows to format document to be presented correctly in various types of media such as screen, print, etc.
- **Methods to specify media dependencies for style sheets**
  - **@media**
  - **@import**
  - **<link>**

Media Types

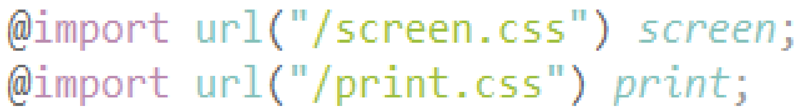
- **@media**
  - Used to define different style rules for different media types
  - @media followed by comma separated list of media types



```
@media screen, print {
    body {
        font-size: 18px;
    }
}
```

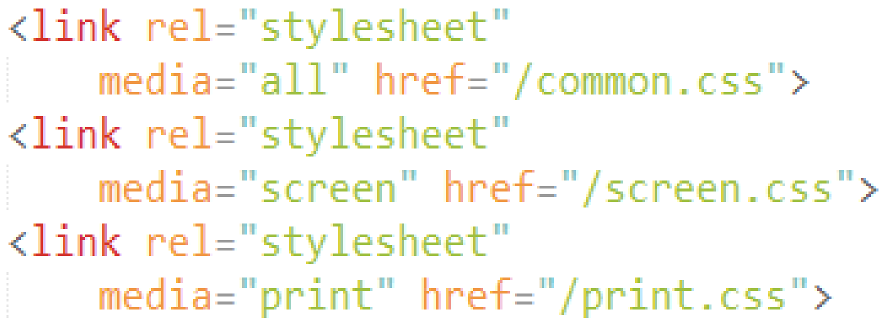
Media Types

- **@import**
  - Used to style information for specific target media
  - @import followed by url of imported stylesheets and comma separated list of media types



```
@import url("/screen.css") screen;
@import url("/print.css") print;
```

- **<link>** element
  - **Media** attribute in link element is used to specify target media



```
<link rel="stylesheet"
    media="all" href="/common.css">
<link rel="stylesheet"
    media="screen" href="/screen.css">
<link rel="stylesheet"
    media="print" href="/print.css">
```

Media Types

- **Media Types are**
  - **all** for all media type devices
  - **aural** for speech and sound synthesizers
  - **braille** for braille tactile feedback devices
  - **embossed** for paged braille printers

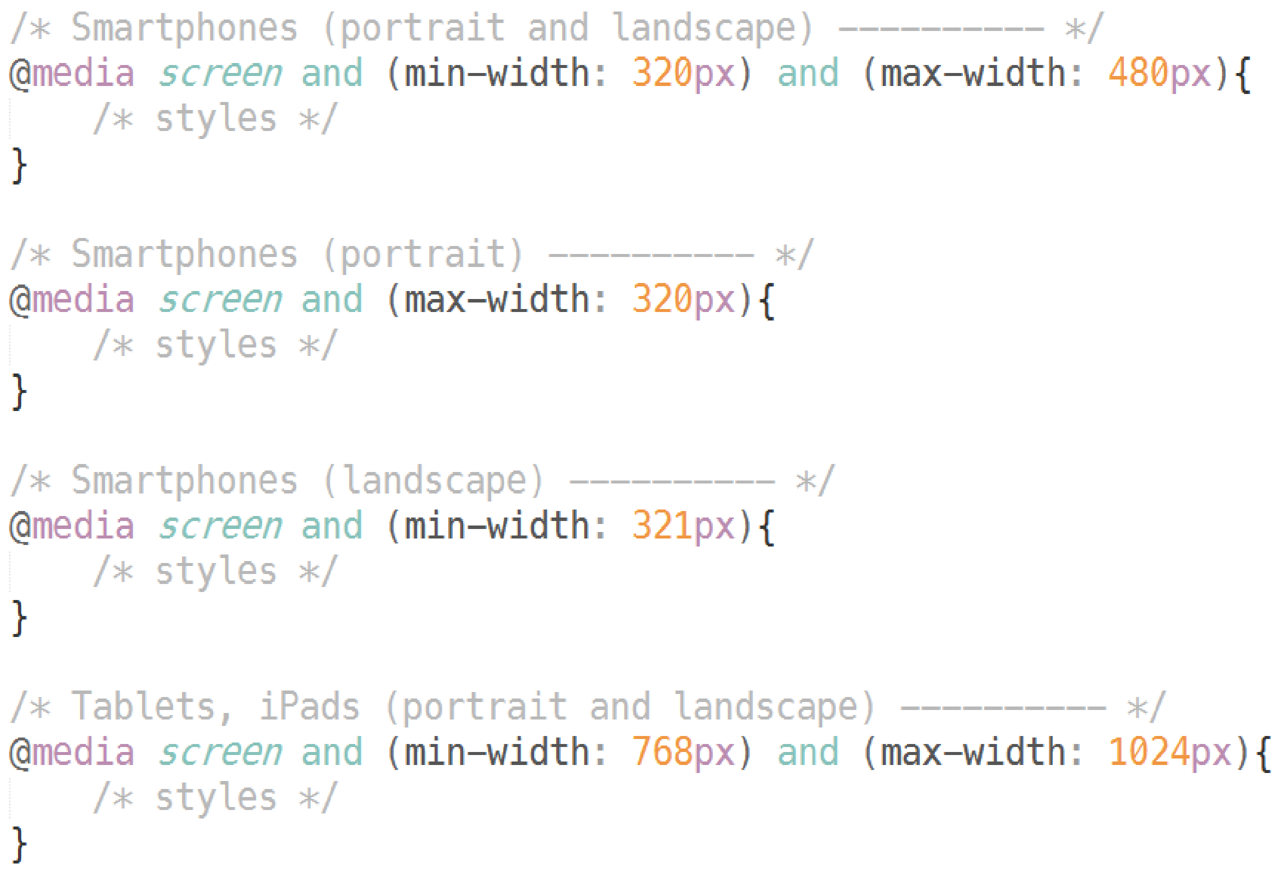
- **handheld** small or handheld devices
- **print** used for printers
- **projection** for projectors
- **screen** for color computer screens
- **tty** for teletypes, terminals, portable devices with limited display capabilities
- **tv** for television type devices

Media Queries

CSS3 by Prashant Chaudhari

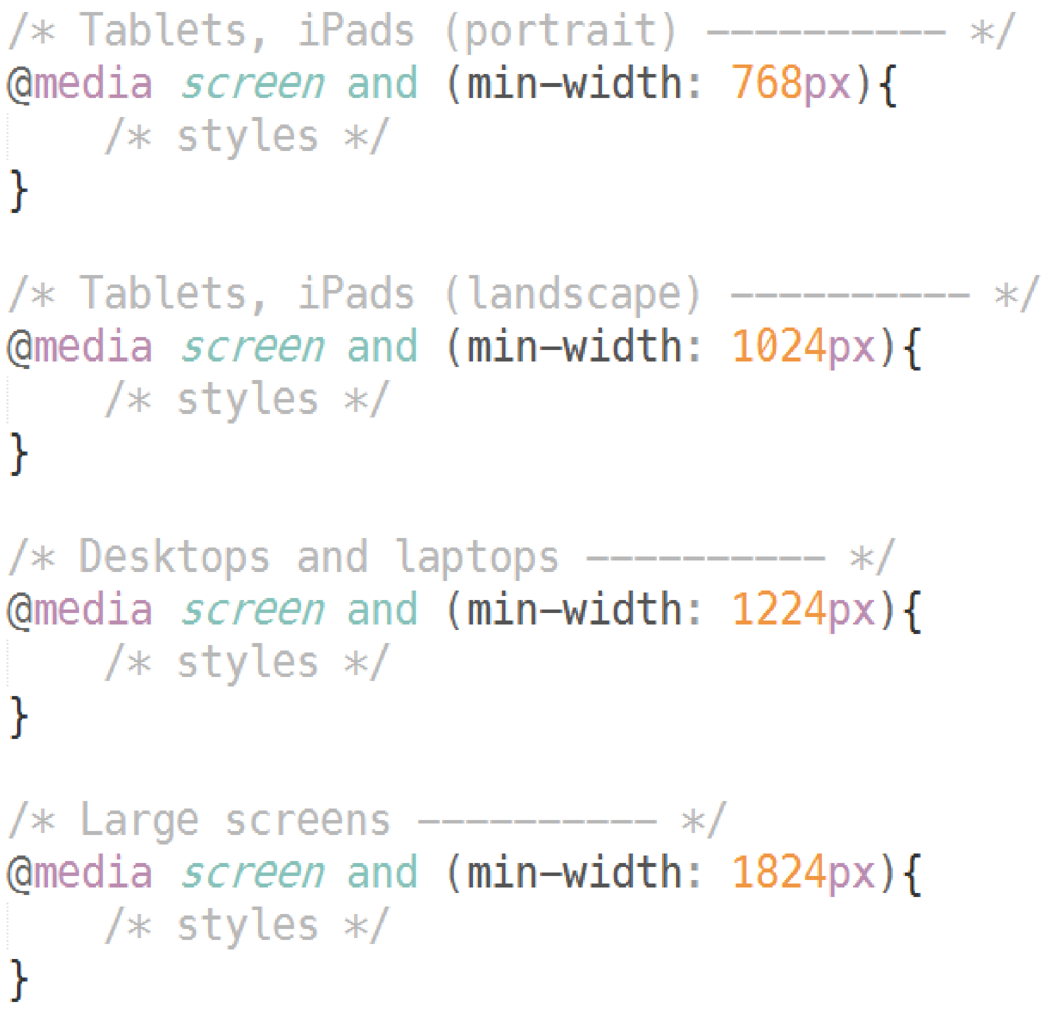
Media Queries

- Used to customize the presentation of web pages for specific range of devices



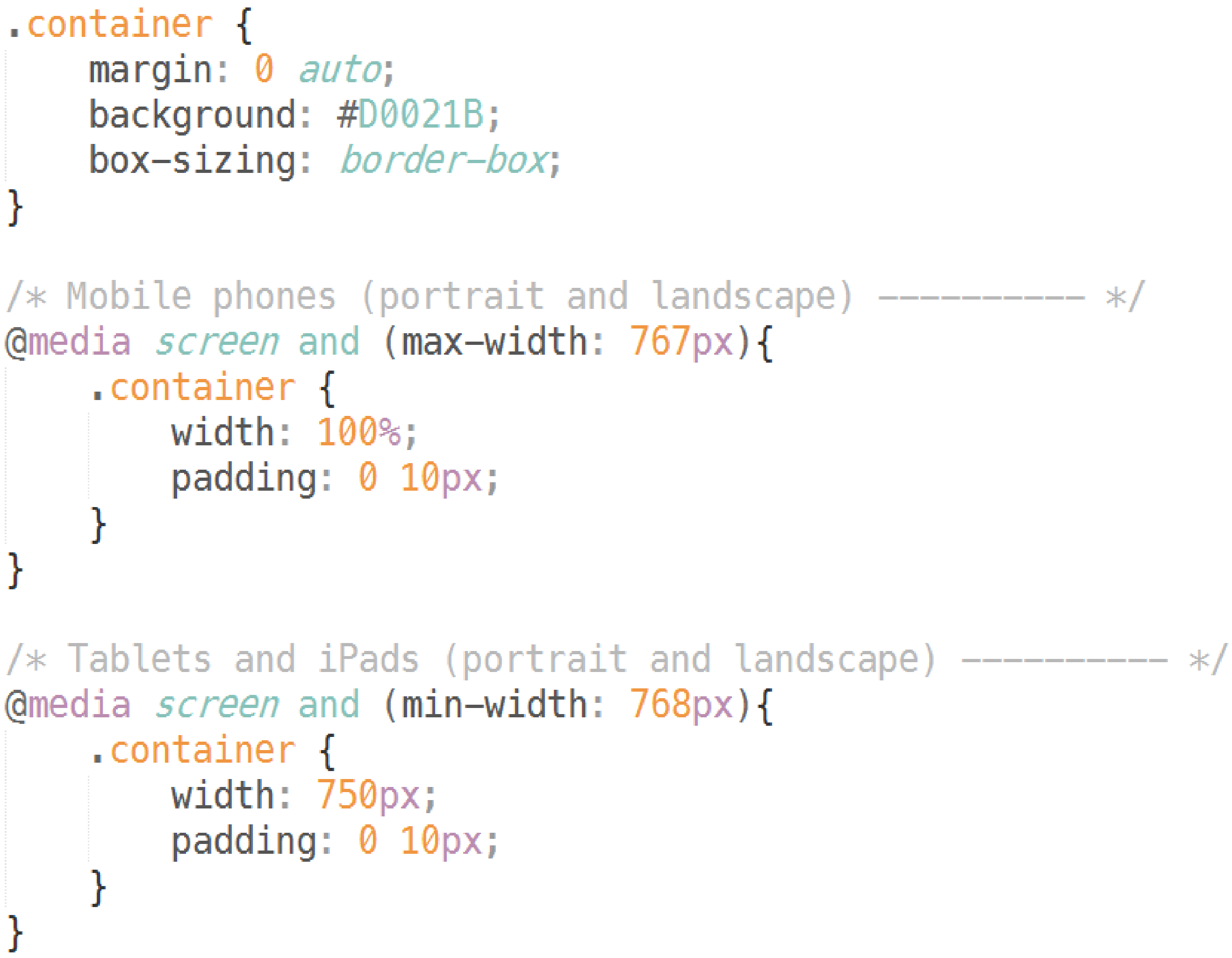
```
/* Smartphones (portrait and landscape) ---------- */
@media screen and (min-width: 320px) and (max-width: 480px){
    /* styles */
}

/* Smartphones (portrait) ---------- */
@media screen and (max-width: 320px){
    /* styles */
}

/* Smartphones (landscape) ---------- */
@media screen and (min-width: 321px){
    /* styles */
}

/* Tablets, iPads (portrait and landscape) ---------- */
@media screen and (min-width: 768px) and (max-width: 1024px){
    /* styles */
}
```

Media Queries



```css
/* Tablets, iPads (portrait) ---------- */
@media screen and (min-width: 768px){
    /* styles */
}

/* Tablets, iPads (landscape) ---------- */
@media screen and (min-width: 1024px){
    /* styles */
}

/* Desktops and laptops ---------- */
@media screen and (min-width: 1224px){
    /* styles */
}

/* Large screens ---------- */
@media screen and (min-width: 1824px){
    /* styles */
}
```

Media Queries - Example



```css
.container {
    margin: 0 auto;
    background: #D0021B;
    box-sizing: border-box;
}

/* Mobile phones (portrait and landscape) ---------- */
@media screen and (max-width: 767px){
    .container {
        width: 100%;
        padding: 0 10px;
    }
}

/* Tablets and iPads (portrait and landscape) ---------- */
@media screen and (min-width: 768px){
    .container {
        width: 750px;
        padding: 0 10px;
    }
}
```

References

- https://www.w3schools.com/css/
- https://www.tutorialspoint.com/css/index.htm