





Styling

jQuery - 스타일링

Decorating

jQuery - CSS를 활용한 꾸미기

.css() 함수

css() 함수는 () 내부에 CSS에서 사용했던 속성, 값을 통해 선택한 대상을 꾸며줄 수 있습니다.

.css() 함수를 통해 CSS 값 읽기(property)

```
$('#celebs tbody tr:even').css('background-color');  
$('#celebs tbody tr:even').css('color');
```

.css() 함수를 통해 CSS 값 설정(property, value)

```
$('#celebs tbody tr:even').css('background-color','#dedcda');  
$('#celebs tbody tr:even').css('color','tan');
```

.css({ map }) 함수 - 복합 선언

```
$('#celebs tbody tr:even').css(  
  {  
    'background-color':'#dedcda',  
    'color':'#3432ca',  
    'font-size':'12px'  
  }  
);
```


1.4

.css(property, function(index, value)) 함수

```
$("#div").click(function() {  
    $(this).css({  
        width: function(index, value) {  
            return parseFloat(value) * 1.2;  
        },  
        height: function(index, value) {  
            return parseFloat(value) * 1.2;  
        };  
    });  
});
```

.css() 함수 사용시 주의 점.

css() 함수는 '동작'을 관리하는 자바스크립트가 '표현'을 포함한 형태이기 때문에 일시적인 스타일 변화가 아닌, 표현에 사용하지 않기를 권합니다.

이유는 '구조', '표현', '동작'은 서로 분리되어야 함이 현대적인 웹 사이트 제작의 핵심이기 때문입니다.

1.4

.addClass(className | function(index, class)) 함수

css() 함수와는 달리 자바스크립트에서 직접 '표현'에 관여하지 않고, CSS에 선언된 .class 파일을 호출하여 적용함으로서 자바스크립트가 '표현'과는 무관하게 '동작'에만 신경쓸 수 있습니다.



.addClass() 함수

```
$('#celebs tr:even').addClass('zebra');
```

```
.zebra {  
    background-color: #ddd;  
    color: #666;  
}
```

.addClass() 함수 - 멀티클래스

```
$('#celebs tr:even').addClass('zebra selected');
```

```
.zebra {  
    background-color: #ddd;  
    color: #666;  
}  
  
.selected {  
    text-decoration: underline;  
}
```

1.4

.removeClass(className | function(index, class)) 함수

선택한 대상에 addClass()로 부여한 클래스이거나, 사전에 부여된 클래스가 있을 경우 removeClass()로 이를 제거할 수 있습니다.

.removeClass() 함수

```
$('#celebs tr.zebra').removeClass('zebra');
```

```
<tr class="zebra pointer market"> .....
```

.removeClass() 함수 - 클래스 모두 제거

```
$('#celebs tr.zebra').removeClass();
```

```
<tr class="zebra pointer market"> .....
```


.toggleClass() 함수

선택한 대상에 toggleClass()로 클래스를 부여하거나, 제거하는 스위치 함수입니다.

. toggleClass(클래스 이름) 함수

```
$(document).ready(function() {  
    $('#celebs tr.zebra p').click(function(){  
        $(this).toggleClass('zebra');  
    }  
});
```

<tr class="zebra pointer market">

. toggleClass(클래스 이름, 스위치) 함수

```
$(document).ready(function() {  
    var count = 0;  
    $('#celebs tr.zebra p').click(function(){  
        $(this).find('span').text('clicks: ' + count++);  
        $(this).toggleClass('zebra', count % 4 == 0);  
    }  
});
```

<tr class="zebra pointer market">

.hasClass() 함수

hasClass()를 통해 선택한 대상에 찾고자 하는 클래스가 있는지를 확인하는 함수입니다.
찾고자 하는 클래스가 있으면 true를 반환하며, 그렇지 않으면 false를 반환합니다.

.hasClass() 함수

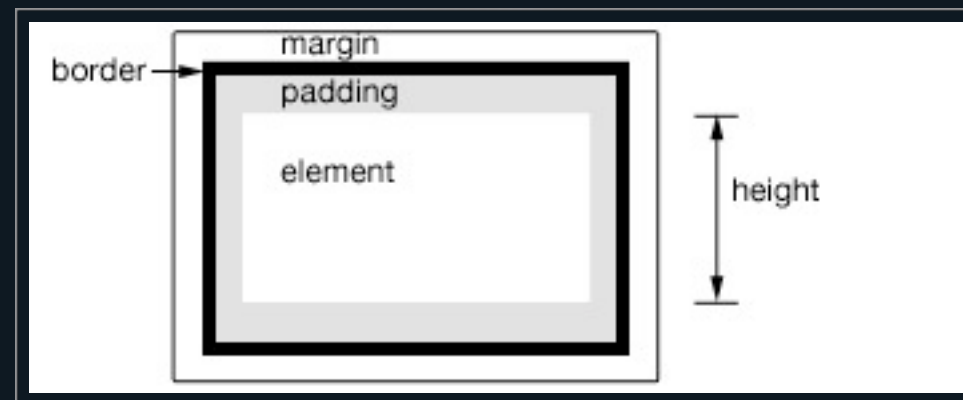
```
$(document).ready(function() {  
    if ($('#aside~p.zebra').hasClass('active-inline')) {  
        return;  
    }  
});
```

Dimensions

jQuery - CSS 치수(넓이, 높이)

.height() 함수

.height() 함수는 선택된 대상의 콘텐츠 높이 값을 반환(return)합니다.



.height() 함수를 사용해 대상의 높이 가져오기

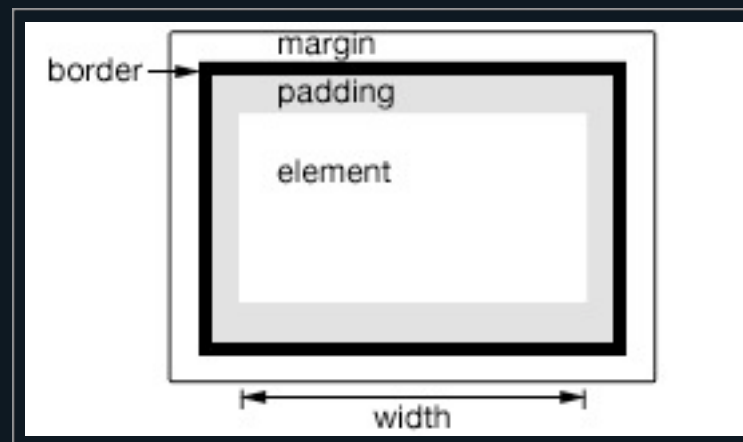
```
$(document).ready(function() {  
    $(window).height(); // 브라우저 뷰포트(viewport) 높이 값 반환  
    $(document).height(); // HTML 문서 높이 값 반환  
});
```


.height(value) 함수를 사용해 높이 설정하기

```
$(document).ready(function() {  
    $('p:last').height(30); // 마지막 단락의 높이를 30 px로 설정한다.  
});
```

.width() 함수

.width() 함수는 선택된 대상의 콘텐츠 넓이 값을 반환(return)합니다.



.width() 함수를 사용해 대상의 넓이 가져오기

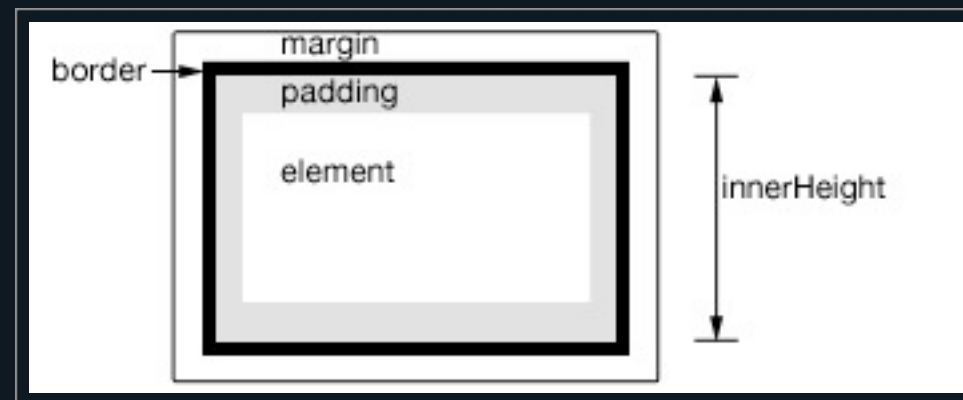
```
$(document).ready(function() {  
    $(window).width(); // 브라우저 뷰포트(viewport) 넓이 값 반환  
    $(document).width(); // HTML 문서 넓이 값 반환  
});
```

.width(value) 함수를 사용해 넓이 설정하기

```
$(document).ready(function() {  
    $('p:even').width(300); // 짝수 단락의 넓이를 300 px로 설정한다.  
});
```

.innerHeight() 함수 : ‘~패딩까지’

.innerHeight() 함수는 선택된 대상의 콘텐츠 + 패딩 높이 값을 반환(return)합니다.



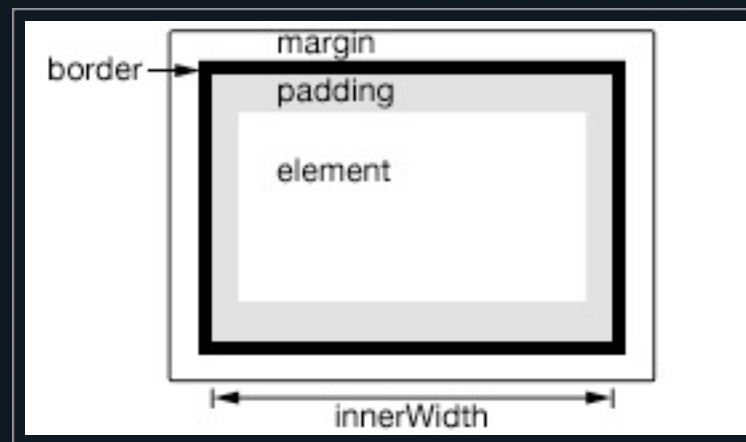
.innerHeight() 함수 : 높이 가져오기

```
$(document).ready(function() {  
    $('p:nth-child(2)').innerHeight();  
});
```

```
p { margin:10px; padding:5px; border:2px solid #666; }
```

.innerWidth() 함수 : ‘~패딩까지’

.innerWidth() 함수는 선택된 대상의 콘텐츠 + 패딩 넓이 값을 반환(return)합니다.



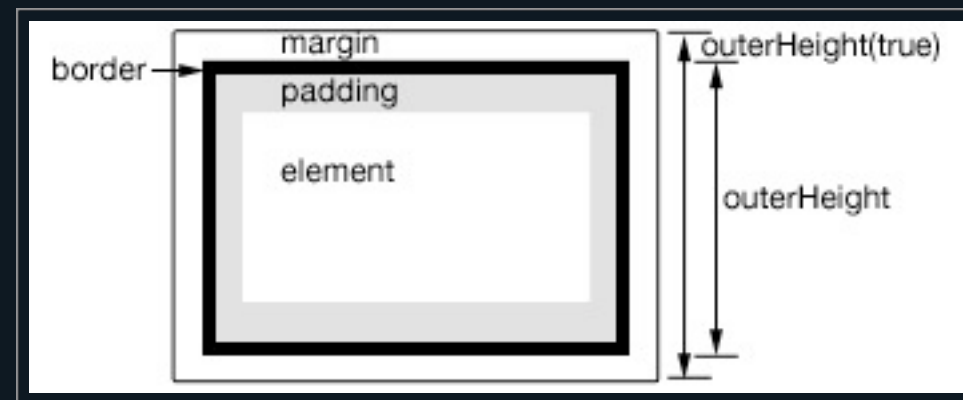
.innerWidth() 함수 : 넓이 가져오기

```
$(document).ready(function() {  
    $('p:nth-child(2)').innerWidth();  
});
```

```
p { margin:10px; padding:5px; border:2px solid #666; }
```


.outerHeight() 함수 : '~보더까지'

.outerHeight() 함수는 선택된 대상의 콘텐츠 + 패딩 + 보더 높이 값을 반환(return)합니다.





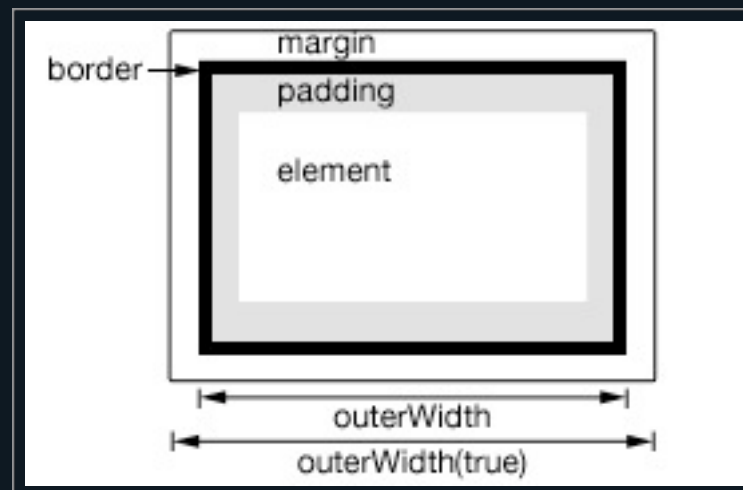
.outerHeight() 함수 : 높이 가져오기

```
$(document).ready(function() {  
    $('p:nth-child(2)').outerHeight();  
});
```

```
p { margin:10px; padding:5px; border:2px solid #666; }
```

.outerWidth() 함수 : ‘~보더까지’

.outerWidth() 함수는 선택된 대상의 콘텐츠 + 패딩 + 보더 넓이 값을 반환(return)합니다.



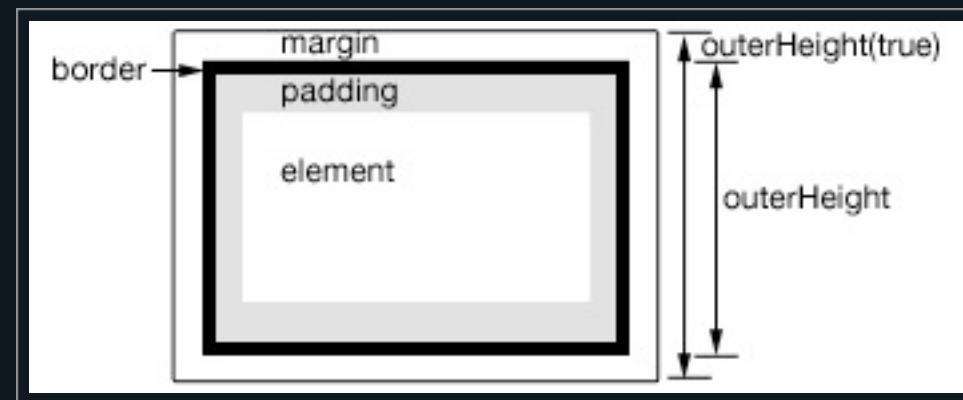
.outerWidth() 함수 : 넓이 가져오기

```
$(document).ready(function() {  
    $('p:nth-child(2)').outerWidth();  
});
```

```
p { margin:10px; padding:5px; border:2px solid #666; }
```

.outerHeight(true) 함수 : ‘~마진까지’

.outerHeight() 함수는 선택된 대상의 콘텐츠 + 패딩 + 보더 + 마진 높이 값을 반환(return)합니다.



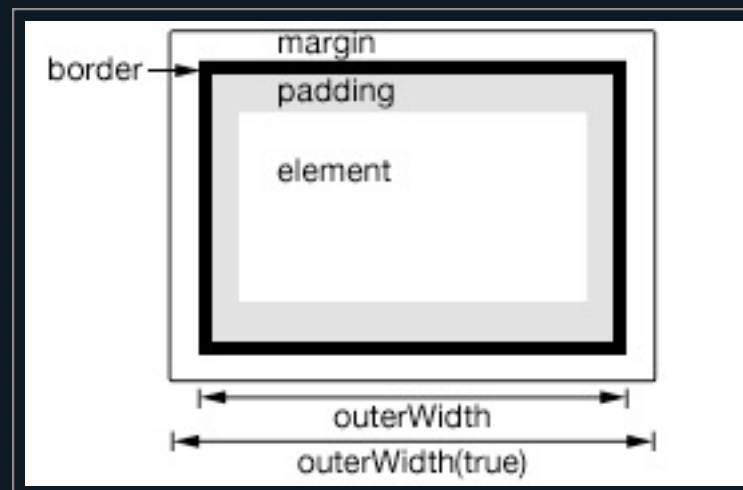
.outerHeight(true) 함수 : 높이 가져오기

```
$(document).ready(function() {  
    $('p:nth-child(2)').outerHeight(true);  
});
```

```
p { margin:10px; padding:5px; border:2px solid #666; }
```

.outerWidth(true) 함수 : ‘~마진까지’

.outerWidth() 함수는 선택된 대상의 콘텐츠 + 패딩 + 보더 + 마진 넓이 값을 반환(return)합니다.



.outerWidth(true) 함수 : 넓이 가져오기

```
$(document).ready(function() {  
    $('p:nth-child(2)').outerWidth(true);  
});
```

```
p { margin:10px; padding:5px; border:2px solid #666; }
```


Offset

jQuery - CSS 좌표(좌측상단 기점)

.offset() 함수

.offset() 함수는 화면 뷰포트(viewport)의 좌측 상단으로부터 대상의 좌표를 반환합니다.

.offset() 함수 : 대상의 좌표 가져오기

```
$(document).ready(function() {  
    var p_offset = $('p').offset();  
    alert('단락의 x축 좌표점은 ' + p_offset.left + '이다.');
```

```
});
```

.offset({top:value, left:value}) : 좌표 설정

```
$(document).ready(function() {  
    $('p').offset({  
        top: 50,  
        left: 300  
    });  
});
```

.offsetParent() 함수

.offsetParent() 함수는 대상의 부모 중 포지션 값을 가지는 대상을 반환(return)합니다.

.offsetParent() 함수 : 포지션 부모찾기

```
$(document).ready(function() {  
    var p_first_offsetP = $('p:first').offsetParent();  
    console.log('첫번째 단락의 포지션 부모는 : ' + p_first_offsetP + '이다.');
```

```
});
```

.position() 함수

.position() 함수는 offsetParent()으로부터의 상대 위치 좌표를 반환(return)합니다.

.position() 함수 : 부모로부터 대상의 위치

```
$(document).ready(function() {  
    var p_first_pos = $('p:first').position();  
    console.log(p_first_pos.top);  
    console.log(p_first_pos.left);  
});
```




.scrollTop() 함수

.scrollTop() 함수는 대상의 세로 스크롤 위치를 반환(return)합니다.

.scrollTop() 함수 : 대상의 스크롤 상단 위치

```
$(document).ready(function() {  
    var win_scrollTop = $(window).scrollTop();  
    console.log(win_scrollTop);  
});
```



.scrollTop() 함수

.scrollTop() 함수는 대상의 가로 스크롤 위치를 반환(return)합니다.

.scrollLeft() 함수 : 대상의 스크롤 왼쪽 위치

```
$(document).ready(function() {  
    var win_scrollTop = $(window).scrollLeft();  
    console.log(win_scrollLeft);  
});
```



.scrollTop(value) 함수

.scrollTop() 함수는 대상의 세로 스크롤 위치를 설정(setting)합니다.

.scrollTop() 함수 : 대상 스크롤 상단 위치 설정

```
$(document).ready(function() {  
    var win_scrollTop = $(window).scrollTop(300);  
});
```



.scrollLeft(value) 함수

.scrollTop() 함수는 대상의 가로 스크롤 위치를 설정(setting)합니다.

.scrollLeft() 함수 : 대상 스크롤 좌측 위치 설정

```
$(document).ready(function() {  
    var win_scrollTop = $(window).scrollLeft(40);  
});
```