



## Chapter 4

# 색상과 글꼴



색상



## ■ RGB 모델

- 빨강(Red), 녹색(Green), 파랑(Blue)의 세 가지 값으로 표현하는 방법
  - 숫자가 클수록 해당 색이 진함
  - 최소값 : 0(00), 최대값 : 255(FF)
- 세 가지의 표현 방법 사용
  - Hexadecimal(16진수) 지정방식
  - rgb() 함수 지정 방식
  - Predefined/Cross-browser color (브라우저 내부에서 정의된 시스템 색상) 지정방식

## ■ HSL 모델

- 색조(Hue), 채도(saturation), 밝기(Lightness/Luminance)의 세 가지 값으로 표현하는 방법
  - 색조는 값으로 표현 (최소값 : 0, 최대값 : 360)
  - 채도와 밝기는 퍼센트로 표현 (최소값 : 0%, 최대값 : 100%) 색상)





## ■ CSS3에서는 투명도를 나타내는 A(alpha)를 포함하여 색상 표현 가능

- **RGBA** : Red, Green, Blue, Alpha
- **HSLA** : Hue, Saturation, Lightness, Alpha
- **Alpha 값의 지정**
  - Alpha의 최소값 : 0 ( 완전 투명 )
  - Alpha의 최대값 : 1 ( 완전 불투명 )





## ■ Hexadecimal 지정방식

- 기호 이후에 3자리(또는 6자리)의 빨간색(r), 녹색(g), 파란색(b)의 16진수 값을 적용
  - 3자리의 표기는 #rgb 형태로 표기
  - 6자리 표기는 #rrggbb 형태로 표기
- : (예) white 색을 나타내는 #ffffff의 6자리 표현은 #fff의 3자리로 표현가능

## ■ Predefined/Cross-browser color 지정방식

- 브라우저 내부에서 정의된 색상의 이름을 직접 지정하는 방법

색상명	색상	16진수 표현	색상명	색상	16진수 표현
navy	감색	#000080	maroon	밤색	#800000
blue	파란색	#0000FF	olive	올리브색	#808000
aqua	하늘색	#00FFFF	teal	청록색	#008080
green	초록색	#008000	purple	보라색	#800080
lime	연두색	#00FF00	black	검은색	#000000
yellow	노란색	#FFFF00	gray	회색	#808080
red	빨간색	#FF0000	silver	은색	#C0C0C0
fuchsia	자주색	#FF00FF	white	흰색	#FFFFFF





## ■ rgb 지정방식

`rgb(red, green, blue)`

- red, green, blue 값은 각각 정수 값(0~255)으로 표현하며, 퍼센트(0%~100%)로도 표현 가능

- [예]

```
: #p1 { background-color : rgb(255, 0, 0) }      /* 빨간색 */  
: #p1 { background-color : rgb(100%, 50%, 0%) }   /* 오렌지색 */
```





## ■ rgba 지정방식

`rgba(red, green, blue, alpha)`

- red, green, blue 값은 각각 정수 값(0~255)으로 표현하며, 퍼센트(0%~100%)로도 표현 가능
- 투명도를 나타내는 alpha는 0~1 사이의 소수로 표현
  - 0 : 완전 투명
  - 1 : 완전 불투명
  - [예]
    - : p1 { background-color : rgba(255, 0, 0, 1) } /\* 불투명도 1을 지정 (=rgb(255, 0, 0) )\*/
    - : #p1 { background-color : rgba(255, 0, 0, 0.5) } /\* 반 투명도를 가진 빨간색 \*/
    - : #p1 { background-color : rgba(100%, 50%, 0%, 0.1) } /\* 매우 투명한 오렌지색 \*/





## ■ hsl 지정방식

- hsl(색조, 채도, 밝기)와 같이 3개의 값으로 지정한
- 색조(Hue)
  - 색상환(Color Circle)의 각도(°)로 지정
  - red=0=36, green=120, blue=240으로 표시
- 채도(saturation)
  - 색상의 선명함을 나타내며 백분율로 표시됨
  - 채도가 100%이면 순색이고 0%이면 무채색임
- 밝기(Lightness)
  - 색상의 밝기를 나타내며 백분율로 표시됨
  - 밝기가 0%이면 검은색, 100%이면 흰색, 50%이면 흰색과 검은색의 중간
  - 밝기를 광도라고도 부르기도 함
- [ 예 ]
  - : #p1 { background-color : hsl(120, 100%, 50%) } /\* 녹색 \*/
  - : #p1 { background-color : hsl(120, 100%, 75%) } /\* 밝은 녹색 \*/
  - : #p1 { background-color : hsl(120, 100%, 25%) } /\* 어두운 녹색 \*/
  - : #p1 { background-color : hsl(120, 60%, 70%) } /\* 파스텔(연하고 부드러운) 녹색 \*/
  - : #p1 { background-color : hsl(0, 100%, 50%) } /\* 빨간색 \*/







## ■ hsla 지정방식

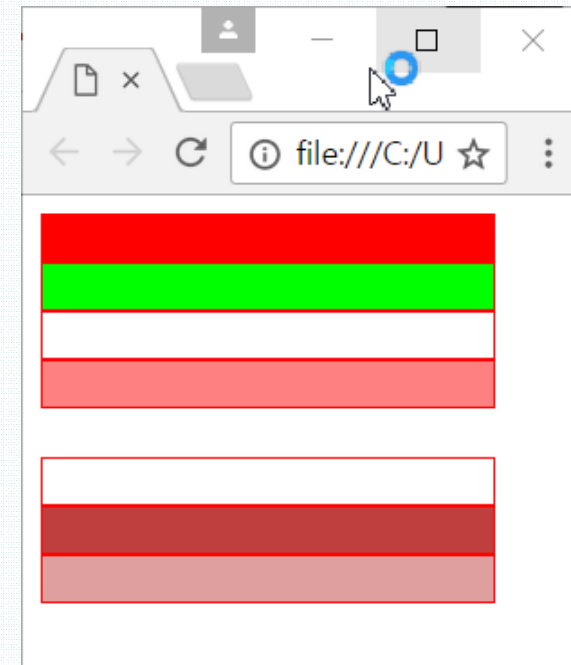
- **색조(Hue)**
  - hsl 방식과 동일
- **채도(saturation)**
  - hsl 방식과 동일
- **밝기(Lightness)**
  - hsl 방식과 동일
- **투명도(alpha)**
  - 색상의 투명도를 0~1.0 사이의 실수로 표현
  - 0에 가까울수록 투명, 1에 가까울수록 불투명
- [ 예 ]
  - : #p1 { background-color : hsl(120, 100%, 50%, 0) }
  - : #p1 { background-color : hsl(120, 100%, 75%, 0.5) }





## ■ 색상 지정 방식 지정의 예

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>색상의 지정 방법</title>
<style type="text/css">
  div { width:200px; height:20px; border:1px solid red;}
  #rgb1 { background-color:red; }
  #rgb2 { background-color:rgb(0,255,0); }
  #rgb3 { background-color:rgb(0,0,100%); }
  #rgb4 { background-color:rgba(255,0,0,0.5); }
  #hsl1 { background-color:hsl(120,100%0%,50%); }
  #hsl2 { background-color:hsl(0,50%,50%); }
  #hsl3 { background-color:hsla(0,50%,50%,0.5); }
</style>
</head>
<body>
  <div id="rgb1"></div>
  <div id="rgb2"></div>
  <div id="rgb3"></div>
  <div id="rgb4"></div><BR>
  <div id="hsl1"></div>
  <div id="hsl2"></div>
  <div id="hsl3"></div>
</body>
</html>
```





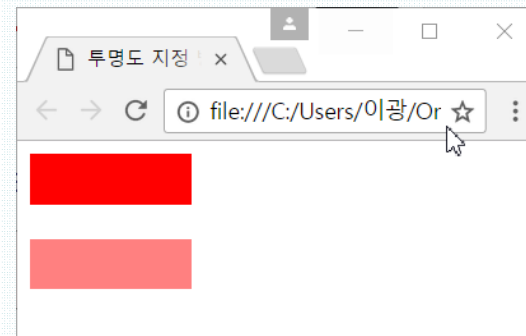
## ■ opacity 속성

- rgba()와 hsla()와 사용하지 않고 특정 객체에 직접 투명도를 지정하는 방식

요소 { opacity:투명도 }

- 투명도는 0부터 1까지의 실수를 사용해 지정 ( 0에 가까울수록 투명 )

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>투명도 지정 방법</title>
<style type="text/css">
  div { width:100px; height:30px; border:1px solid red;}
  #rgb1 { background-color:red; }
  #rgb2 { background-color:red; opacity:0.5}
</style>
</head>
<body>
  <div id="rgb1"></div> <br>
  <div id="rgb2"></div>
</body>
</html>
```





## ■ 그라데이션의 종류

### • 선형 그라데이션

- 두 개 이상의 색상을 지정하고 좌표간의 사이를 직선으로 이동하면서 색상을 변화시키는 경우
- linear-gradient()를 사용

### • 원형 그라데이션

- 두 개 이상의 색상을 지정해 두고 특정 좌표를 중심으로 색상을 변화시키는 경우
- radial-gradient()를 사용

### • 브라우저의 종류와 버전에 따른 벤더 프리픽스(vendor prefix)를 사용해야 하는 경우도 있음

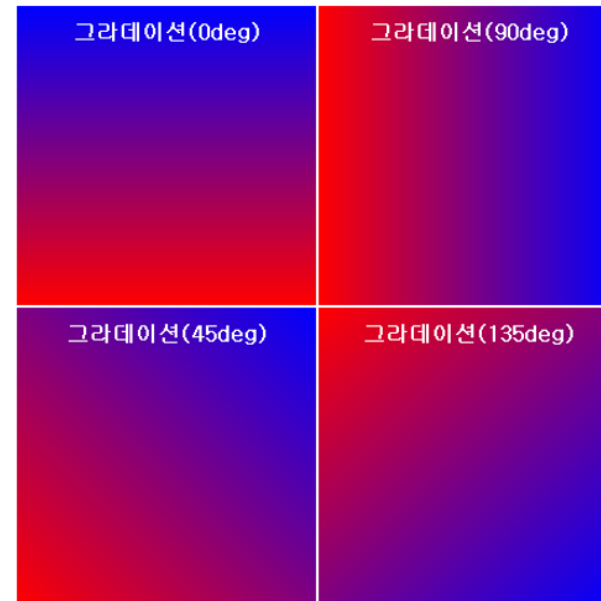
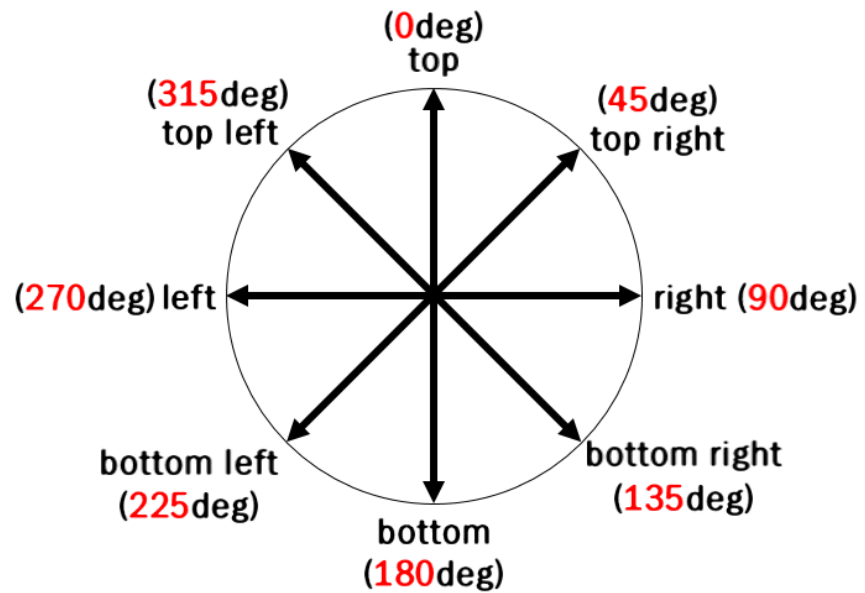
프리픽스	설명
-webkit-	웹킷(webkit) 엔진을 사용하는 크롬(chrom)과 사파리(safari)
-o-	오페라(opera)
-moz-	모질라 계열의 파이어폭스(FireFox)
-ms-	마이크로소프트의 인터넷 익스플로러



# 그라데이션( gradation )의 개요



## ■ 그라데이션의 방향





## ■ linear-gradient()

- 선형 그라데이션을 작성하기 위한 함수

linear-gradient( 방향, 색1 (범위), 색2 (범위), ... , 색n (범위) )

인자		설명
방향	각도의 사용	[ 각도deg ] 형태로 사용 12시 방향이 0deg이며, 시계방향으로 각도를 표현 - [예] 90deg : 오른쪽 방향
	방향명칭의 사용	[to 방향명칭] 형태로 사용 - [예] to right : 오른쪽 방향 (=90deg) 방향 명칭은 혼용하여 사용할 수 있음 - [예] to right top ( = 45deg )
색		그라데이션을 구성하는 색을 의미하며 하나 이상의 색을 지정함
범위		각각의 색이 그라데이션 내에서 표현될 수 있는 범위를 의미 그라데이션 영역 전체를 100%로 하여 각 색의 범위는 %로 표현 - [예] linear-gradient(to right, red 10%, green 20%, blue);

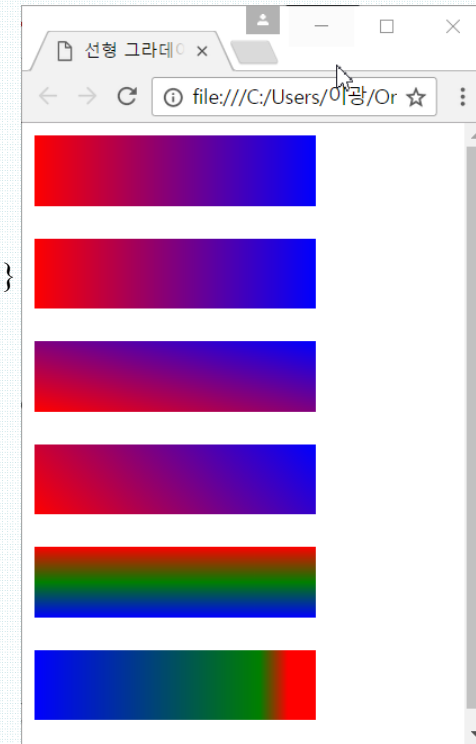


# 선형 그라데이션 ( linear gradation )



## ■ linear-gradient()의 예

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>선형 그라데이션</title>
  <style type="text/css">
    div { width:200px; height:50px; border:1px solid white;}
    #grad1 { background:linear-gradient(to right, red, blue); }
    #grad2 { background:linear-gradient(90deg, red, blue); }
    #grad3 { background:linear-gradient(to right top, red, blue); }
    #grad4 { background:linear-gradient(45deg, red, blue); }
    #grad5 { background:linear-gradient(180deg, red, green, blue); }
    #grad6 { background:linear-gradient(270deg, red 10%, green 20%, blue); }
  </style>
</head>
<body>
  <div id="grad1"></div><br>
  <div id="grad2"></div><br>
  <div id="grad3"></div><br>
  <div id="grad4"></div><br>
  <div id="grad5"></div><br>
  <div id="grad6"></div><br>
</body>
</html>
```





## ■ repeating-linear-gradient()

- 동일한 패턴의 선형 그라데이션을 반복해서 표현하는 함수

`repeating-linear-gradient( 방향, 색1 (범위), 색2 (범위), ... , 색n (범위) )`

- 전체 그라데이션 영역인 100% 를 기준으로 반복되는 그라데이션 영역의 크기가 지정  
: 마지막으로 지정된 색의 범위가 반복되는 그라데이션 영역의 임

### • [ 그라데이션 반복 지정의 예 ]

- `repeating-linear-gradient(to right, red, blue 50%)`
  - : 왼쪽에서 오른쪽으로 진행되는 그라데이션을 작성
  - : red에서 blue로 그라데이션 진행
  - : 전체의 그라데이션 영역 중 50%의 영역에 그라데이션이 발생하며 반복됨( 총 2회 반복 )
- `repeating-linear-gradient(to right, red, green 25%, blue 25%)`
  - : 왼쪽에서 오른쪽으로 진행되는 그라데이션을 작성
  - : red에서 blue로 green으로, green에서 blue로 그라데이션 진행
  - : 전체의 그라데이션 영역 중 25%의 영역에 그라데이션이 발생하며 반복됨( 총 4회 반복 )
  - : 반복되는 하나의 영역 중 25%를 green 영역으로 지정





# 선형 그라데이션 ( linear gradation )

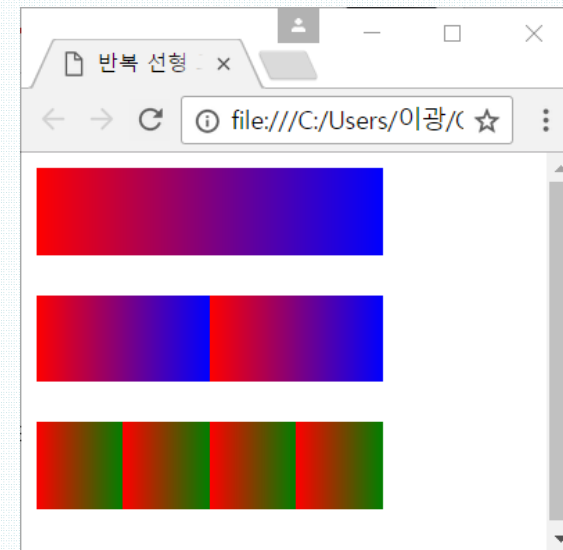


## ■ repeating-linear-gradient()의 예

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>반복 선형 그라데이션</title>
  <style type="text/css">
    div { width:200px; height:50px; border:1px solid white;}
    #grad1 { background:linear-gradient(to right, red, blue); }
    #grad2 { background:repeating-linear-gradient(to right, red, blue 50%); }
    #grad3 { background:repeating-linear-gradient(to right, red, green 25%, blue 25%); }
  </style>
</head>

<body>
  <div id="grad1"></div><br>
  <div id="grad2"></div><br>
  <div id="grad3"></div><br>
</body>

</html>
```





## ■ radial-gradient()

### • 원형 그라데이션을 작성하기 위한 함수

radial-gradient( 종류, [size키워드 at x좌표 y좌표,] 색1 (범위), 색2 (범위), ... , 색n (범위) )

- 중심 위치(center)를 기준으로 0~100%가 계산되기 때문에 중심에서 시작하여 원 바깥쪽으로 색상을 점점 변화시킴

### • 원형 그라데이션의 종류

- circle
  - : 원형 그라데이션 작성
- ellipse
  - : 타원형 그라데이션 작성(기본값)

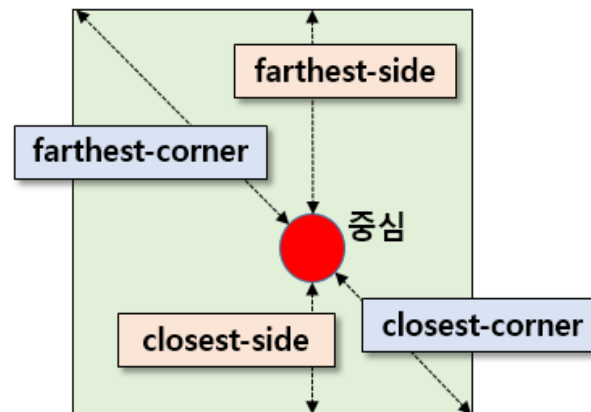


# 원형 그라데이션 ( radial gradation )



## • size 키워드

size 키워드	설명
closest-side	중심에서 가장 가까운 박스 측면과 정확하게 일치하도록 그라데이션 모양의 크기를 형성
farthest-side	중심에서 가장 먼 박스 측면이나 가장 먼 세로 및 가로 측면과 일치하도록 그라데이션 모양의 크기를 형성
closest-corner	중심에서 가장 가까운 박스 모서리와 정확하게 일치하도록 그라데이션 모양의 크기를 형성
farthest-corner (기본값)	중심에서 가장 먼 박스 모서리와 정확하게 일치하도록 그라데이션 모양의 크기를 형성





## ▣ repeating-radial-gradient()

- 동일한 패턴의 원형 그라데이션을 반복하기 위한 함수

repeating-radial-gradient( 종류, 색1 (범위), 색2 (범위), ... , 색n (범위) )

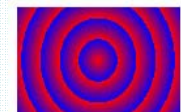
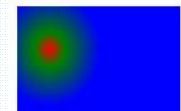
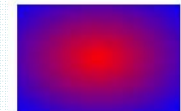
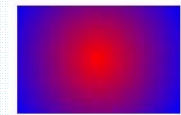
- 전체 그라데이션 영역인 100% 를 기준으로 반복되는 그라데이션 영역의 크기가 지정되어야 함
  - : 마지막으로 지정된 색의 범위가 반복되는 그라데이션 영역의 임





## ■ radial-gradient(), repeating-radial-gradient()의 예

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>원형 그라데이션</title>
  <style type="text/css">
    div { width:100px; height:60px; border:1px solid white;}
    #radial1 { background:radial-gradient(circle, red, blue); }
    #radial2 { background:radial-gradient(ellipse, red, blue); }
    #radial3 { background:radial-gradient(circle, red, green 30%, blue); }
    #radial4 { background:radial-gradient(circle closest-corner at 20% 40%, red, green 30%, blue); }
    #radial5 { background:repeating-radial-gradient(circle, red, blue 20%); }
  </style>
</head>
<body>
  <div id="radial1"></div><br>
  <div id="radial2"></div><br>
  <div id="radial3"></div><br>
  <div id="radial4"></div><br>
  <div id="radial5"></div><br>
</body>
</html>
```





끝 끝



## ■ font-family 속성

### • 글자체를 지정하기 위한 속성

font-family : 글자체 리스트

- : [예-CSS] font-family : 굴림체;
- : [예-HTML] <FONT FACE="굴림체"></FONT>

- **family-name(글꼴그룹) 이나 generic-family(기본 글꼴그룹)를 값으로 가짐**
  - generic-family : serif(명조체계열), sans-serif(고딕체계열), cursive(필기체계열) 등
  - 하나의 폰트 이름이 하나 이상의 문자로 구성된 경우 이중인용부호 내에 표현
- **하나 이상의 글자체를 지정할 경우**
  - 각 글자체를 쉼표로 구분하여 표현
  - 대체 글자체를 표현하기 위해 사용
    - : 사용자 시스템 환경에서 출력할 글자체를 지정 ( 앞 글자체가 우선 순위를 가짐 )
    - : [예] <DIV style="font-family:굴림체, 궁서체, 바탕체">한국교통대학교</DIV>





## ■ font-size 속성

### • 글자의 크기를 지정하기 위한 속성

font-size : 글자크기

- HTML에서는 <FONT> 태그의 size 속성을 사용하거나 헤딩 태그(<H>)를 사용
  - : <FONT> 태그의 size 속성 값 지정 범위 : 1 ~ 7
  - : 헤딩 태그의 범위 : <H1> ~ </H6>
- HTML에서보다 다양한 방법으로 세밀한 크기 지정 가능
  - : [예] BODY { font-size : 100px

### • 글자 크기 지정 방법

- 절대적 크기 단위 사용 방법
- 상대적 크기 단위 사용 방법
- 절대적 크기
- 상대적 크기(기본 크기 기준)







## ■ 절대적 크기 지정 방법

- 절대적(물리적)인 크기 단위를 사용

- pt(포인트), cm(센티미터), mm(밀리미터), in(인치)  
: [예] <DIV style="font-size:50px">한국교통대학교</DIV>

## ■ 상대적 크기 지정 방법

- 특정 기준에 대한 상대적인 크기 단위를 사용

- px : 픽셀 단위 크기 ( 해상도가 높을수록 글자 크기는 작아짐 )
- em : 기준크기에 대한 상대적인 크기 (1em = <body>의 기준 폰트 크기 )
- % : 기준크기에 대한 상대적인 크기 (100% = <body>의 기준 폰트 크기 )
- [예]  
: <DIV style="font-size:50%">한국교통대학교</DIV>



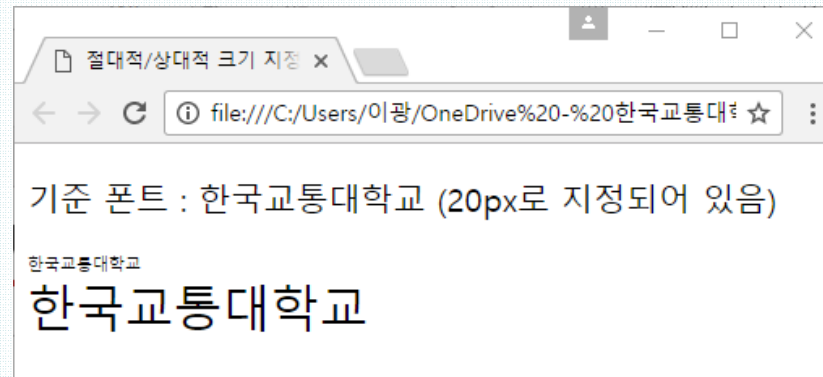


## ■ 절대적/상대적 크기 지정 방법

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>절대적/상대적 크기 지정</title>

  <style type="text/css">
    body { font-size:20px; }
    div#t1 { font-size:50%; }
    div#t2 { font-size:1.5em; }
  </style>

<body>
<div style="font-size:em">
  <p>기준 폰트 : 한국교통대학교 (20px로 지정되어 있음)</p>
  <div id="t1">한국교통대학교</div>
  <div id="t2">한국교통대학교</div>
</div>
</body>
</html>
```



기준 폰트(<body>에서 지정되어 있음)의 크기가 달라지면 %와 em의 크기도 달라짐





## ■ 절대적 크기

- **xx-small**(가장 작게), **x-small**(매우 작게), **small**(작게)
- **medium**(중간크기 )
- **large** (크게), **x-large** (매우 크게), **xx-large** (가장 크게)

- [예] <DIV style="font-size:small">한국교통대학교</DIV>

## ■ 상대적 크기

- **smaller** (기본 크기보다 작게), **larger** (기본 크기보다 크게)

- [예] <DIV style="font-size:smaller">한국교통대학교</DIV>



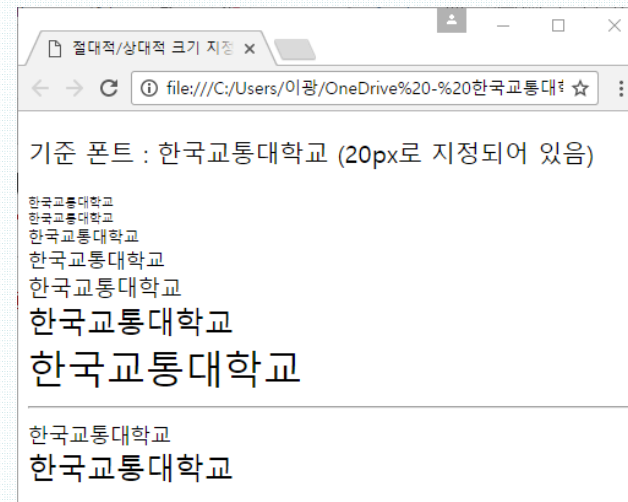


## ■ 절대적/ 상대적 크기의 예

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>절대적/상대적 크기 지정</title>

  <style type="text/css">
    body { font-size:20px; }
  </style>

<body>
<div style="font-size:em">
  <p>기준 폰트 : 한국교통대학교 (20px로 지정되어 있음)</p>
  <div style="font-size:xx-small">한국교통대학교</div>
  <div style="font-size:x-small">한국교통대학교</div>
  <div style="font-size:small">한국교통대학교</div>
  <div style="font-size:medium">한국교통대학교</div>
  <div style="font-size:large">한국교통대학교</div>
  <div style="font-size:x-large">한국교통대학교</div>
  <div style="font-size:xx-large">한국교통대학교</div>
  <hr>
  <div style="font-size:smaller">한국교통대학교</div>
  <div style="font-size:larger">한국교통대학교</div>
</div>
</body>
</html>
```





## ■ font-style 속성

- 주로 기울임체 글씨를 표현하기 위해 사용

```
font-style : [ normal, no, oblique, italic ]
```

- normal, no (스타일을 지정하지 않음)
- oblique, italic (이탤릭 체로 지정)

## ■ font-variant 속성

- 영문자 소문자를 대문자로 표시하지만 소문자의 크기로 표현

```
font-variant : [ small-caps, normal ]
```

- small-caps(소문자를 대문자로 변환하되 소문자 크기로 변환)
- normal(변환하지 않음)
- 대문자에는 적용되지 않음



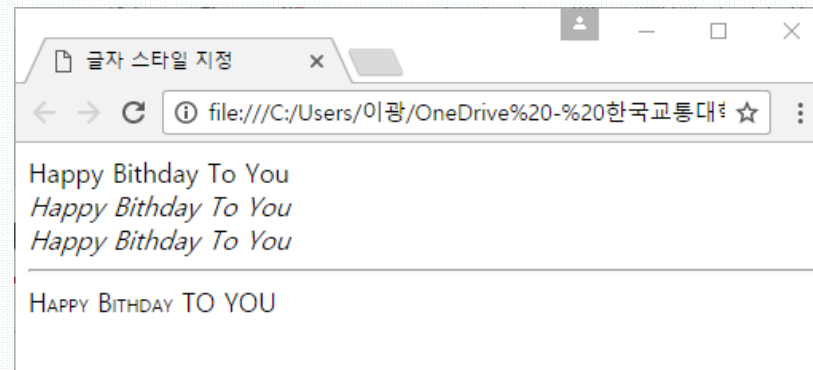


## ■ font-style, font-variant 속성의 예

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>글자 스타일 지정</title>

<body>

  <div style="font-style:normal">Happy Bithday To You</div>
  <div style="font-style:oblique">Happy Bithday To You</div>
  <div style="font-style:italic">Happy Bithday To You</div>
  <hr>
  <div style="font-variant:small-caps">Happy Bithday TO YOU</div>
</body>
</html>
```





## ■ font-weight 속성

- 글자의 굵기를 지정하기 위한 속성

font-weight : 가중치

- 가중치 표현 방법

- 수치 표현 지정 방법
- 상대적 표현 지정 방법





## ■ 수치 표현 지정 방법

```
font-weight : [ 100, 200, 300, 400, 500, 600, 700, 800, 900 ]
```

- 브라우저마다 다소 차이가 있으며, 육안으로 구분이 어려움
  - 100~500 : 굵게 표현하지 않음 (육안으로 식별이 어려움)
  - 600~900 : 굵게 표현함 (육안으로 식별이 어려움)
  - 진하게 표현하지 않은 기본값은 400, HTML의 <B>와 같은 효과 값은 600임

## ■ 상대적 표현 지정 방법

```
font-weight : [lighter, normal, bold, bolder ]
```

- 브라우저마다 다소 차이가 있으며, 육안으로 구분이 어려움
  - lighter(한 단계 아래)
  - normal( =400, 기본)
  - bold (진하게), bolder(한 단계 더 진하게)
- 일반적으로 bold를 사용







## ■ @font-face 속성

- 개발자가 정의한 글꼴을 웹 사이트 이용자가 사용할 수 있도록 하는 속성
  - 개발자가 개발한 페이지에서 사용되는 폰트는 사용자의 시스템에도 존재해야 함
    - : 존재하지 않으면 의도되지 않는 기본 폰트로 표현됨 ( 한글의 경우 심각함 )
  - 개발자가 사용한 폰트를 웹 서버에 보관하고 사용자가 다운로드하여 사용하는 방법
    - : 브라우저에 따라 폰트의 형식이 다양하므로 접근성이 있는 방법을 사용해야 함

## • 지정 가능한 폰트의 종류

font 확장자	글꼴의 종류	글꼴의 파일 유형
.TTF	트루 타입 (TrueType)	truetype
.OTF	오픈 타입 (OpenType)	opentype
.WOFF	웹 오픈 타입 (Web Open Font Format)	woff
.EOT	임베디드 오픈 타입 (Embedded OpenType)	embedded-opentype
.SVG	스케일러블벡터 (Scalable Vector Graphic)	svg





## ■ @font-face 속성의 형식

```
@font-face {  
    font-family: '생성할 폰트 이름';  
    src: local(폰트파일의 경로와 이름) format("폰트파일유형");  
    src: url(폰트파일의 경로와 이름) format("폰트파일유형");  
}
```

### - font-family

- : 지정할 각종 형태의 폰트들을 대표하는 이름
- : 임의 지정 가능하나 생략은 불가능함

### - src

- : 폰트의 경로와 이름, 유형을 지정
- : local - 해당 폰트가 시스템 내에 있는 경우 폰트파일의 경로와 이름을 지정
- : 해당 폰트가 시스템 내에 없는 경우 웹 서버 상의 폰트파일의 경로와 이름을 지정

```
@font-face {  
    font-family: 'MyFont';  
    src: url(font/MyFont.ttf) format("truetype");  
    src: url(font/MyFont.otf) format("opentype");  
    src: url(font/MyFont.woff) format("woff");  
}
```

```
body { font-family: MyFont; }
```

