

# 6장

## CSS 필수 속성 다루기

6.1 CSS의 특징 살펴보기

6.2 텍스트 속성으로 텍스트 꾸미기

6.3 박스 모델을 구성하는 속성 다루기

6.4 배경 속성으로 요소의 배경 설정하기

6.5 위치 속성으로 HTML 요소 배치하기

6.6 **전환 효과 속성 적용하기**

6.7 애니메이션 속성으로 전환 효과 제어하기

6.8 변형 효과 적용하기

6.9 웹 폰트와 아이콘 폰트 사용하기

## 6.6 전환 효과 속성 적용하기

### 1. 전환이란

- 한 요소에 적용된 스타일 속성값을 다른 속성 값으로 변하게 하는 것

### 2. transition-property 속성

- 전환 효과의 대상 속성명을 값으로 지정

**형식** `transition-property:<속성값>;`

- 속성값
  - none 전환 효과 속성을 지정하지 않음
  - all 모든 속성을 전환 효과 대상으로 지정

## 6.6 전환 효과 속성 적용하기

예를 들어, `background-color` 속성에 전환 효과를 주고 싶다면 다음과 같이 작성하면 됩니다.

```
transition-property:background-color;
```

전환 효과를 여러 속성에 지정하고 싶다면 쉼표로 구분해 작성하면 됩니다.

```
transition-property:background-color, color, width;
```

개별 속성이 아닌 모든 속성에 전환 효과를 넣고 싶으면 `all`로 지정하면 됩니다.

```
transition-property:all;
```

## 6.6 전환 효과 속성 적용하기

### 3. transition-duration 속성

- 전환 효과가 진행되는 시간을 지정

형식 `transition-duration:<시간>;`

- 어떤 요소에 전환 효과를 지정하려면 반드시 `transition-property` 속성과 `transition-duration` 속성을 함께 사용해야 함

아래의 코드처럼 사용합니다.

```
transition-property:background-color, color, width;
```

```
transition-duration:1s, 500ms, 2s;
```

## 6.6 전환 효과 속성 적용하기

### 4. transition-delay 속성

- 전환 효과가 지연되는 시간을 지정

**형식** `transition-delay:<시간>;`

```
<style>
  .red-box{
    width:100px;
    height:100px;
    background-color:■red;
    transition-property:background-color, color, width;
    transition-duration:1s;
    transition-delay:1s;
  }
  .red-box:hover{
    background-color:■blue;
  }
</style>
</head>
<body>
  <div class="red-box"></div>
```

## 6.6 전환 효과 속성 적용하기

### 5. transition-timing-function 속성

- 전환 효과 속도를 지정

**형식** `transition-timing-function`:<속성값>;

- 속성값
  - linear : 처음 속도와 마지막 속도가 일정
  - ease : 처음에는 속도가 점점 빨라지다가 중간부터 점점 느려짐
  - ease-in : 처음에는 속도가 느리지만 완료될 때까지 점점 빨라짐
  - ease-out : 처음에는 속도가 빠르지만 완료될 때까지 점점 느려짐
  - ease-in-out : 처음에는 속도가 느리지만 점점 빨라지다가 다시 점점 느려짐

## 6.6 전환 효과 속성 적용하기 예제 실습

080313-main\06\06\transition-timing-function.html

```
<style>
.container{
  padding:10px;
  color:□white;
  border:1px solid ■black;
}
.bar{
  width:100px;
  background-color:■red;
  transition-property:width;
  transition-duration:1s;
  margin-bottom:10px;
}
.container:hover .bar{
  width:200px;
}
.bar:nth-child(1){
  transition-timing-function:linear;
}
.bar:nth-child(2){
  transition-timing-function:ease;
}
.bar:nth-child(3){
  transition-timing-function:ease-in
}
.bar:nth-child(4){
  transition-timing-function:ease-out;
}
.bar:nth-child(5){
  transition-timing-function:ease-in-out;
}
</style>
```

```
</head>
<body>
  <div class="container">
    <div class="bar">linear</div>
    <div class="bar">ease</div>
    <div class="bar">ease-in</div>
    <div class="bar">ease-out</div>
    <div class="bar">ease-in-out</div>
  </div>
```

linear

ease

ease-in

ease-out

ease-in-out

## 6.6 전환 효과 속성 적용하기

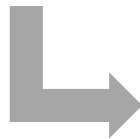
### 6. transition 속성으로 한 번에 지정하기

- 모든 전환 효과 속성을 한 번에 지정할 수 있는 단축 속성

**형식** `transition:<property>, <duration>, <timing-function>, <delay>`

**예**

```
transition-property:width;  
transition-duration:1s;  
transition-timing-function:ease-in;  
transition-delay:1s;
```



```
transition:width, 1s, ease-in, 1s;
```



## 6.7 애니메이션 속성으로 전환 효과 제어하기

### 1. 키 프레임 정의하기

- @keyframes 애니메이션의 전체 속성 정의

**형식** @keyframes <키 프레임명>{  
    0%{ /\* CSS 코드 \*/}  
    n%{ /\* CSS 코드 \*/}  
    100%{ /\* CSS 코드 \*/}  
}

### 2. animation-name 속성

- 애니메이션을 지정할 키 프레임명 지정

**형식** animation-name:<키 프레임명>;

## 6.7 애니메이션 속성으로 전환 효과 제어하기

### 3. animation-duration 속성

- 애니메이션의 지속 시간을 지정

**형식** `animation-duration`:<지속 시간>;

- 애니메이션은 키 프레임, `animation-name` 속성, `animation-duration` 속성 3가지 중 하나라도 빠지면 전환 효과는 적용되지 않음

### 4. animation-delay 속성

- 애니메이션의 지연 시간을 지정

**형식** `animation-delay`:<지연 시간>;

## 6.7 애니메이션 속성으로 전환 효과 제어하기

### 5. animation-fill-mode 속성

- 애니메이션 실행 전과 종료 후의 상태 지정

속성값	상태	설명
none	실행 전	시작 시점(0%, from)의 스타일을 적용하지 않고 대기
	실행 후	실행되기 전의 스타일 적용 상태로
forwards	실행 전	시작 시점(0%, from)의 스타일을 적용하지 않고 대기
	실행 후	키 프레임에 정의된 종료 시점(100%, to)의 스타일을 적용하고 대기
backwards	실행 전	키 프레임에 정의된 시작 시점(0%, from)의 스타일을 적용하고 대기
	실행 후	실행되기 전의 스타일 적용 상태로
both	실행 전	키 프레임에 정의된 시작 시점(0%, from)의 스타일을 적용하고 대기
	실행 후	키 프레임에 정의된 종료 시점(100%, to)의 스타일을 적용하고 대기

## 6.7 애니메이션 속성으로 전환 효과 제어하기

### 6. animation-iteration-count 속성

- 애니메이션의 반복 횟수 지정

**형식** `animation-iteration-count`:<횟수>;

### 7. animation-play-state 속성

- 애니메이션의 진행/정지 상태 정의

**형식** `animation-play-state`:<속성값>;

- 속성값
  - `paused` : 애니메이션의 실행을 일시 정지
  - `running` : 애니메이션을 실행

## 6.7 애니메이션 속성으로 전환 효과 제어하기

### 7. animation-play-state 속성 예제

```
<style>
div{
  width:100px;
  height:100px;
  background-color:■red;
  position:relative;
  animation-name:move;
  animation-duration:10s;
  animation-fill-mode:forwards;
  animation-play-state:paused;
}
@keyframes move{
  from{
    left:0;
  }
  to{
    left:300px;
  }
}
</style>
</head>
<body>
<div></div>
<button id="start">start</button>
<button id="paused">paused</button>
<script>
const box = document.querySelector("div");
document.getElementById("start").addEventListener("click", function(){
  box.style.animationPlayState = "running";
});
document.getElementById("paused").addEventListener("click", function(){
  box.style.animationPlayState = "paused";
})
</script>
```



start paused

## 6.7 애니메이션 속성으로 전환 효과 제어하기

### 8. animation-direction 속성

- 애니메이션의 진행 방향을 지정

```
animation-direction:reverse;
```

**형식** `animation-direction`:<속성값>;

- 속성값
  - normal : 애니메이션의 진행 방향을 키 프레임에 정의된 시간 순서대로 진행( to → from)
  - reverse : 애니메이션의 진행 방향을 키 프레임에 정의된 시간 순서의 역으로 진행(from → to)
  - alternate : 애니메이션이 1회 이상 실행될 경우 홀수 번째는 normal로, 짝수 번째는 reverse로 진행
  - alternate-reverse : 애니메이션이 1회 이상 실행될 경우 홀수 번째는 reverse로, 짝수 번째는 normal로 진행

## 6.7 애니메이션 속성으로 전환 효과 제어하기

### 9. animation 속성으로 한 번에 지정하기

- 모든 애니메이션 관련 속성 지정

**형식** `animation:<name> <duration> <timing-function> <delay>`  
`<iteration-count> <direction> <fill-mode> <play-state>;`

**예** `animation:bgchange 5s 3 ease-in;`

## 6.8 변형 효과 적용하기

### 1. transform 속성

- 요소에 특정 변형 효과를 지정

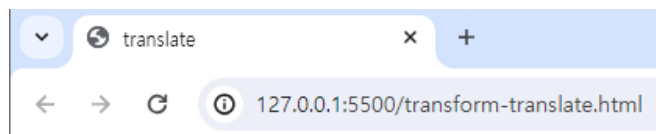
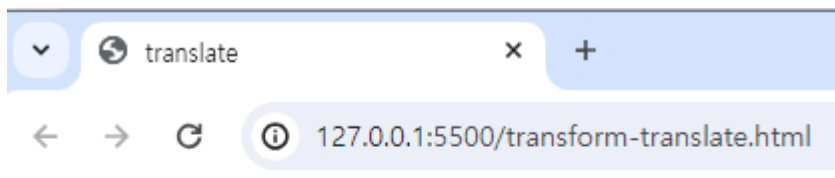
**형식** `transform:<함수>;`

- 2차원 좌표 이동하기
  - `translate(x, y)` : 요소를 현재 위치에서 x(x축)와 y(y축)만큼 이동
  - `translateX(n)` : 요소를 현재 위치에서 n만큼 x축으로 이동
  - `translateY(n)` : 요소를 현재 위치에서 n만큼 y축으로 이동

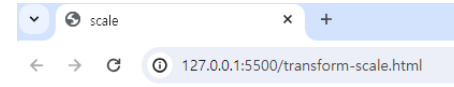


## 6.8 변형 효과 적용하기

```
<style>
  div{
    width:100px;
    height:100px;
    background-color:■red;
  }
  div:hover{
    transform:translate(100px, 200px); /* 오른쪽으로 100px, 아래쪽으로 200px 이동 */
  }
</style>
</head>
<body>
  <div></div>
```



## 6.8 변형 효과 적용하기



- 2차원 확대 또는 축소하기



- `scale(x, y)` : 요소를  $x$ ( $x$ 축)와  $y$ ( $y$ 축)만큼 확대 또는 축소
- `scaleX(n)` : 요소를  $n$ 만큼  $x$ 축으로 확대 또는 축소
- `scaleY(n)` : 요소를  $n$ 만큼  $y$ 축으로 확대 또는 축소

- 2차원 기울이기

- `skew(xdeg, ydeg)` : 요소를  $x$ 축과  $y$ 축으로  $xdeg$ ,  $ydeg$ 만큼 기울임
- `skewX(deg)` : 요소를  $deg$ 만큼  $x$ 축 방향으로 기울임
- `skewY(deg)` : 요소를 주어진  $deg$ 만큼  $y$ 축 방향으로 기울임

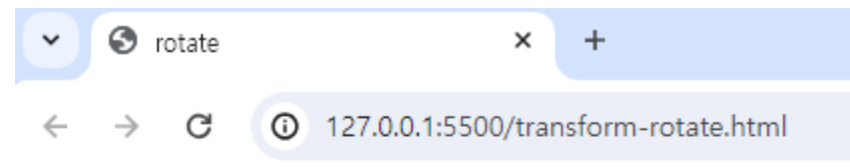
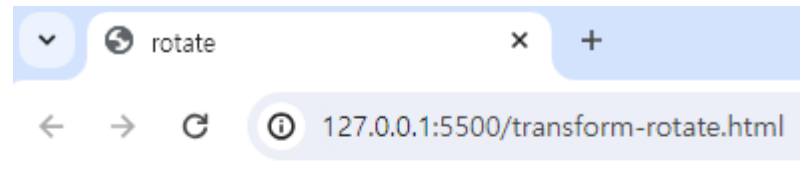
- 2차원 회전하기

- `rotate(deg)` : 요소를  $deg$ 만큼 회전

## 6.8 변형 효과 적용하기

- 2차원 회전하기 예제 실습

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>rotate</title>
  <style>
    .container{
      border:1px solid black;
      display:inline-block;
      margin:20px;
    }
    .box{
      width:100px;
      height:100px;
      background-color:red;
    }
    .box:hover{ /* 마우스를 올리면 회전 */
      transform:rotate(30deg); /* 오른쪽으로 회전 */
    }
  </style>
</head>
<body>
  <div class="container">
    <div class="box"></div>
  </div>
</body>
</html>
```



## 6.8 변형 효과 적용하기

### 2. transform-origin 속성

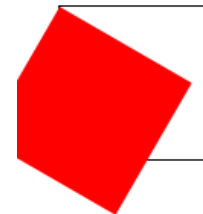
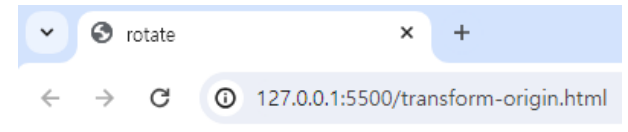
- 변형 효과의 기준점을 변경할 때

**형식** `transform-origin`:<x축 위치> <y축 위치>;

- 속성값

- x축 기준 위치 : left 왼쪽 / center 중앙 / right 오른쪽
- y축 기준 위치 : top 위쪽 / center 중앙 / bottom 아래쪽

```
.box:hover{  
  transform:rotate(30deg); /* 오른쪽으로 회전 */  
  transform-origin:top left; /* 왼쪽 위 모서리를 기준으로 지정 */  
}
```

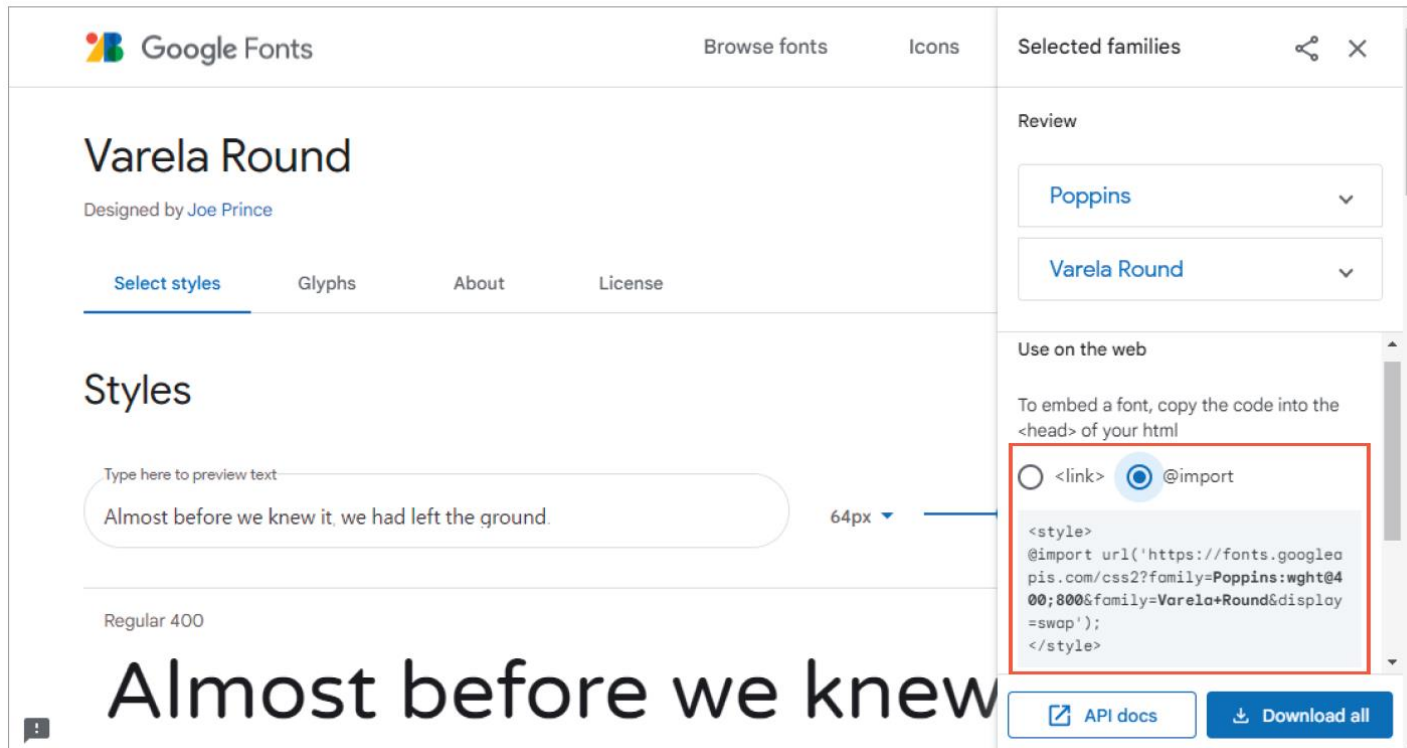


## 6.9 웹 폰트와 아이콘 폰트 사용하기

### 1. 구글 폰트 적용하기

- 구글 웹 폰트 <https://fonts.google.com>

그림 6-87 글꼴 사용



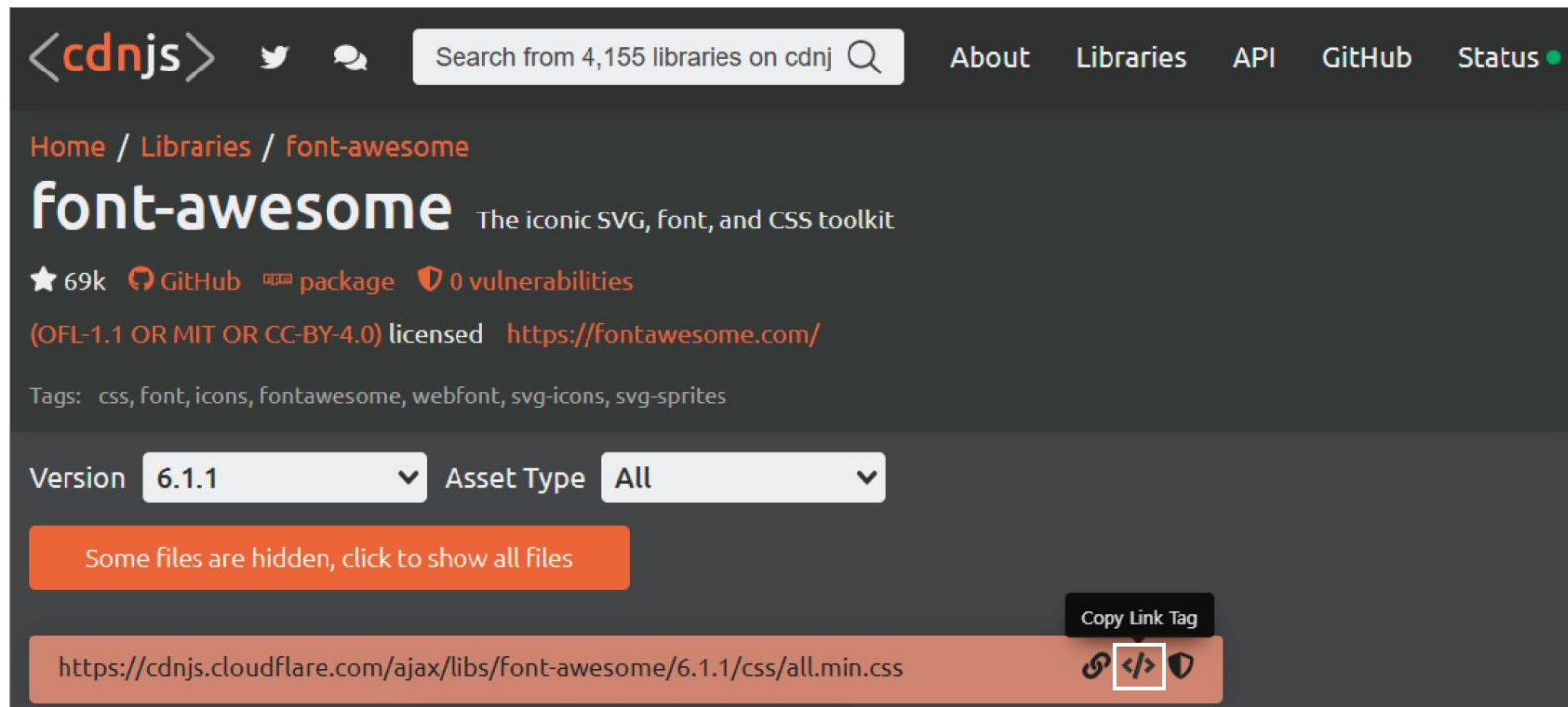
## 6.9 웹 폰트와 아이콘 폰트 사용하기

### 2. 아이콘 폰트 적용하기

- Font Awesome 라이브러리

<https://cdnjs.com/libraries/fontawesome>

그림 6-89 CDNJS에서 제공하는 Font Awesome 라이브러리



## 6.9 웹 폰트와 아이콘 폰트 사용하기

06/09/webfont.html

```
<style>
  @import url('https://fonts.googleapis.com/css2?family=Poppins:wght@400;700&family=Varela+Round&display=swap');
  .poppins{
    font-family:"Poppins", sans-serif;
  }
  .varela{
    font-family:"Varela Round", sans-serif;
  }
</style>
(중략)
<p class="poppins">Lorem ipsum dolor, sit amet consectetur adipisicing elit.</p>
<p class="varela">Lorem ipsum, dolor, sit amet consectetur adipisicing elit.</p>
```