

## 8. 테두리(Border)와 그림자

### 테두리 스타일, 색상, 두께

CSS에서는 `border` 속성을 통해 요소의 테두리 스타일, 색상, 두께를 제어할 수 있다.

테두리는 시각적 구분, 강조, 버튼/박스 디자인 등에 자주 사용된다.

#### 1. 기본 문법

```
1 selector {  
2   border: <두께> <스타일> <색상>;  
3 }
```

예시:

```
1 .box {  
2   border: 2px solid black;  
3 }
```

- 두께: `2px`
- 스타일: `solid` (실선)
- 색상: `black`

#### 2. 테두리 스타일 (`border-style`)

스타일 이름	설명
<code>none</code>	테두리 없음 (기본값)
<code>solid</code>	실선
<code>dashed</code>	점선 (긴 대시)
<code>dotted</code>	점선 (작은 점)
<code>double</code>	이중선 (두 줄)
<code>groove</code>	입체적으로 파인 효과
<code>ridge</code>	입체적으로 솟은 효과
<code>inset</code>	요소가 눌린 것처럼 보이는 내부 테두리
<code>outset</code>	요소가 튀어나온 것처럼 보이는 외부 테두리

```
1 | .border-demo {  
2 |   border-style: dashed;  
3 | }
```

---

### 3. 테두리 색상 (border-color)

```
1 | .box {  
2 |   border-color: red;  
3 | }
```

- 모든 면에 같은 색 적용
- 색상은 `red`, `#ff0000`, `rgb(255,0,0)` 등 다양하게 지정 가능

#### 각 방향별 지정:

```
1 | border-top-color: red;  
2 | border-right-color: blue;  
3 | border-bottom-color: green;  
4 | border-left-color: yellow;
```

---

### 4. 테두리 두께 (border-width)

```
1 | .box {  
2 |   border-width: 5px;  
3 | }
```

#### 각 방향별 지정:

```
1 | border-top-width: 3px;  
2 | border-right-width: 2px;  
3 | border-bottom-width: 4px;  
4 | border-left-width: 1px;
```

---

### 5. 전체 축약형: border

```
1 | .box {  
2 |   border: 4px dotted blue;  
3 | }
```

```
1 | border: [두께] [스타일] [색상];
```

---

## 6. 방향별 속성

전체 속성	방향별 속성
<code>border</code>	<code>border-top</code> , <code>border-right</code> , <code>border-bottom</code> , <code>border-left</code>
<code>border-style</code>	<code>border-top-style</code> , 등
<code>border-color</code>	<code>border-left-color</code> , 등
<code>border-width</code>	<code>border-right-width</code> , 등

## 7. 실전 예제

```
1 .card {
2   border-top: 5px solid #2c3e50;
3   border-bottom: 2px dashed #e74c3c;
4   border-left: 10px double #8e44ad;
5   border-right: none;
6 }
```

## 8. border vs outline

- `border`: 요소 크기에 포함됨 (box-sizing 영향 받음)
- `outline`: 요소 외부에 그려짐, 공간 차지하지 않음

## 결론

`border`는 요소에 테두리 형태, 색상, 두께를 조절하여 시각적 구분과 인터페이스 디자인에 핵심적인 역할을 한다. 필요에 따라 축약형이나 방향별 설정을 유연하게 활용하면 복잡한 테두리 스타일도 쉽게 구현할 수 있다.

## 둥근 테두리: border-radius

`border-radius` 속성은 HTML 요소의 모서리를 둥글게 만들어주는 CSS 속성이다. 사각형 요소의 모서리를 둥글게 잘라내듯 표현하며, 카드형 UI, 버튼, 프로필 이미지 등을 꾸밀 때 널리 사용된다.

## 1. 기본 문법

```
1 selector {
2   border-radius: <값>;
3 }
```

```
1 .box {
2   border-radius: 10px;
3 }
```

- 네 모서리가 모두 `10px` 둥글게 처리됨

## 2. 단일 값 vs 다중 값

### ▶ 단일 값

```
1 | border-radius: 15px;
```

- 네 모서리에 동일하게 적용

### ▶ 두 개의 값

```
1 | border-radius: 10px 20px;
```

- 왼쪽 위와 오른쪽 아래: `10px`
- 오른쪽 위와 왼쪽 아래: `20px`

### ▶ 네 개의 값

```
1 | border-radius: 10px 15px 20px 5px;
```

- 시계 방향: 왼쪽 위 → 오른쪽 위 → 오른쪽 아래 → 왼쪽 아래

## 3. 개별 모서리 지정

속성	설명
<code>border-top-left-radius</code>	왼쪽 위 모서리
<code>border-top-right-radius</code>	오른쪽 위 모서리
<code>border-bottom-right-radius</code>	오른쪽 아래 모서리
<code>border-bottom-left-radius</code>	왼쪽 아래 모서리

```
1 | .box {  
2 |   border-top-left-radius: 10px;  
3 |   border-bottom-right-radius: 30px;  
4 | }
```

## 4. 타원 형태 (비율 지정)

/ 를 사용해 수평/수직 반지름을 따로 설정할 수 있음:

```
1 | border-radius: 50px / 25px;
```

- 수평 반지름: 50px
- 수직 반지름: 25px

이를 통해 타원, 물방울, 완두콩 모양도 표현 가능

---

## 5. 원형 만들기

정사각형 요소에 아래와 같이 설정:

```
1 .profile-img {
2   width: 100px;
3   height: 100px;
4   border-radius: 50%;
5 }
```

정사각형 요소 + `border-radius: 50%` = 완벽한 원

---

## 6. 실전 예제

### 카드형 UI

```
1 .card {
2   border: 1px solid #ccc;
3   border-radius: 12px;
4   padding: 20px;
5   background-color: white;
6 }
```

### 한쪽 모서리만 둥글게

```
1 .tab {
2   border-radius: 12px 12px 0 0;
3 }
```

---

## 7. 브라우저 지원

- ☒ 모든 주요 브라우저 완벽 지원 (IE9 이상 포함)
- ☒ 반응형 디자인에서도 안전하게 사용 가능

---

## 결론

`border-radius`는 간단한 구문만으로도 요소의 **시각적 분위기와 부드러움**을 크게 개선할 수 있다. 버튼, 이미지, 카드 등 거의 모든 UI 요소에 적용 가능하며, 타원/원형 형태도 자유롭게 구현할 수 있다.

# 내부 간격: padding

CSS의 `padding` 속성은 요소의 콘텐츠(content)와 테두리(border) 사이의 내부 여백을 설정하는 데 사용된다. 즉, 콘텐츠로부터 테두리까지의 안쪽 간격을 의미하며, 요소 크기에 영향을 준다.

## 1. 기본 문법

```
1 selector {  
2   padding: <값>;  
3 }
```

```
1 .box {  
2   padding: 20px;  
3 }
```

- 모든 방향(상하좌우)에 20픽셀 간격이 생긴다.

## 2. 값 지정 방식

### ▶ 하나의 값

```
1 padding: 10px;
```

- 모든 방향 동일하게 적용

### ▶ 두 개의 값

```
1 padding: 10px 20px;
```

- 상하(top/bottom): 10px
- 좌우(left/right): 20px

### ▶ 세 개의 값

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

- top: 10px
- left/right: 20px
- bottom: 30px

## ▶ 네 개의 값 (시계 방향)

```
1 padding: 10px 20px 30px 40px;
```

- **top:** 10px
- **right:** 20px
- **bottom:** 30px
- **left:** 40px

## 3. 방향별 속성

속성명	설명
<code>padding-top</code>	위쪽 여백
<code>padding-right</code>	오른쪽 여백
<code>padding-bottom</code>	아래쪽 여백
<code>padding-left</code>	왼쪽 여백

예:

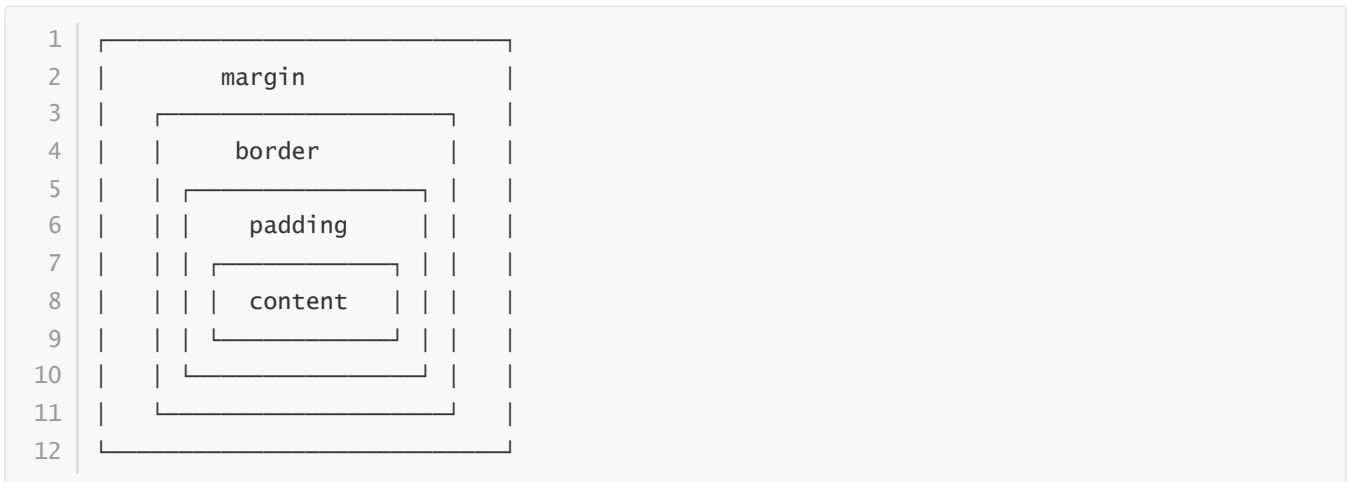
```
1 .box {  
2   padding-top: 10px;  
3   padding-left: 15px;  
4 }
```

## 4. 예제

```
1 .container {  
2   padding: 20px;  
3   background-color: lightgray;  
4 }
```

```
1 <div class="container">  
2   이 텍스트는 테두리로부터 20px 떨어져 있습니다.  
3 </div>
```

## 5. 시각적 구조



## 6. box-sizing 과의 관계

기본적으로 padding 은 요소의 width 나 height 에 추가되므로, 전체 요소 크기를 증가시킨다.

```
1 .box {  
2   width: 300px;  
3   padding: 20px;  
4   box-sizing: content-box; /* 기본값 */  
5 }
```

해결: **box-sizing: border-box**

```
1 .box {  
2   width: 300px;  
3   padding: 20px;  
4   box-sizing: border-box;  
5 }
```

- 이 설정을 하면 패딩을 포함한 전체 너비가 300px로 고정된다.

## 7. 단위

- px: 고정 간격
- %: 부모 요소의 너비 기준
- em, rem: 글꼴 크기 기준

```
1 .box {  
2   padding: 5%;  
3 }
```



## 결론

`padding`은 콘텐츠를 둘러싼 **내부 공간을 확보**하여 콘텐츠의 가독성, 배치, 디자인을 조절하는 핵심 속성이다.

`box-sizing: border-box`와 함께 사용하면 레이아웃 제어가 쉬워진다.

## 그림자: `box-shadow`

`box-shadow` 속성은 HTML 요소에 **그림자(새도우)**를 생성하는 데 사용되며,

입체감, 부드러움, 강조 효과를 주기 위해 널리 활용된다.

버튼, 카드, 팝업 등 거의 모든 UI에서 시각적 깊이를 표현할 때 사용한다.

### 1. 기본 문법

```
1 selector {  
2   box-shadow: <x-offset> <y-offset> <blur-radius> <spread-radius> <color> <inset>;  
3 }
```

#### 각 구성 요소 설명

항목	설명	필수 여부
<code>x-offset</code>	수평 이동 (양수: 오른쪽, 음수: 왼쪽)	✓
<code>y-offset</code>	수직 이동 (양수: 아래, 음수: 위)	✓
<code>blur-radius</code>	그림자의 흐림 정도 (선택, 기본: 0)	○
<code>spread-radius</code>	그림자의 확장 정도 (선택, 기본: 0)	○
<code>color</code>	그림자 색상	○
<code>inset</code>	그림자를 내부로 적용	○

### 2. 예시

```
1 .box {  
2   box-shadow: 5px 5px 15px rgba(0, 0, 0, 0.3);  
3 }
```

- 오른쪽 5px, 아래쪽 5px 이동
- 흐림 반경 15px
- 반투명 검정색 그림자

### 3. inset 내부 그림자

```
1 .inner-box {  
2   box-shadow: inset 0 0 10px rgba(0, 0, 0, 0.5);  
3 }
```

- 그림자가 안쪽으로 들어감 → 내부 강조 느낌

### 4. 여러 그림자 중첩

침표(,)로 구분하여 여러 그림자를 한 요소에 동시에 적용할 수 있음

```
1 .card {  
2   box-shadow:  
3     0 1px 3px rgba(0,0,0,0.12),  
4     0 1px 2px rgba(0,0,0,0.24);  
5 }
```

- 그림자가 쌓여서 더 풍부한 입체감

### 5. 실전 예제

#### 버튼 강조

```
1 .button {  
2   background-color: #3498db;  
3   color: white;  
4   padding: 10px 20px;  
5   border: none;  
6   box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);  
7   transition: box-shadow 0.3s ease;  
8 }  
9  
10 .button:hover {  
11   box-shadow: 0 6px 12px rgba(0, 0, 0, 0.3);  
12 }
```

### 6. 활용 패턴

목적	예시
부드러운 카드 스타일	<code>0 2px 5px rgba(0,0,0,0.1)</code>
강한 강조 효과	<code>0 8px 20px rgba(0,0,0,0.4)</code>
내부 놀림 효과	<code>inset 0 2px 6px rgba(0,0,0,0.3)</code>

## 7. 퍼포먼스 참고

- `box-shadow`는 GPU 가속이 **제한적**이므로 그림자 수가 많거나 blur가 큰 경우 **렌더링 비용 증가**
- 성능에 민감한 UI에서는 **단순하고 얇은 그림자**만 사용하는 것이 좋음

## 8. 브라우저 호환성

- ☒ 모든 주요 브라우저에서 지원
- ☒ 모바일 브라우저에서도 안정적으로 사용 가능

## 결론

`box-shadow`는 CSS에서 **시각적 깊이와 입체감**을 표현하는 대표적인 도구다.  
간단한 문법으로도 깔끔한 UI를 구성할 수 있으며, `inset`, 다중 그림자 등 다양한 활용이 가능하다.  
`transition`, `hover`와 함께 사용하면 **인터랙티브한 효과**를 쉽게 구현할 수 있다.

## outline과 border의 차이

CSS에서 `outline`과 `border`는 모두 **요소의 외곽선을 그리는 데 사용**되지만, 용도와 동작 방식에는 중요한 차이가 있다.

### 1. 정의

속성	설명
<code>border</code>	요소의 테두리이며, <b>레이아웃에 포함</b> 되는 시각 요소
<code>outline</code>	요소 외곽을 표시하는 선이며, <b>레이아웃에 영향을 주지 않음</b>

### 2. 핵심 차이점 요약

항목	<code>border</code>	<code>outline</code>
레이아웃에 포함	<input checked="" type="checkbox"/> 포함됨 ( <code>box-sizing</code> 영향 받음)	<input checked="" type="checkbox"/> 포함되지 않음
요소 크기 증가 여부	<code>border</code> 를 설정하면 요소 너비/높이 증가	<code>outline</code> 은 요소 밖에 그려지며 크기 변화 없음
개별 면 설정	가능 ( <code>border-top</code> , <code>border-left</code> , 등)	불가능 (항상 전체 테두리에 적용)
둥근 모서리	<code>border-radius</code> 로 둥글게 가능	<code>border-radius</code> 적용 안 됨
겹침 우선순위	콘텐츠보다 <b>안쪽에 있음</b>	콘텐츠보다 <b>바깥쪽에 있음</b>
시각적 효과	디자인 강조, 요소 영역 구분	주로 <b>접근성 포커스</b> 표시 용도

항목	border	outline
브라우저 포커스 표시	❌ 수동 설정 필요	✅ <button>, <a>, <input> 등에 기본 사용됨

### 3. 기본 예시 비교

```
1 /* border */
2 .box-border {
3     border: 2px solid red;
4 }
5
6 /* outline */
7 .box-outline {
8     outline: 2px solid red;
9 }
```

- `.box-border`: 요소의 크기가 **2px만큼 커짐**
- `.box-outline`: 요소 크기 **변화 없음**, 외곽선만 표시됨

### 4. outline 예제: 접근성

```
1 button:focus {
2     outline: 3px solid orange;
3 }
```

- 키보드 사용 시 접근성 강조
- 마우스 클릭만으로는 포커스 스타일이 나타나지 않을 수 있음

### 5. outline-offset 속성

`outline`은 요소 외부에 그려지므로 `outline-offset`으로 거리 조절 가능:

```
1 .button:focus {
2     outline: 2px solid blue;
3     outline-offset: 4px;
4 }
```

### 6. 사용 용도 비교

사용 목적	적합한 속성
UI 디자인, 스타일 강조	<code>border</code>

사용 목적	적합한 속성
접근성 포커스 표시	<code>outline</code>
요소 외곽 표시 (공간 차지 없이)	<code>outline</code>
버튼이나 입력창 테두리 커스터마이징	<code>border</code>
키보드 포커스 인식 가능하게 만들기	<code>outline</code>

## 결론

`border`는 요소의 **시각적 테두리**와 **공간 배치**에 중요한 역할을 하고,  
`outline`은 **접근성과 포커스 시각화**에 유용하다.  
두 속성은 모두 테두리를 그리지만, **레이아웃에 영향을 주는지 여부, 적용 대상, 표현 목적**이 다르다.