



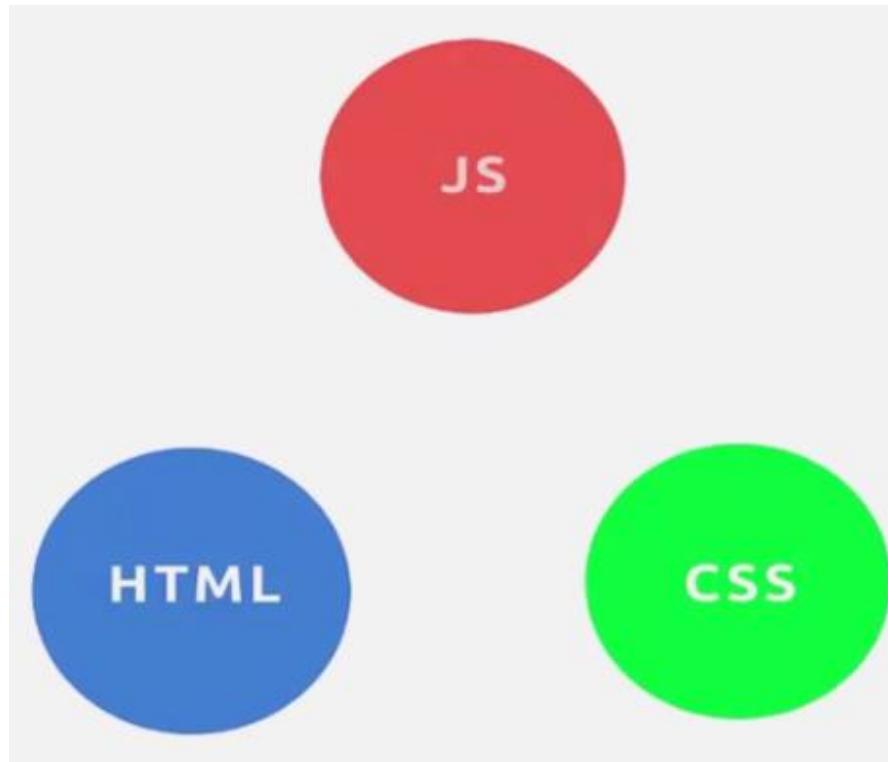
Download a text editor

- | | |
|----------------------|----------------------|
| 1 Sublime Text | Windows, OS X, Linux |
| 2 Visual Studio Code | Windows, OS X, Linux |
| 3 Atom | Windows, OS X, Linux |
| 4 Notepad ++ | Windows |

Download: <https://www.sublimetext.com/3>



Download: <https://www.google.com/chrome/>



Download: <https://www.google.com/chrome/>



CSS = Cascading Style Sheets

CSS: Why Do We Need It?



css ref pages: <http://www.csszengarden.com/>



A demonstration of what can be accomplished through [CSS](#)-based design. Select any style sheet from the list to load it into this page.

Download the example [HTML FILE](#) and [CSS FILE](#)

THE ROAD TO ENLIGHTENMENT

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible [DOMs](#), broken [CSS](#) support, and abandoned browsers.

We must clear the mind of the past. Web enlightenment has been achieved thanks to the tireless efforts of folk like the [W3C](#), [WASP](#), and the major browser creators.

The CSS Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the time-honored techniques in new and invigorating fashion. Become one with the web.

SO WHAT IS THIS ABOUT?

There is a continuing need to show the power of [CSS](#). The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The [HTML](#) remains the same, the only thing that has changed is the external [CSS](#) file. Yes, really.

[CSS](#) allows complete and total control over the style of a hypertext document. The only way this can be illustrated in a way that gets people excited is by demonstrating what it can truly be, once the reins are placed in the hands of those able to create beauty from structure. Designers and coders alike have contributed to the beauty of the web; we can always push it further.

MID CENTURY MODERN

by Andrew Lohman

GARMENTS

by Dan Mall

STEEL

by Steffen Knoeller

APOTHECARY

by Trent Walton

SCREEN FILLER

by Elliot Jay Stocks

FOUNTAIN KISS

by Jeremy Carlson

A ROBOT NAMED JIMMY

by meltmedia

css ref pages: <http://www.csszengarden.com/>



css Zen Garden

The Beauty of CSS Design

A demonstration of what can be accomplished through [CSS-based design](#). Select any style sheet from the list to load it into this page.

Download the [example html file](#) and [css file](#)

The Road to Enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible [DOMs](#), broken [CSS support](#), and abandoned browsers.

We must clear the mind of the past. Web enlightenment has been achieved thanks to the tireless efforts of folk like the [W3C](#), [WaSP](#), and the major browser creators.

css ref pages: <http://www.csszengarden.com/>



MADE LOCALLY
CSS ZEN GARMENTS
IMPECCABLE
QUALITY

A DEMONSTRATION OF WHAT CAN BE ACCOMPLISHED THROUGH `css`-BASED DESIGN. SELECT ANY STYLE SHEET FROM THE LIST TO LOAD IT INTO THIS PAGE.

↓ ⚡ <> ☺

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible `pxms`, broken `css` support, and abandoned browsers. We must clear the mind of the past. Web enlightenment has been achieved thanks to the tireless efforts of folk like the `W3C`, `WHATWG`, and the major browser creators. The CSS Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the time-honored techniques in new and invigorating fashion. Become one with the web.

There is a continuing need to show the power of `css`. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The `HTML` remains the same, the only thing that has changed is the external `css` file. Yes, really. `css` allows complete and total control over the style of a hypertext document. The only way this can be illustrated in a way that gets people excited is by demonstrating what it can truly be; once the reins are placed in the hands of those able to create beauty from structure. Designers and coders alike have contributed to the beauty of the web; we can always push it further.

Strong visual design has always been our focus. You are modifying this page, so strong `css` skills are necessary too, but the example files are commented well enough that even `css` novices can use them as starting points. Please see the [css Resource Guide](#) for advanced tutorials and tips on working with `css`.

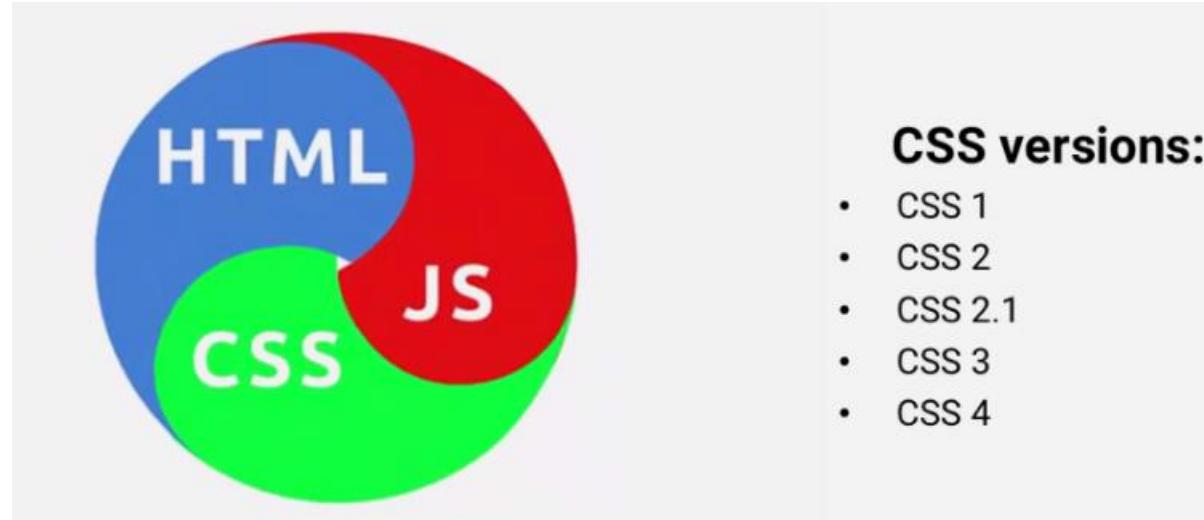
You may modify the style sheet in any way you wish, but not the `HTML`. This may seem daunting at first if you've never worked this way before, but follow the listed links to learn more, and use the sample files as a guide.

A FASHION-FORWARD FUTURE

SEE YOURSELF IN A DIFFERENT WAY

GET INTO A BRAND NEW PAIR

css ref pages: <http://www.csszengarden.com/>



CSS versions:

- CSS 1
- CSS 2
- CSS 2.1
- CSS 3
- CSS 4

css ref pages: <http://www.csszengarden.com/>

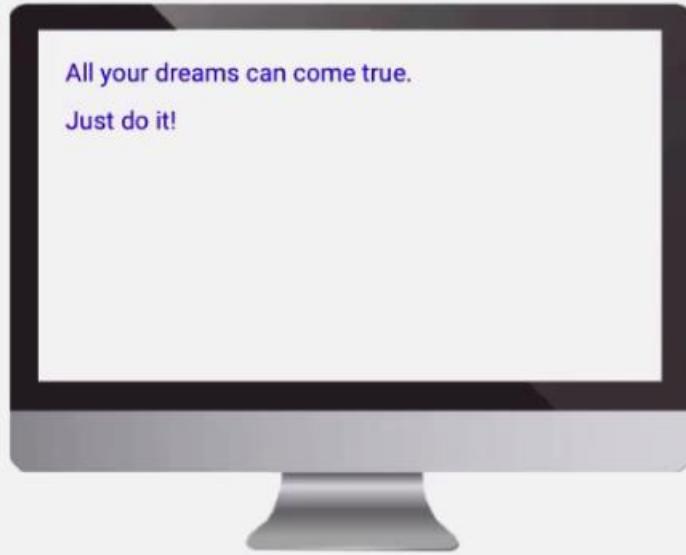


CSS Properties

Property	What it does	Value examples
color	Sets the color of an element	blue red green yellow
font-family	Sets a font-family for text	Arial “Times New Roman” Georgia

Make STYLE your HTML tags

1



HTML file

```
<body>
  <p style="color:blue;">All your dreams can come true.</p>
  <p style="color:blue;">Just do it!</p>
</body>
```

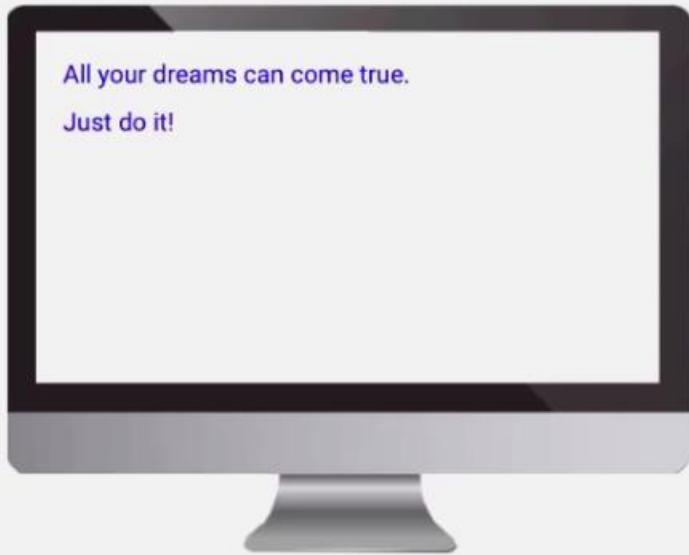
style = "property:value;property:value;..."

Attribute

Value

INLINE CSS

2



HTML file

```
<head>
  <meta charset="utf-8">
  <title> CSS </title>
  <style>
    p {
      color: blue;
    }
  </style>
</head>

<body>
  <p>All your dreams can come true.</p>
  <p>Just do it!</p>
</body>
```

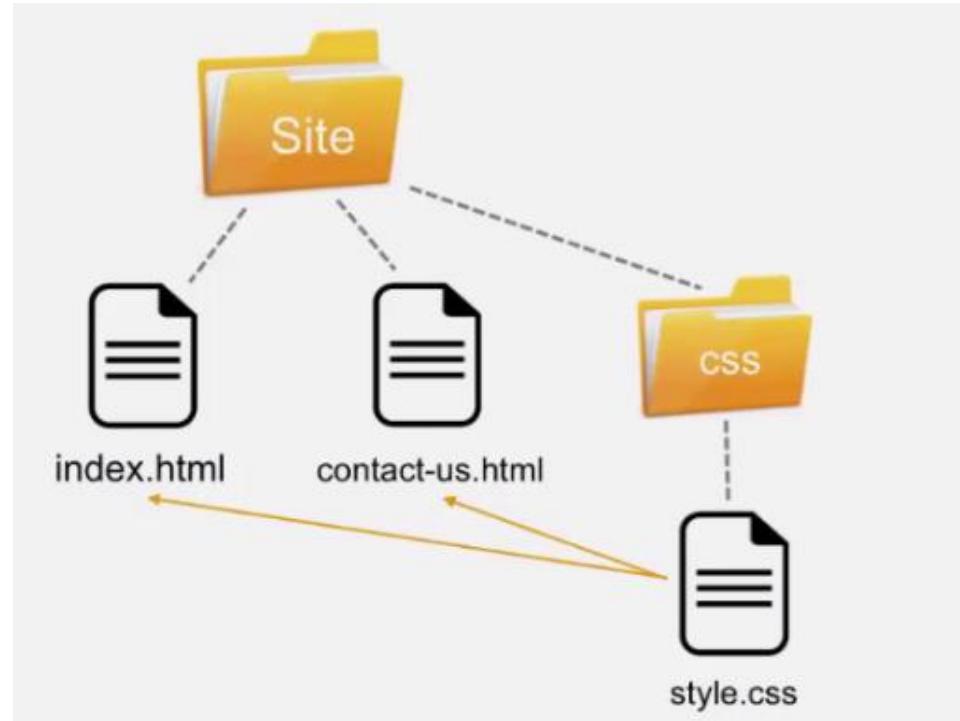
INTERNAL CSS



```
<style>
  selector
    P {
      font-family: Arial;
      color: yellow;
      property   value
    }
</style>
```

Internal CSS

Put in head tag



EXTERNAL CSS

3



HTML file

```
<head>
  <meta charset="utf-8">
  <title> CSS </title>
  <link rel="stylesheet" href="css/style.css">
</head>
<body>
  <p>All your dreams can come true.</p>
  <p>Just do it!</p>
</body>
```

CSS file

```
p {
  color: blue;
}
```

External CSS

Put in head tag



1

HTML file

```
<p style="color:blue;">All your dreams can come true.</p>
<p style="color:blue;">Just do it!</p>
```

2

HTML file

```
<head>
  <meta charset="utf-8">
  <title> CSS </title>
  <style>
    p {
      color: blue;
    }
  </style>
</head>
<body>
  <p >All your dreams can come true.</p>
  <p >Just do it!</p>
</body>
```

3

HTML file

```
<head>
  <meta charset="utf-8">
  <title> CSS </title>
  <link rel="stylesheet" href="css/style.css">
</head>

<body>
  <p >All your dreams can come true.</p>
  <p >Just do it!</p>
</body>
```

CSS file

```
p {
  color: blue;
}
```



Inline

Internal

External



4 main font families

4 main font families

Strokes with serifs

Serif



Example:

Vertex (Times New Roman)
Vertex (Georgia)
Vertex (Cambria)

Straight strokes.

Sans-Serif



Example:

Vertex (Tahoma)
Vertex (Arial)
Vertex (Trebuchet MS)

Resembles handwriting

Handwritten



Example:

Vertex (Monotype corsiva)
Vertex (Magneto)
Vertex (Vivaldi)

Fancy fonts

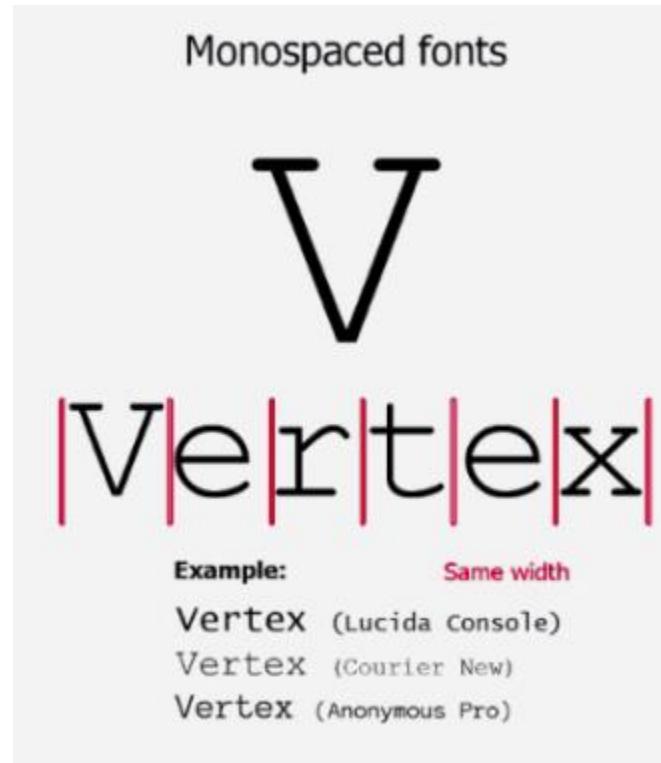
Decorative



Example:

VERTEX (Goudy Stout)
Vertex (Snap ITC)
Vertex (Old English Text MT)

<https://blackwoodcreative.com/the-4-main-font-families/>



<https://blackwoodcreative.com/the-4-main-font-families/>



Windows 10

- Calibri
- Cambria
- Comic Sans



- Dingbats
- FreeMono
- Ubuntu



Mac OS

- Helvetica
- Gill Sans
- Lucida Grande



Web Safe Fonts

1	Arial	Text with this kind of font
2	Arial Black	Text with this kind of font
3	Comic Sans MS	Text with this kind of font
4	Courier New	Text with this kind of font
5	Georgia	Text with this kind of font
6	Impact	Text with this kind of font
7	Times New Roman	Text with this kind of font
8	Trebuchet MS	Text with this kind of font
9	Verdana	Text with this kind of font

https://www.w3schools.com/cssref/css_websafe_fonts.asp



Fallback

https://www.w3schools.com/cssref/css_websafe_fonts.asp

Fallback

`font-family: Tahoma, Geneva, sans-serif;`

`font-family: font1,font2,...;`

https://www.w3schools.com/cssref/css_websafe_fonts.asp

Eurocine AaBbCcDdEeFfGgHh

Font Squirrel > Eurocine > Paulo Goode > 2 Styles

[DOWNLOAD OTF \(OFFSITE\)](#)

INTRO RUST ABCDEFGHIJKLMNOP

Font Squirrel > Intro Rust > Fontfabric > 3 Styles

[DOWNLOAD OTF \(OFFSITE\)](#)

Source Sans Pro AaBbCcDdEeFfGgHh

Font Squirrel > Source Sans Pro > Adobe > 12 Styles

[DOWNLOAD OTF](#)

Open Sans AaBbCcDdEeFfGgHhIijj

Font Squirrel > Open Sans > Ascender Fonts > 10 Styles

[DOWNLOAD TTF](#)

<https://www.fontsquirrel.com/>

<https://www.fontsquirrel.com/fonts/list/tag/calligraphic>

<https://www.fontsquirrel.com/fonts/Chantelli-Antiqua>



woff

ttf/otf

<https://www.fontsquirrel.com/>

<https://www.fontsquirrel.com/fonts/list/tag/calligraphic>

<https://www.fontsquirrel.com/fonts/Chantelli-Antiqua>



Browser Compatibility

Format	Name	Browser					
		Chrome	Firefox	Safari	Opera	Edge	IE
ttf	truetype	4.0	3.5	3.1	10.0	9.0	9.0 (partly)
otf	opentype	4.0	3.5	3.1	10.0	9.0	9.0 (partly)
woff	woff	5.0	3.6	5.1	11.5	12.0	9.0
eot	embedded-opentype	-	-	-	-	-	6.0

woff

ttf/otf

<https://www.fontsquirrel.com/>

<https://www.fontsquirrel.com/fonts/list/tag/calligraphic>

<https://www.fontsquirrel.com/fonts/Chantelli-Antiqua>



How to Connect a Font



```
@font-face {  
    font-family: "font family name";  
    src: url(path to font) format("name"),  
         url(path to font) format("name");  
}
```

<https://www.fontsquirrel.com/>

<https://www.fontsquirrel.com/fonts/list/tag/calligraphic>

<https://www.fontsquirrel.com/fonts/Chantelli-Antiqua>

Source Sans Pro

[Download family](#)



Select styles

Glyphs

About

License

Pairings

Extra-light 200 italic

Almost before we knew it, we ha + Select this style

Light 300

Almost before we knew it, we h - Remove this style

Light 300 italic

Almost before we knew it, we ha + Select this style

Regular 400

A B C D E F G H I J K L M N O P Q R S T U V W Z + Select this style

Selected family



Review

Source Sans Pro

Light 300

Add more styles Remove all

Header of your html

<link> @import

```
<link rel="preconnect" href="https://fonts.gstatic.com">
<link href="https://fonts.googleapis.com/css2?family=Source+Sans+Pro:wght@300&display=swap" rel="stylesheet">
```

CSS rules to specify families

[API docs](#)

[Download all](#)

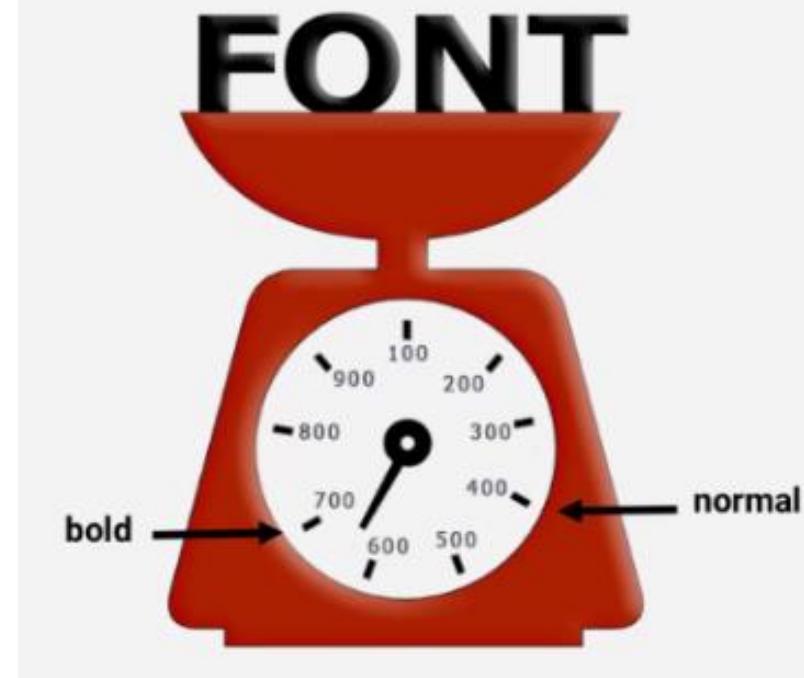
<https://fonts.google.com/>



Everest is **the highest** mountain on Earth

P tag

font-weight



Font-family: Arial

Font-weight: bold



font-style

normal	font-style:normal
italic	<i>font-style:italic</i>
oblique	<i>font-style:oblique</i>

Font-family: Arial

Font-weight: 500

Font-size: 30

Font-style: italic



text-decoration	
underline	<u>underline</u>
overline	overline
line-through	line-through
none	none

Font-family: Arial

Font-weight: 500

Font-size: 30

Font-style: italic

Text-decoration: underline



font-weight: 100...900
bold
normal

text-decoration: underline
overline
line-through
none

font-style: normal
italic
oblique

Font-family: Arial
Font-weight: 500
Font-size: 30
Font-style: italic
Text-decoration: underline



text-transform

capitalize	First Letter Of Each Word Is Capitalized
lowercase	all letters are lowercase
uppercase	ALL LETTERS ARE UPPERCASE
none	Cancels any text transformations

<p>Coding is fun :'(</p>



TEXT

The highest mountain on Earth

letter-spacing

word-spacing

<p>Coding is fun :(</p>

Text-transformation: upper-case

word-spacing: 15px;

letter-spacing: normal;

text-align



left



center

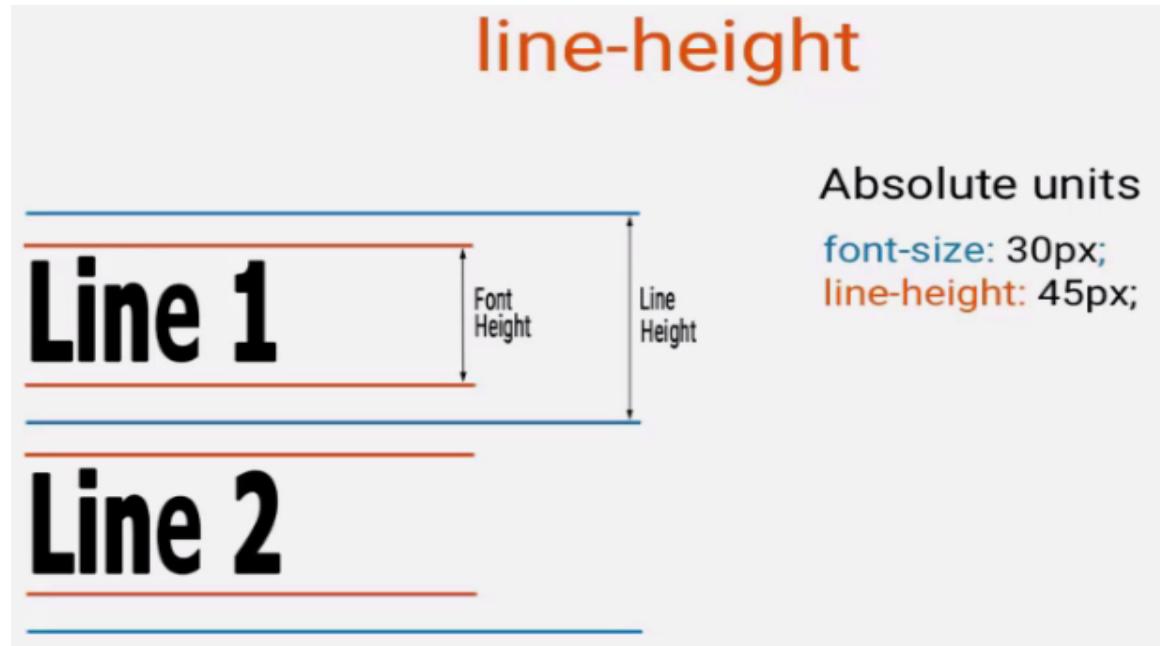


right



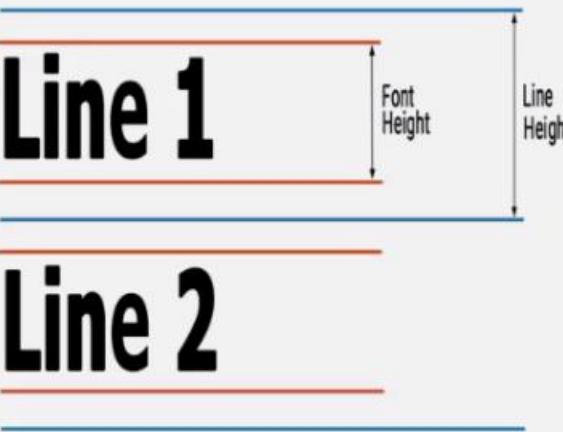
justify

P tag with lemonade



P tag with lemonade
Text-align: justify
Font-size: 30px
Line-height: 45px

line-height



Absolute units
`font-size: 30px;
line-height: 45px;`

Multiplier
`font-size: 30px;
line-height: 1.5;`

Percentage
`font-size: 30px;
line-height: 150%;`

Best Practice
Put line-height equals with height
of parent element to make text
vertical center

P tag with lemonade
Text-align: justify
Font-size: 30px
Line-height: 45px



1

Color Name

147 colors

red blue yellow black grey green orange lavender darkolivegreen ...

https://www.w3schools.com/colors/colors_picker.asp

https://www.w3schools.com/colors/colors_names.asp

P Have a nice day

Font-size: 30px

Color: yellow

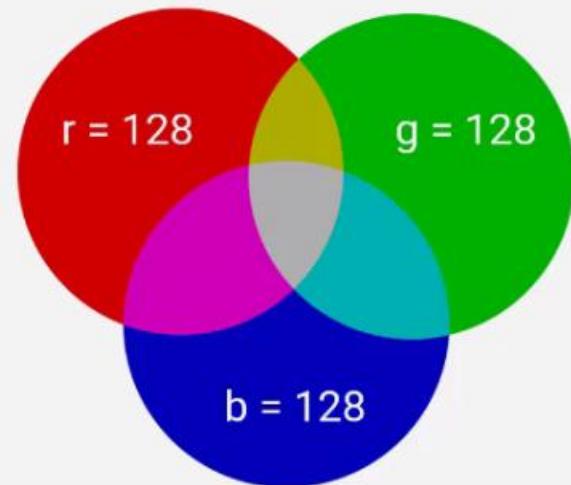
2

RGB = Red Green Blue

0 – 255

0 – 255

0 – 255



Gray

rgb(128, 128, 128)

↑
50% 50% 50%

https://www.w3schools.com/colors/colors_picker.asp

https://www.w3schools.com/colors/colors_names.asp

P Have a nice day

Font-size: 30px

Color: rgb(128,128, 128)



3

Hexadecimal (HEX)

#RedGreenBlue

0 – FF 0 – FF 0 – FF

White

rgb(255, 255, 255)

#FFFFFF

Black

rgb(0, 0, 0)

#000000

Grey

rgb(128, 128, 128)

#808080

https://www.w3schools.com/colors/colors_picker.asp

https://www.w3schools.com/colors/colors_names.asp

P Have a nice day

Font-size: 30px

Color: #FFFFFF

Convert: <http://www.javascripter.net/faq/rgbtohex.htm>



list-style-type

list-style-image

list-style-position



list-style-type

Favorite sports:

- Football
- Baseball
- Hockey

Summer trip:

1. Paris
2. Rome
3. Sri Lanka

list-style-type

Favorite sports:

- Football
- Baseball
- Hockey

● disc

○ circle

■ square

none



list-style-type

Summer trip:

1. Paris
2. Rome
3. Sri Lanka

Value	Result
decimal	1 2 3 4 5
decimal-leading-zero	01 02 03 04
lower-alpha	a b c
upper-alpha	A B C D E
lower-roman	i. ii. iii. iv. v.
upper-roman	I II III IV V

list-style-image



```
ul {  
    list-style-image: url(world.png);  
}
```

list-style-image



Summer trip:

- Paris
- Rome
- Sri Lanka



list-style-position

outside

- When in Rome, do as Romans do
- A picture is worth a thousand words
- All things are difficult before they are easy

inside

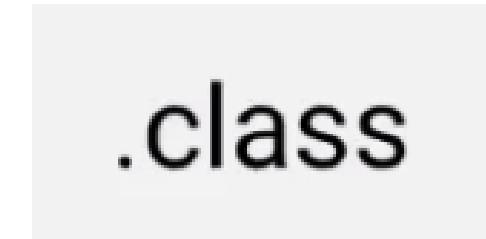
- When in Rome, do as Romans do
- A picture is worth a thousand words
- All things are difficult before they are easy

H4 Favorite sports ul Football, Hockey, Tennis

H4 Summer trip ol Para, Rome, Sri Lanka

```
ul {  
    type: none; image=world, position=inside  
}
```

<https://www.iconfinder.com/search/?q=world&price=free>



Definition and Usage

The `.class` selector selects elements with a specific class attribute.

To select elements with a specific class, write a period (.) character, followed by the name of the class.

```
<p class="intro">My best friend is Mickey.</p>
```

<https://lipsum.com/>

HTML elements can also refer to more than one class (look at Example 2 below).

This `<p>` element will be styled according to `class="center"` AND to `class="large"`:

```
<p class="center large">This paragraph refers to two classes.</p>
```



```
<h1>
  LOREM IPSUM
</h1>
<p class="special">
  Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
</p>
<p>
  Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
</p>
```

```
h1 {
  color: blue;
  text-transform: uppercase;
}

p {
  color: orange;
  font-style: italic;
}

.special {
  color: black;
  font-weight: bold;
}
```

<https://lipsum.com/>

```
<ul>  <li class="change">
      • Lorem
    </li>
    <li class="change">
      • Lorem
    </li>
    <li>
      • Lorem
    </li>
</ul>
<ol>
  <li class="change">
    1. Lorem
  </li>
  <li>
    2. Lorem
  </li>
  <li>
    3. Lorem
  </li>
</ol>
```

```
li {
  color: green;
}

.change{
  color: orange;
  font-weight: bold
}
```

<https://lipsum.com/>

H1 with 2 P

Class bold-text, green-text



```
<h1>
  LOREM IPSUM
</h1>
<p id="special">
  Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
</p>
<p>
  Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
</p>
```



```
h1 {
  color: blue;
  text-transform: uppercase;
}

p {
  color: orange;
  font-style: italic;
}

#special {
  color: black;
  font-weight: bold;
}
```

<https://lipsum.com/>

H1 with 2 P

bold with h1 main-heading

A href block – Back to top page



float: left
right
none

clear: both
left
right
none



The float Property

The `float` property is used for positioning and formatting content e.g. let an image float left to the text in a container.

The `float` property can have one of the following values:

- `left` - The element floats to the left of its container
- `right` - The element floats to the right of its container
- `none` - The element does not float (will be displayed just where it occurs in the text). This is default
- `inherit` - The element inherits the float value of its parent



The clear Property

The `clear` property specifies what elements can float beside the cleared element and on which side.

The `clear` property can have one of the following values:

- `none` - Allows floating elements on both sides. This is default
- `left` - No floating elements allowed on the left side
- `right` - No floating elements allowed on the right side
- `both` - No floating elements allowed on either the left or the right side
- `inherit` - The element inherits the clear value of its parent



Normal flow

HTML

```
<div> </div>
<p> Lorem ipsum dolor sit amet,
consectetur adipisicing elit.
</p>
<p> Lorem ipsum dolor sit amet,
consectetur adipisicing elit.
</p>
```

CSS

```
div{
background-color: orange;
height: 200px;
width: 20%;}
```



Normal flow

HTML

```
<div> </div>
<p> Lorem ipsum dolor sit amet,
consectetur adipisicing elit.
</p>
<p> Lorem ipsum dolor sit amet,
consectetur adipisicing elit.
</p>
```

CSS

```
div{
background-color: orange;
height: 200px;
width: 20%;
}
```



float

HTML

```
<div> </div>
<p> Lorem ipsum dolor sit amet,
consectetur adipisicing elit.
</p>
<p> Lorem ipsum dolor sit amet,
consectetur adipisicing elit.
</p>
```

CSS

```
div{
background-color: orange;
height: 200px;
width: 20%;
float: right;
}
```





Normal flow

HTML

```
<div> </div>
<p> Lorem ipsum dolor sit amet, consectetur adipisicing elit.
<p> Lorem ipsum dolor sit amet, consectetur adipisicing elit.
```

CSS

```
div{
    background-color: orange;
    height: 200px;
    width: 20%;}
```



clear: both

```
<div> </div>
<p> Lorem ipsum dolor sit amet, consectetur adipisicing elit.
<p> Lorem ipsum dolor sit amet, consectetur adipisicing elit.
```

```
div{
    background-color: orange;
    height: 200px;
    width: 20%;
    float: left;}
```

The diagram illustrates two scenarios of element rendering on a web page:

Normal flow

HTML

```
<div> </div>
<p> Lorem ipsum dolor sit amet, consectetur
adipisicing elit.
<p> Lorem ipsum dolor sit amet, consectetur
adipisicing elit.
```

CSS

```
div{
background-color: orange;
height: 200px;
width: 20%;}
```

clear: both

HTML

```
<div> </div>
<p> Lorem ipsum dolor sit amet,
consectetur adipisicing elit.
</p>
<p class="clear">
    Lorem ipsum dolor sit amet,
    consectetur adipisicing elit.
</p>
```

CSS

```
div{
background-color: orange;
height: 200px;
width: 20%;
float: left;}
.clear{
clear: both;}
```

Image with 2 P



A screenshot of the Exoticar website's header. It features a dark grey background with a red border around the main content area. On the left is the 'EXOTiCAR' logo. To the right are five menu items: HOME, ABOUT, SERVICES, INVESTORS, and CONTACT. The 'SERVICES' and 'INVESTORS' items are connected by a thin red horizontal line.

Height: 66px
Background: #2b2b2b

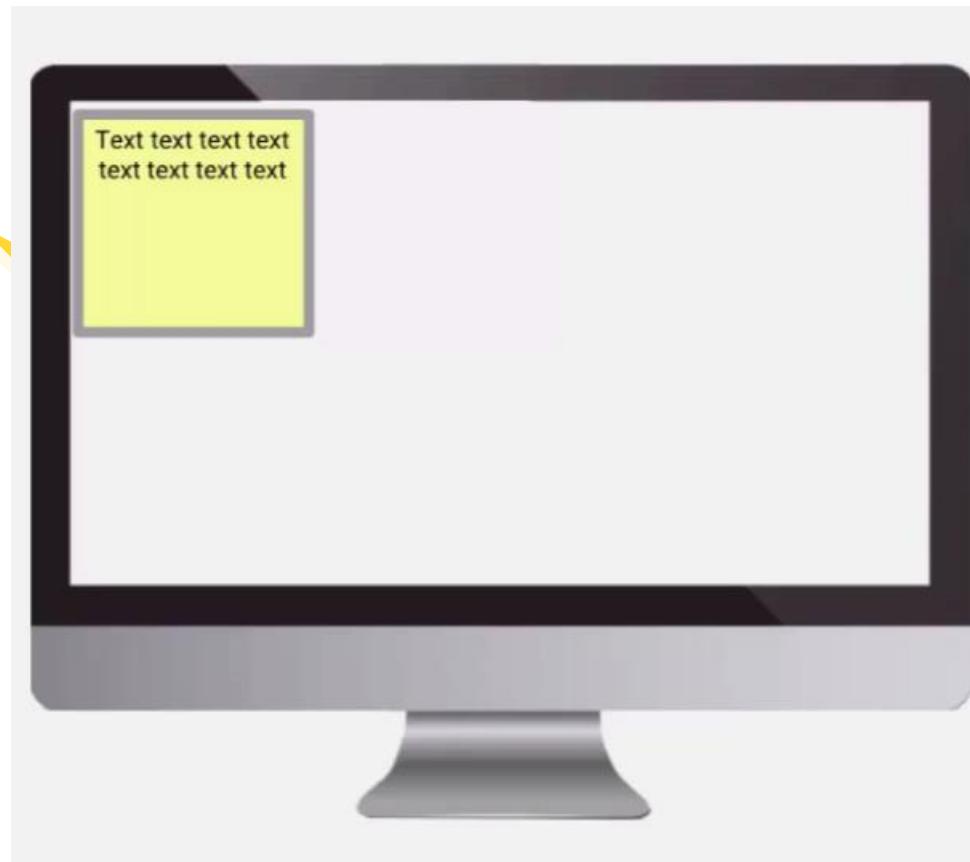
Font-size: 32px – 2rem
Font-family: corbel

Margin: 44px

Demo Float

- 
- 
1. border
 2. padding
 3. margin

- 
- 
- 1. border
 - border-width
 - border-color
 - border-style
 - 2. padding
 - 3. margin



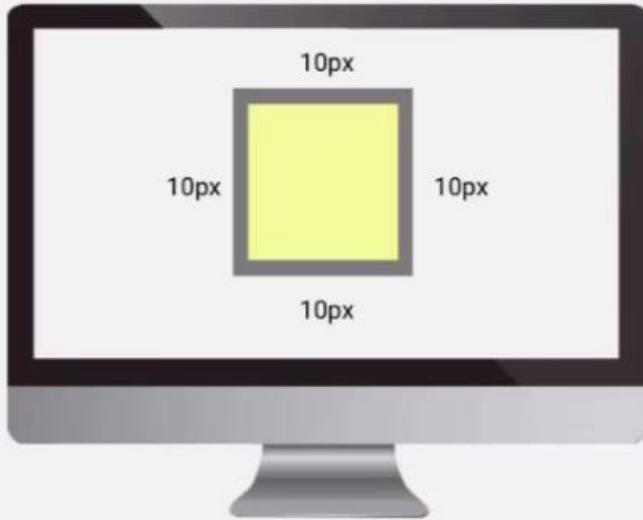
HTML

```
<div class ="a">
  <p>
    Text ... text
  </p>
</div>
```

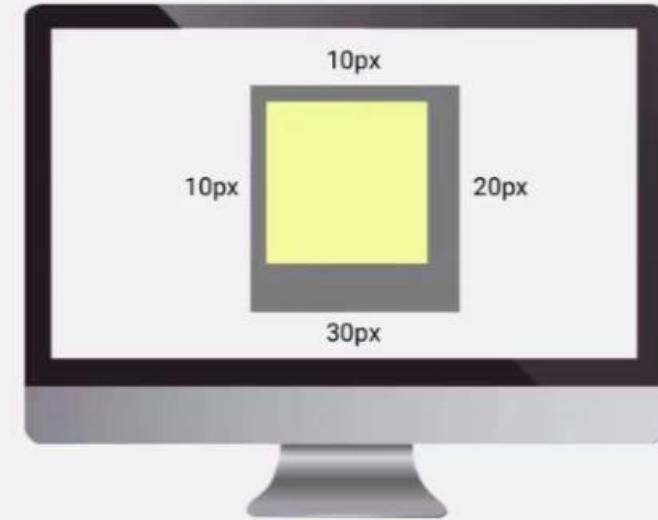
CSS

```
div{
  height: 300px;
  width: 200px;
}
.a {
  background-color: yellow;
  border-width: 10px;
  border-color: grey;
}
```

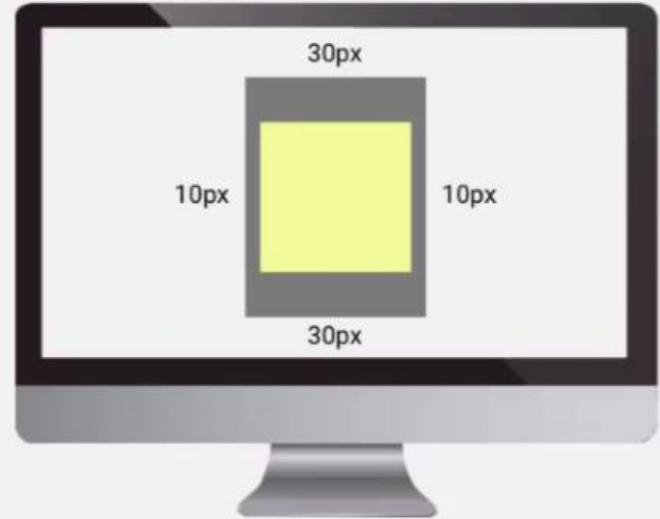
`border-width: 10px;`



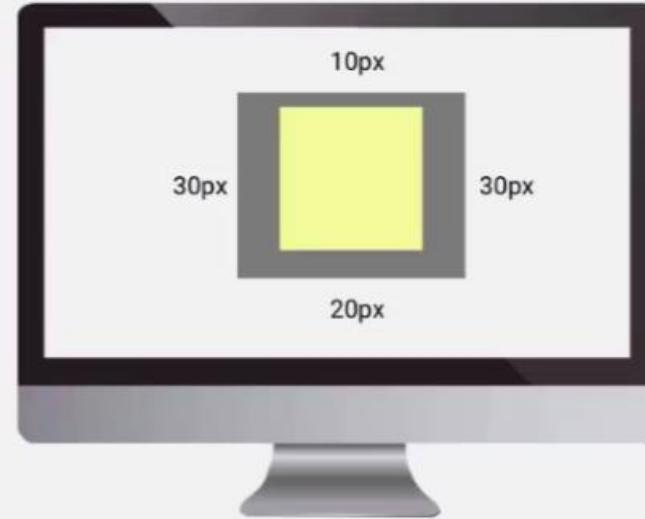
`border-width: 10px 20px 30px 10px;`

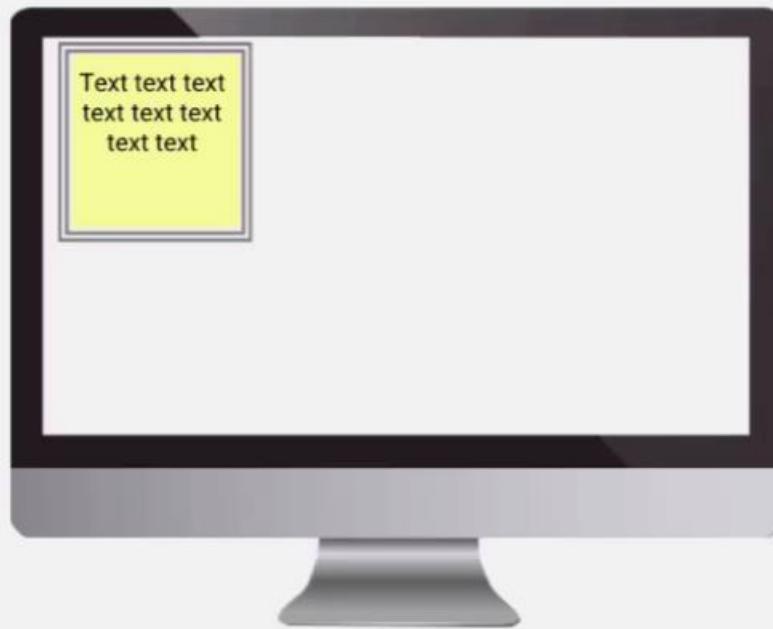
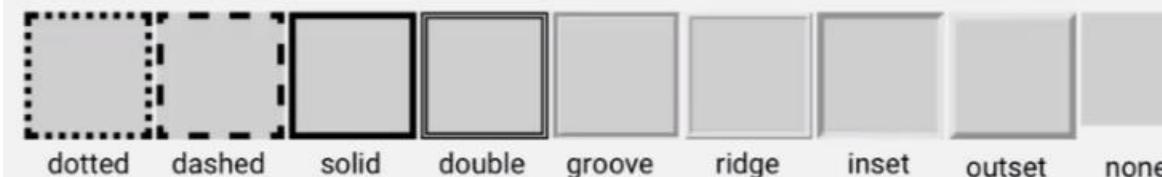


`border-width: 30px 10px;`



`border-width: 10px 30px 20px;`





HTML

```
<div class = "a">
  <p> Text ... text
  </p>
</div>
```

CSS

```
div{
  height: 300px;
  width: 200px;
}
.a {
  background-color: yellow;
  border-width: 10px;
  border-color: grey;
  border-style: double;
}
```



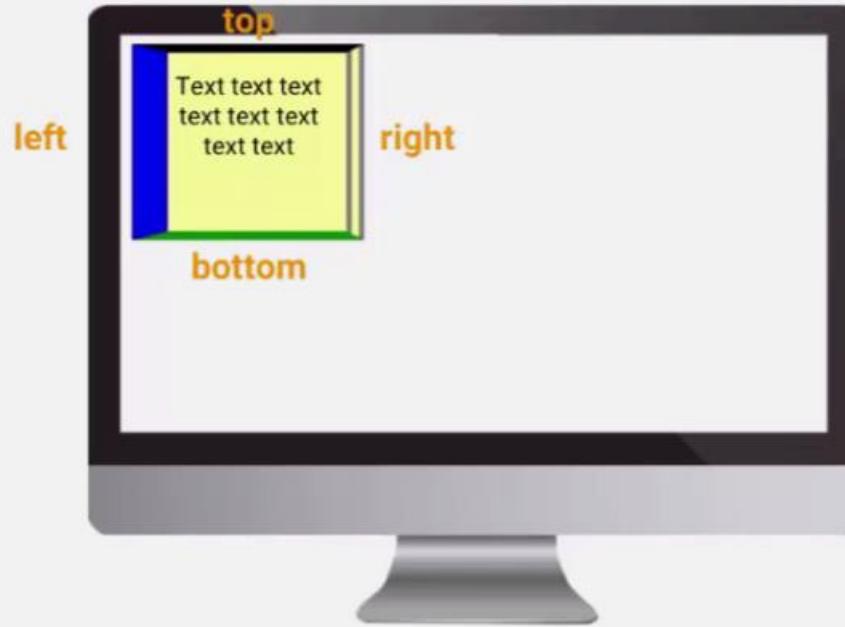
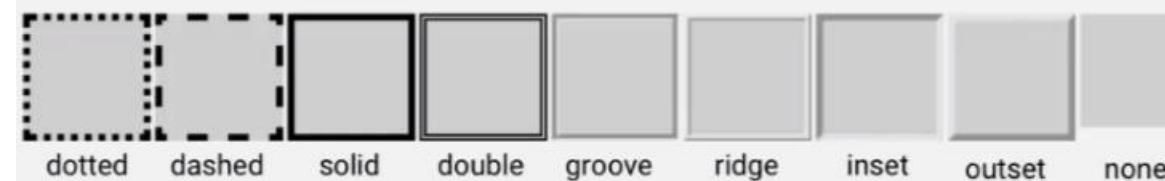
```
.a {  
    border-width: 10px;  
    border-color: grey;  
    border-style: double;  
}
```

Shorthand

```
.a {  
    border: 10px double grey;  
}
```



border-width
border-color
border-style



HTML

```
<div class = "a">
  <p> Text ... text
  </p>
</div>
```

CSS

```
div{
  height: 300px;
  width: 200px;
}
.a {
  background-color: yellow;
  border-top: 5px solid black;
  border-right: 10px double grey;
  border-bottom: 5px solid green;
  border-left: 20px solid blue;
}
```

- 
- 
1. border
 2. padding
 3. margin

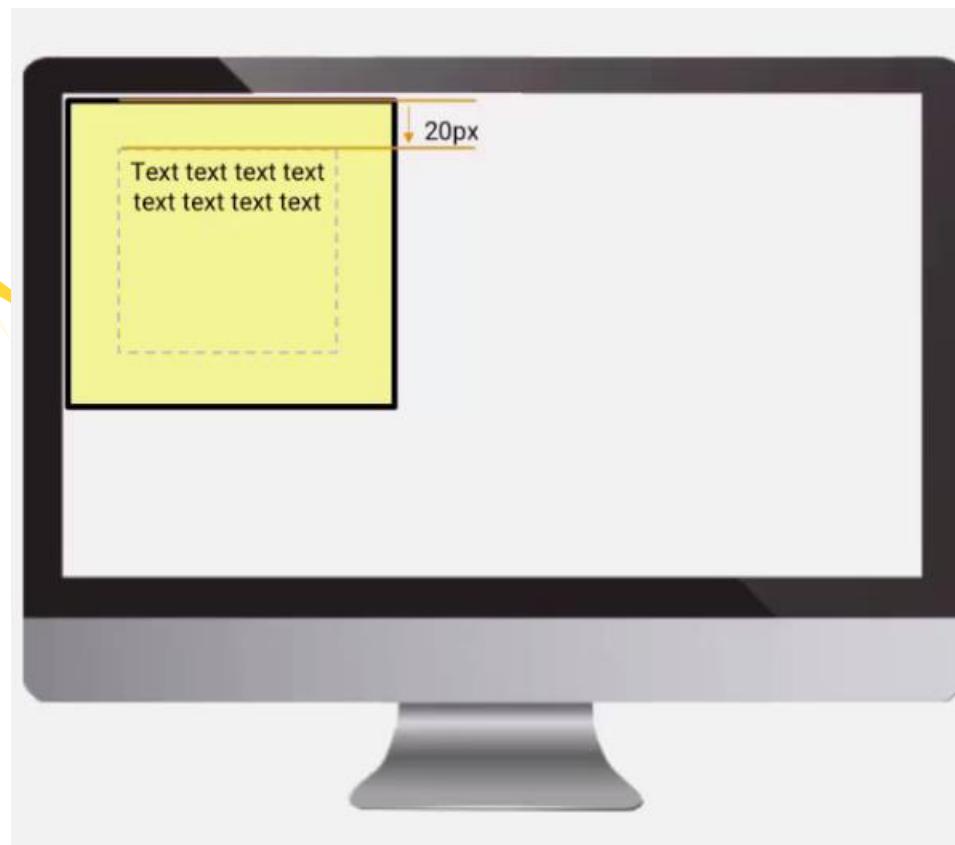


HTML

```
<div class = "a">
  <p>
    Text ... text
  </p>
</div>
```

CSS

```
div{
  height: 300px;
  width: 200px;
}
.a {
  background-color: yellow;
  border: 5px solid black;
}
```



The diagram shows a computer monitor displaying a yellow rectangular box. Inside the box, there is some placeholder text: "Text text text text text text text". A dashed red border surrounds the text area. An orange arrow points from the text area to the right edge of the box, with the value "20px" written next to it, indicating the padding.

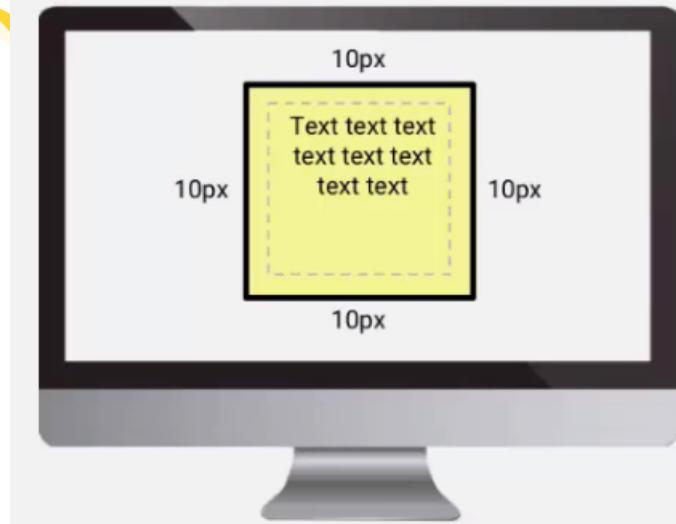
HTML

```
<div class = "a">
  <p>
    Text ... text
  </p>
</div>
```

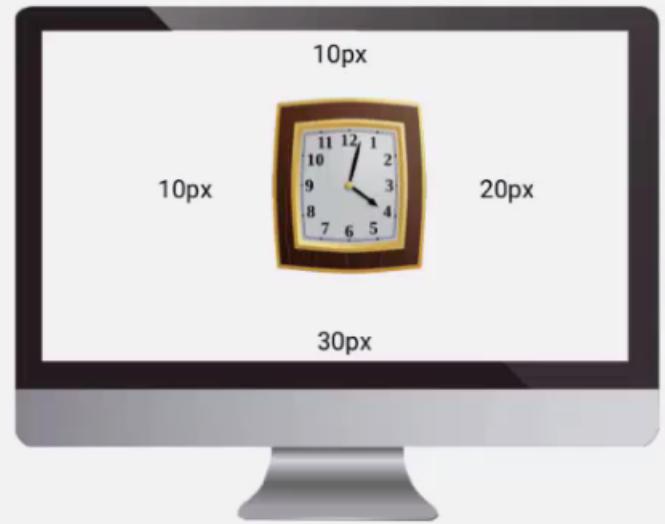
CSS

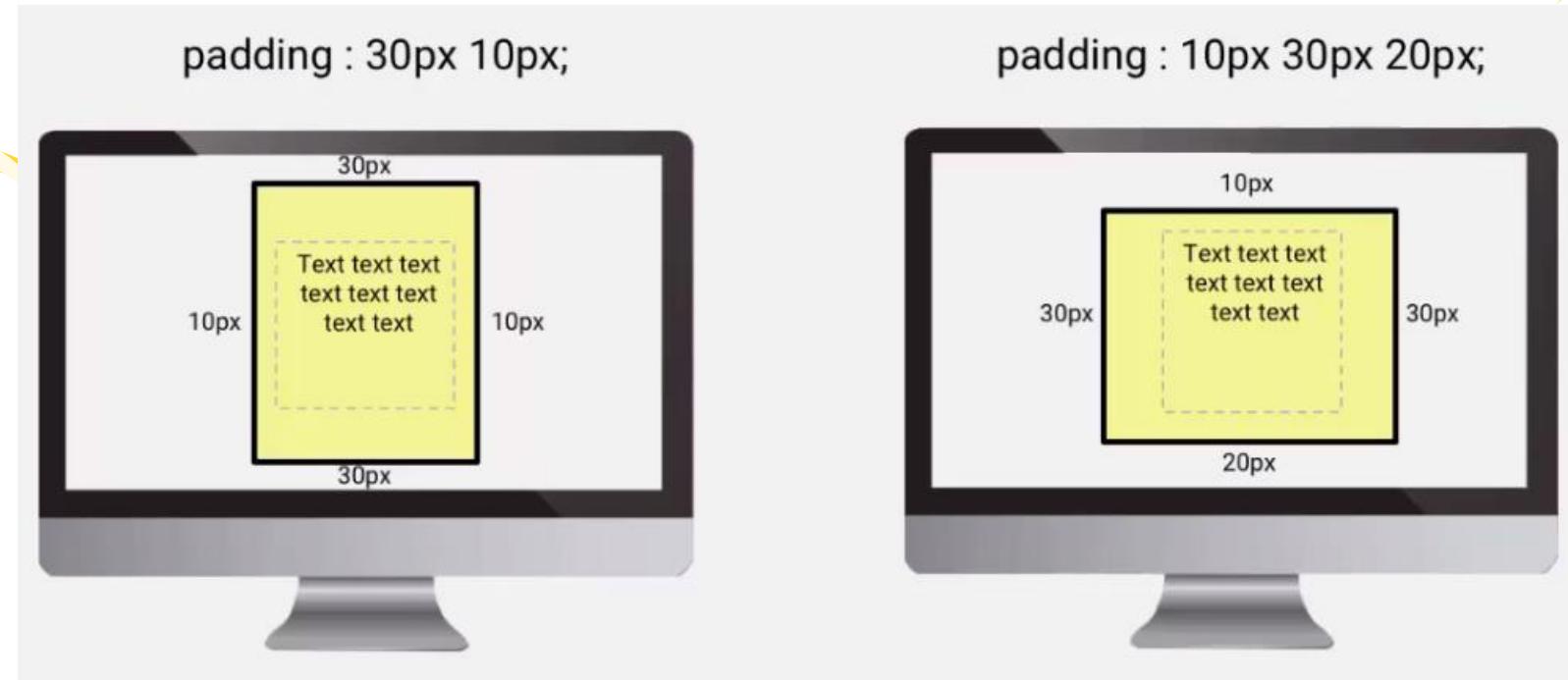
```
div{
  height: 300px;
  width: 200px;
}
.a {
  background-color: yellow;
  border: 5px solid black;
  padding: 20px;
}
```

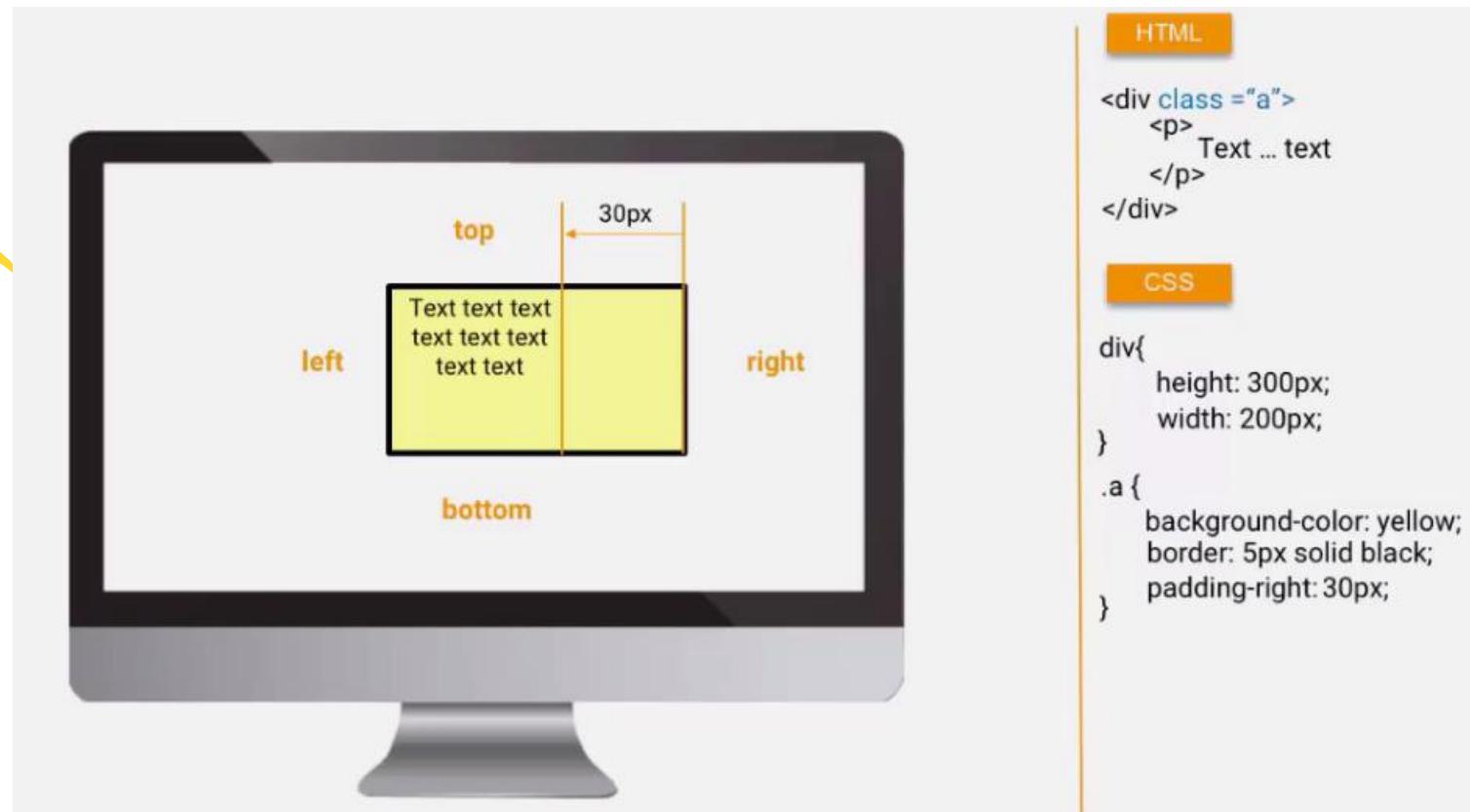
`padding: 10px;`



`padding : 10px 20px 30px 10px;`



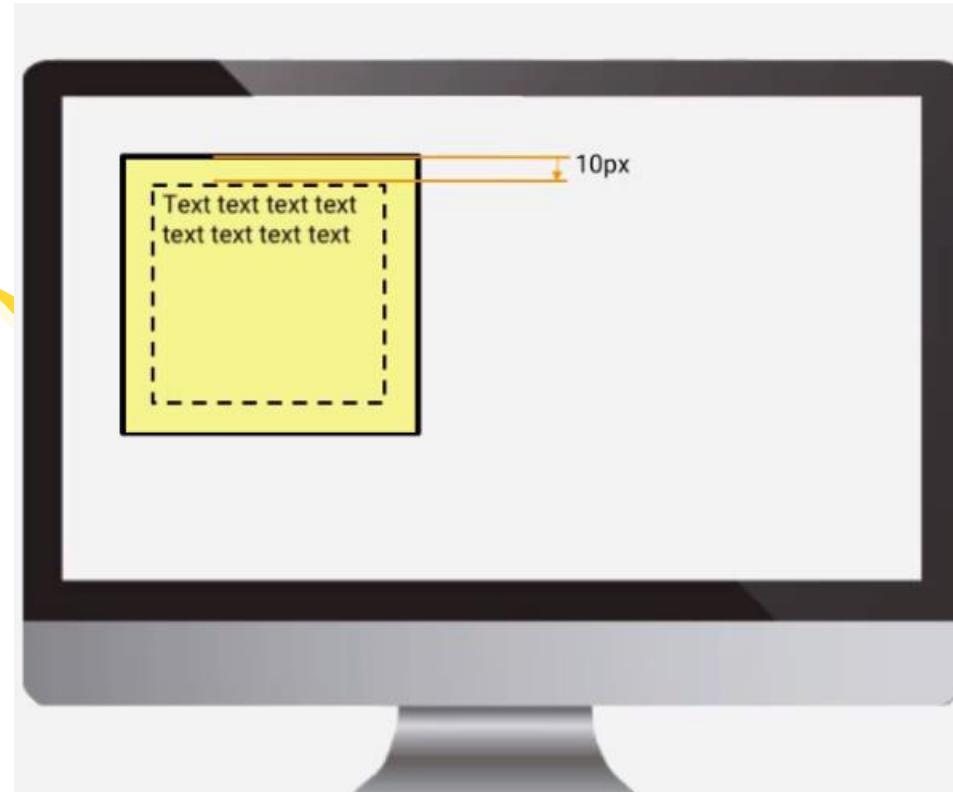




The diagram illustrates the CSS box model on a computer monitor. A central yellow box represents the content area, with a black border around it. The width of the box is 200px and its height is 300px. The text "Text text text text text text" is displayed within the box. A vertical line passes through the center of the box, with arrows pointing to the left and right sides labeled "left" and "right" respectively. At the top of the box, there is a horizontal line with an arrow pointing to the right labeled "top", and at the bottom, another horizontal line with an arrow pointing to the right labeled "bottom". A dimension line on the right side of the box indicates a padding-right of 30px.

```
HTML
<div class = "a">
  <p> Text ... text
  </p>
</div>

CSS
div{
  height: 300px;
  width: 200px;
}
.a {
  background-color: yellow;
  border: 5px solid black;
  padding-right: 30px;
```



HTML

```
<div class = "a">
<p> Text ... text
</p>
</div>
```

CSS

```
div{
    height: 300px;
    width: 200px;
}
.a {
    background-color: yellow;
    border: 5px solid black;
    padding: 10px;
    box-sizing: border-box;
}
```

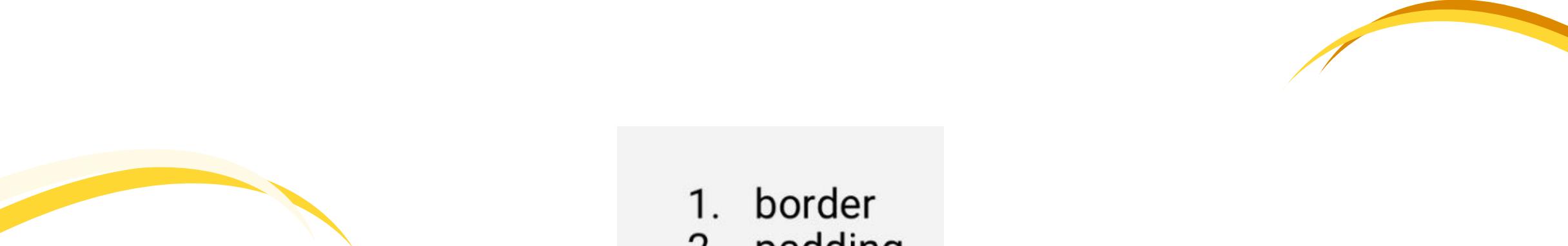
box-sizing: border-box || content-box

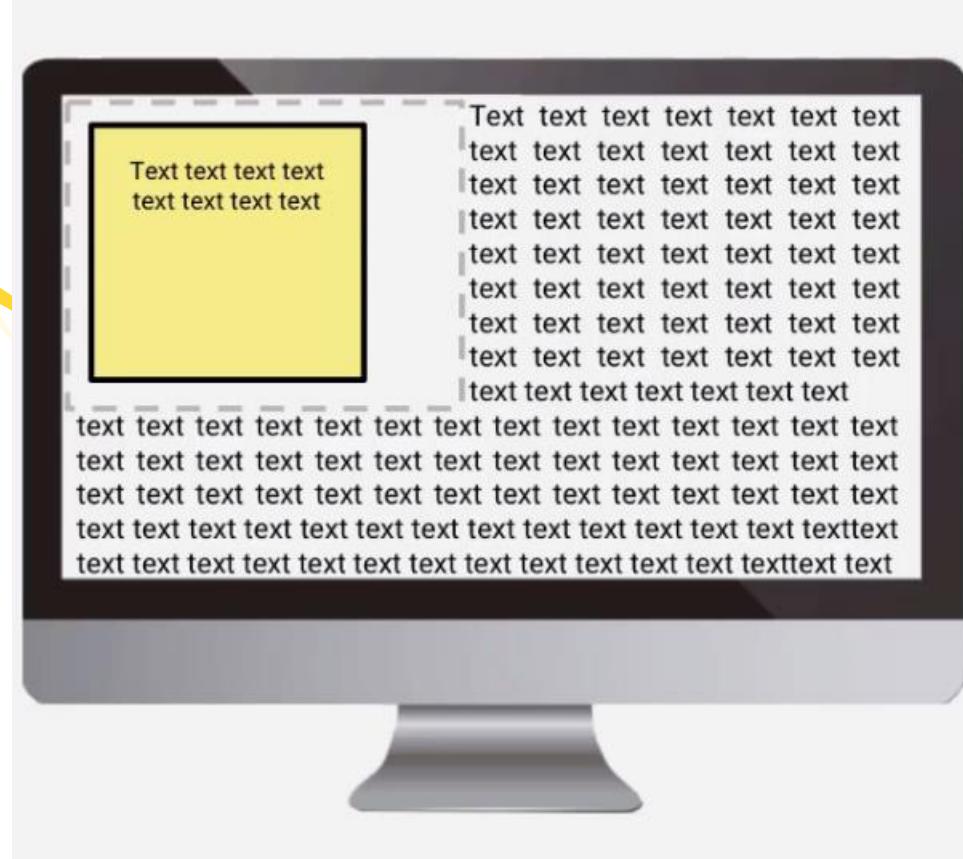


Width: 300, Height: 200

Padding: 10

Padding-left: 40

- 
1. border
 2. padding
 3. margin

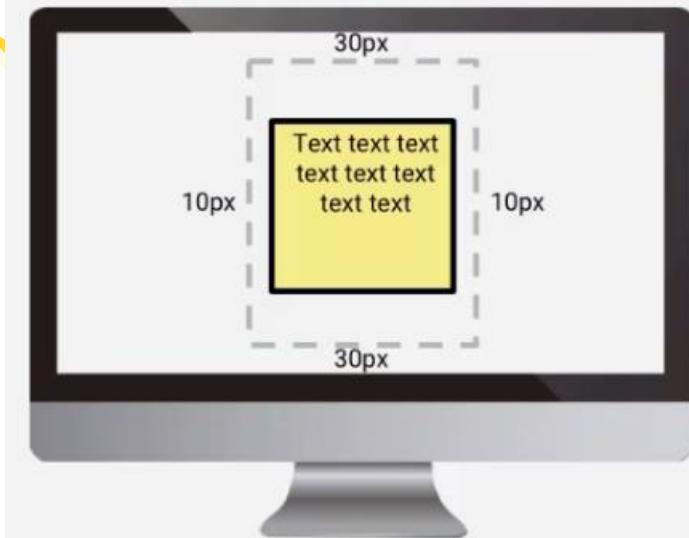


HTML

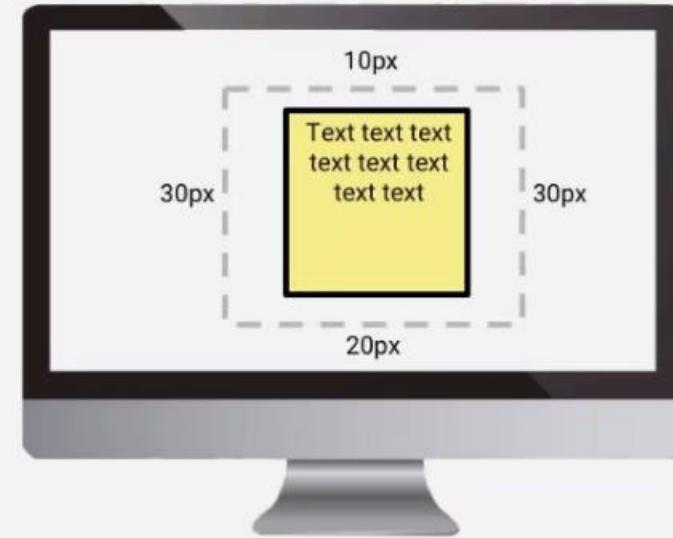
```
<div class="a" >
  <p> Text ... text
  </p>
</div>
<p> Text ... text </p>
```

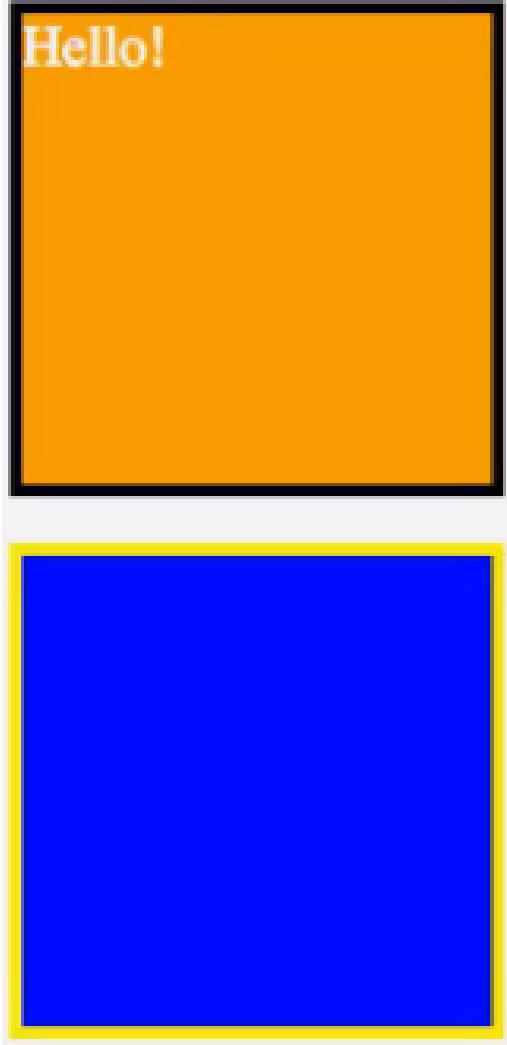
```
div{  
    height: 300px;  
    width: 200px;  
    float: left;  
}  
.a {  
    background-color: yellow;  
    border: 5px solid black;  
    padding: 10px;  
    margin: 10px;  
    margin-right: 30px;  
}
```

`margin: 30px 10px;`



`margin: 10px 30px 20px;`



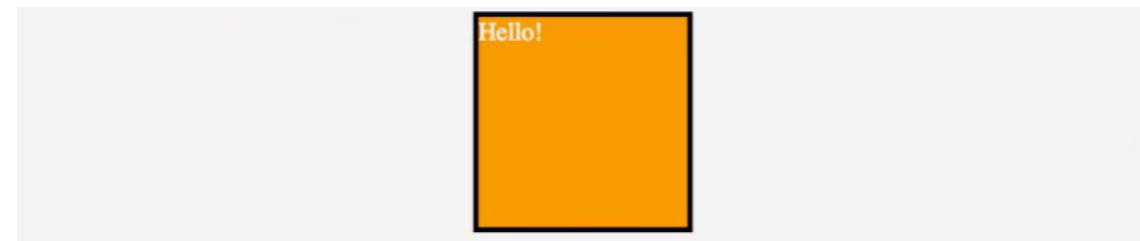


Can apply a negative value

Margin-left: auto

Margin-right: auto

Margin: 0 auto || 50 auto



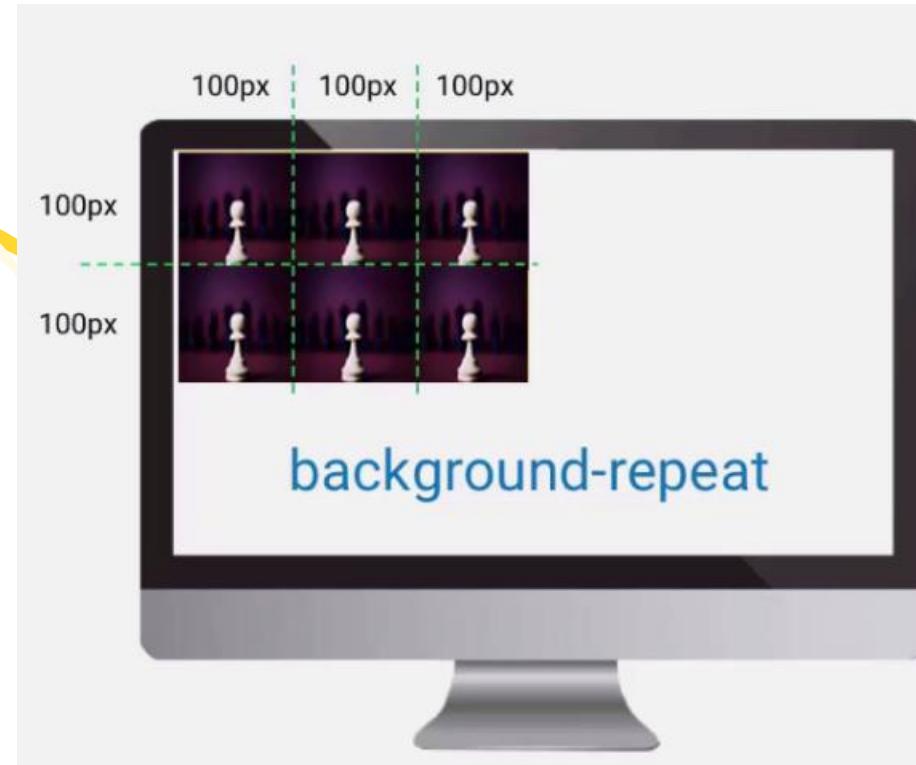
Can apply a negative value

Margin-bottom: Block 1

Marge-top: Block 2

To see what happens

- 
- 
1. background-color
 2. background-image
 3. background-size
 4. background-repeat
 5. background-position
 6. background-attachment

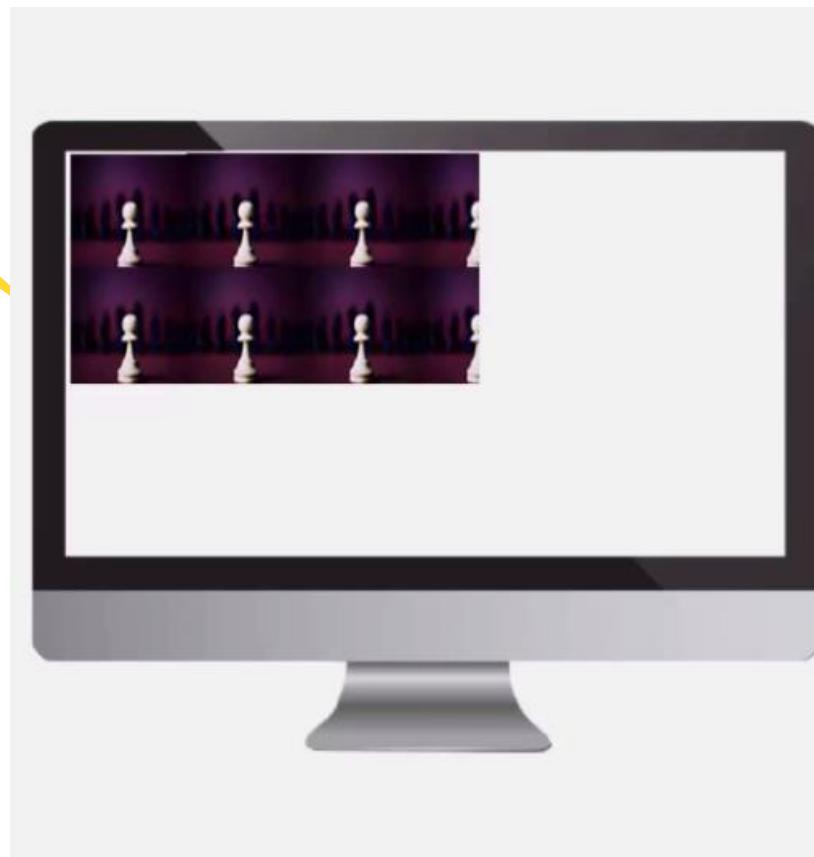


HTML

```
<div class="A" > </div>
```

CSS

```
div{  
    height: 200px;  
    width: 300px;  
}  
.A{  
    background-color: orange;  
    background-image: url(figure.jpg);  
    background-size: 100px 100px;  
}
```



HTML

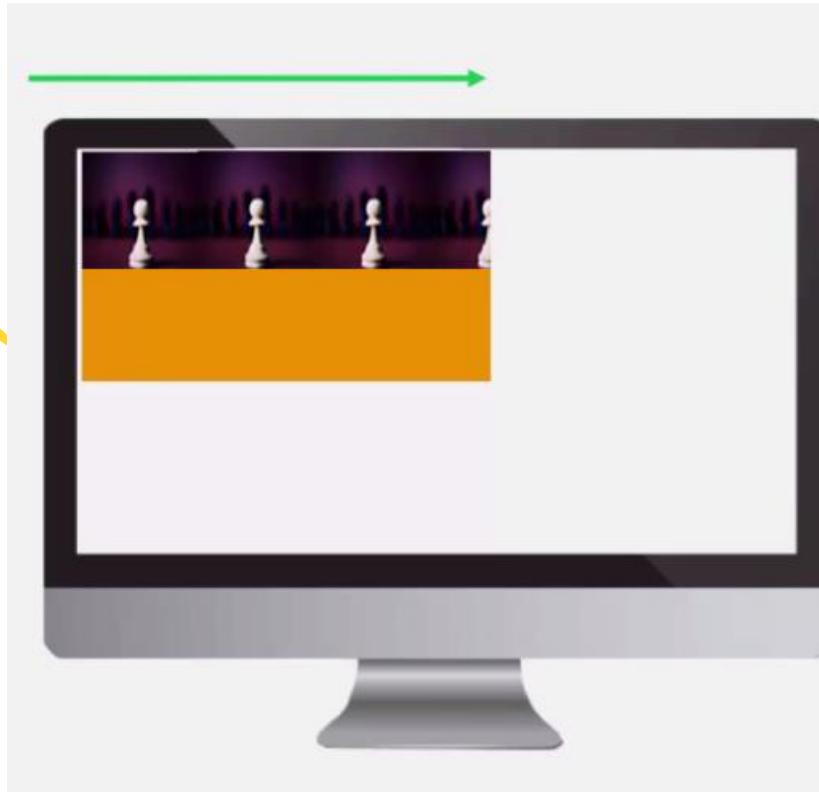
```
<div class="A" > </div>
```

CSS

```
div{  
    height: 200px;  
    width: 350px;  
}  
.A{  
    background-color: orange;  
    background-image: url(figure.jpg);  
    background-size: 100px 100px;  
    background-repeat: no-repeat;  
}
```

Background-repeat

Non-repeat, repeat-x, repeat y



HTML

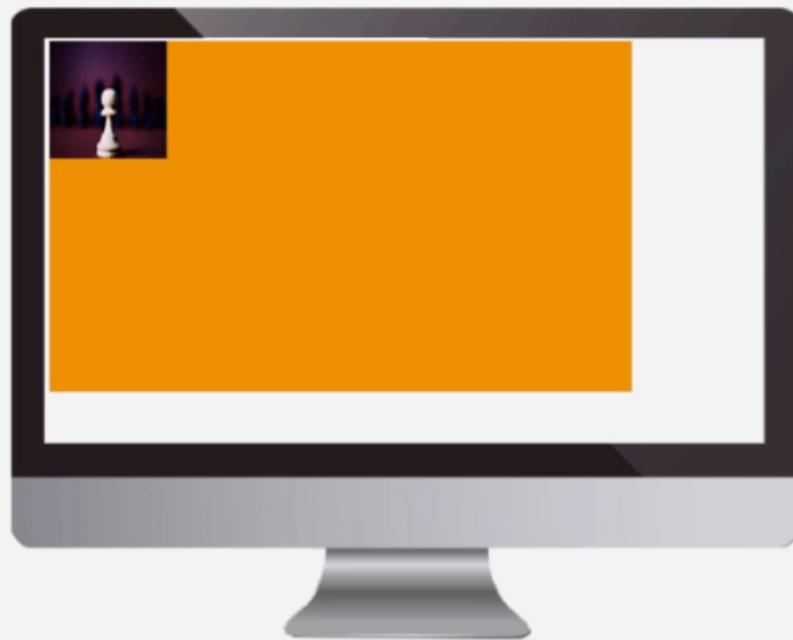
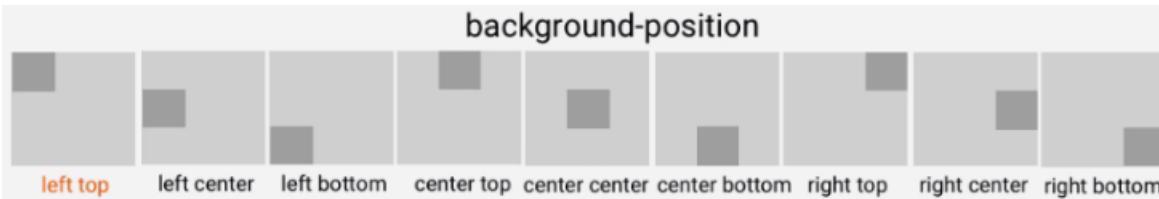
```
<div class="A" > </div>
```

CSS

```
div{  
    height: 200px;  
    width: 350px;  
}  
.A{  
    background-color: orange;  
    background-image: url(figure.jpg);  
    background-size: 100px 100px;  
    background-repeat: repeat-x;  
}
```

Background-repeat

Non-repeat, repeat-x, repeat y



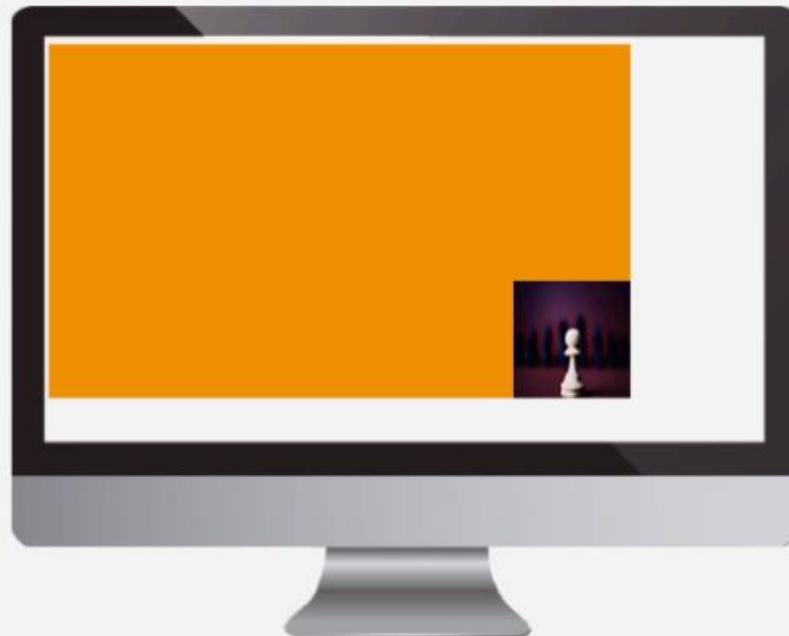
HTML

```
<div class="A" > </div>
```

CSS

```
div{  
    height: 300px;  
    width: 500px;  
}  
.A{  
    background-color: orange;  
    background-image: url(pawn.jpg);  
    background-size: 100px 100px;  
    background-repeat: no-repeat;  
    background-position: left top;  
}
```

background-attachment



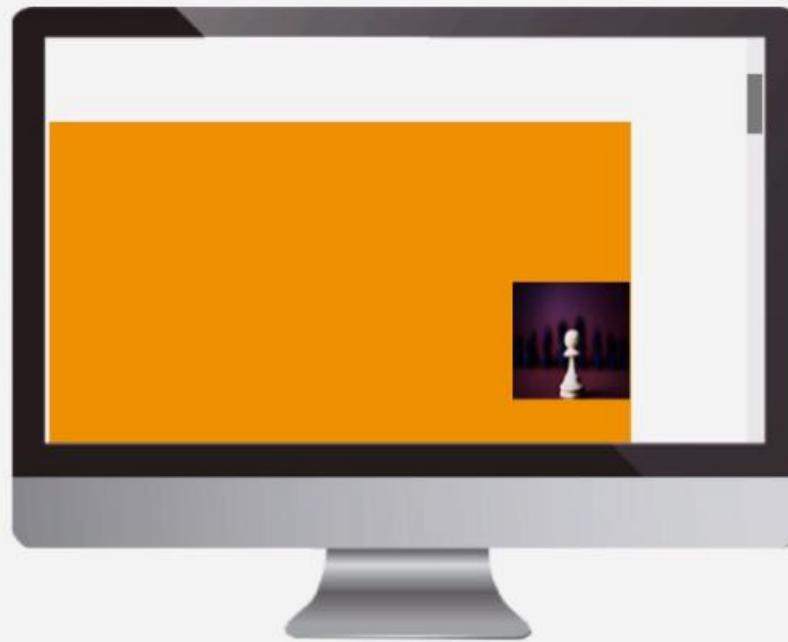
HTML

```
<div class="A" > </div>
```

CSS

```
div{  
    height: 300px;  
    width: 500px;  
}  
.A{  
    background-color: orange;  
    background-image: url(pawn.jpg);  
    background-size: 100px 100px;  
    background-repeat: no-repeat;  
    background-position: right bottom;  
    background-attachment: fixed;  
}
```

background-attachment



HTML

```
<div class="A" > </div>
```

CSS

```
div{  
    height: 300px;  
    width: 500px;  
}  
.A{  
    background-color: orange;  
    background-image: url(pawn.jpg);  
    background-size: 100px 100px;  
    background-repeat: no-repeat;  
    background-position: right bottom;  
    background-attachment: fixed;  
}
```



position: static
absolute
relative
fixed



1. Static

`position: static` is the **default value**. Whether we declare it or not, elements are positioned in a normal order on the webpage. Let's give an example:

position: static

Normal flow

HTML

```
<div class="a" > </div>
<div class="b" > </div>
<div class="c" > </div>
```

CSS

```
div{
    height: 100px;
    width: 100px;
}
.a {
    background-color: orange;
}
.b {
    background-color: green;
}
.c {
    background-color: blue;
}
```



2. Relative

`position: relative`: An element's new position relative to its normal position.

Starting with `position: relative` and for all **non-static** position values, we are able to change an element's **default** position by using the **helper properties** that I've mentioned above.

The diagram shows a computer monitor displaying three colored boxes: orange, green, and blue. The orange box is at the top, the green box is in the middle, and the blue box is at the bottom. A dashed box highlights the green and blue boxes. A horizontal double-headed arrow labeled '70px' indicates the distance from the left edge of the highlighted area to the left edge of the blue box. A vertical double-headed arrow labeled '40px' indicates the distance from the top edge of the highlighted area to the top edge of the green box.

position: relative

HTML

```
<div class="a" > </div>
<div class="b" > </div>
<div class="c" > </div>
```

CSS

```
div{
    height: 100px;
    width: 100px;
}
.a {
    background-color: orange;
}
.b {
    background-color: green;
    position: relative;
    left: 70px;
    top: 40px;
}
.c {
    background-color: blue;
}
```

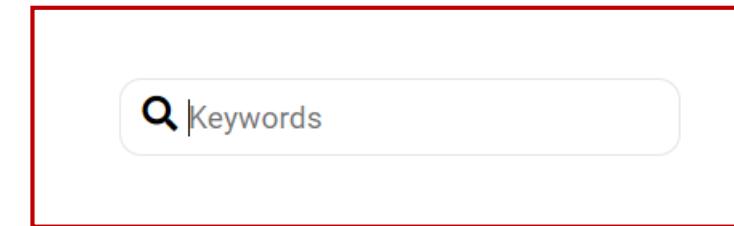
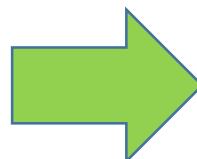
3. Absolute

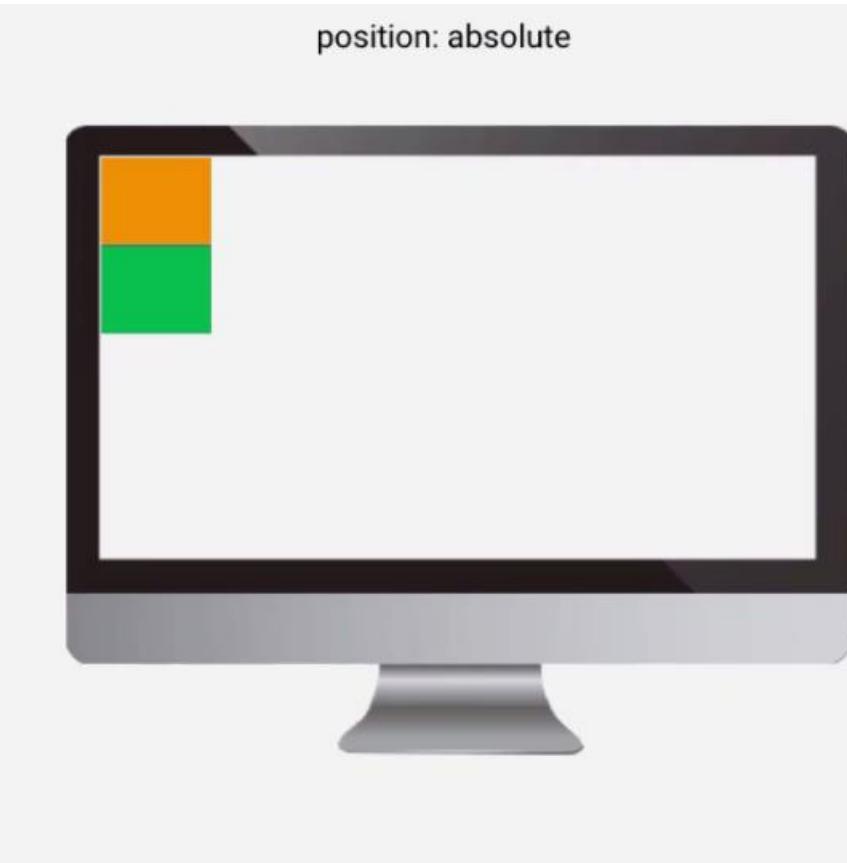
In `position: relative`, the element is positioned **relative to itself**. However, an **absolutely** positioned element is **relative to its parent**.

An element with `position: absolute` is removed from the normal document flow. It is positioned automatically to the starting point (**top-left corner**) of its parent element. If it doesn't have any parent elements, then the initial document `<html>` will be its parent.

Since `position: absolute` removes the element from the document flow, other elements **are affected** and behave as the element is removed completely from the webpage.

```
▼<div class="search-box">
  ▼<label for="search">
    ▶<i class="fas fa-search">...</i>
    </label>
    <input type="text" id="search" placeholder="Keywords">
  </div>
```





position: absolute

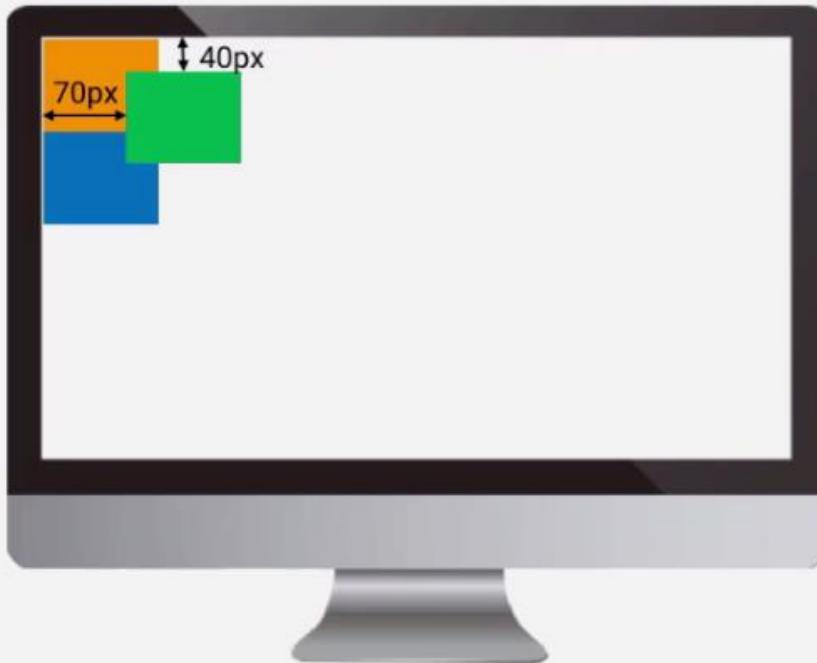
HTML

```
<div class="a" > </div>
<div class="b" > </div>
<div class="c" > </div>
```

CSS

```
div{
    height: 100px;
    width: 100px;
}
.a {
    background-color: orange;
}
.b {
    background-color: green;
    position: absolute;
}
.c {
    background-color: blue;
}
```

position: absolute

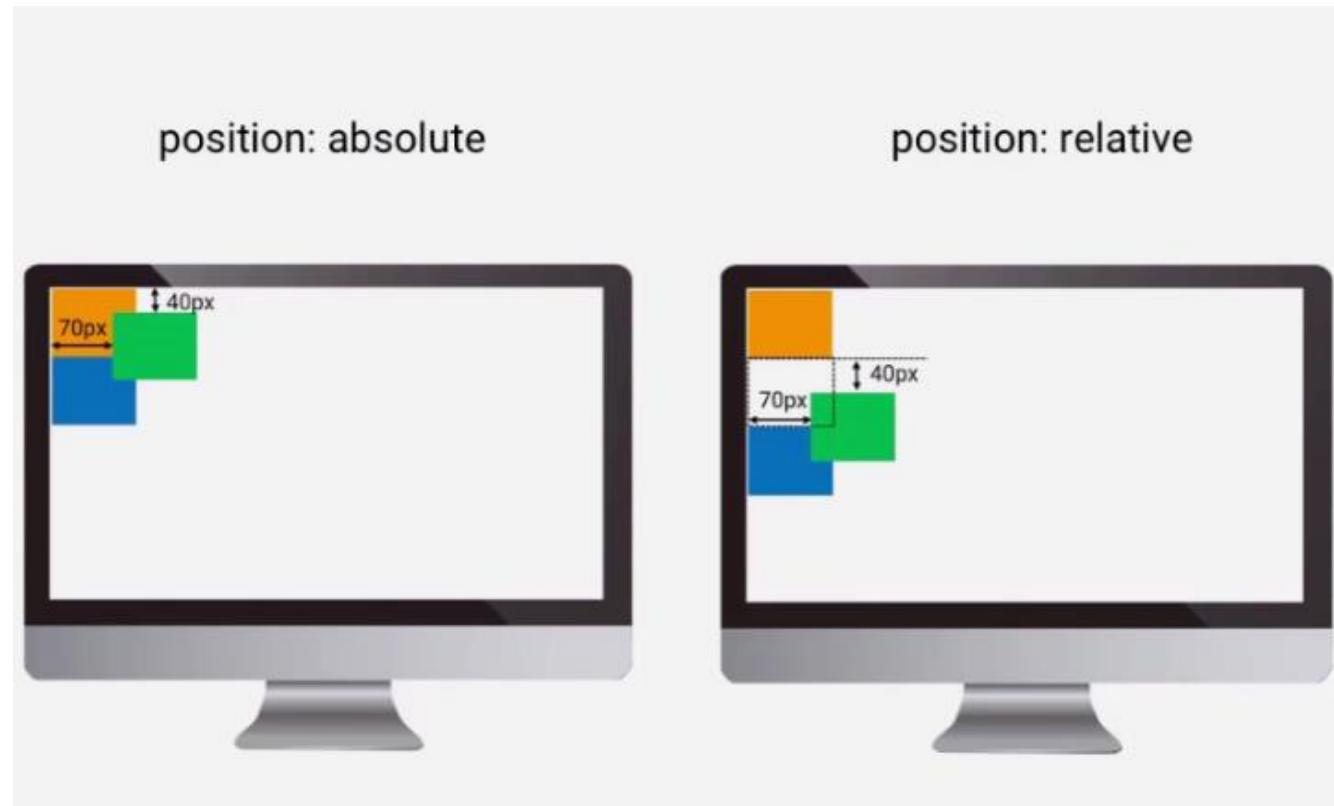


HTML

```
<div class="a" > </div>
<div class="b" > </div>
<div class="c" > </div>
```

CSS

```
div{
    height: 100px;
    width: 100px;
}
.a {
    background-color: orange;
}
.b {
    background-color: green;
    position: absolute;
    left: 70px;
    top: 40px;
}
.c {
    background-color: blue;
}
```





HTML

```
<div class="a" >
  <div class="b" > </div>
</div>
```

CSS

```
.a {
  height: 400px;
  width: 400px;
  background-color: orange;
  position: relative;
  left: 70px;
  top: 40px;
}

.b {
  height: 100px;
  width: 100px;
  background-color: green;
  position: absolute;
  left: 70px;
  top: 40px;
}
```

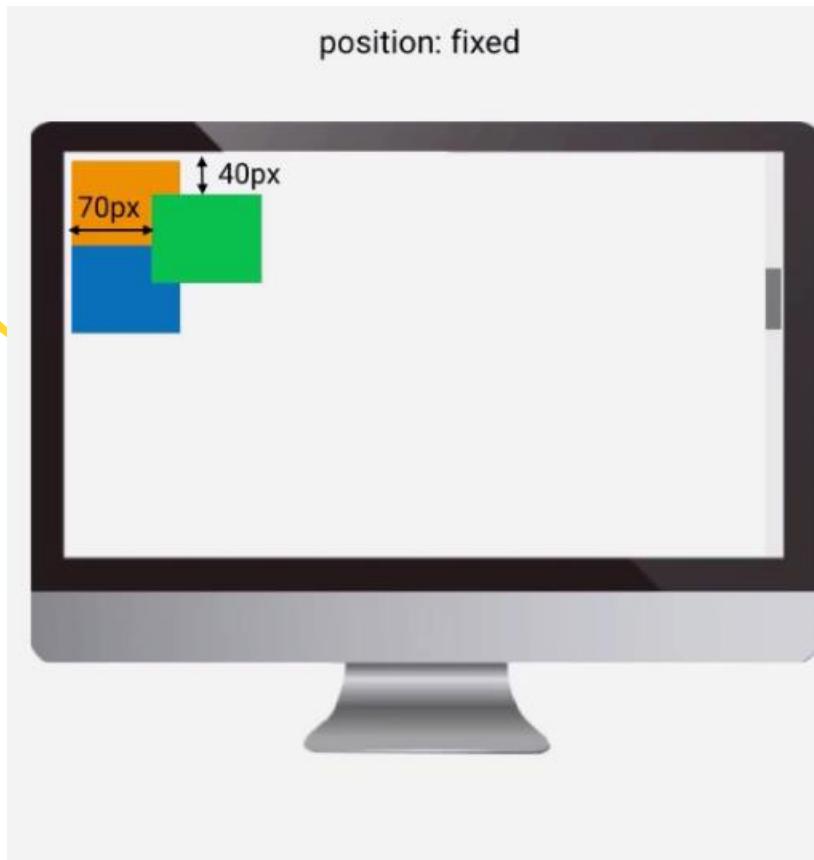
Google search box



4. Fixed

Like `position: absolute`, fixed positioned elements are also removed from the normal document flow. The differences are:

- They are **only relative to the <html> document**, not any other parents.
- They are **not affected by scrolling**.



position: fixed

```
<div class="a" > </div>
<div class="b" > </div>
<div class="c" > </div>
```

HTML

```
div{ height: 100px; width: 100px; }
.a { background-color: orange; }
.b { background-color: green; position: fixed; left: 70px; top: 40px; }
.c { background-color: blue; }
```

CSS



position: fixed

HTML

```
<div class="a" > </div>
<div class="b" > </div>
<div class="c" > </div>
```

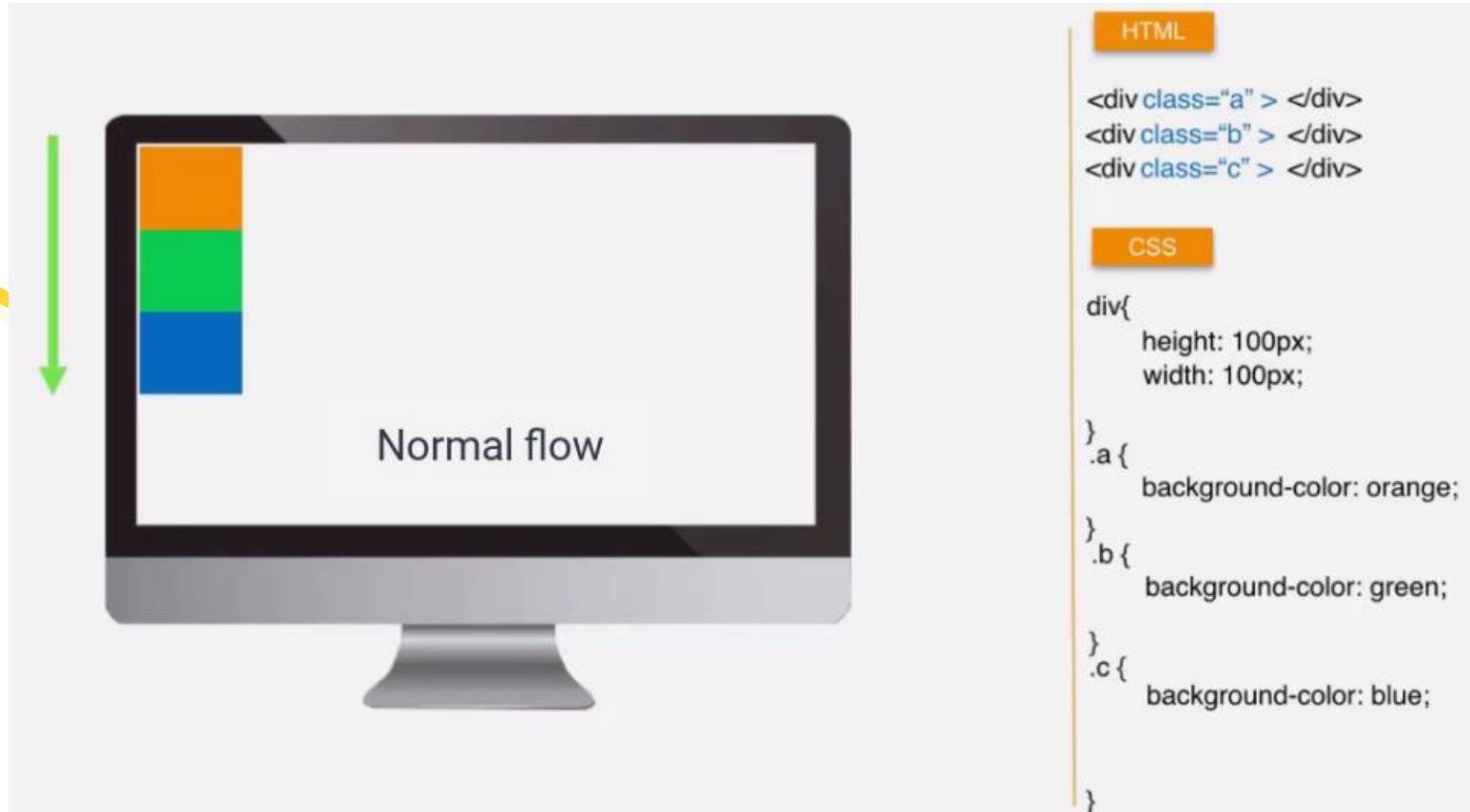
CSS

```
div{
    height: 100px;
    width: 100px;
}
.a {
    background-color: orange;
}
.b {
    background-color: green;
    position: fixed;
    left: 70px;
    top: 40px;
}
.c {
    background-color: blue;
}
```

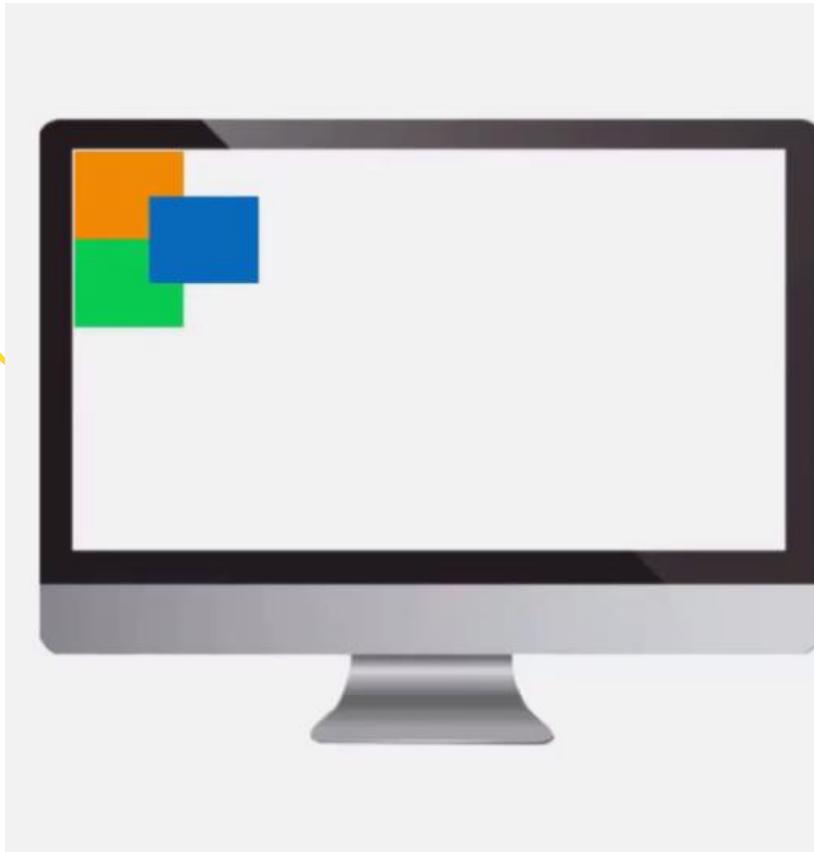
Google search box



z-index: number



Divide webpage to multiple layers with z-index
Go with relative position

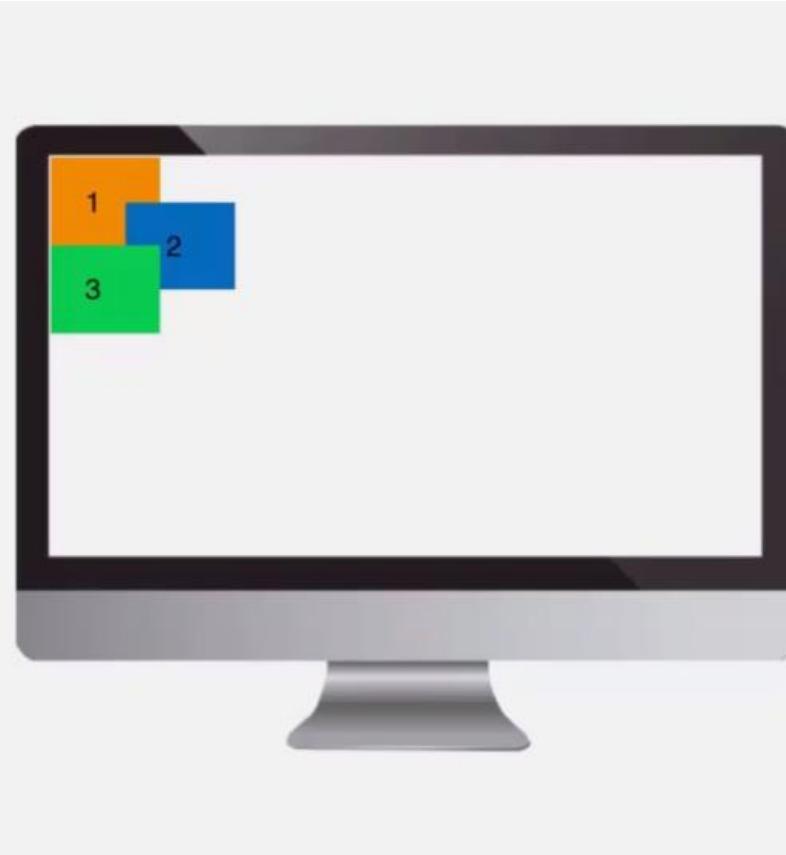


HTML

```
<div class="a" > </div>
<div class="b" > </div>
<div class="c" > </div>
```

CSS

```
div{
    height: 100px;
    width: 100px;
    position: relative;
}
.a{
    background-color: orange;
}
.b{
    background-color: green;
}
.c{
    background-color: blue;
    left: 70px;
    bottom: 150px;
}
```



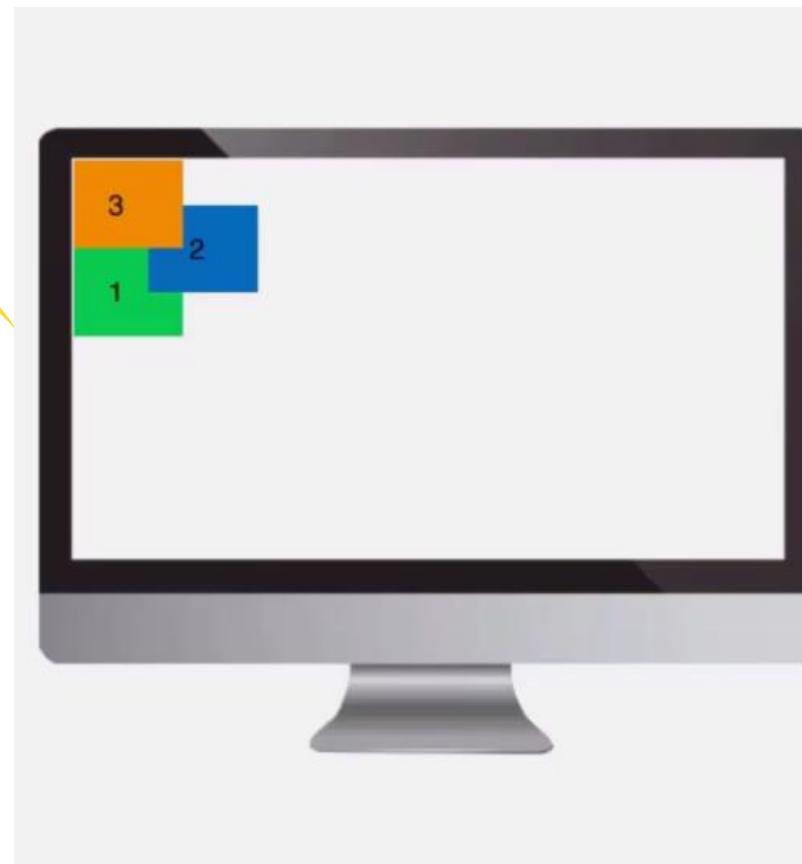
The image shows a computer monitor displaying three overlapping colored squares. The top-left square is orange with the number '1'. The bottom-left square is green with the number '3'. The top-right square is blue with the number '2'. This visual represents the stacking order of elements based on their z-index values.

HTML

```
<div class="a" > </div>
<div class="b" > </div>
<div class="c" > </div>
```

CSS

```
div{
    height: 100px;
    width: 100px;
    position: relative;
}
.a{
    background-color: orange;
    z-index: 1;
}
.b{
    background-color: green;
    z-index: 3;
}
.c{
    background-color: blue;
    left: 70px;
    bottom: 150px;
    z-index: 2;
}
```



The image shows a computer monitor displaying three overlapping colored boxes. The orange box (z-index: 3) is at the top, the green box (z-index: 1) is in the middle, and the blue box (z-index: 2) is partially visible behind them. This visual representation demonstrates how the CSS z-index property affects the stacking order of elements.

HTML

```
<div class="a" > </div>
<div class="b" > </div>
<div class="c" > </div>
```

CSS

```
div{
    height: 100px;
    width: 100px;
    position: relative;
}
.a{
    background-color: orange;
    z-index: 3;
}
.b{
    background-color: green;
    z-index: 1;
}
.c{
    background-color: blue;
    left: 70px;
    bottom: 150px;
    z-index: 2;
}
```

Inline
Block
Inline-Block
None
Visibility: hidden

Normal flow

```
HTML
<div class="a" > </div>
<div class="b" > </div>
<div class="c" > </div>

CSS
div{
    height: 100px;
    width: 100px;
    display: block;
}

.a {
    background-color: orange;
}

.b {
    background-color: green;
}

.c {
    background-color: blue;
}
```



HTML

```
<div class="a" > </div>
<div class="b" > </div>
<div class="c" > </div>
```

CSS

```
div{
    height: 100px;
    width: 100px;
    display: inline-block;
}
.a{
    background-color: orange;
}
.b{
    background-color: green;
}
.c{
    background-color: blue;
}
```



HTML

```
<div class="a"> Text </div>
<div class="b"> Text </div>
<div class="c"> Text </div>
```

CSS

```
div{
    height:100px;
    width:100px;
    display: inline;
}
.a{
    background-color: orange;
}
.b{
    background-color: green;
}
.c{
    background-color: blue;
}
```



An illustration of a computer monitor with a black bezel and a silver base. On the screen, there are three small rectangular boxes arranged horizontally. The first box is green with the word "Text" in white. The second box is blue with the word "Text" in white. The third box is orange with the word "Text" in white. The monitor is set against a light gray background.

HTML

```
<div class="a" > Text </div>
<div class="b" > Text </div>
<div class="c" > Text </div>
```

CSS

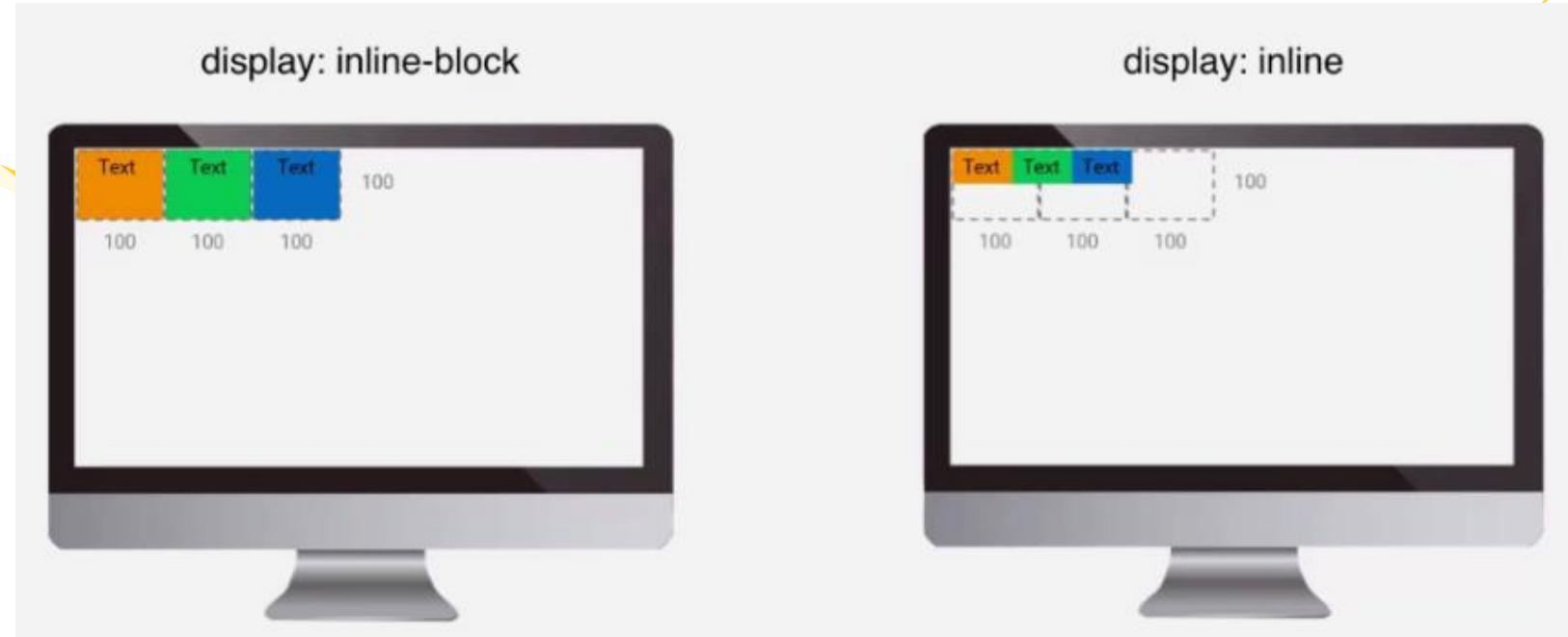
```
div{
    height: 100px;
    width: 100px;
    display: inline;
}

.a {
    background-color: orange;
    display: none;
}

.b {
    background-color: green;
}

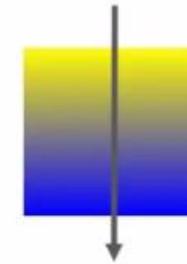
.c {
    background-color: blue;
}
```







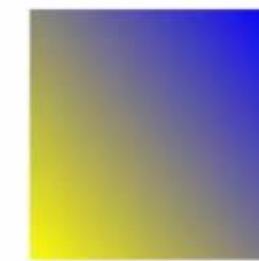
1. linear-gradient()



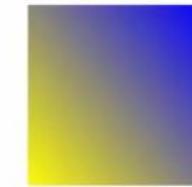
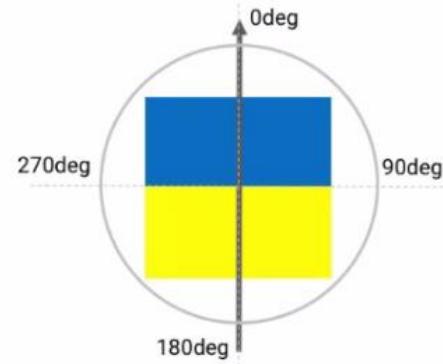
2. radial-gradient()



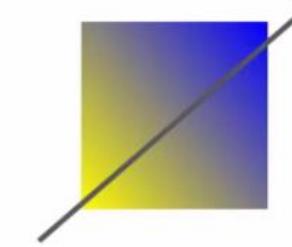
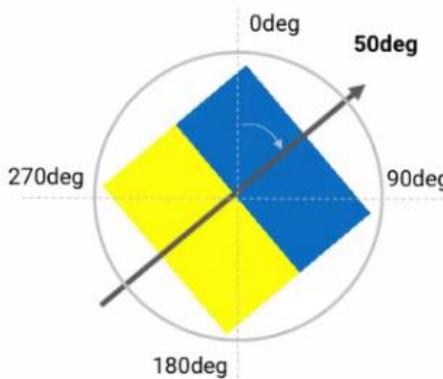
`background-image: linear-gradient(50deg, yellow,blue);`



`background-image: linear-gradient(50deg, yellow,blue);`

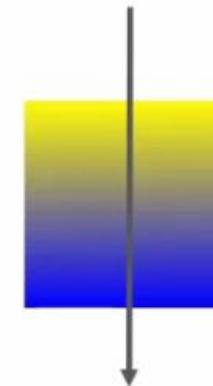
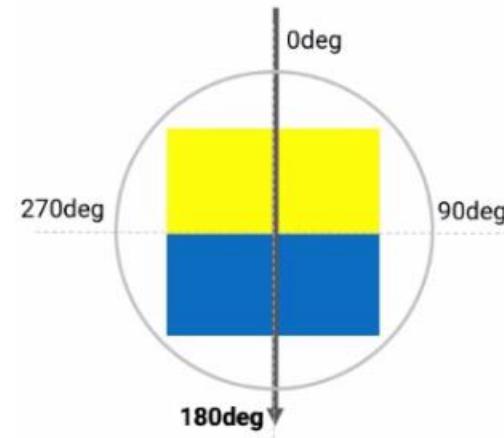


`background-image: linear-gradient(50deg, yellow,blue);`



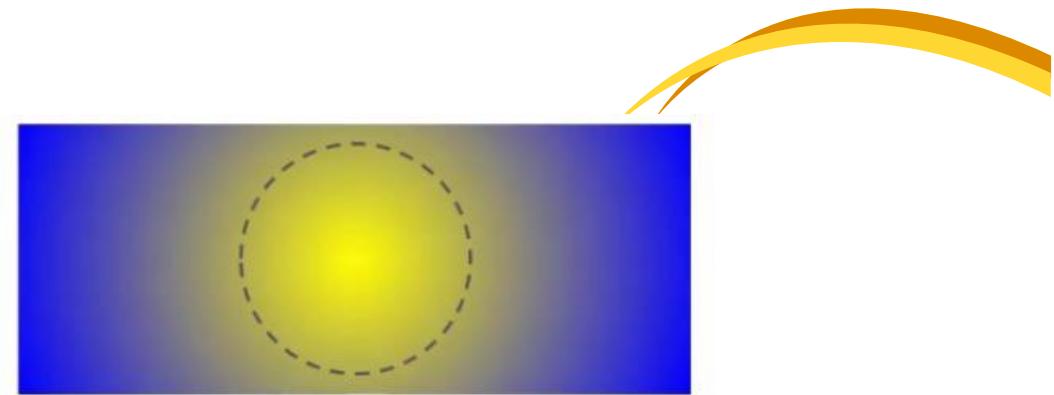


`background-image: linear-gradient(180deg, yellow,blue);`

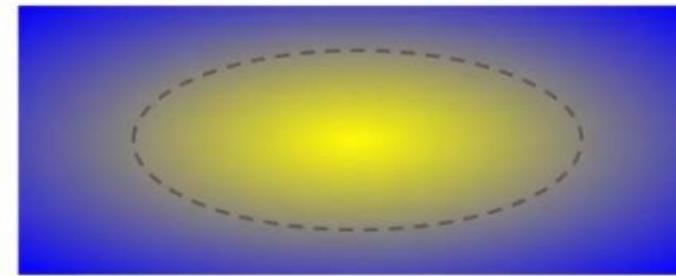




background-image: radial-gradient(**circle**, yellow, blue);



background-image: radial-gradient(**ellipse**, yellow, blue);



Block: 400 400 with Border
Background-Image

colorzilla gradient generator:
<https://www.colorzilla.com/gradient-editor/>
Replace with background image



Sports Car

What is a sports car? A car may be a sporting automobile without being a sports car.

A sports car, or sportscar, is a small, usually two-seater, two-door automobile designed for spirited performance and nimble handling. The term "sports car" was used in *The Times*, London in 1919. According to USA's Merriam-Webster dictionary, USA's first known use of the term was in 1928. Sports cars started to become popular during the 1920s.

Sports cars may be spartan or luxurious, but high maneuverability and light weight are requisite. Sports cars are usually aerodynamically shaped (since the 1950s), and have a low center of gravity compared to standard models. Steering and suspension are typically designed for precise control at high speeds. Traditionally sports cars were open roadsters, but closed coupés also started to become popular during the 1930s, and the distinction between a sports car and a grand tourer is not absolute.



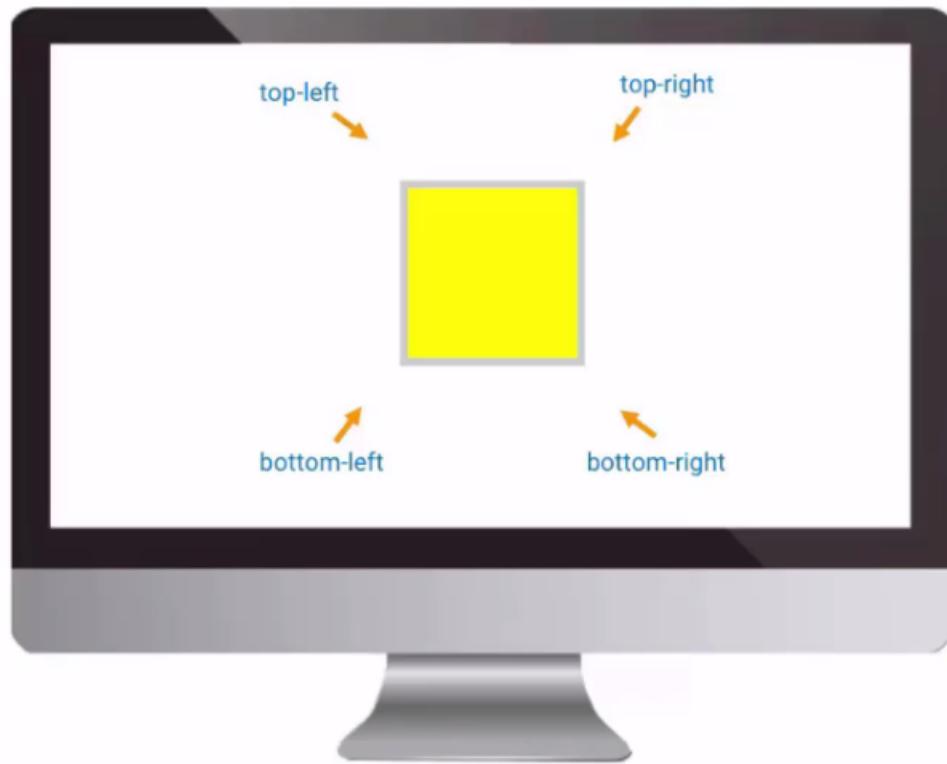
Seating Layout

Traditional sports cars were typically two-seat roadsters.

Although the first sports cars were derived from fast tourers, and early sporting regulations often demanded four seats (even three-seaters were often produced by coachbuilders), two seats became common from about the mid-1920s. Modern sports cars may also have small back seats that are often really only suitable for luggage or small children; such a configuration is referred to as a 2+2 (two full seats + two "occasional" seats).

Over the years, some manufacturers of sports cars have sought to increase the practicality of their vehicles by increasing the seating room. One method is to place the driver's seat in the center of the car, which allows two full-sized passenger seats on each side and slightly behind the driver. The arrangement was originally considered for the Lamborghini Miura, but abandoned as impractical because of the difficulty for the driver to enter/exit the vehicle. McLaren used the design in their F1.

Do the exercises – Exoticar Body

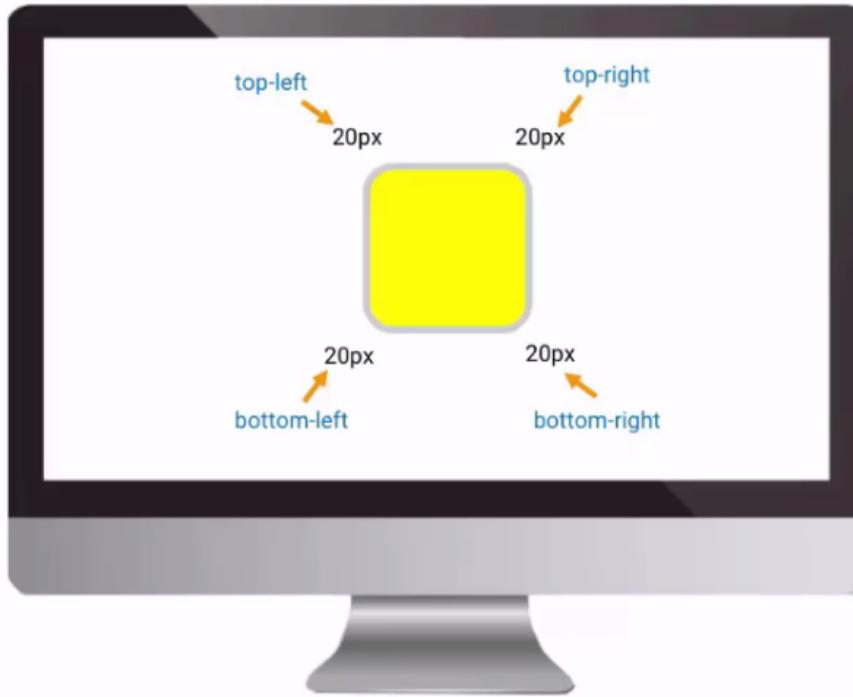


HTML

```
<div class = "block"> </div>
```

CSS

```
.block {  
    width: 200px;  
    height: 200px;  
    background: yellow;  
    border: 5px solid grey;  
}
```

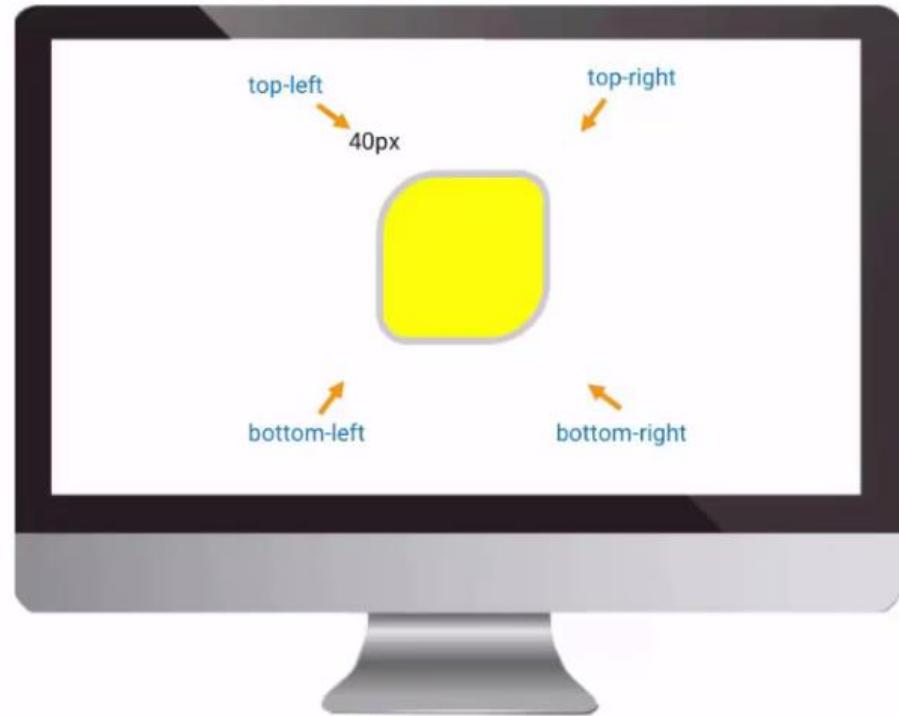


HTML

```
<div class = "block"> </div>
```

CSS

```
.block {  
    width: 200px;  
    height: 200px;  
    background: yellow;  
    border: 5px solid grey;  
  
    border-radius: 20px;  
}
```



HTML

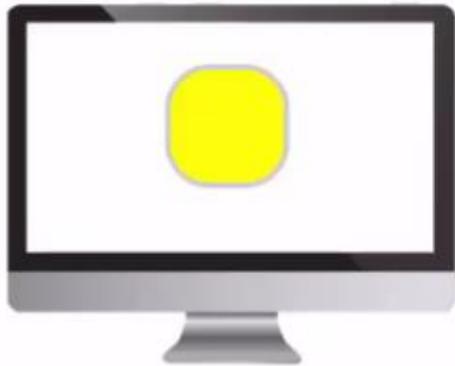
```
<div class = "block"> </div>
```

CSS

```
.block {  
    width: 200px;  
    height: 200px;  
    background: yellow;  
    border: 5px solid grey;  
  
    border-radius: 40px 20px;  
}
```



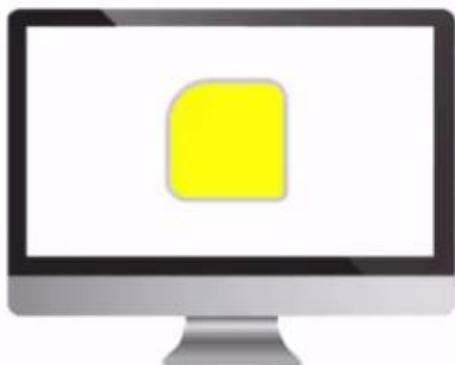
border-radius: 40px;



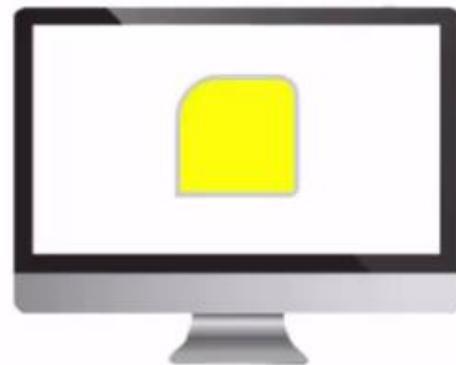
border-radius: 40px 20px;

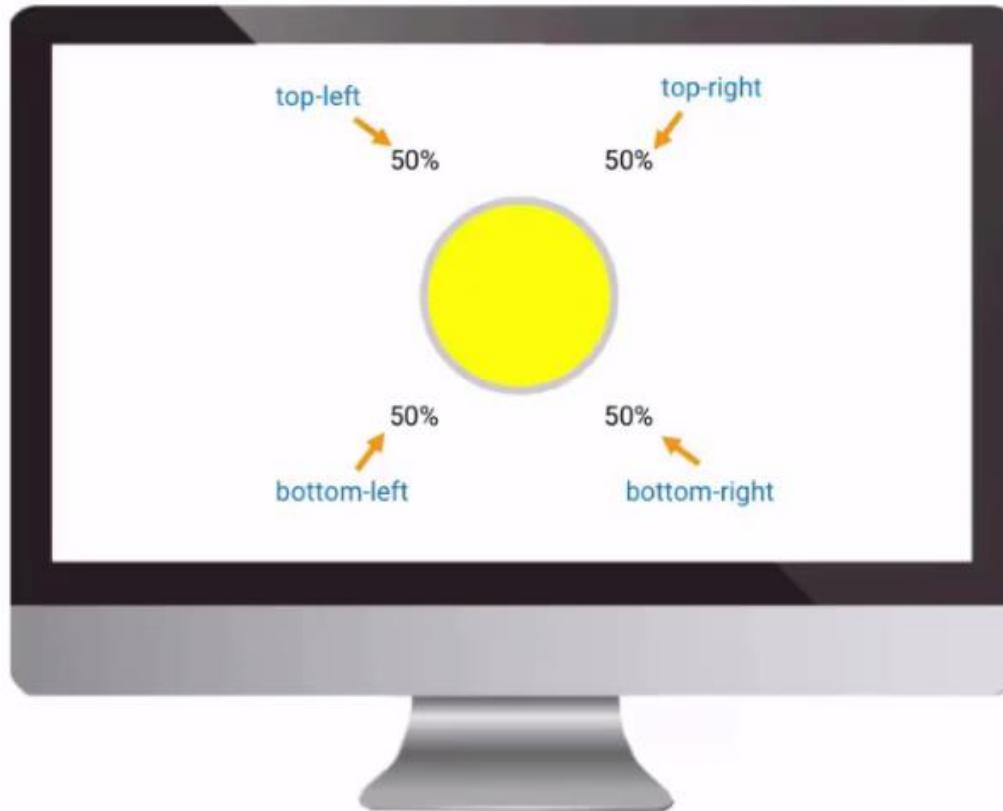


border-radius: 40px 20px 10px;



border-radius: 40px 20px 10px 0;





HTML

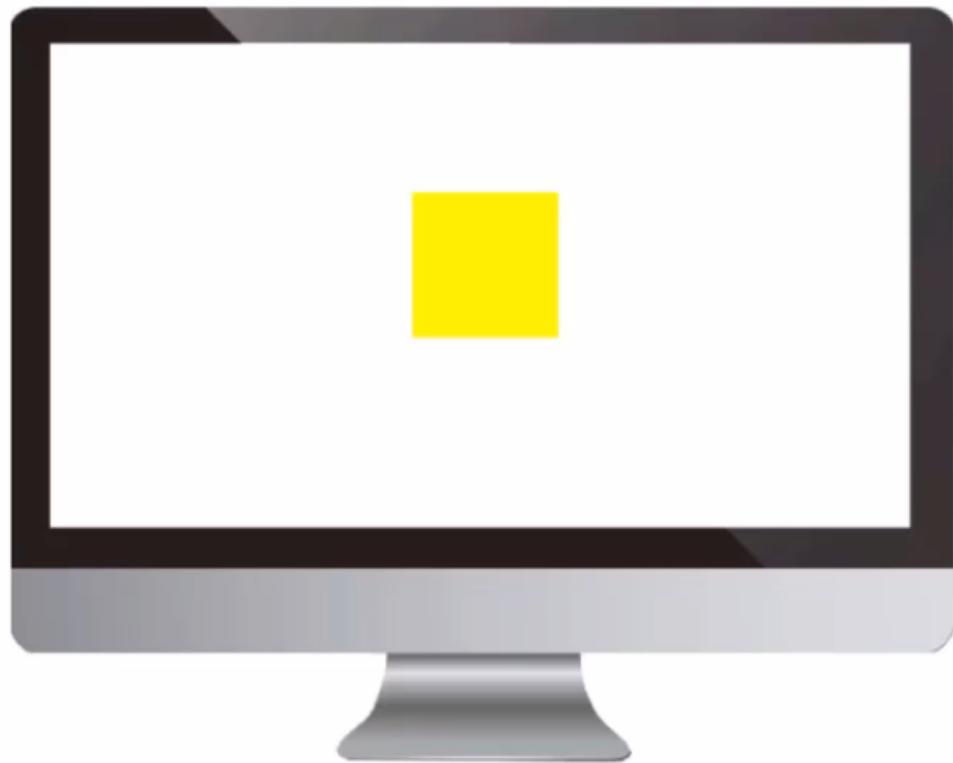
```
<div class = "block"> </div>
```

CSS

```
.block {  
    width: 200px;  
    height: 200px;  
    background: yellow;  
    border: 5px solid grey;  
  
    border-radius: 50%;  
}
```

Create an division w200 h200
Margin: 100px 0 0 300px;





HTML

```
<div class = "block"> </div>
```

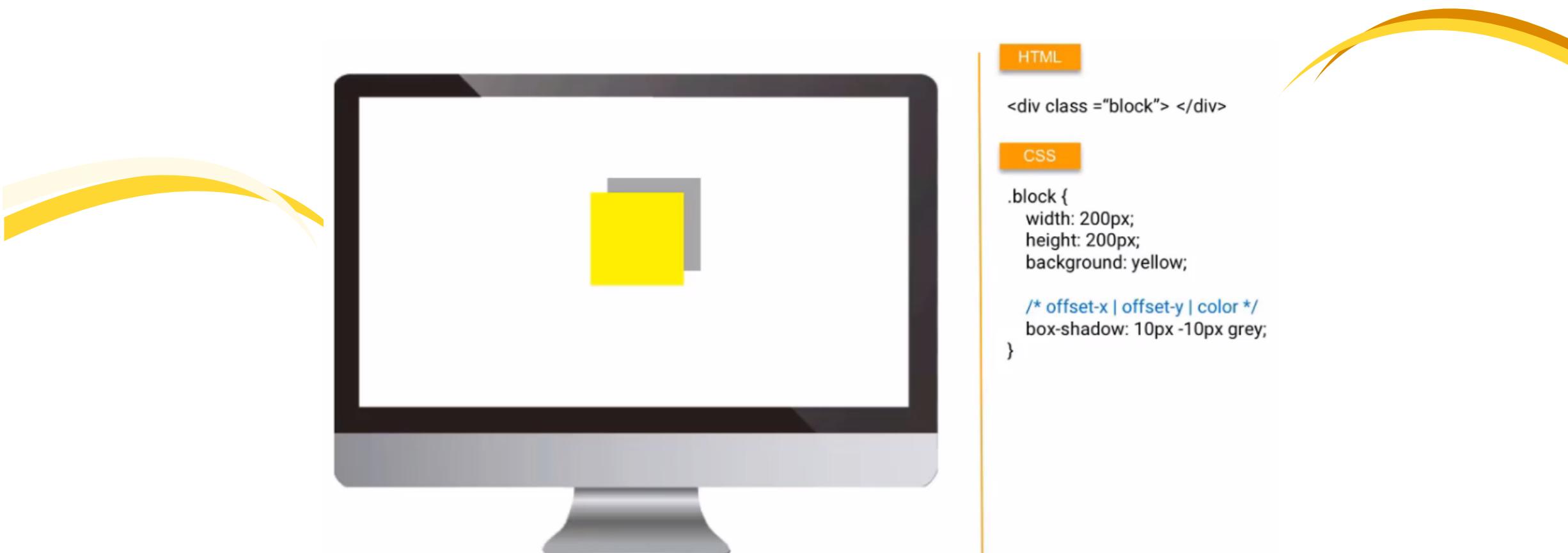
CSS

```
.block {  
    width: 200px;  
    height: 200px;  
    background: yellow;  
}
```





Value	Description
none	Default value. No shadow is displayed
<i>h-offset</i>	Required. The horizontal offset of the shadow. A positive value puts the shadow on the right side of the box, a negative value puts the shadow on the left side of the box
<i>v-offset</i>	Required. The vertical offset of the shadow. A positive value puts the shadow below the box, a negative value puts the shadow above the box
<i>blur</i>	Optional. The blur radius. The higher the number, the more blurred the shadow will be
<i>spread</i>	Optional. The spread radius. A positive value increases the size of the shadow, a negative value decreases the size of the shadow
<i>color</i>	Optional. The color of the shadow. The default value is the text color. Look at CSS Color Values for a complete list of possible color values. Note: In Safari (on PC) the color parameter is required. If you do not specify the color, the shadow is not displayed at all.
<i>inset</i>	Optional. Changes the shadow from an outer shadow (outset) to an inner shadow
<i>initial</i>	Sets this property to its default value. Read about initial
<i>inherit</i>	Inherits this property from its parent element. Read about inherit



HTML

```
<div class = "block"> </div>
```

CSS

```
.block {  
    width: 200px;  
    height: 200px;  
    background: yellow;  
  
    /* offset-x | offset-y | color */  
    box-shadow: 10px -10px grey;  
}
```



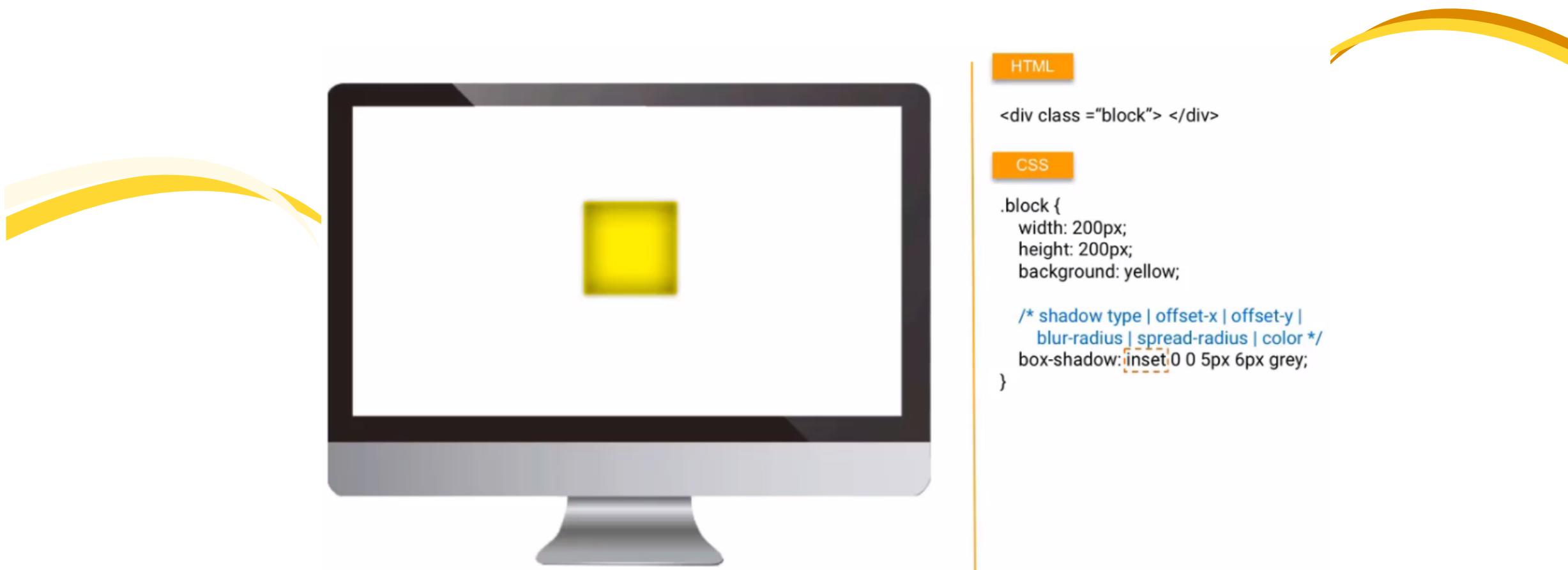
HTML

```
<div class = "block"> </div>
```

CSS

```
.block {  
    width: 200px;  
    height: 200px;  
    background: yellow;  
  
    /* offset-x | offset-y | blur-radius | color */  
    box-shadow: 10px -10px 5px grey;  
}
```





HTML

```
<div class = "block"> </div>
```

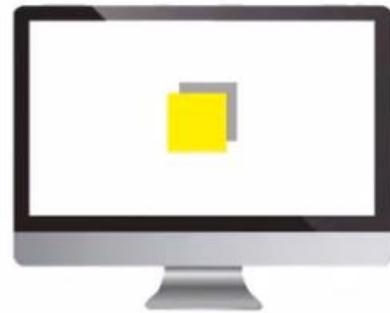
CSS

```
.block {  
    width: 200px;  
    height: 200px;  
    background: yellow;  
  
    /* shadow type | offset-x | offset-y |  
     blur-radius | spread-radius | color */  
    box-shadow: inset 0 0 5px 6px grey;  
}
```





```
/* offset-x | offset-y | color */  
box-shadow: 10px -10px grey;
```



```
/* offset-x | offset-y | blur-radius | color */  
box-shadow: 10px -10px 5px grey;
```



```
/* offset-x | offset-y | blur-radius | spread-radius | color */  
box-shadow: 10px -10px 5px 6px grey;
```



```
/* shadow type | offset-x | offset-y | blur-radius | spread-radius | color */  
box-shadow: inset 0 0 5px 5px grey;
```

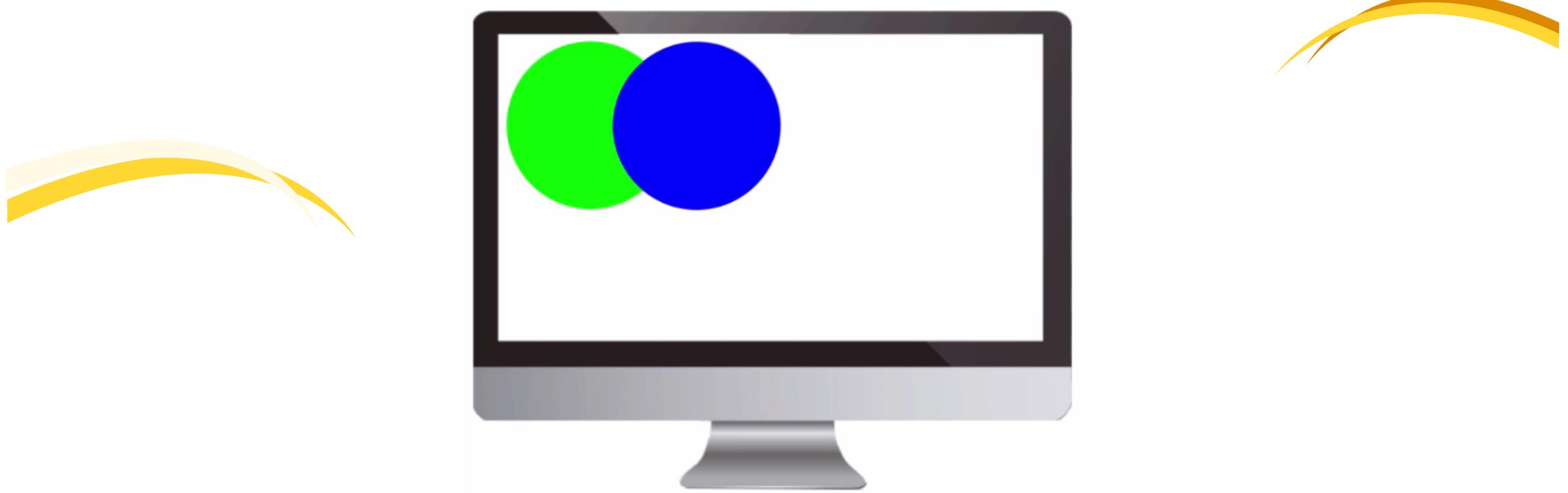


```
/*offset-x offset-y blur-radius color position-shadow(inner|outer)*/  
box-shadow: 20px -20px 60px orange inset;
```



We can add many box shadow styles for an element

Box-shadow: a b c d, a b c d





HTML

```
<div class = "circle-1"> </div>  
<div class = "circle-2"> </div>
```

CSS

```
div{  
    height: 400px;  
    width: 400px;  
    border-radius: 50%;  
}  
.circle-1{  
    background: rgb(0,255,0);  
}  
.circle-2{  
    position: absolute;  
    top: 0;  
    left: 300px;  
    background: rgba(0,0,255,0.5);  
}
```





RGB = Red Green Blue
0 – 255 0 – 255 0 – 255

RGBA = Red Green Blue Alfa
0 – 255 0 – 255 0 – 255 0 – 1





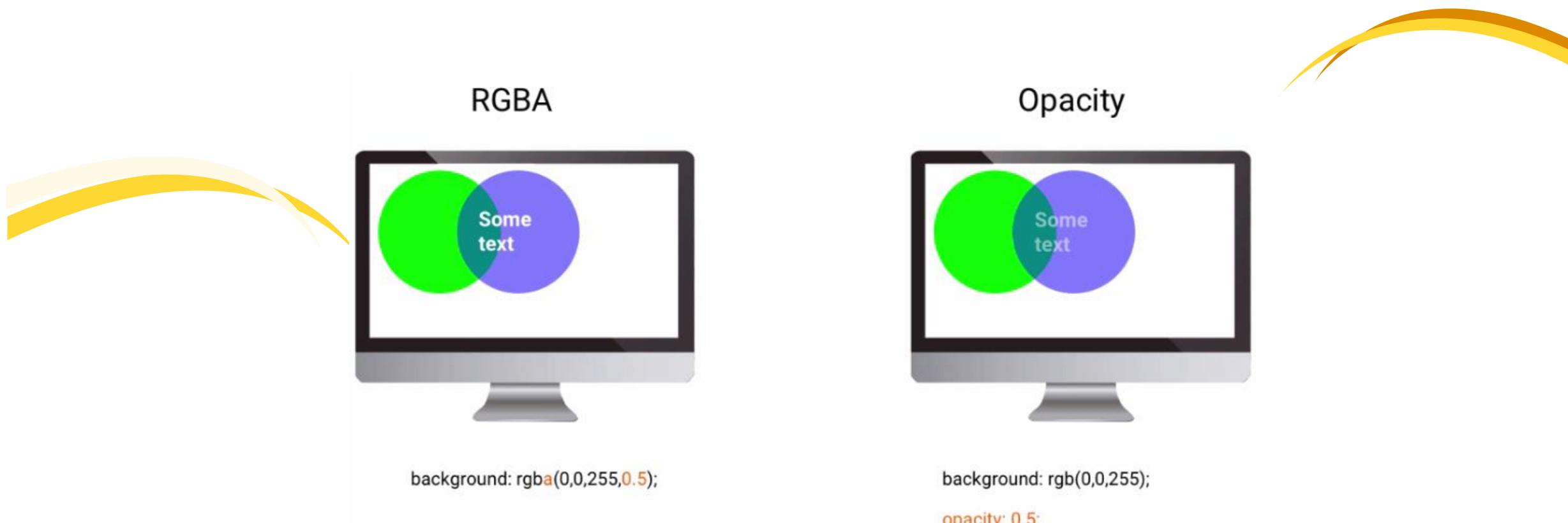
HTML

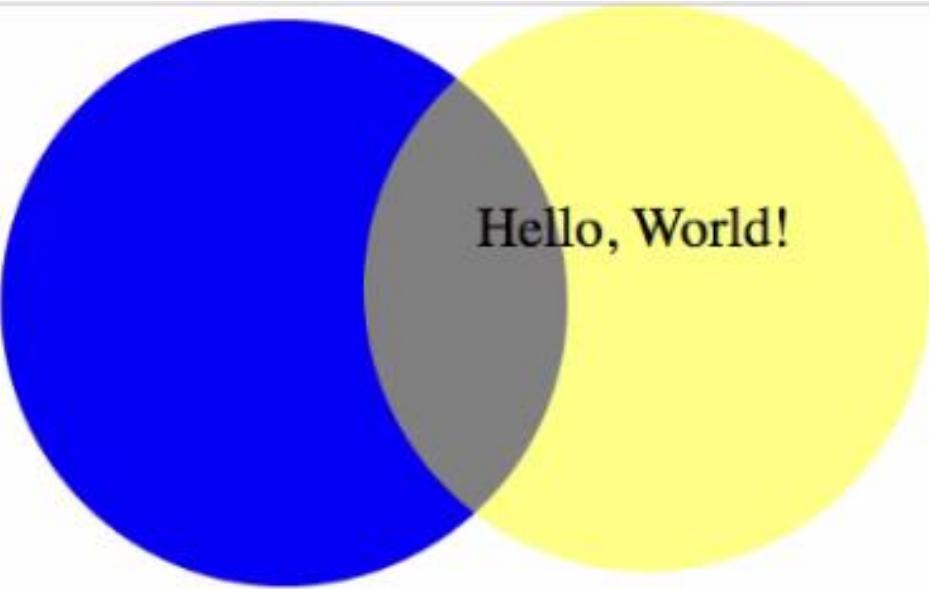
```
<div class = "circle-1"> </div>
<div class = "circle-2"> </div>
```

CSS

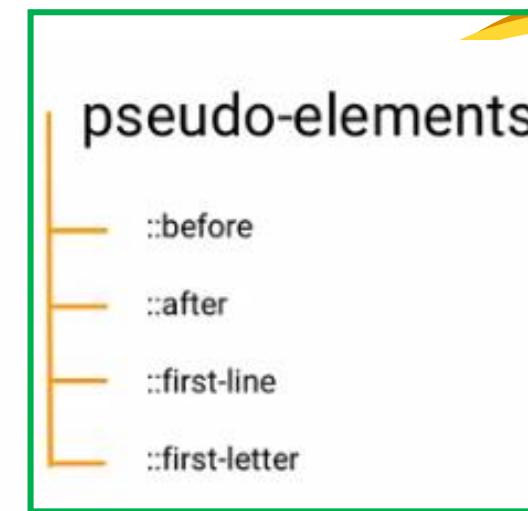
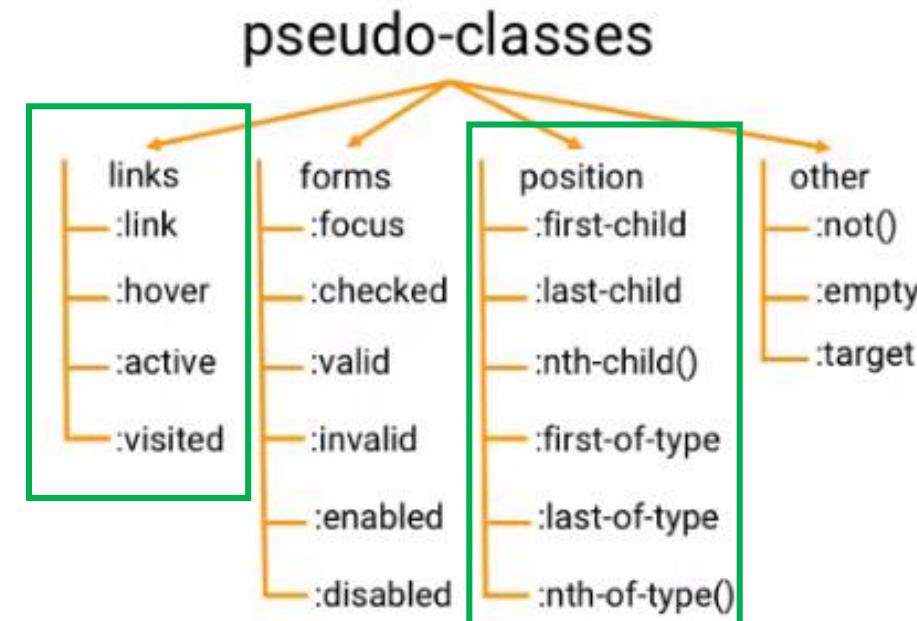
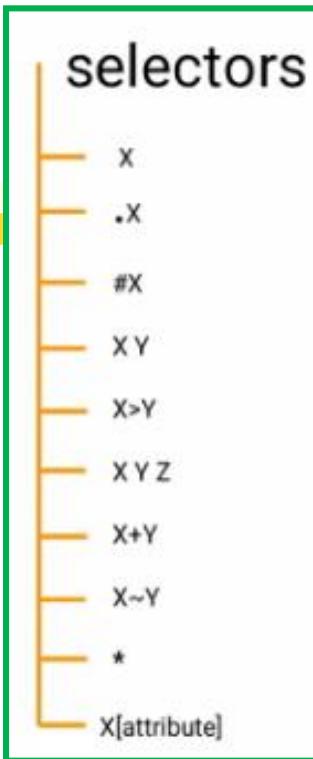
```
div{
    height: 400px;
    width: 400px;
    border-radius: 50%;
}
.circle-1{
    background: rgb(0,255,0);
}
.circle-2{
    position: absolute;
    top: 0;
    left: 300px;
    background: rgb (0,0,255);
    opacity: 0.5;
}
```

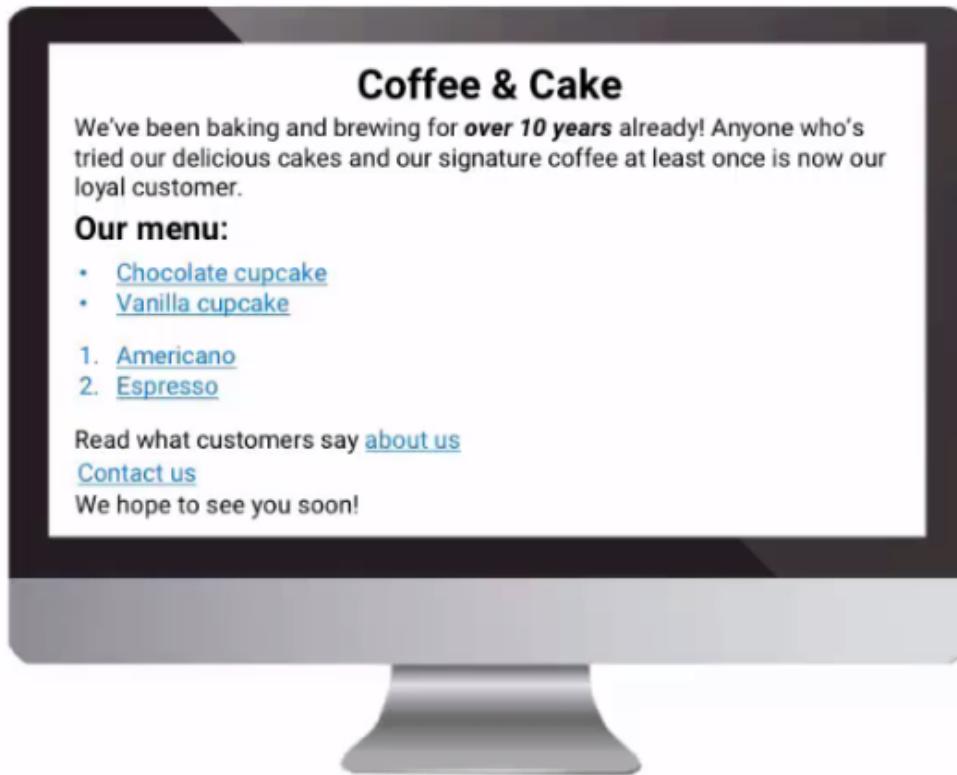






Demo





HTML

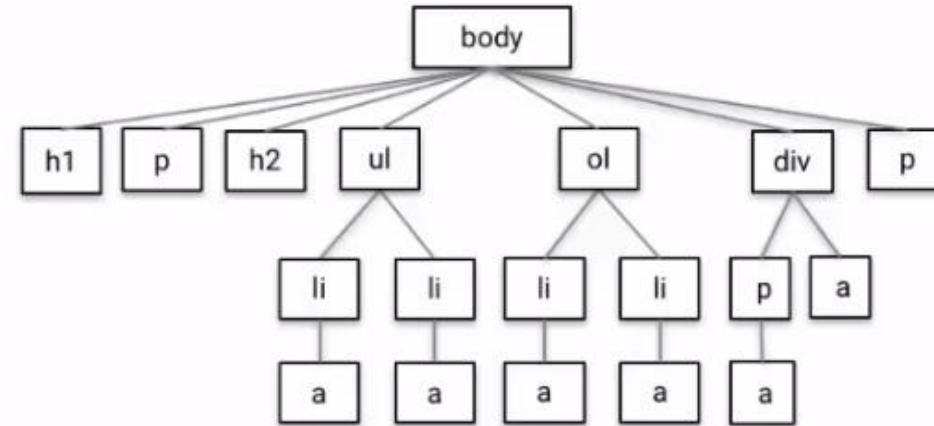
```
<body>
  <h1>Coffee & Cake</h1>
  <p>We've been baking and brewing...</p>
  <h2>Our menu:</h2>
  <ul>
    <li><a href="">Chocolate cupcake</a></li>
    <li><a href="">Vanilla cupcake</a></li>
  </ul>
  <ol>
    <li><a href="">Americano</a></li>
    <li><a href="">Espresso</a></li>
  </ol>
  <div>
    <p>Read what customers say
      <a href="">about us</a>
    </p>
    <a href="">Contact us</a>
  </div>
  <p>We hope to see you soon!</p>
</body>
```



Create input together

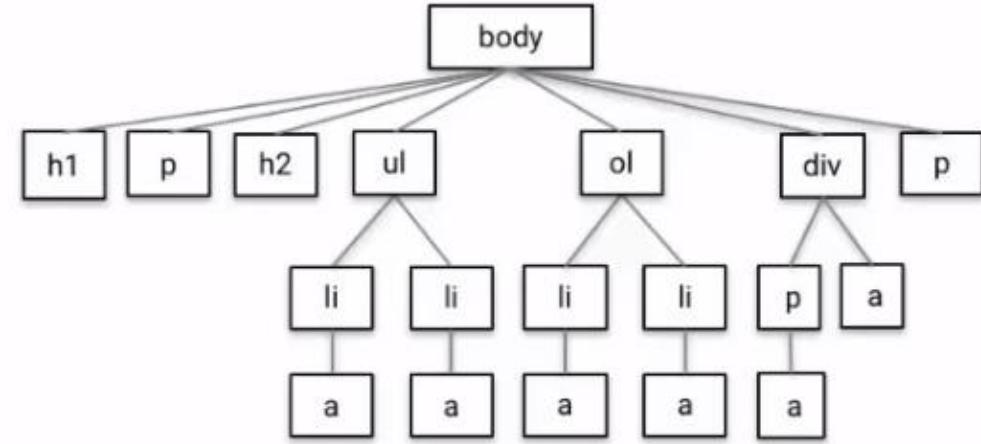
HTML

```
<body>
  <h1>Coffee & Cake</h1>
  <p>We've been baking and brewing...</p>
  <h2>Our menu:</h2>
  <ul>
    <li><a href="">Chocolate cupcake</a></li>
    <li><a href="">Vanilla cupcake</a></li>
  </ul>
  <ol>
    <li><a href="">Americano</a></li>
    <li><a href="">Espresso</a></li>
  </ol>
  <div>
    <p>Read what customers say
      <a href="">about us</a>
    </p>
    <a href="">Contact us</a>
  </div>
  <p>We hope to see you soon!</p>
</body>
```



HTML

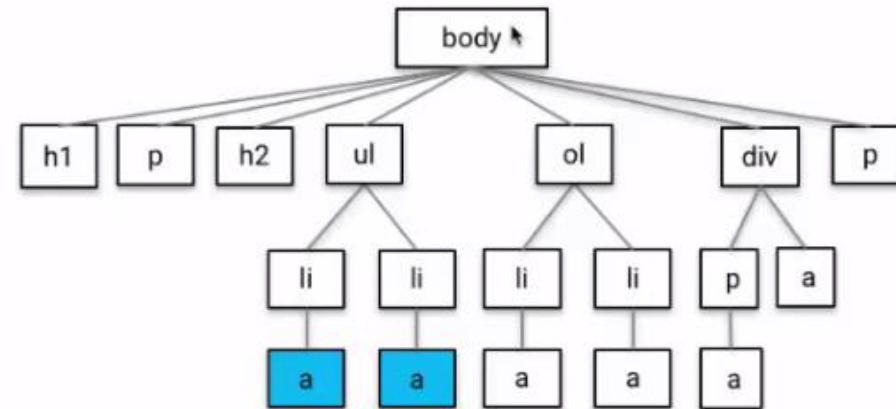
- | | |
|------------------|-----------|
| 1. X | <tag> |
| 2. .X | class |
| 3. #X | id |
| 4. X Y | |
| 5. X > Y | |
| 6. X Y Z | |
| 7. X + Y | |
| 8. X ~ Y | |
| 9. * | all |
| 10. X[attribute] | attribute |





HTML

1. X	<tag>
2. .X	class
3. #X	id
4. XY	↓
5. X > Y	↓
6. XYZ	↓
7. X + Y	→
8. X ~ Y	→
9. *	all
10. X[attribute]	attribute



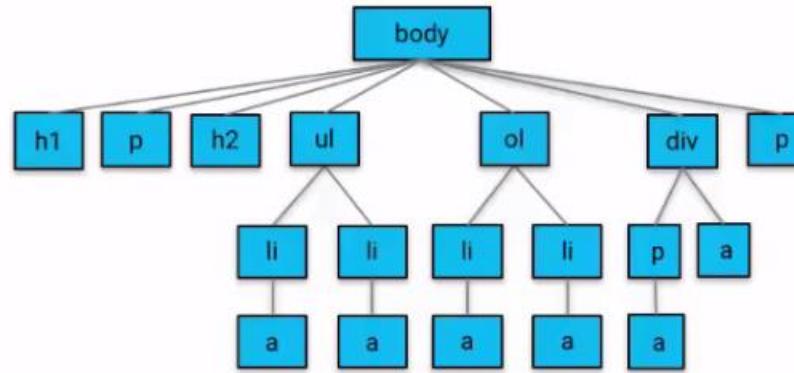
ul li a

↑ ↑ ↑
X Y Z



HTML

1. X	<tag>
2. .X	class
3. #X	id
4. X Y	
5. X > Y	
6. X Y Z	
7. X + Y	→
8. X ~ Y	→
9. *	all
10. X[attribute]	attribute



Background: rgba(255, 255, 0, 0.5)

HTML

```
<input type="text">  
<input type="checkbox">  
<a href="http://my-website-world.org">Link 1</a>  
<a href="https://freesite.com">Link 2</a>
```

CSS

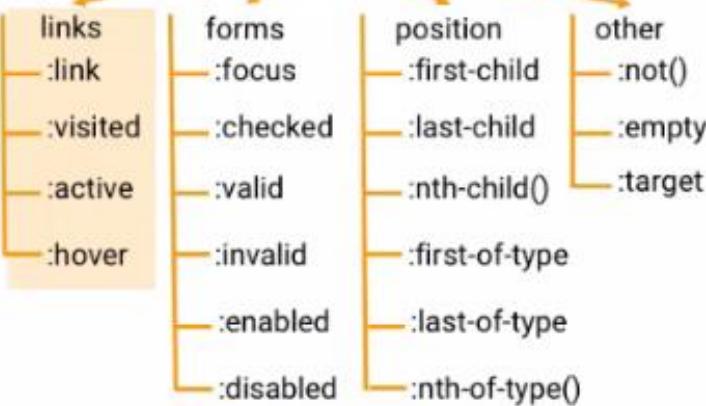
	Symbol	Description	
1.	=	the exact value	input[type = text] { ... }
2.	^=	begins with	a[href ^= https] { ... }
3.	\$=	ends with	a[href \$= org] { ... }
4.	*=	has the value anywhere	a[href *= site] { ... }



selectors

- X
- .X
- #X
- XY
- X>Y
- XYZ
- X+Y
- X~Y
- *
- X[attribute]

pseudo-classes



pseudo-elements

- ::before
- ::after
- ::first-line
- ::first-letter

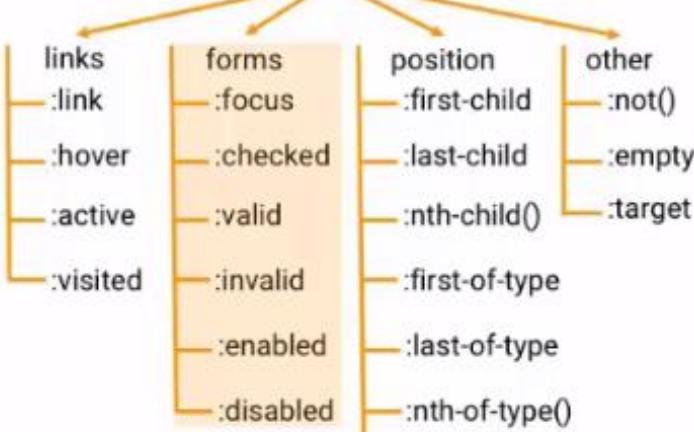
4444ff
3333cc
2222cc
& box-shadow



selectors

- x
- .x
- #x
- XY
- X>Y
- XYZ
- X+Y
- X~Y
- *
- X[attribute]

pseudo-classes



pseudo-elements

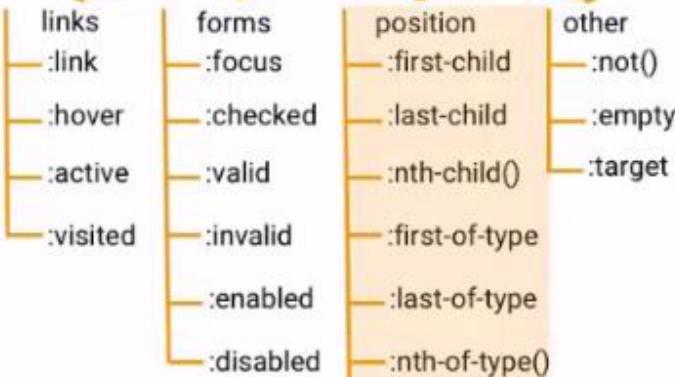
- ::before
- ::after
- ::first-line
- ::first-letter

Text[outline:none, font-size:18px]
Ip1:focus{ box-shadow: 0 0 10px rgba(0,0,0,0.8); }
Ip2:checked + label { color, background } }

selectors

- x
- .x
- #x
- XY
- X>Y
- XYZ
- X+Y
- X~Y
- *
- X[attribute]

pseudo-classes



pseudo-elements

- ::before
- ::after
- ::first-line
- ::first-letter

Example

Using a formula $(an + b)$. Description: a represents a cycle size, n is a counter (starts at 0), and b is an offset value.

Here, we specify a background color for all p elements whose index is a multiple of 3:

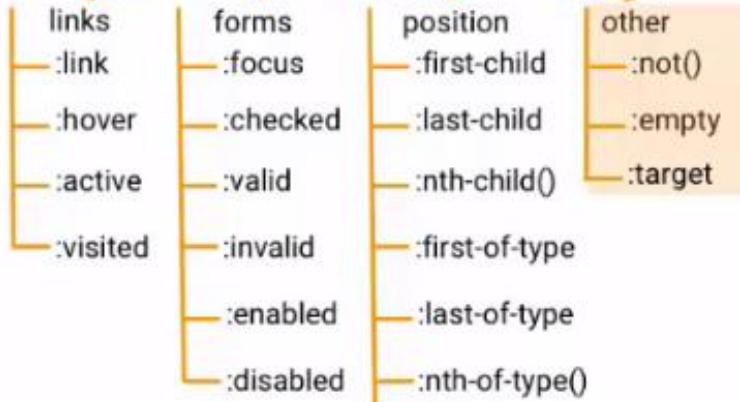
```
p:nth-child(3n+0) {  
    background: red;  
}
```



selectors

- x
- .x
- #x
- X Y
- X>Y
- X Y Z
- X+Y
- X~Y
- *
- X[attribute]

pseudo-classes



pseudo-elements

- ::before
- ::after
- ::first-line
- ::first-letter

6 li
li:not(:first-child)
li:hover
li:not(:hover)
li:empty



selector :: before
after
first-letter
first-line



HTML

```
<body>
  <h1> Headline1 </h1>
  <p>Text text text ... text text
  </p>
</body>
```

CSS

```
p::before {
  content: "Start ";
}
p::after {
  content: " End";
}
p:first-line {
  color: orange;
}
p:first-letter {
  color: blue;
}
```

Coffee & Cake

We have been banking and brewing

Our menu:

- Chocolate cupcake
- Vanilla cupcake

1. Americano
2. Espresso

Read what customer say [about us](#)

We hope to see you soon

fullname

- football
 volleybal

email

```
/* SELECTORS */
.title::before {
    content: "";
    display: block;
    width: 20px;
    height: 20px;
    background: red;
    margin-bottom: 30px;
}

.title::after {
    content: "";
}

ul>li>a {
    text-decoration: none;
}

ol>* {
    color: green;
}

h1~p {
    color: red;
}

/*a.href.contains("link")*/
a[href*="link2"] {
    background: yellow;
}

/*pseudo-classes form*/
.inp-1:focus {
    box-shadow: inset 0 0 2px green;
    outline: none;
    border: none;
}

.cbx-group input[type="checkbox"]:checked + label {
    background: yellow;
}

/*check valid while texting >> keypress event*/
.inp-3:invalid {
    box-shadow: 0 0 5px red;
}

.inp-3:valid {
    box-shadow: 0 0 5px green;
}

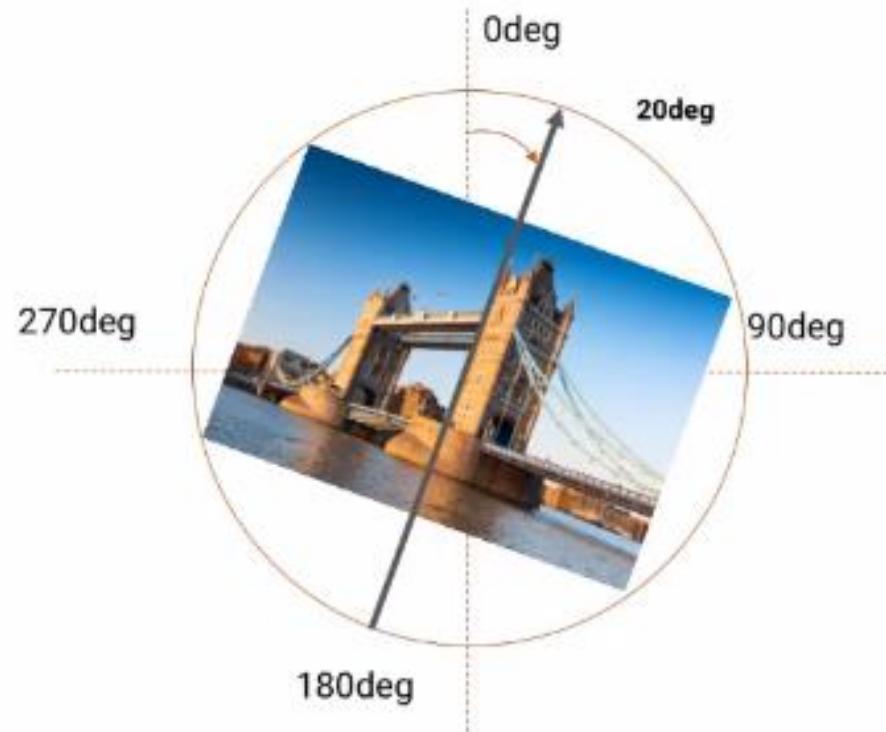
.inp-4:disabled {
    color: black;
}

.inp-4:enabled {
    color: blue;
}
```



transform: **rotate**
scale
translate
skew
rotateX
rotateY

`transform: rotate(20deg)`



Rotate: 20, 200, 380

transform: scale(2)





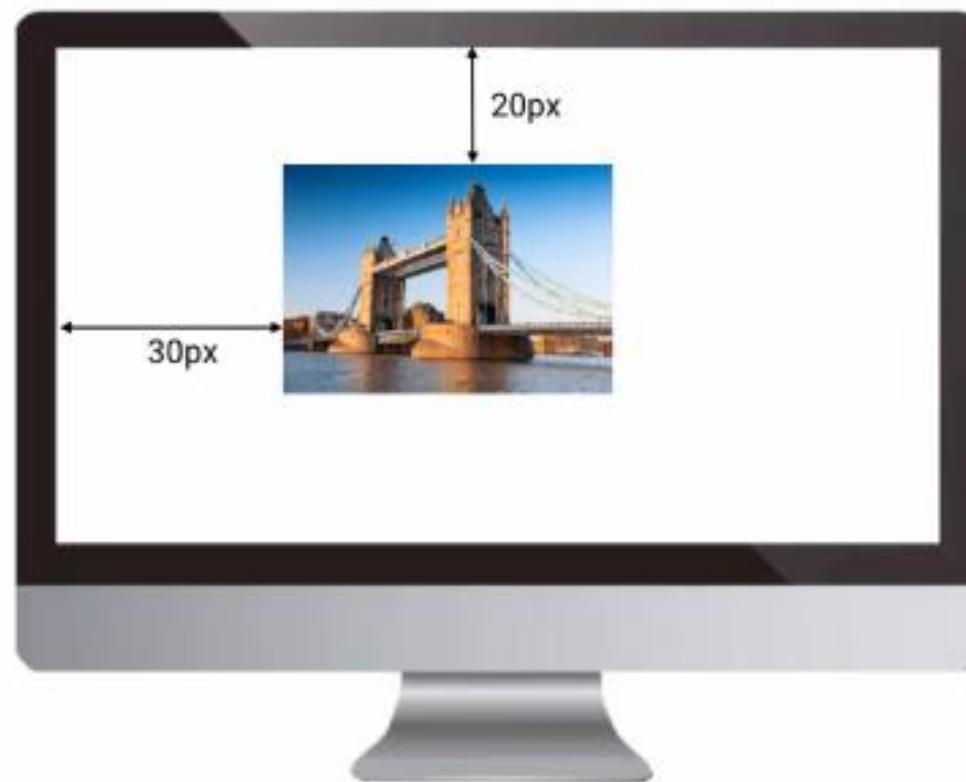
transform: **scale(1.5, 1)**



transform: **scale(1, 1.5)**



`transform: translate(30px, 20px)`



transform: skew(20deg, 0)



https://www.w3schools.com/css/css3_2dtransforms.asp



transform: rotateY(180deg)

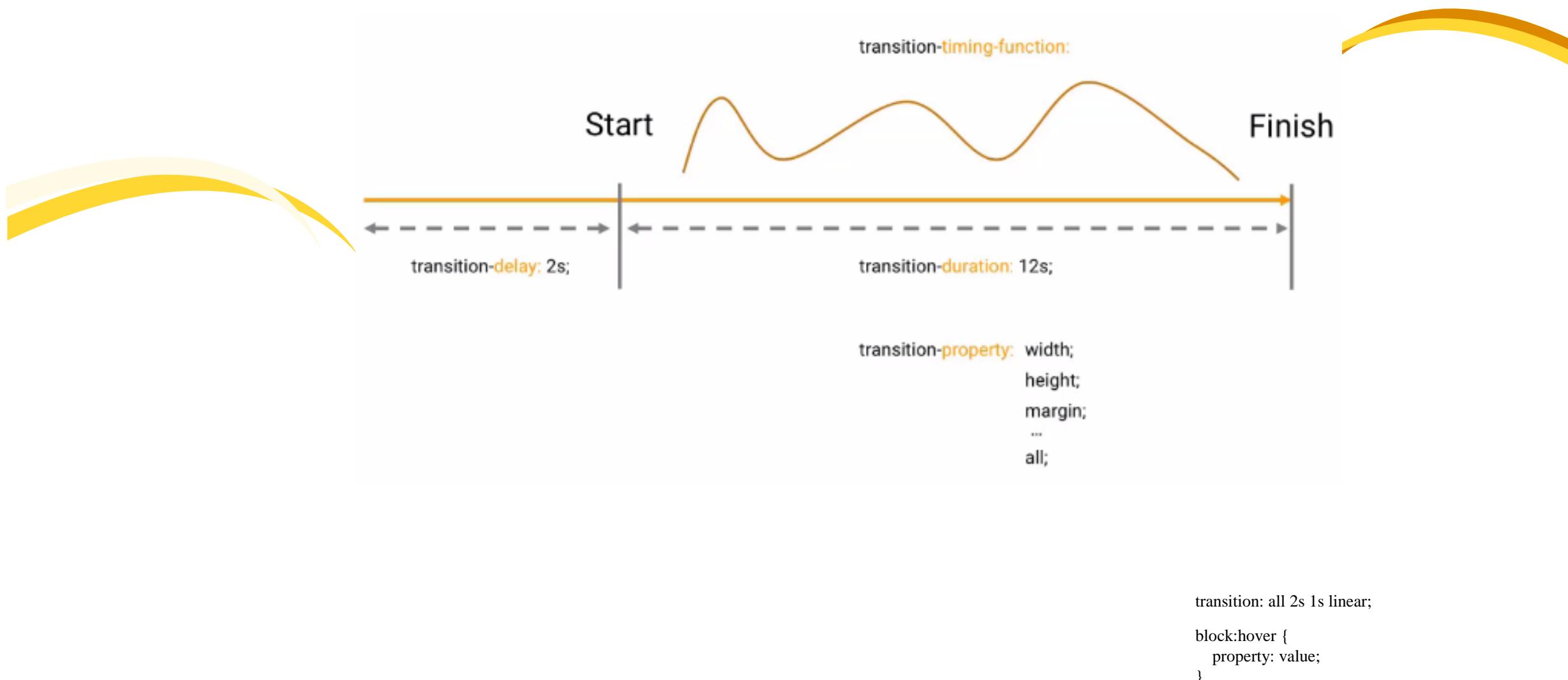


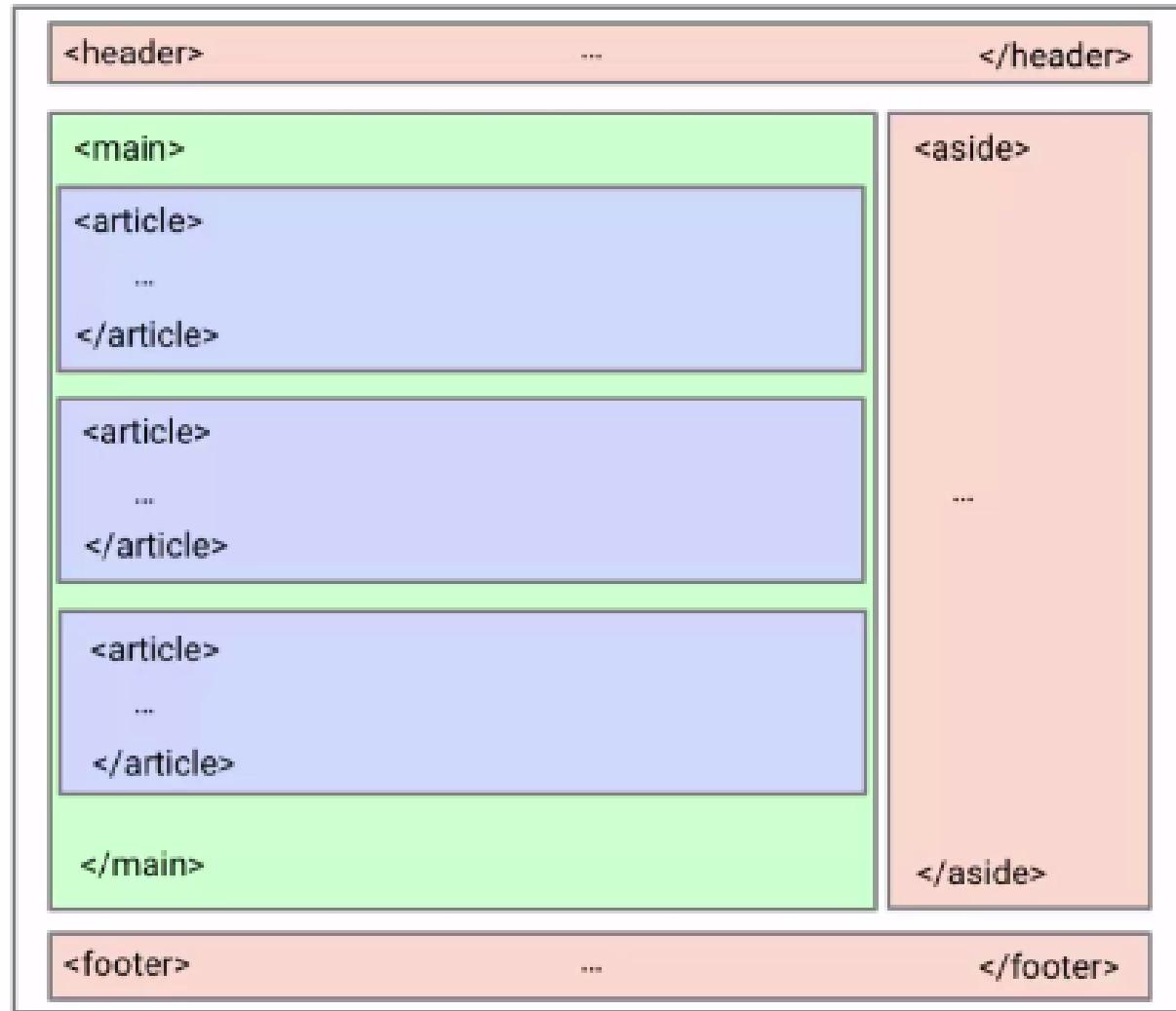


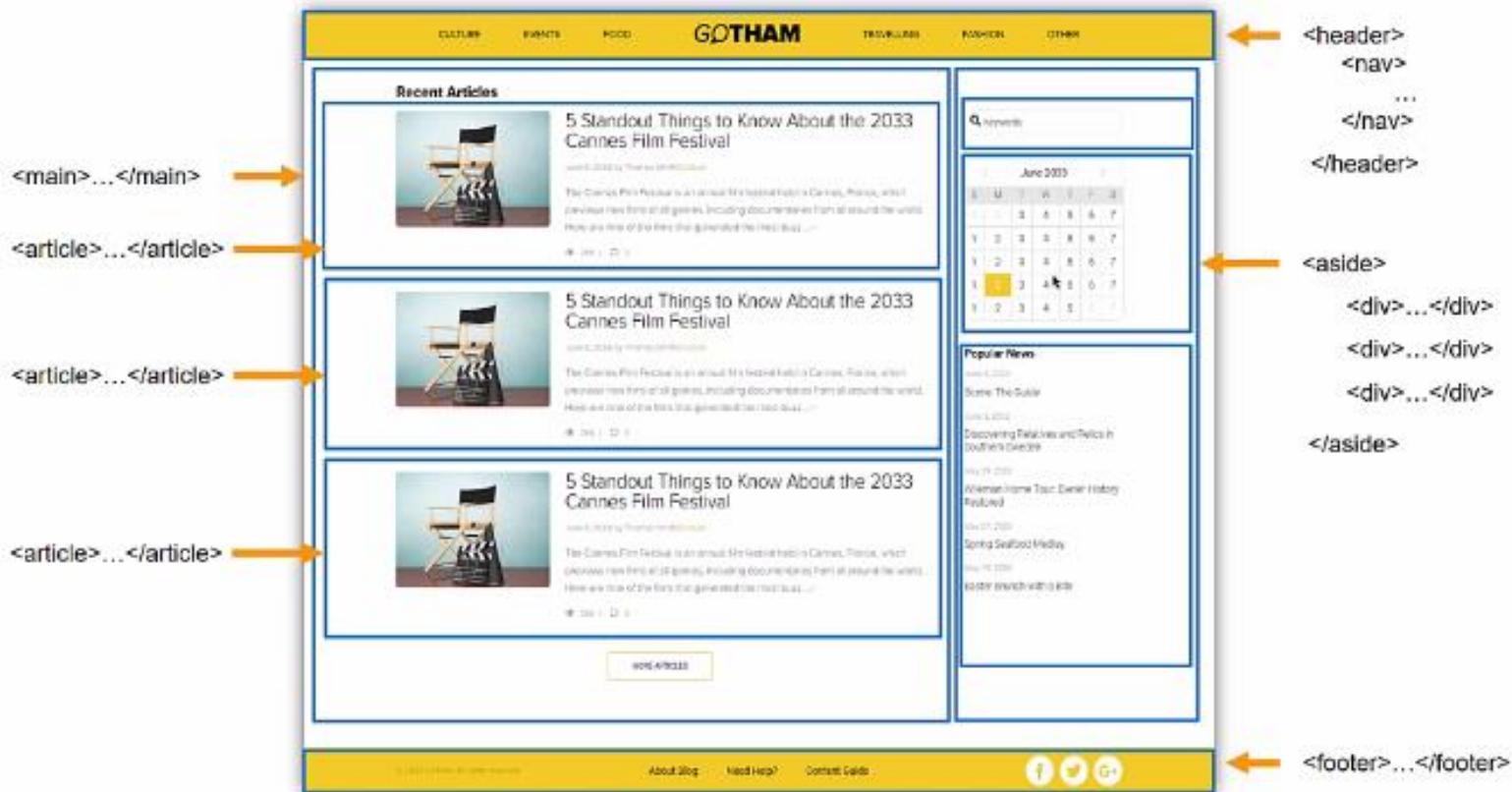
Transition

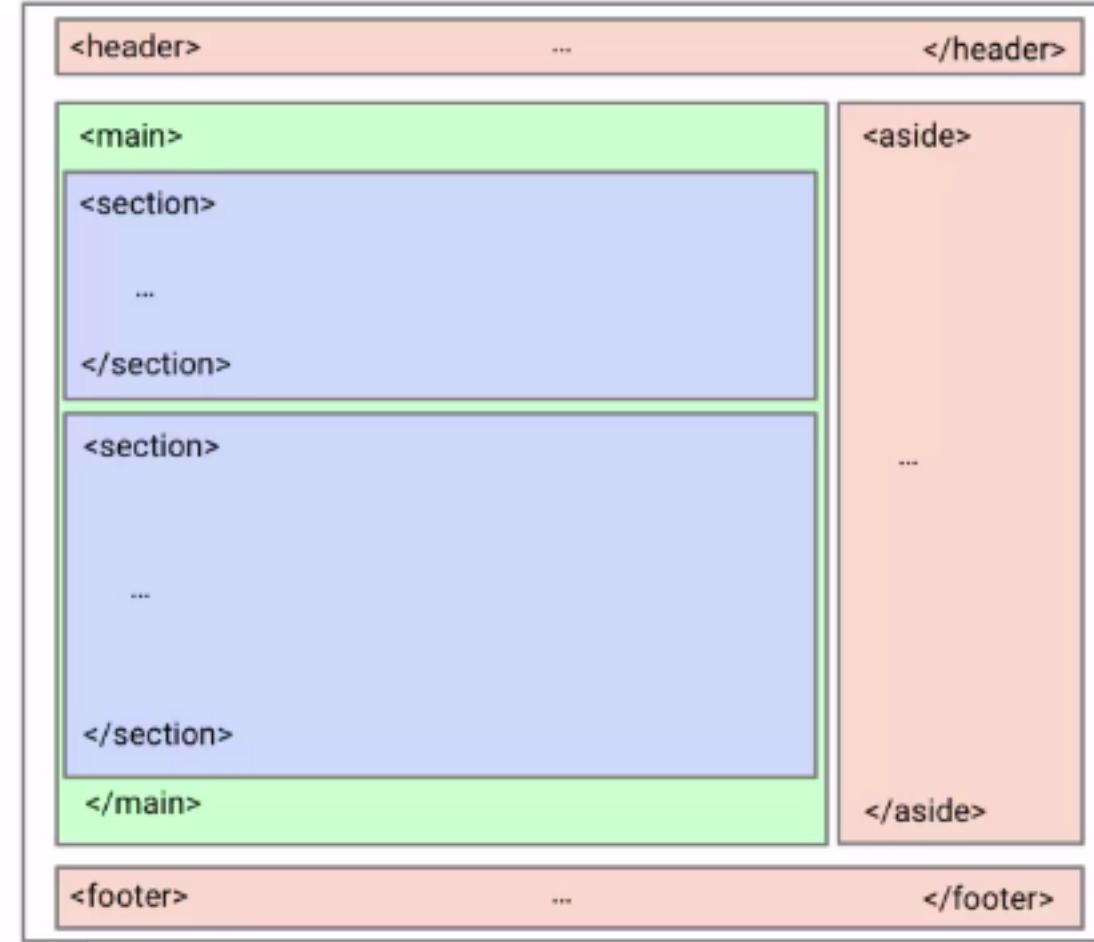
- 
- 1. transition-property :hover
 - 2. transition-duration :focus
 - 3. transition-delay :active
 - 4. transition-timing-function

+









<https://edition.cnn.com/>