

# Table property

CSS에는 테이블 표현에 관한 몇가지 속성이 제공됩니다.

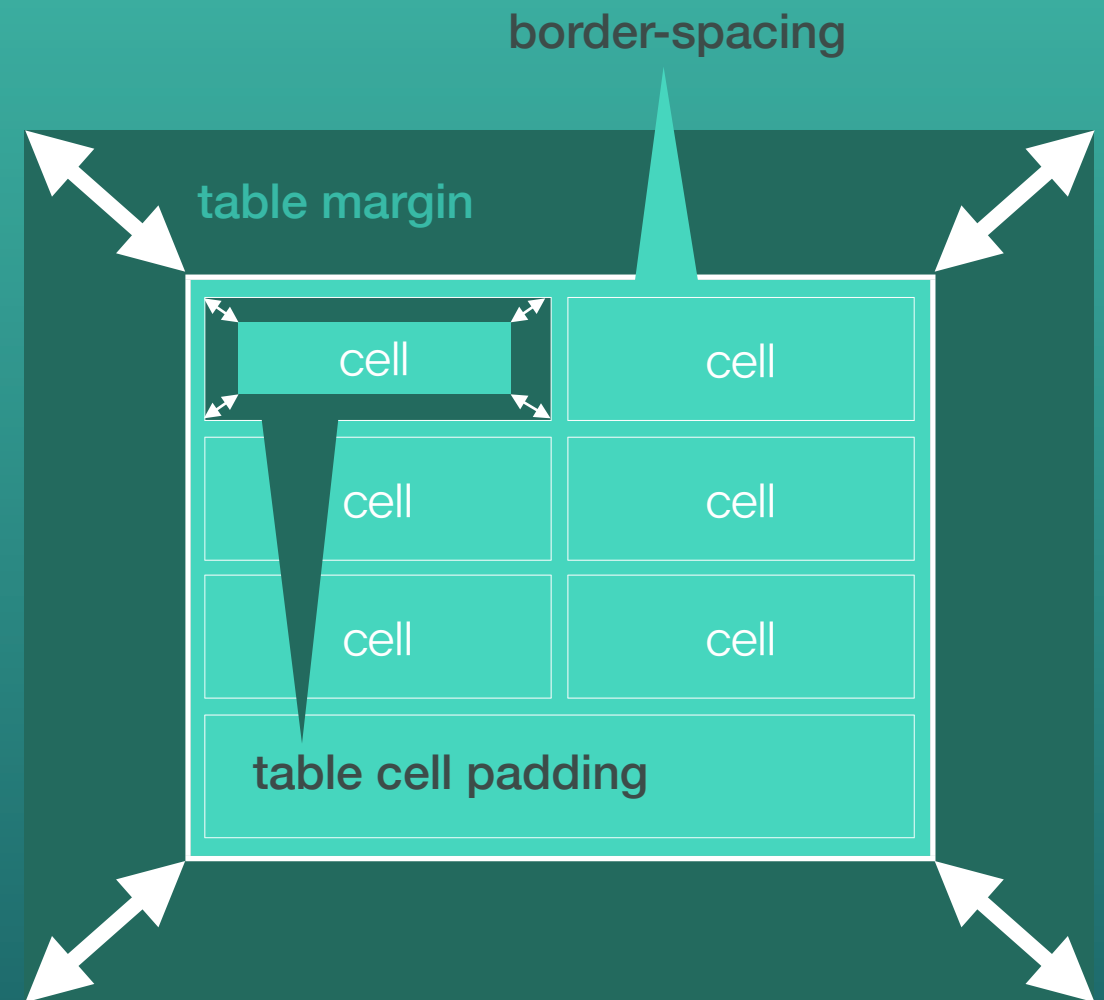




# 테이블 셀은 다른 요소와 다르게 마진 값이 적용되지도 않습니다.

```
<style type="text/css">
  table {
    margin: 10px;
    padding: 10px;
  }

  th, td {
    margin: 10px;
    padding: 10px;
  }
</style>
```

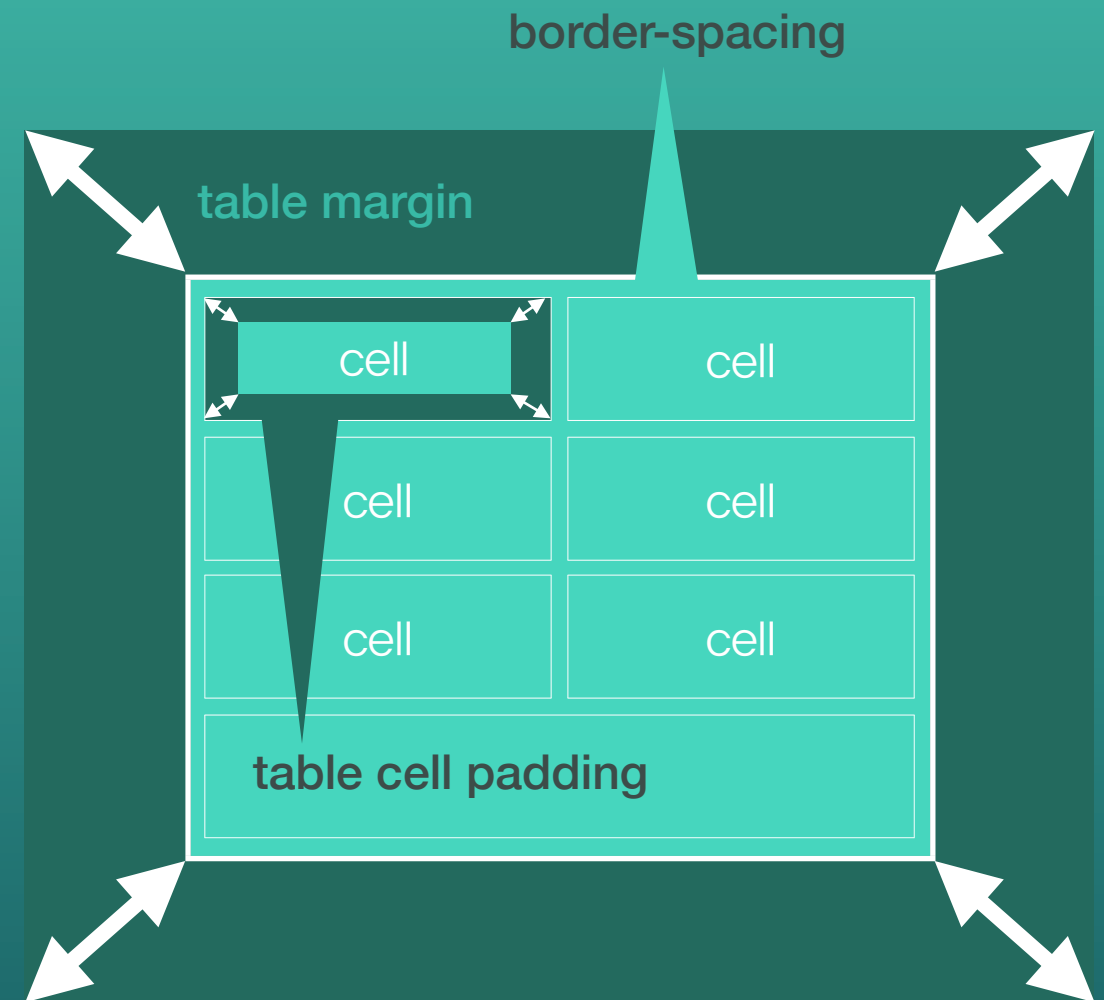




# 테이블 셀은 다른 요소와 다르게 마진 값이 적용되지도 않습니다.

```
<style type="text/css">
  table {
    margin: 10px;
    padding: 10px;
  }

  th, td {
    margin: 10px; 테이블 셀은 마진이 적용되지 않습니다.
    padding: 10px;
  }
</style>
```





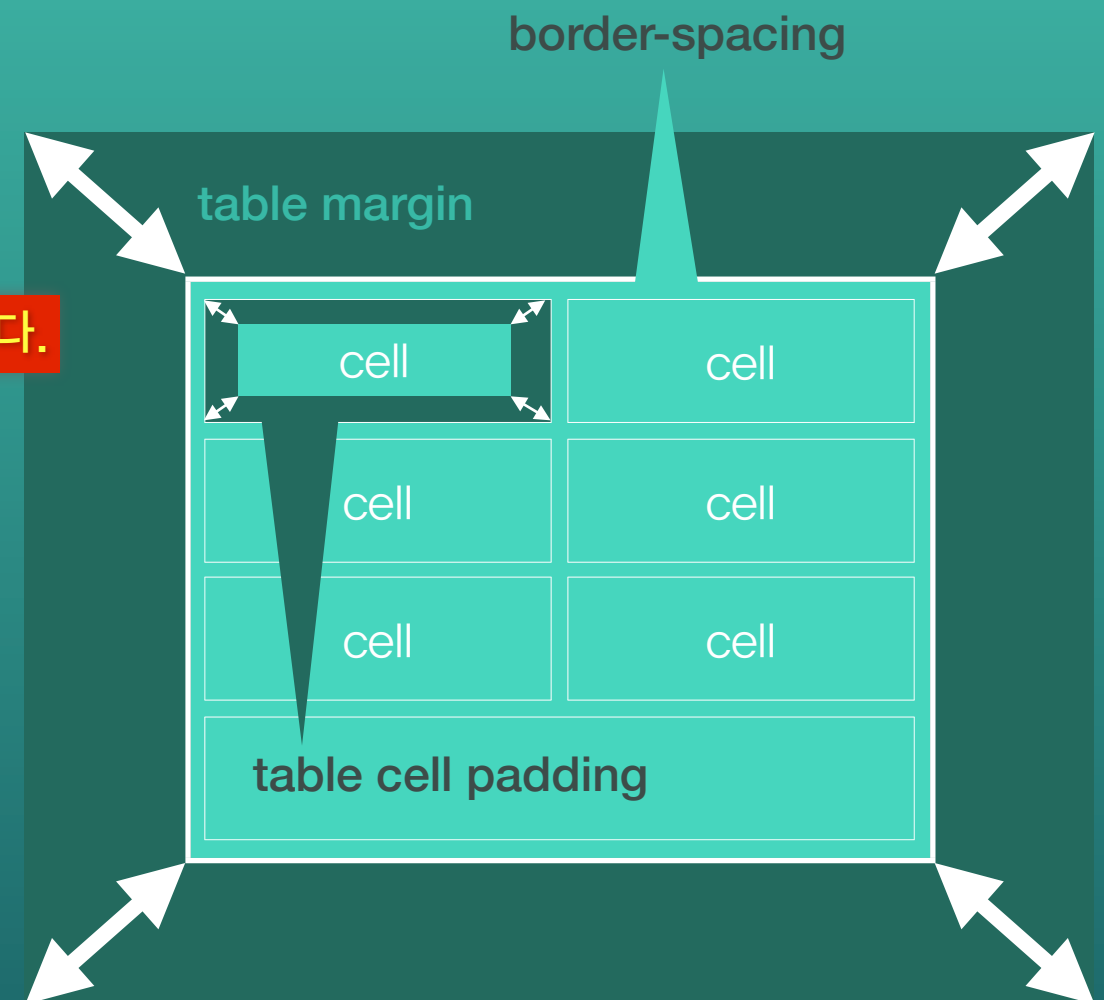
# 테이블 셀은 다른 요소와 다르게 마진 값이 적용되지도 않습니다.

```
<style type="text/css">  
table {  
  margin: 10px;  
  padding: 10px;  
}
```

padding: 10px; IE6,7 테이블에 패딩이 적용되지 않습니다.

```
th, td {  
  margin: 10px;  
  padding: 10px;  
}  
</style>
```

margin: 10px; 테이블 셀은 마진이 적용되지 않습니다.





# CSS언어를 이용하여 테이블의 셀 테두리를 통합해볼까요?



Style code

대상

속성

값

selector {border-collapse: value}

<head>

<style> table {border-collapse: collapse} </style>

</head>



# CSS언어를 이용하여 테이블의 셀 테두리를 통합해볼까요?



Style code

대상

속성

값

selector {border-collapse: value}

collapse

<head>

<style> table {border-collapse: collapse} </style>

</head>



# CSS언어를 이용하여 테이블의 셀 테두리를 통합해볼까요?



Style code

대상

속성

값

selector {border-collapse: value}

collapse

separate

<head>

<style> table {border-collapse: collapse} </style>  
</head>



# CSS언어를 이용하여 테이블의 셀 테두리를 통합해볼까요?



Style code

selector {border-collapse: value}

대상 속성 값

collapse  
separate  
inherit

<head>  
    <style> table {border-collapse: collapse} </style>  
</head>





# CSS언어를 이용하여 테이블의 셀 테두리를 통합해볼까요?



## IE7 이하에서는 불완전하게 작동합니다.

IE7 이하 브라우저에서는 border-collapse 스타일 적용이 요소에 직접 적용한 cellspacing보다 우선하지 못합니다. 표준을 잘 준수한다면 스타일이 항상 스타일 속성보다 우선하여야 하지만 IE6, IE7은 그렇지 못합니다.

```
<style> table {border-collapse: collapse} </style>  
</head>
```

collapse

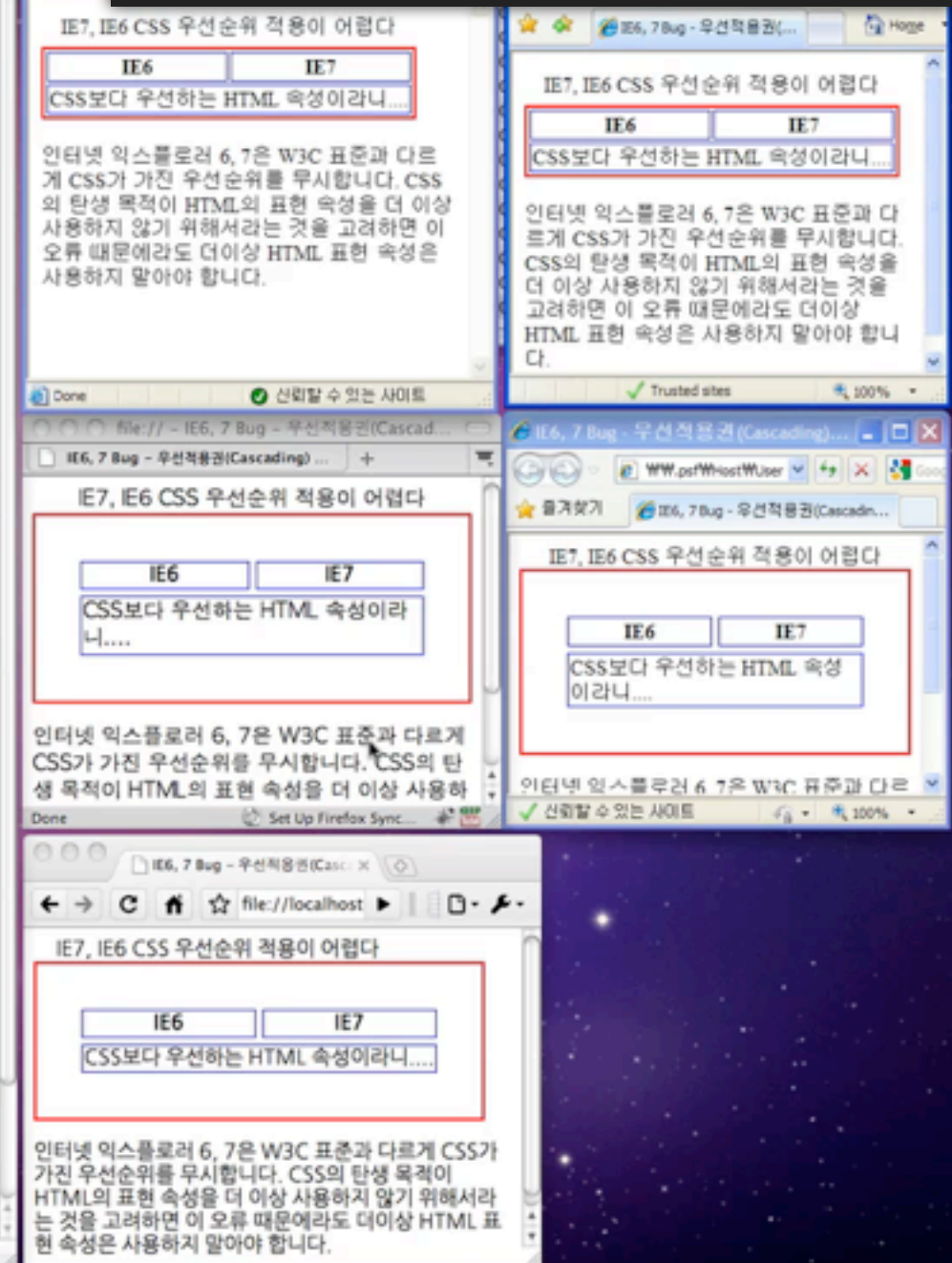
separate

inherit

## IE6, IE7. Bug cascading error

```
Internet Explorer
05_IE-bug_non-cascading_fin.html

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/
  DTD/xhtml1-transitional.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ko">
3   <head>
4     <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5     <title>IE6, 7 Bug - 우선적용권(Cascading) 무시 오류</title>
6     <style type="text/css">
7     <!--
8       /* IE6, 7 Bug - 우선적용권(Cascading) 무시오류 */
9       table, th, td {
10         border:0;
11       }
12       table {
13         padding:30px; /* */
14         border:2px solid red; /* border 1px을 사용하면 ie6, 7 다르게 보인다. */
15         border-spacing:5px; /* IE6, 7 미적용 오류 */
16         /* border-collapse:collapse; */
17       }
18       th, td {
19         border:1px solid blue;
20       }
21     -->
22   </style>
23 </head>
24 <body>
25   <table summary="IE7, IE6 CSS 우선순위 적용이 어렵다는 것을 실험" border="1">
26     <caption>IE7, IE6 CSS 우선순위 적용이 어렵다</caption>
27     <tr><th>IE6</th><th>IE7</th></tr>
28     <tr><td colspan="2">CSS보다 우선하는 HTML 속성이라니....</td></tr>
29   </table>
30   <p>인터넷 익스플로러 6, 7은 W3C 표준과 다르게 CSS가 가진 우선순위를 무시합니다.
  CSS의 탄생 목적이 HTML의 표현 속성을 더 이상 사용하지 않기 위해서라는 것을 고려하면
  이 오류 때문에라도 더이상 HTML 표현 속성은 사용하지 말아야 합니다.</p>
31
32
```





# CSS언어를 이용하여 테이블의 셀 간격을 설정해볼까요?



Style code

대상

속성

값

selector {border-spacing: value}

<head>

<style> table {border-spacing: 0 10px} </style>  
</head>



# CSS언어를 이용하여 테이블의 셀 간격을 설정해볼까요?



<head>  
    <style> table {border-spacing: 0 10px} </style>  
</head>





# CSS언어를 이용하여 테이블의 셀 간격을 설정해볼까요?



Style code

대상

속성

값

selector {border-spacing: value}

pixel

em

<head>

<style> table {border-spacing: 0 10px} </style>  
</head>



# CSS언어를 이용하여 테이블의 셀 간격을 설정해볼까요?



Style code

selector {border-spacing: value}

대상 속성 값

pixel  
em  
inherit

<head>  
    <style> table {border-spacing: 0 10px} </style>  
</head>



# CSS언어를 이용하여 테이블의 제목 위치를 설정해볼까요?



Style code

대상

속성

값

selector {caption-side: value}

<head>

<style> table {caption-side: bottom} </style>

</head>



# CSS언어를 이용하여 테이블의 제목 위치를 설정해볼까요?



Style code

대상

속성

값

selector {caption-side: value}

top

<head>

<style> table {caption-side: bottom} </style>

</head>





# CSS언어를 이용하여 테이블의 제목 위치를 설정해볼까요?



Style code

대상

속성

값

selector {caption-side: value}

top

bottom

<head>

<style> table {caption-side: bottom} </style>

</head>



# CSS언어를 이용하여 테이블의 제목 위치를 설정해볼까요?



Style code

대상

속성

값

selector {caption-side: value}

top

bottom

left

right

<head>

<style> table {caption-side: bottom} </style>

</head>



# CSS언어를 이용하여 테이블의 제목 위치를 설정해볼까요?



Style code

selector {caption-side: value}

<head>  
    <style> table {caption-side: bottom} </style>  
</head>

top  
bottom  
left  
right  
inherit



# CSS언어를 이용하여 테이블의 비어있는 셀들의 표현을 설정해볼까요?



Style code

대상

속성

값

selector {empty-cells: value}

<head>

<style> td, th {empty-cells: hide} </style>

</head>



# CSS언어를 이용하여 테이블의 비어있는 셀들의 표현을 설정해볼까요?



Style code

대상

속성

값

selector {empty-cells: value}

show

<head>



<style> td, th {empty-cells: hide} </style>

</head>



# CSS언어를 이용하여 테이블의 비어있는 셀들의 표현을 설정해볼까요?



Style code

대상

속성

값

selector {empty-cells: value}

show

hide

<head>

<style> td, th {empty-cells: hide} </style>  
</head>



# CSS언어를 이용하여 테이블의 비어있는 셀들의 표현을 설정해볼까요?



Style code

대상

속성

값

selector {empty-cells: value}

show

hide

inherit

<head>  
    <style> td, th {empty-cells: hide} </style>  
</head>

`table {empty-cells:show;}`

cell	cell
cell	

---

`table {empty-cells:hide;}`

cell	cell
cell	



`table {empty-cells:show;}`

cell	cell
cell	



`table {empty-cells:hide;}`

cell	cell
cell	

비어 있는 셀은 눈에 보이지 않는다.



# CSS언어를 이용하여 테이블의 레이아웃을 그려주는 방법을 설정해볼까요?



Style code

대상

속성

값

selector {table-layout: value}

<head>

<style> table {table-layout: fixed} </style>  
</head>



# CSS언어를 이용하여 테이블의 레이아웃을 그려주는 방법을 설정해볼까요?



Style code

대상

속성

값

selector {table-layout: value}

auto

<head>

<style> table {table-layout: fixed} </style>  
</head>



# CSS언어를 이용하여 테이블의 레이아웃을 그려주는 방법을 설정해볼까요?



Style code

대상

속성

값

selector {table-layout: value}

auto

fixed

<head>

<style> table {table-layout: fixed} </style>  
</head>



# CSS언어를 이용하여 테이블의 레이아웃을 그려주는 방법을 설정해볼까요?



Style code

대상

속성

값

selector {table-layout: value}

auto

fixed

inherit

<head>

<style> table {table-layout: fixed} </style>  
</head>

`table {table-layout: auto;}`

테이블 셀 내용이 많아지면 자동으로 늘어납니다.

보조 셀의 내용

`<td>테이블 셀 내용1</td><td>테이블 셀 내용2</td>`

`table {table-layout: fixed; width: 600px;}`

테이블 셀 내용이  
 많아져도 자동으로  
 늘어나지 않습니다.

보조 셀의 내용

`<td>테이블 셀 내용1</td><td>테이블 셀 내용2</td>`

`table {table-layout: auto;}`



테이블 셀 내용이 많아지면 자동으로 늘어납니다.	보조 셀의 내용
----------------------------	----------

`<td>테이블 셀 내용1</td><td>테이블 셀 내용2</td>`

`table {table-layout: fixed; width: 600px;}`



테이블 셀 내용이 많아져도 자동으로 늘어나지 않습니다.	보조 셀의 내용
--------------------------------	----------

`<td>테이블 셀 내용1</td><td>테이블 셀 내용2</td>`

table {table-layout: auto;}



테이블 셀 내용이 많아지면 자동으로 늘어납니다.	보조 셀의 내용
----------------------------	----------

<td>테이블 셀 내용1</td><td>테이블 셀 내용2</td>

table {table-layout: fixed; width: 600px;}

width 값을 명시하지 않으면  
fixed는 작동하지 않습니다.



테이블 셀 내용이 많아져도 자동으로 늘어나지 않습니다.	보조 셀의 내용
--------------------------------	----------

<td>테이블 셀 내용1</td><td>테이블 셀 내용2</td>