

Front-end Bootcamp

CSS3

shadow

기본 그림자

```
.box {  
  box-shadow: 5px 10px red;  
}
```

블러(blur)와 확산(spread)

```
.box {  
  box-shadow: 가로위치 세로위치 블러 확산 색상  
}
```

inset(내부 그림자)

```
.box {  
  box-shadow: inset ...  
}
```

rgba()

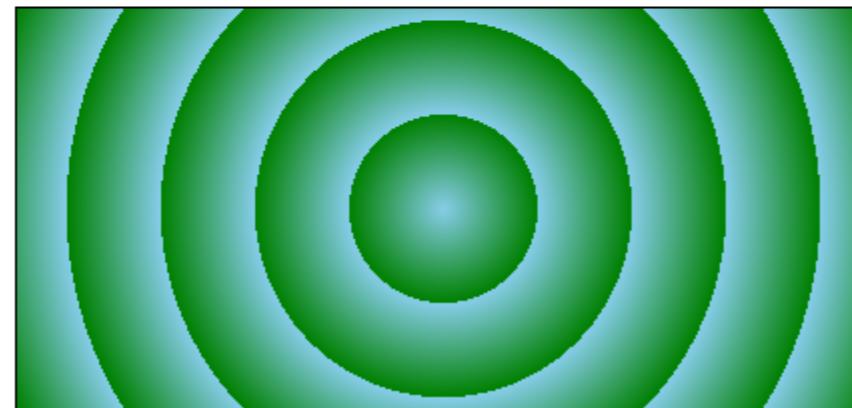
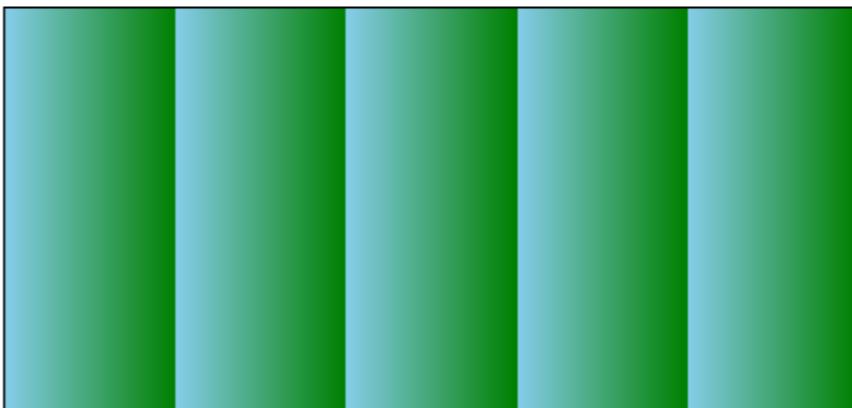
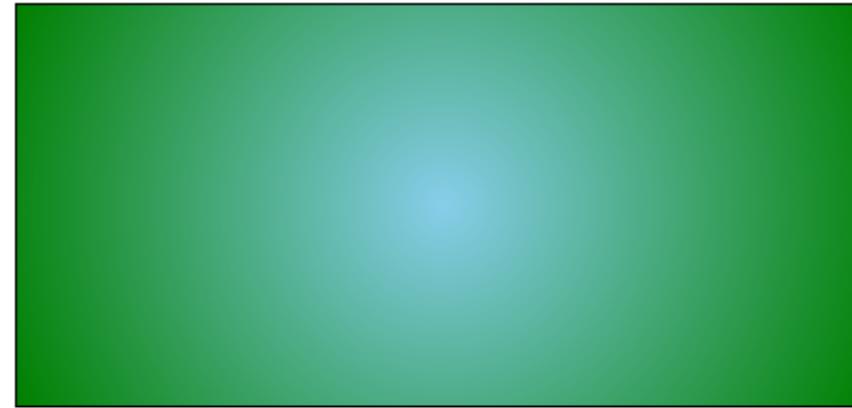
CSS의 컬러

- 컬러명 직접 입력
- hex(hexdecimal) color
- rgba → 투명도를 설정 가능
- hls, hlsa → 투명도를 설정 가능

빨간색을 표현하는 방법

```
background: red;  
background: #FF0000;  
background: #F00;  
background: rgba(255, 0, 0, );  
background: rgba(255, 0, 0, 50%);  
background: hsl(0, 100%, 50%);  
background: hsl(0, 100%, 50%, 50%);
```

gradient



```
.box1 {  
  background: linear-gradient(to right, skyblue, green);  
}  
  
.box2 {  
  background: radial-gradient(circle, skyblue, green);  
}  
  
.box3 {  
  background: repeating-linear-gradient(to right, skyblue, green 20%);  
}  
  
.box4 {  
  background: repeating-radial-gradient(circle, skyblue, green 20%);  
}
```

linear-gradient(선형)

```
background: linear-gradient(to right, #1e5799 25%,#7db9e8 60%);  
background: linear-gradient(방향, 색상1 점유율,색상2 점유율, ...);
```

- 방향
 - to right
 - to bottom
 - to bottom right
 - 45deg..
- 점유율은 생략 가능
- 색상은 rgba()로 투명도 설정 가능

radial-gradient

```
background: radial-gradient(모양 크기및위치, 색상1 점유율, 색상2 점유율, ...);
```

- 모양
 - circle
 - ellipse
- 크기 및 위치
 - closest-side : 가장 가까운 선, 요소
 - closest-corner : 가장 가까운 모서리
 - farthest-side : 가장 먼 선, 요소
 - farthest-corner : 가장 먼 모서리

repeating-linear/radial-gradient

`background: repeating-linear-gradient(방향, 색상1 점유율, ... , 마지막색상 점유율);`

`background: repeating-radial-gradient(모양 크기및위치, 색상1 점유율, ... , 마지막 색상 점유율);`

- 마지막 색상의 점유율을 기준으로 반복

| 작성이 어려울 때는? [gradient editor](#)

animation in web

- jquery의 animate()
- jquery 라이브러리
 - [velocity.js](#), [mo.js](#)
- CSS3 transition, animation

CSS animation

- (지원하는) 브라우저의 내장 엔진 사용
- 빠르고, 부드럽다 (*myth busting*)
- 비교적 쉽다

CSS animation

1. transition
2. transform
3. keyframes & animation

transition

- CSS로 이루어지는 상태의 변화를 부드럽게
- 변화
 - 크기
 - 색상
 - 위치
 - . . .
- 부드럽게
 - 시간(duration)

transition

```
transition-property: opacity, left, top, height;  
transition-duration: 3s, 5s;  
transition-timing-function: linear;  
transition-delay: 2s;  
  
transition: <property> <duration> <timing-function> <delay>;
```

- **property**: 트랜지션될 요소
- **duration**: 지속 시간
- **timing-function**: 이펙트, 이징
- **delay**: 딜레이

transition (hover) 예제

```
.hover-moving-box {  
    width:100px;  
    height:100px;  
    position:relative;  
    background: red;  
  
    transition: all 3s ease;  
}  
  
.hover-moving-box:hover {  
    width:200px;  
    height:200px;  
    left:80%;  
    background: blue;  
}
```

transform

- 애니메이션만을 위한 스펙은 아님
- 요소의 크기, 위치 등을 상세하게 수정 가능

transform

```
transform: none  
transform: translate(12px, 50%) // 이동  
transform: translateX(2em) // X좌표 이동  
transform: translateY(3in) // Y좌표 이동  
transform: scale(2, 0.5) // 크기 조정  
transform: scaleX(2) // X좌표(가로) 크기 조정  
transform: scaleY(0.5) // Y좌표(세로) 크기 조정  
transform: rotate(45deg) // 회전  
transform: skewX(30deg) // 기울기(X축)  
transform: skewY(30deg) // 기울기(Y축)  
  
transform: translateX(10px) rotate(10deg) translateY(5px)
```

transition & transform

```
.hover-moving-box {  
  width:100px;  
  height:100px;  
  position:relative;  
  background: red;  
  
  transition: all 3s ease;  
}  
  
.hover-moving-box:hover {  
  transform: translateX(100px), rotate(45deg), scale(2, 2);  
}
```

keyframes & animation

- 이벤트가 아닌 움직임 구현해야 할 때
- keyframes를 미리 설정하고, 이를 animation 속성에서 참조
- keyframes는 재사용이 가능

keyframes 정의

```
@keyframes spin {  
    from {  
        transform:rotate(0deg);  
    }  
    to {  
        transform:rotate(360deg);  
    }  
}
```

```
@keyframes leftAndRight {  
    0% {  
        transform:translate(0,0);  
    }  
    50% {  
        transform:translate(70%,0);  
    }  
    0% {  
        transform:translate(0,0);  
    }  
}
```

animation 속성 작성

```
animation-name: example;  
animation-duration: 4s;  
animation-delay: 2s;  
animation-iteration-count: infinite;  
animation-timing-function: linear;  
  
animation: <name> <duration> <iteration-count> <timing-function> <delay>;
```

- name: 미리 작성된 keyframes 이름
- duration: 지속 시간
- iteration-count: 반복 횟수, (무한반복: infinite)
- timing-function: 이펙트, 이징
- delay: 딜레이

정리 예제

- <https://codepen.io/yunkimoon/pen/0geMEY>

