

Color – Hex / RGB / HSL – Multimedia Technology 2020 (Lab)

Wneel

All Sections (6)

- Introduction
- 1. สีและการบ่งชีพ
- 2. RGB ล้ำแสงสานสืบห้องเรียน
- 3. เมล็ด RGB เป็นก่อสร้างฯ

Raspberry 330° 255-0-125

Magenta 300° 255-0-255

Violet 270° 125-0-255

Blue 240° 0-0-255

Ocean 210° 0-125-255

Cyan 180° 0-125-255

Green 120° 0-255-0

Turquoise 150° 0-255-125

Yellow 60° 255-255-0

Orange 30° 255-125-0

White 0° 255-0-0

▶ น้ำตาลเปลือกไม้

ลองผสมในหัวใจเพื่อเข้าใจ RGB ค่ะ

</> MT20-Unit2-Color-1 Jabont

```

1 <style>
2   body{
3     background:rgb(0,0,0);
4   }
5 </style>
6

```

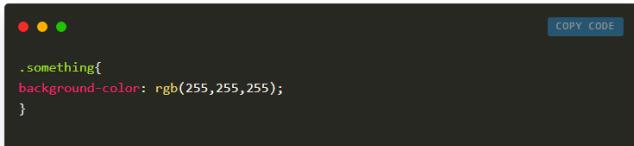
◀ Color – Hex / RGB / HSL 🔍 – Multimedia Technology 2020 (Lab)



ดังนั้น นักประดิษฐ์ก็ได้ประดิษฐ์หน้าจอที่เราใช้ ซึ่งจริงๆ แต่ละจอดีคือความสว่างจริงๆ ไม่เหมือนกันหรอกนะ แต่ที่ให้เข้าใจตรงกันว่าต้องการสีสีอย่างไรก็ตาม เราจะมีมาตรฐาน RGB ที่ได้บอกไว้ว่าใช้ความสว่าง 256 ระดับ ถ้าแต่ 0-255

ดังนั้นมีสีทั้งหมด $256 * 256 * 256$ (R^*G*B) เท่ากับ $16,777,216$ หรือประมาณสิบหลักน้ำสีที่เราได้อ่านบ่อยๆ นั้นเองงงๆ

ซึ่งวิธีการเขียน CSS ที่เขียนได้จะเป็นดังนี้ค่ะ

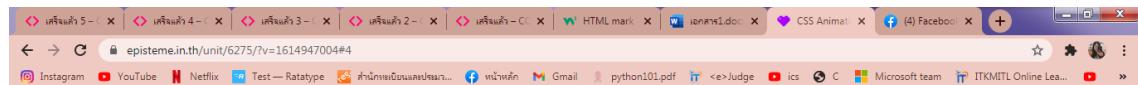


```
.something{  
background-color: rgb(255,255,255);  
}
```

ด้านบนนี้ จะเท่ากับสีขาว เพราะทุกแสงสว่างถึงขีดสูงสุดทุกแสงเลยค่ะ

ดังนั้น ถ้าต้องการเป็นสีแดง ก็จะได้เท่ากับ $rgb(255,0,0)$ แบบนี้ค่ะ

7:17 PM
3/5/2021



< CSS Animation Transform & Transition – Multimedia Technology 2020 (Lab)

All Sections (5)

- Introduction
- ▶ 1. Animation และ Transition บนเว็บทำงานยังไงนะ
- ▶ 2. มาเรียนจาก Transition กันบีกกว่า
- 3. มาสร้างกับ cubic-bezier กันต่ออethylene

Week 3 – Transform

```
1 <html>
2   <head>
3     <title>Let's Start
4   </title>
5   <style>
6     body{
7       background:#fff;
8       margin:0;
9       font-family:sans-serif;
10      display:flex;
11      justify-content:center;
12      align-items:center;
13    }
14    .box{
15      box-sizing:border-box;
16    }
17    .box{
18      position:absolute;
19      width:50vw;
20      height:50vw;
21      background:#f4f4f4;
22      border:4px solid #ddd; /* transform:translateY(0vw) rotate(-30deg) skew(30deg); */
23      transition:all 1s;
24      border-radius:2vw;
25      cursor:zoom-in;
26    }
27    .element{
28      background:#ddd;
29    }
```

7:32 PM
3/5/2021

< CSS Animation Transform & Transition – Multimedia Technology 2020 (Lab)

All Sections (5)

- Introduction
- ▶ 1. Animation และ Transition บนเว็บทำงานยังไงนะ
- ▶ 2. มาเรียนจาก Transition กันบีกกว่า
- 3. มาสร้างกับ cubic-bezier กันต่ออethylene

Week 3 – Transform

```
31   width:100%;
32   height:100%;
33   position: absolute;
34   transition: all .3s;
35   border-radius:2vw;
36 }
37 .box:hover{
38   transform:translateY(10vw)
39   rotate(-30deg) skew(30deg);
40 ;
41 }
42 .e1{
43   background:hsl(270,100%,60%,75%);
44   hsl(270,100%,60%,100%);
45 }
46 .e2{
47   background:hsl(270,80%,60%,75%);
48   border:4px solid
49   hsl(270,80%,60%,100%);
50 }
51 .e2{
52   background:hsl(270,60%,60%,75%);
53   border:4px solid
54   hsl(270,60%,60%,100%);
55 }
56 .e1{
57   background:hsl(270,40%,60%,75%);
58   border:4px solid
59   hsl(270,40%,60%,100%);
```

7:32 PM
3/5/2021

< CSS Animation Transform & Transition – Multimedia Technology 2020 (Lab)

All Sections (5)

- Introduction
- ▶ 1. Animation และ Transition บนเว็บทำงานยังไงนะ
- ▶ 2. มาเรียนจาก Transition กันบีกกว่า
- 3. มาสร้างกับ cubic-bezier กันต่ออethylene

Week 3 – Transform

```
56   border:4px solid
57   hsl(270,40%,60%,100%);
58 }
59 .box:hover .e1{
60   transform:translate(4vw,-4vw);
61 }
62 .box:hover .e2{
63   transform:translate(8vw,-8vw);
64 }
65 .box:hover .e3{
66   transform:translate(12vw,-12vw);
67 }
68 .box:hover .e4{
69   transform:translate(16vw,-16vw);
70 }
71 </style>
72 </head>
73 <body>
74   <div class="box">
75     <div class="element e1">
76     </div>
77     <div class="element e2">
78     </div>
79     <div class="element e3">
80     </div>
81     <div class="element e4">
82     </div>
83   </div>
84 </body>
85 </html>
```

7:32 PM
3/5/2021

```

1 <html>
2   <head>
3     <style>
4       body{
5         font-size:0;
6         display:flex;
7         flex-wrap:wrap;
8         margin:0;
9       }
10      div{
11        color:purple;
12        width:20%;
13        height:20%;
14        /* ลักษณะ */
15        background:hsla(109, 100%, 40%, 0.5);
16      }
17      div:nth-child(even) {
18        background:hsla(109, 100%, 40%, 1);
19      }
20    </style>
21  </head>
22  <body>
23    <!-- ห้ามแก้ HTML -->
24    <div>
25    </div>
26    <div>
27    </div>
28    </div>
29  </body>

```

File menu: E4.jpg, E3.jpg (Cancelled), E2.jpg, E1.jpg (Cancelled)

Bottom bar: G Login with Google, Show all, EN, 6:59 PM, 3/5/2021

Left side: 100 seconds of CSS ANIMATION video player. Buttons: All Sections (5), Introduction, 1. Animation และ Transition บนเว็บทำงานยังไงบ้าง, 2. ในการเขียน Transition กันตื้อกว่า, 3. ทำรูปจั๊กจั๊ก cubic-bezier กันต่ออีก

Right side: Week 3 - Animation code editor. Preview shows yellow squares moving in a circular path.

```

1 <html>
2   <head>
3     <title>Let's Start</title>
4     <style>
5       body{
6         background:#001;
7         margin:0;
8       }
9       @keyframes kraprib{
10         from{opacity:.2}
11         to{opacity:1}
12       }
13       @keyframes moon{
14         from{transform:rotate(0deg)}
15         to{transform:rotate(360deg)}
16       }
17       .star{
18         position:absolute;
19         width:1em;
20         height:1em;
21         background:yellow;
22         opacity:1;
23         box-shadow:0px 0px 20px 10px #ff03;
24       }
25       .star:nth-child(1){
26         top:30%;
27         left:80%;
28         animation:kraprib 1s infinite
29           alternate,moon 1s infinite linear;
30       }

```

Bottom bar: G Login with Google, TH, 7:25 PM, 3/5/2021

◀ Introduction to HTML and CSS – Multimedia Technology 2020 (Lab)

```

<style>
body{
margin:0;
background:#eee;
}
.small{
/* ຂາດເລື້ອງ */
width:40px;
height:40px;
background:pink;
}
.big{
/* ຂາດໃຫຍ່ */
width:100px;
height:100px;
}
.circle{
/* ເປັນກອນ*/
border-radius:100%;
background:green;
}
.red{
/* ສີແດງ */
background:red;
}
.yellow{
/* ສີເພີດອົງ */
background:yellow;
}
</style>

```

◀ CSS Animation Transform & Transition – Multimedia Technology 2020 (Lab)

```

<html>
<head>
<title>Let's Start</title>
<style>
body{
margin:0;
}
.box{
width:20%;
height:10%;
background:blueviolet;
}
/* ພັນລົວທັງໝົດເຊື້ອໂຄນາ ເພຣະຫອວນ */
.box:hover{
background:gold;
/* ພັກກອນກາ */
width:90%;
}
</style>
</head>
<body>


</div>
<div class="box"></div>
<div class="box"></div>
<div class="box"></div>
<div class="box"></div>


```

Color – Hex / RGB / HSL – Multimedia Technology 2020 (Lab)

All Sections (6)

HTML name	RGB		HTML name	RGB		HTML name	RGB	
	Hex	Decimal		Hex	Decimal		Hex	Decimal
Pink colors								
MediumVioletRed	C7 15 85	199 21 133	DarkGreen	00 64 00	0 100 0	Indigo	4B 00 82	75 0 130
DeepPink	FF 14 93	255 20 147	Green	00 80 00	0 128 0	Purple	80 00 80	128 0 128
PaleVioletRed	DB 70 93	219 112 147	DarkOliveGreen	55 68 2F	85 107 47	DarkMagenta	8B 00 8B	139 0 139
HotPink	FF 69 B4	255 105 180	ForestGreen	22 88 22	34 139 34	DarkViolet	94 00 D3	148 0 211
LightPink	FF B6 C1	255 182 193	SeaGreen	28 88 57	46 139 87	DarkSlateBlue	4B 3D 8B	72 61 139
Pink	FF C0 CB	255 192 203	Olive	80 80 00	128 128 0	BlueViolet	8A 2B E2	138 43 226
Red colors								
DarkRed	8B 00 00	139 0 0	OliveDrab	6B 88 23	107 142 35	DarkOrchid	99 32 C0	153 50 204
Red	FF 00 00	255 0 0	MediumSeaGreen	3C B3 71	60 179 113	Fuchsia	FF 00 FF	255 0 255
Firebrick	B2 22 22	178 34 34	LimeGreen	32 CD 32	50 205 50	Magenta	FF 00 FF	255 0 255
Crimson	DC 14 3C	220 20 60	Lime	00 FF 00	0 255 0	SlateBlue	6A 5A CD	106 90 205
IndianRed	CD 5C 5C	205 92 92	SpringGreen	00 FF 7F	0 255 127	MediumSlateBlue	7B 68 EE	123 104 238
LightCoral	F0 80 80	240 128 128	MediumSpringGreen	00 FA 9A	0 250 154	MediumOrchid	9B 55 D3	186 85 211
Salmon	FA 80 72	250 128 114	DarkSeaGreen	8F BC 8F	143 188 143	MediumPurple	93 70 DB	147 112 219
DarkSalmon	E9 96 7A	233 150 122	MediumAquamarine	66 CD AA	102 205 170	Orchid	DA 70 DE	218 112 214
LightSalmon	FF A0 7A	255 160 122	YellowGreen	9A CD 32	154 205 50	Violet	EE 82 EE	238 130 238
			LawnGreen	7C FC 00	124 252 0	Plum	DD A0 DD	221 160 221

◀ CSS Animation Transform & Transition – Multimedia Technology 2020 (Lab)



คลิปนี้จะเป็นการลองใช้ value ต่างๆ กันออกไปในค่า transform นี่จะแตกต่างกับข้าวบ้านเดาเบอยเล่ย์คือ ปกติข้าวบ้านเดาเบอยเล่ย์ค่าสัมประสิทธิ์เป็นค่าคงที่หรือคงที่ ค่าผ่านการสีแบบคอมมา(.) หรือถ้าเขียนหลายๆ property มาเป็น property แนว property (เดียว) แต่ทราบส์ฟอร์มหนึ่นเป็นเมื่ອในการอกร่างใจใช้ property ย่อยๆ ของ transform มากกว่า ที่นี่จะใช้กันนี้คือ

- ▶ scale() หมายถึงย่อหรือขยายรูปสิ่งที่มีนี่ 1 คือ 1 เท่า 0.5 คือเล็กลงซึ่งนี่ 2 คือขยาย 2 เท่า เป็นต้นนี้จะ เวลาใช้ก็ scale(2) แบบนี้คือ หรือจะใช้แค่ X หรือ Y ก็ได้ ลองเดาเว็บดูนะครับ ซึ่งขึ้นต้องการใช้ scale คือ ภาพจะลดมูลลงไม่ กว่าตอนมันยัง แข็งของข่ายตัวจากจุดศูนย์กลาง หรือจุดไหนก็ได้หากเราทำขนาดจุดไว้คือ ทำให้มีกระหطم element อีก ดูว่าอย่าที่นี่
- ▶ rotate() อันนี้เหมือนค่า มีทั้งหมุนแกน X Y และ Z เลย แต่ค่าเริ่มต้นจะเป็นแกน Z คือ ปกติเราจะใช้แกนนี้บัน เวลาใส่ค่า ในวงล้อสามารถใช้ deg rad หรือ turn ก็ได้คือ อย่างตัวต่างกันยังไง ลองดู [ดูว่าอย่างตรงนี้](#) ได้เลย
- ▶ skew() ทำให้ห้องร้องคาย เรายสามารถที่ก้าวเดินไปให้เป็นสี่เหลี่ยมแบบเอียงๆ ได้ ทำรูปช้าๆ ตามตัด รวมทั้งห้องสระวังภาพ สามมิติหลอกตาด้วยอีกหนึ่งส่วนนี่ให้ได้ก้าวคือ [ดูว่าอย่างนี่](#)
- ▶ translate() อันนี้ก้าวได้สิ่งคือ ครอบจางจะอยู่ให้แนวเดือนไปที่อื่น มีหลายวิธี แต่ที่เรียนจะไม่กระหطمตัวอื่นๆ ตัวคือ คูณเพิ่มๆ [ที่นี่เลย](#)
- ▶ อันนี้ เช่น matrix อาจจะยกไป เพราต้องจำได้เหมือนเดือนคือขอไร่ ไม่ต้องได้ใช้ค่า
- ▶ อันนี้แรกออกแบบ คือ transform-origin คือการกำหนดจุดเริ่มต้นของการทราบส์ฟอร์มคือ เราอาจจะอยากหมุนวัดดู จากจุดมุ่งด้านขวาหากว่าก้าวได้ [ดูว่าอย่างนี่เลย](#)

Note!

The screenshot shows a web browser window with multiple tabs open. The active tab is titled "Color – Hex / RGB / HSL – Multimedia Technology 2020 (Lab)". On the left, there is a color wheel with a play button in the center, and below it, a list of "All Sections (6)" including "Introduction", "1. สีและการของกิน", "2. RGB สำหรับสื่อสาร", and "3. เปรียบเทียบ RGB". On the right, a code editor displays the following CSS code:

```
<style>
body{
background:#333;
margin:0;
}
div{
width:100%;
height:10vh;
background:ccc;
}
/* ไล่สีตามลักษณะสี ลักษณะสีที่มีความสัมภัย */
|
</style>
<div style="background:hsl(230,100%,50%)>
</div>
<div style="background:hsl(430,100%,50%)>
</div>
<div style="background:hsl(830,100%,50%)>
</div>
<div style="background:hsl(730,100%,50%)>
</div>
<div style="background:hsl(330,100%,50%)>
</div>
<div style="background:hsl(330,100%,50%)>
</div>
<div style="background:hsl(330,100%,50%)>
</div>
```

All About Web Positioning – Epis... x episteme.in.th/unit/140/?v=1614947622#4

Instagram YouTube Netflix Test — Ratatype ก้าวเดินและประเมิน... Gmail python101.pdf <e>Judge ics C Microsoft team ITKMIT Online Lea... YouTube

นาย A

HTML

```
<div>นาย A</div>
```

CSS

```
div{ position: absolute; top:90px; left: 120px; }
div{ background: pink; width: 100px; line-height: 100px; }
```

JS

All Sections (9)

- Introduction
- 1. การอ้างอิง และจุดอ้างอิง
- 2. Position Relative
- 3. Position Absolute ใน element รุ่นที่ - 1

EN 7:34 PM 3/5/2021



ดังนั้น นักประดิษฐ์ก็ได้ประดิษฐ์หน้าจอที่เราใช้ ซึ่งจริงๆ แต่ละจอ ก็ให้ความสว่างจริงๆ ไม่เหมือนกันหรอกค่ะ แต่เพื่อให้เข้าใจตรงกันว่าต้องการส่งสีอะไรออกมานะ เราจึงมีมาตรฐาน RGB ที่ได้บอกไว้ว่า ให้ความสว่าง 256 ระดับ ตั้งแต่ 0-255

ดังนั้นจะมีสีทั้งหมด $256 * 256 * 256$ ($R * G * B$) เท่ากับ 16,777,216 หรือประมาณสิบหกล้านสี ที่เราได้ยินบ่อยๆ นั้นเองงง ซึ่งวิธีการเขียน CSS ก็เขียนได้ดังนี้ค่ะ

```
.something{
background-color: rgb(255,255,255);}
```

COPY CODE

ด้านบนนี้ จะเท่ากับสีขาว เพราะทุกแสงสว่างถึงขีดสูงสุดทุกแสงเลยค่ะ

ดังนั้น ถ้าต้องการเป็นสีแดง ก็จะได้เท่ากับ $rgb(255,0,0)$ แบบนี้ค่ะ

Color – Hex / RGB / HSL – Multimedia Technology 2020 (Lab)



▶ น้ำตาลเปลือกฟ้า

ลองผสมในทั่วๆ เอื้อไปใน RGB ค่า

MT20-Unit2-Color-1 Jabont

```
<style>
body{
background:rgb(0,0,0);
}
</style>
```

All Sections (6)

- Introduction
- 1. สีและการมองเห็น
- 2. RGB สำหรับสีและสีที่มีอยู่
- 3. การแปลงสี RGB เป็นสี CMYK

TH 7:18 PM 3/5/2021

This element has a margin of 70px.

[Try it Yourself »](#)

CSS Margins

The CSS `margin` properties are used to create space around elements, outside of any defined borders.

With CSS, you have full control over the margins. There are properties for setting the margin for each side of an element (top, right, bottom, and left).

Margin - Individual Sides

CSS has properties for specifying the margin for each side of an element:

- `margin-top`
- `margin-right`
- `margin-bottom`
- `margin-left`

The rotateZ() Method

The `rotateZ()` method rotates an element around its Z-axis at a given degree:

Example

```
#myDiv {  
  transform: rotateZ(90deg);  
}
```

CSS Radial Gradients

A radial gradient is defined by its center.

To create a radial gradient you must also define at least two color stops.

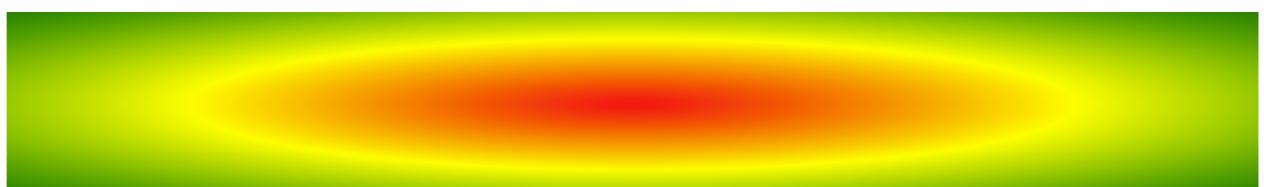
Syntax

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

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

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

The following example shows a radial gradient with evenly spaced color stops:



Specify the Speed Curve of the Animation

The `animation-timing-function` property specifies the speed curve of the animation.

The `animation-timing-function` property can have the following values:

- `ease` - Specifies an animation with a slow start, then fast, then end slowly (this is default)
- `linear` - Specifies an animation with the same speed from start to end
- `ease-in` - Specifies an animation with a slow start
- `ease-out` - Specifies an animation with a slow end
- `ease-in-out` - Specifies an animation with a slow start and end
- `cubic-bezier(n,n,n,n)` - Lets you define your own values in a cubic-bezier function

The following example shows some of the different speed curves that can be used:

CSS Transform Properties

The following table lists all the 3D transform properties:

Property	Description
<code>transform</code>	Applies a 2D or 3D transformation to an element
<code>transform-origin</code>	Allows you to change the position on transformed elements
<code>transform-style</code>	Specifies how nested elements are rendered in 3D space
<code>perspective</code>	Specifies the perspective on how 3D elements are viewed
<code>perspective-origin</code>	Specifies the bottom position of 3D elements
<code>backface-visibility</code>	Defines whether or not an element should be visible when not facing the screen

CSS Border - Shorthand Property

Like you saw in the previous page, there are many properties to consider when dealing with borders.

To shorten the code, it is also possible to specify all the individual border properties in one property.

The `border` property is a shorthand property for the following individual border properties:

- `border-width`
- `border-style` (required)
- `border-color`

Specify the fill-mode For an Animation

CSS animations do not affect an element before the first keyframe is played or after the last keyframe is played. The `animation-fill-mode` property can override this behavior.

The `animation-fill-mode` property specifies a style for the target element when the animation is not playing (before it starts, after it ends, or both).

The `animation-fill-mode` property can have the following values:

- `none` - Default value. Animation will not apply any styles to the element before or after it is executing
- `forwards` - The element will retain the style values that is set by the last keyframe (depends on `animation-direction` and `animation-iteration-count`)
- `backwards` - The element will get the style values that is set by the first keyframe (depends on `animation-direction`), and retain this during the `animation-delay` period
- `both` - The animation will follow the rules for both forwards and backwards, extending the animation properties in both directions

position: relative;

An element with `position: relative;` is positioned relative to its normal position.

Setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position. Other content will not be adjusted to fit into any gap left by the element.

This `<div>` element has `position: relative;`:

Here is the CSS that is used:

Example

```
div.relative {  
  position: relative;  
  left: 30px;  
  border: 3px solid #73AD21;  
}
```

[Try it Yourself »](#)

iv> elem

More Transition Examples

The CSS transition properties can be specified one by one, like this:

Example

```
div {  
  transition-property: width;  
  transition-duration: 2s;  
  transition-timing-function: linear;  
  transition-delay: 1s;  
}
```

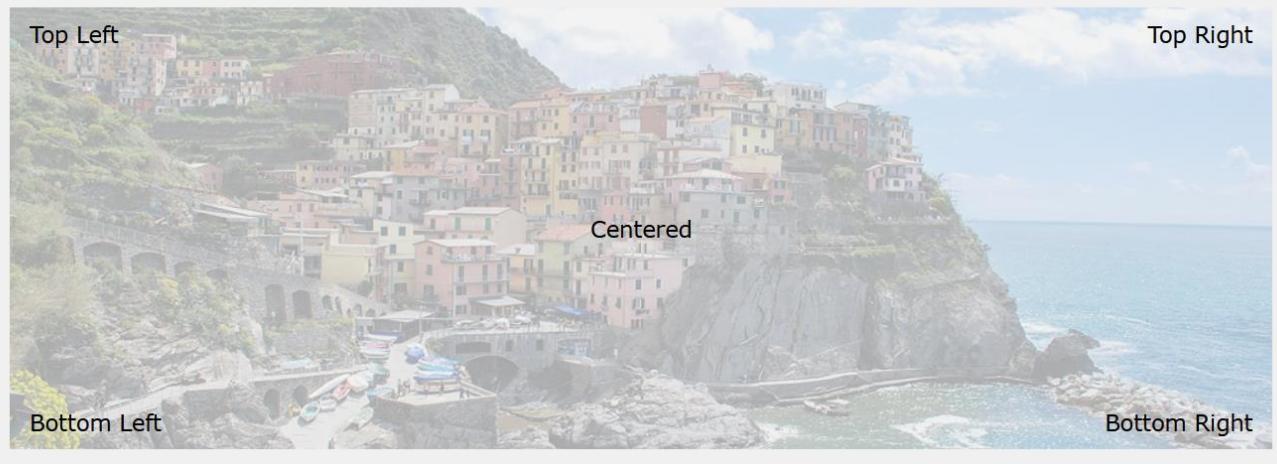
CSS 2D Transform Methods

Function	Description
matrix(<i>n,n,n,n,n,n</i>)	Defines a 2D transformation, using a matrix of six values
translate(<i>x,y</i>)	Defines a 2D translation, moving the element along the X- and the Y-axis
translateX(<i>n</i>)	Defines a 2D translation, moving the element along the X-axis
translateY(<i>n</i>)	Defines a 2D translation, moving the element along the Y-axis
scale(<i>x,y</i>)	Defines a 2D scale transformation, changing the elements width and height
scaleX(<i>n</i>)	Defines a 2D scale transformation, changing the element's width
scaleY(<i>n</i>)	Defines a 2D scale transformation, changing the element's height
rotate(<i>angle</i>)	Defines a 2D rotation, the angle is specified in the parameter
skew(<i>x-angle,y-angle</i>)	Defines a 2D skew transformation along the X- and the Y-axis
skewX(<i>angle</i>)	Defines a 2D skew transformation along the X-axis
skewY(<i>angle</i>)	Defines a 2D skew transformation along the Y-axis

Positioning Text In an Image

How to position text over an image:

Example



Run Animation in Reverse Direction or Alternate Cycles

The `animation-direction` property specifies whether an animation should be played forwards, backwards or in alternate cycles.

The `animation-direction` property can have the following values:

- `normal` - The animation is played as normal (forwards). This is default
- `reverse` - The animation is played in reverse direction (backwards)
- `alternate` - The animation is played forwards first, then backwards
- `alternate-reverse` - The animation is played backwards first, then forwards

The following example will run the animation in reverse direction (backwards):

position: sticky;

An element with `position: sticky;` is positioned based on the user's scroll position.

A sticky element toggles between `relative` and `fixed`, depending on the scroll position. It is positioned relative until a given offset position is met in the viewport - then it "sticks" in place (like `position:fixed`).

Note: Internet Explorer does not support sticky positioning. Safari requires a `-webkit-` prefix (see example below). You must also specify at least one of `top`, `right`, `bottom` or `left` for sticky positioning to work.

This `<div>` element

Function	Description
<code>matrix3d (n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n)</code>	Defines a 3D transformation, using a 4x4 matrix of 16 values
<code>translate3d(x,y,z)</code>	Defines a 3D translation
<code>translateX(x)</code>	Defines a 3D translation, using only the value for the X-axis
<code>translateY(y)</code>	Defines a 3D translation, using only the value for the Y-axis
<code>translateZ(z)</code>	Defines a 3D translation, using only the value for the Z-axis
<code>scale3d(x,y,z)</code>	Defines a 3D scale transformation
<code>scaleX(x)</code>	Defines a 3D scale transformation by giving a value for the X-axis
<code>scaleY(y)</code>	Defines a 3D scale transformation by giving a value for the Y-axis
<code>scaleZ(z)</code>	Defines a 3D scale transformation by giving a value for the Z-axis
<code>rotate3d(x,y,z,angle)</code>	Defines a 3D rotation
<code>rotateX(angle)</code>	Defines a 3D rotation along the X-axis
<code>rotateY(angle)</code>	Defines a 3D rotation along the Y-axis
<code>rotateZ(angle)</code>	Defines a 3D rotation along the Z-axis
<code>perspective(n)</code>	Defines a perspective view for a 3D transformed element

The screenshot shows a browser window displaying the w3schools.com website. The URL in the address bar is https://www.w3schools.com/css/css3_3dtransforms.asp. The page title is "CSS 3D Transform Methods". On the left, there is a sidebar with a navigation menu for CSS topics, including "CSS 3D Transforms" which is currently selected. The main content area contains a table with 16 rows, each representing a different 3D transform function and its description. The functions listed are: matrix3d, translate3d(x,y,z), translateX(x), translateY(y), translateZ(z), scale3d(x,y,z), scaleX(x), scaleY(y), scaleZ(z), rotate3d(x,y,z,angle), rotateX(angle), rotateY(angle), rotateZ(angle), and perspective(n).

Function	Description
matrix3d (n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n)	Defines a 3D transformation, using a 4x4 matrix of 16 values
translate3d(x,y,z)	Defines a 3D translation
translateX(x)	Defines a 3D translation, using only the value for the X-axis
translateY(y)	Defines a 3D translation, using only the value for the Y-axis
translateZ(z)	Defines a 3D translation, using only the value for the Z-axis
scale3d(x,y,z)	Defines a 3D scale transformation
scaleX(x)	Defines a 3D scale transformation by giving a value for the X-axis
scaleY(y)	Defines a 3D scale transformation by giving a value for the Y-axis
scaleZ(z)	Defines a 3D scale transformation by giving a value for the Z-axis
rotate3d(x,y,z,angle)	Defines a 3D rotation
rotateX(angle)	Defines a 3D rotation along the X-axis
rotateY(angle)	Defines a 3D rotation along the Y-axis
rotateZ(angle)	Defines a 3D rotation along the Z-axis
perspective(n)	Defines a perspective view for a 3D transformed element

The @keyframes Rule

When you specify CSS styles inside the `@keyframes` rule, the animation will gradually change from the current style to the new style at certain times.

To get an animation to work, you must bind the animation to an element.

The following example binds the "example" animation to the `<div>` element. The animation will last for 4 seconds, and it will gradually change the background-color of the `<div>` element from "red" to "yellow":

Example

```
/* The animation code */
@keyframes example {
  from {background-color: red;}
  to {background-color: yellow;}
}

/* The element to apply the animation to */
div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
}
```

CSS Border Style

The `border-style` property specifies what kind of border to display.

The following values are allowed:

- `dotted` - Defines a dotted border
- `dashed` - Defines a dashed border
- `solid` - Defines a solid border
- `double` - Defines a double border
- `groove` - Defines a 3D grooved border. The effect depends on the border-color value
- `ridge` - Defines a 3D ridged border. The effect depends on the border-color value
- `inset` - Defines a 3D inset border. The effect depends on the border-color value
- `outset` - Defines a 3D outset border. The effect depends on the border-color value
- `none` - Defines no border
- `hidden` - Defines a hidden border

The `border-style` property can have from one to four values (for the top border, right border, bottom border, and the left border).

All CSS Positioning Properties

Property	Description
<code>bottom</code>	Sets the bottom margin edge for a positioned box
<code>clip</code>	Clips an absolutely positioned element
<code>left</code>	Sets the left margin edge for a positioned box
<code>position</code>	Specifies the type of positioning for an element
<code>right</code>	Sets the right margin edge for a positioned box
<code>top</code>	Sets the top margin edge for a positioned box
<code>z-index</code>	Sets the stack order of an element

All CSS Dimension Properties

Property	Description
<code>height</code>	Sets the height of an element
<code>max-height</code>	Sets the maximum height of an element
<code>max-width</code>	Sets the maximum width of an element
<code>min-height</code>	Sets the minimum height of an element
<code>min-width</code>	Sets the minimum width of an element
<code>width</code>	Sets the width of an element

Using Angles

If you want more control over the direction of the gradient, you can define an angle, instead of the predefined directions (to bottom, to top, to right, to left, to bottom right, etc.). A value of 0deg is equivalent to "to top". A value of 90deg is equivalent to "to right". A value of 180deg is equivalent to "to bottom".

Syntax

```
background-image: linear-gradient(angle, color-stop1, color-stop2);
```

Text shadow effect!

Example

```
h1 {  
  color: white;  
  text-shadow: 1px 1px 2px black, 0 0 25px blue, 0 0 5px darkblue;  
}
```

[Try it Yourself »](#)

You can also use the text-shadow property to create a plain border around some text (without shadows):

Border around text!

Example

```
h1 {  
  color: yellow;  
  text-shadow: -1px 0 black, 0 1px black, 1px 0 black, 0 -1px black;  
}
```

CSS Transform Properties

The following table lists all the 2D transform properties:

Property	Description
<u>transform</u>	Applies a 2D or 3D transformation to an element
<u>transform-origin</u>	Allows you to change the position on transformed elements

Next, add a blur effect to the shadow:

This is a yellow <div> element
with a blurred, grey box-shadow

Example

```
div {  
  box-shadow: 10px 10px 5px grey;  
}
```

`position: fixed;`

An element with `position: fixed;` is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled. The top, right, bottom, and left properties are used to position the element.

A fixed element does not leave a gap in the page where it would normally have been located.

Notice the fixed element in the lower-right corner of the page. Here is the CSS that is used:

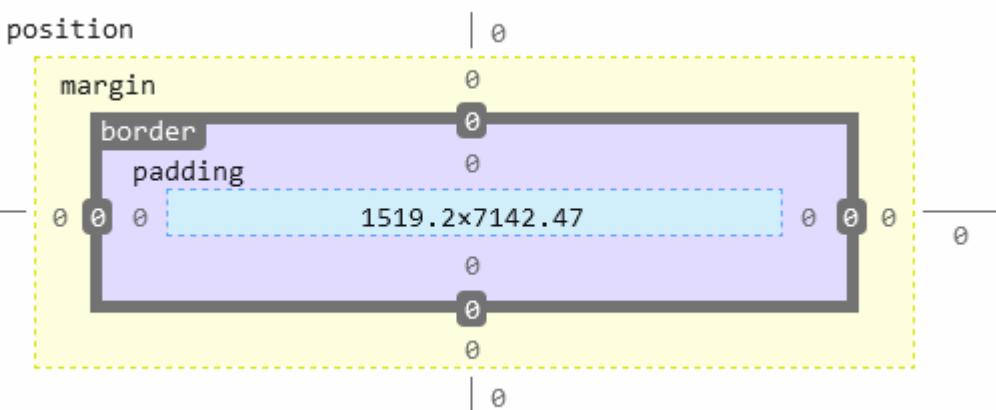
Example

```
div.fixed {  
  position: fixed;  
  bottom: 0;  
  right: 0;  
  width: 300px;  
  border: 3px solid #73AD21;  
}
```

[Try it Yourself »](#)

This <div> elem

QUESTION



position: absolute;

An element with `position: absolute;` is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like `fixed`).

However; if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.

Note: A "positioned" element is one whose position is anything except `static`.

Here is a simple example:

This <div> element has position: relative;

This <div> element has
position: absolute;

In these chapters, you will learn about the following CSS background properties:

- `background-color`
- `background-image`
- `background-repeat`
- `background-attachment`
- `background-position`
- `background` (shorthand property)

position: static;

HTML elements are positioned static by default.

Static positioned elements are not affected by the top, bottom, left, and right properties.

An element with `position: static;` is not positioned in any special way; it is always positioned according to the normal flow of the page:

This <div> element has position: static;

Here is the CSS that is used:

Example

```
div.static {  
  position: static;  
  border: 3px solid #73AD21;  
}
```

[Try it Yourself »](#)

This <div> elen