INTERACTION PROGRAMMING 2

인터랙션 프로그래밍 2

jQuery Events

2018. 10. 4.

jQuery Manipulation

DOM Replacement

DOM Replacement

```
//DOM Replacement
$(this).replaceAll('p');
```

DOM Replacement

```
//DOM Replacement
$(this).replaceAll('p');
$(this).replaceWith();
```

DOM Replacement - .replaceAll()

```
$('<b>Hello</b>').replaceAll('p');
```

DOM Replacement - .replaceWith()

```
$('p').replaceWith('<b>Hello</b>');
```

Copying

Copying

```
//Copying.
$(this).clone();
```

Copying

```
$('.hello').clone().appendTo('.goodbye');
$('.goodbye').append($('.hello').clone());
```

General Attributes

General Attributes

```
//General Attributes
$(this).attr('checked');
```

General Attributes

```
//General Attributes
$(this).attr('checked');
$(this).prop('checked');
```

General Attributes

```
//General Attributes
$(this).attr('checked');
$(this).prop('checked');
$(this).is(':checked');
```

General Attributes - .attr()

```
$(this).attr('title', 'Attribute title');
$(this).attr('alt', 'Alt text');
```

General Attributes - .attr()

```
$(this).removeAttr('title');
$(this).removeAttr('alt');
```

General Attributes - .prop()

```
$(this).prop('disabled', false);
$(this).prop('checked', true);
```

General Attributes - .prop()

```
$('input[type="checkbox"]').prop('disabled', true);
```

General Attributes - .prop()

```
$('input[type="checkbox"]').prop('disabled', true);
$(this).prop();
```

General Attributes - .prop(), .val()

```
$('input[type="checkbox"]').prop('disabled', true);
$(this).prop();
$(this).val();
```

Style Properties

Style Properties

```
//Style Properties
$(this).css();
```

Style Properties

```
$(this).css('color', '#000');
$(this).css({'background-color' : '#000', 'colro' : '#fff'});
$(this).css('width', '+=200');
```

Style Properties - Dimension

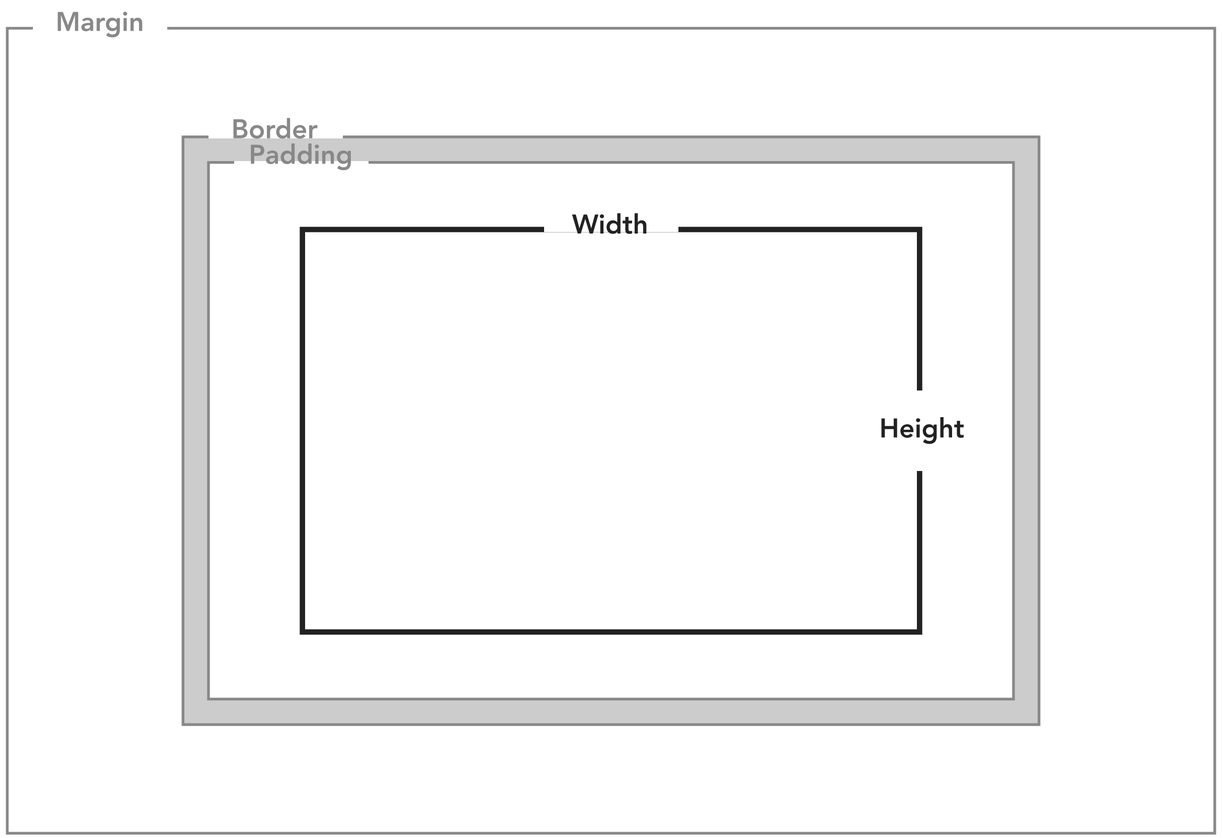
```
$(this).width();
$(this).innerWidth();
$(this).outerWidth();
$(this).outerWidth(true);
```

Style Properties - Dimension

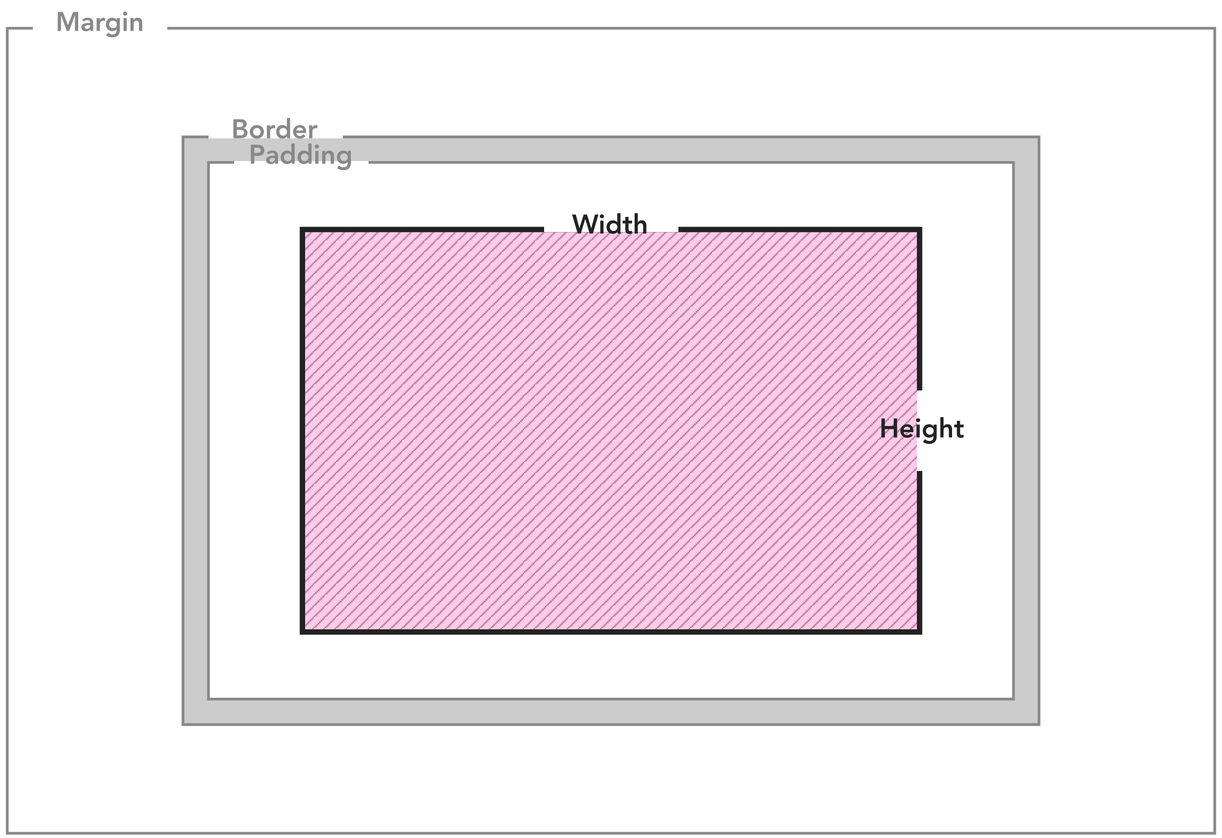
```
$(this).width();
$(this).innerWidth();
$(this).outerWidth();
$(this).outerWidth(true);

$(this).height();
$(this).innerHeight();
$(this).outerHeight();
$(this).outerHeight(true);
```

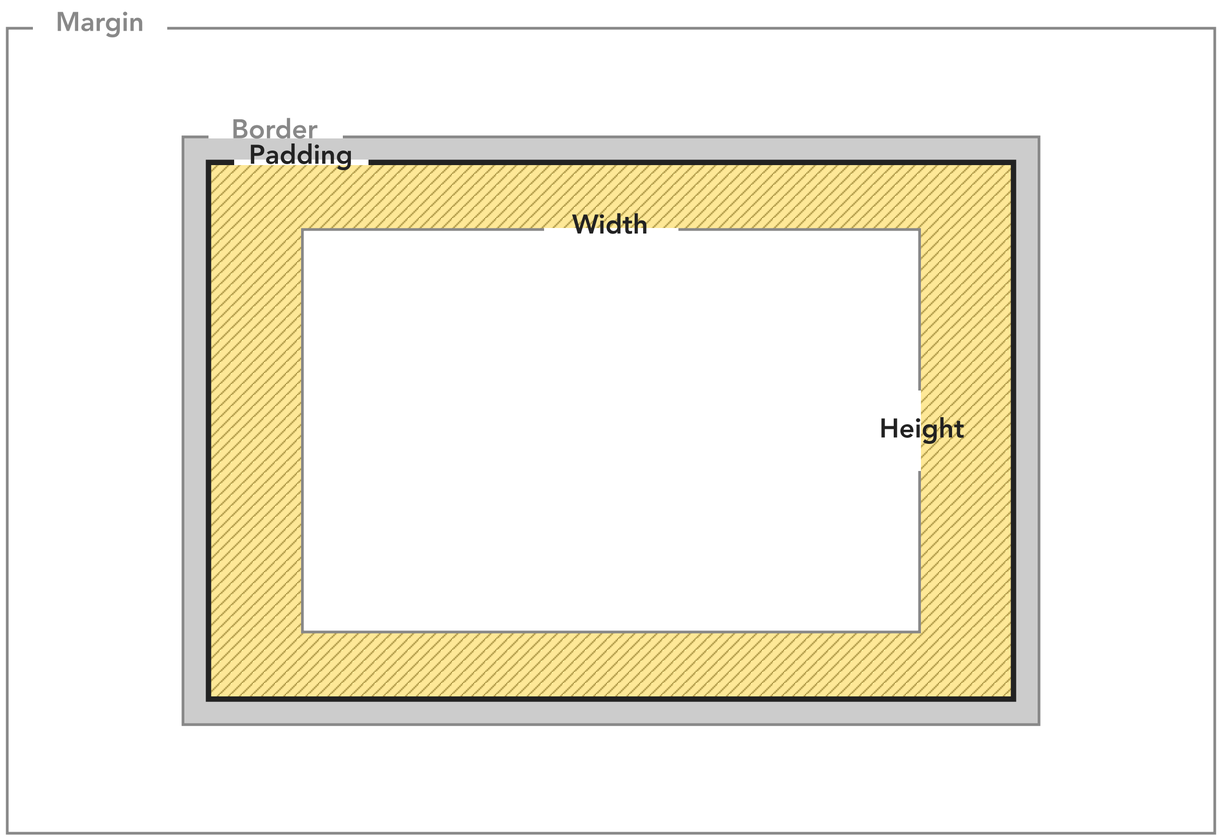
Style Properties - Dimension



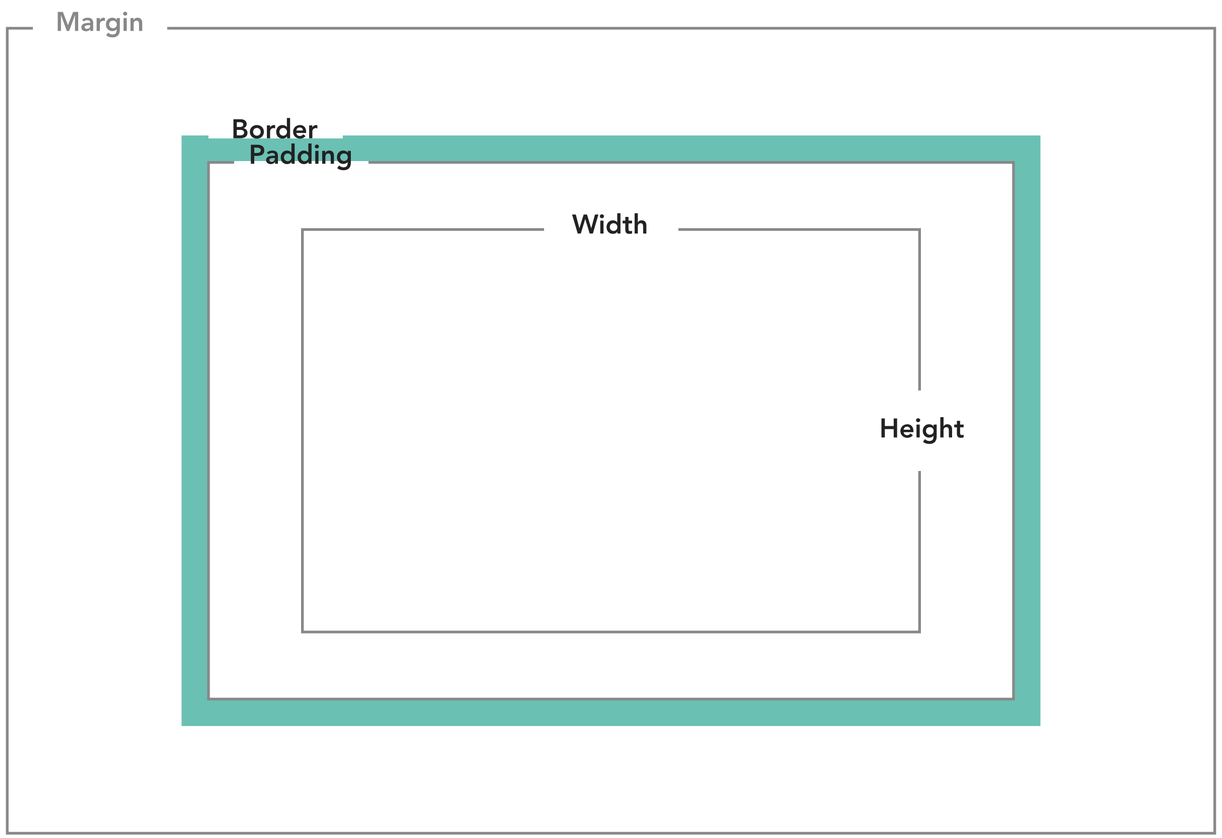
Style Properties - Dimension



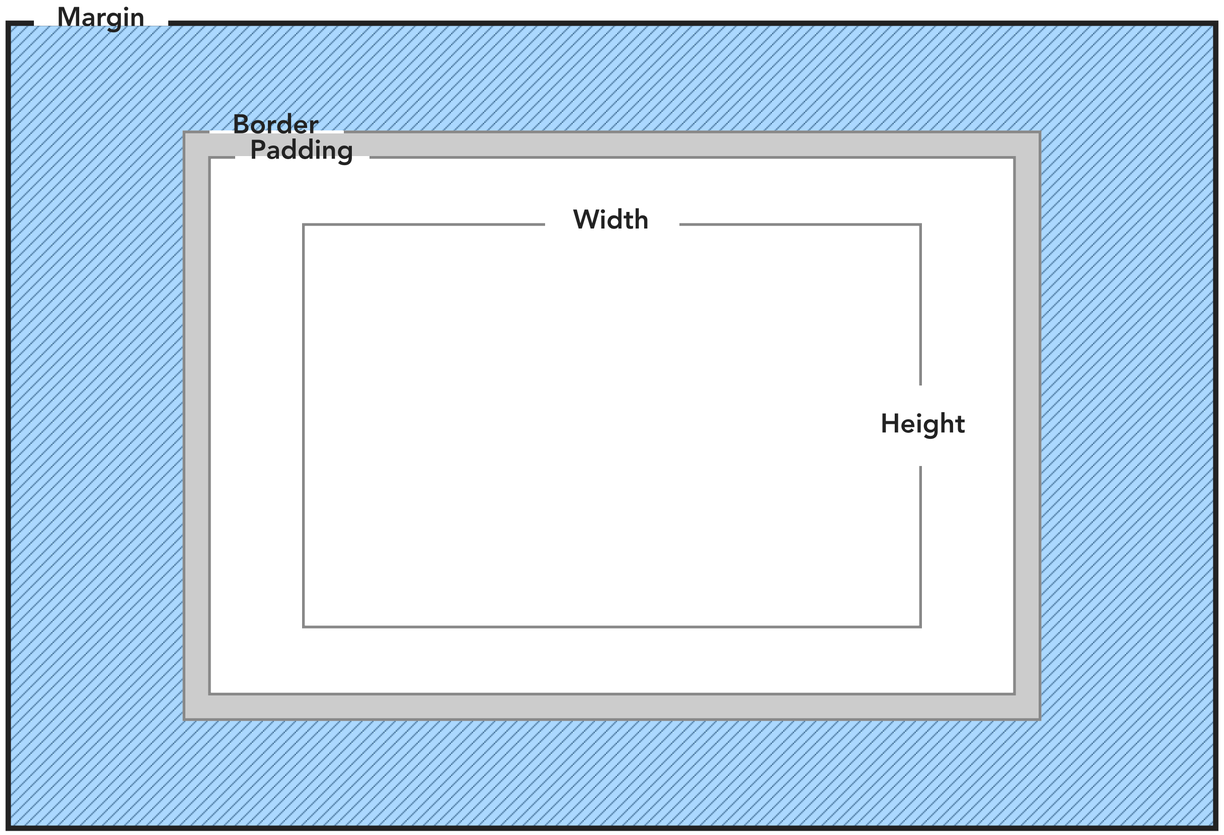
Style Properties - Dimension



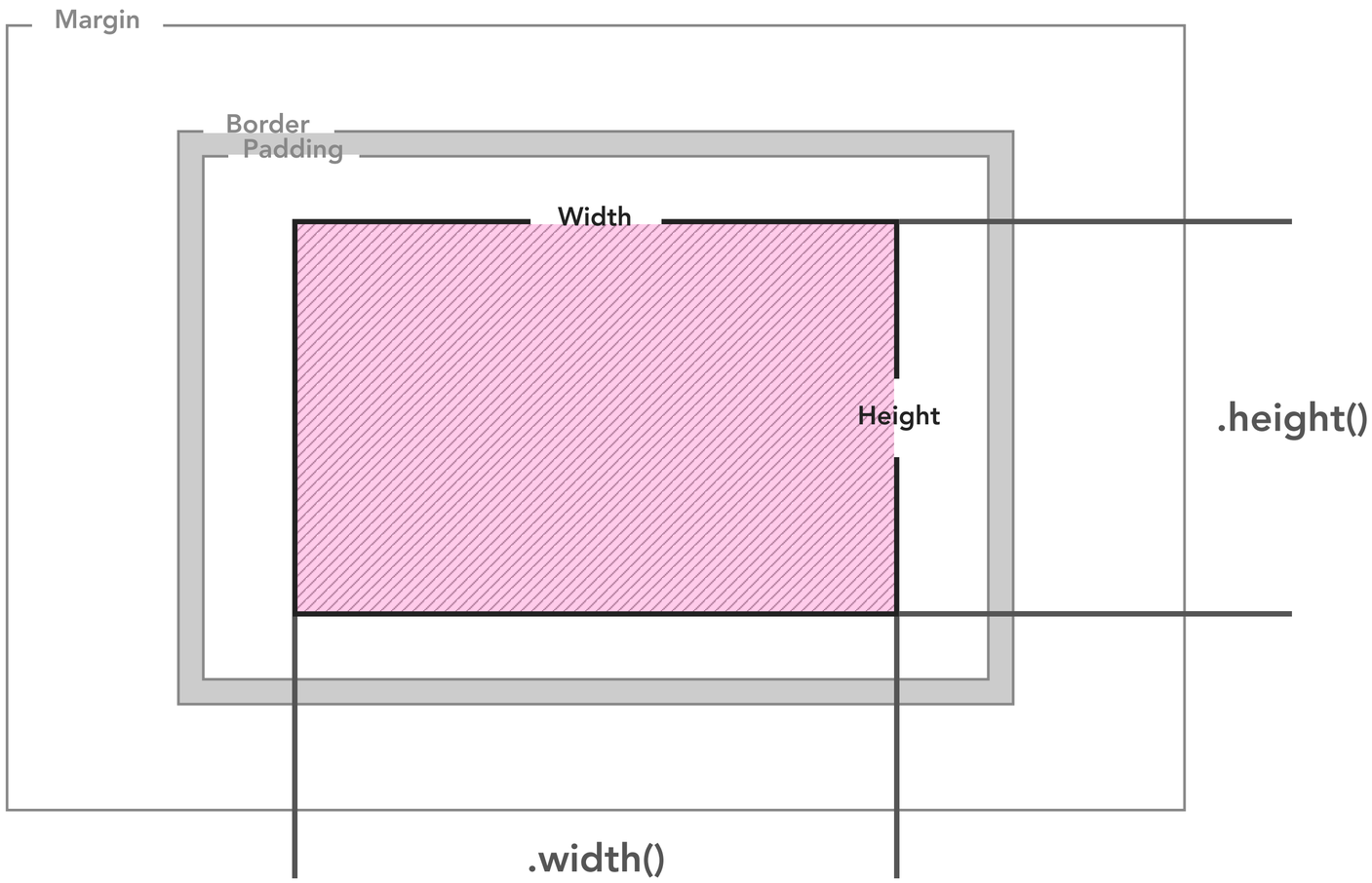
Style Properties - Dimension



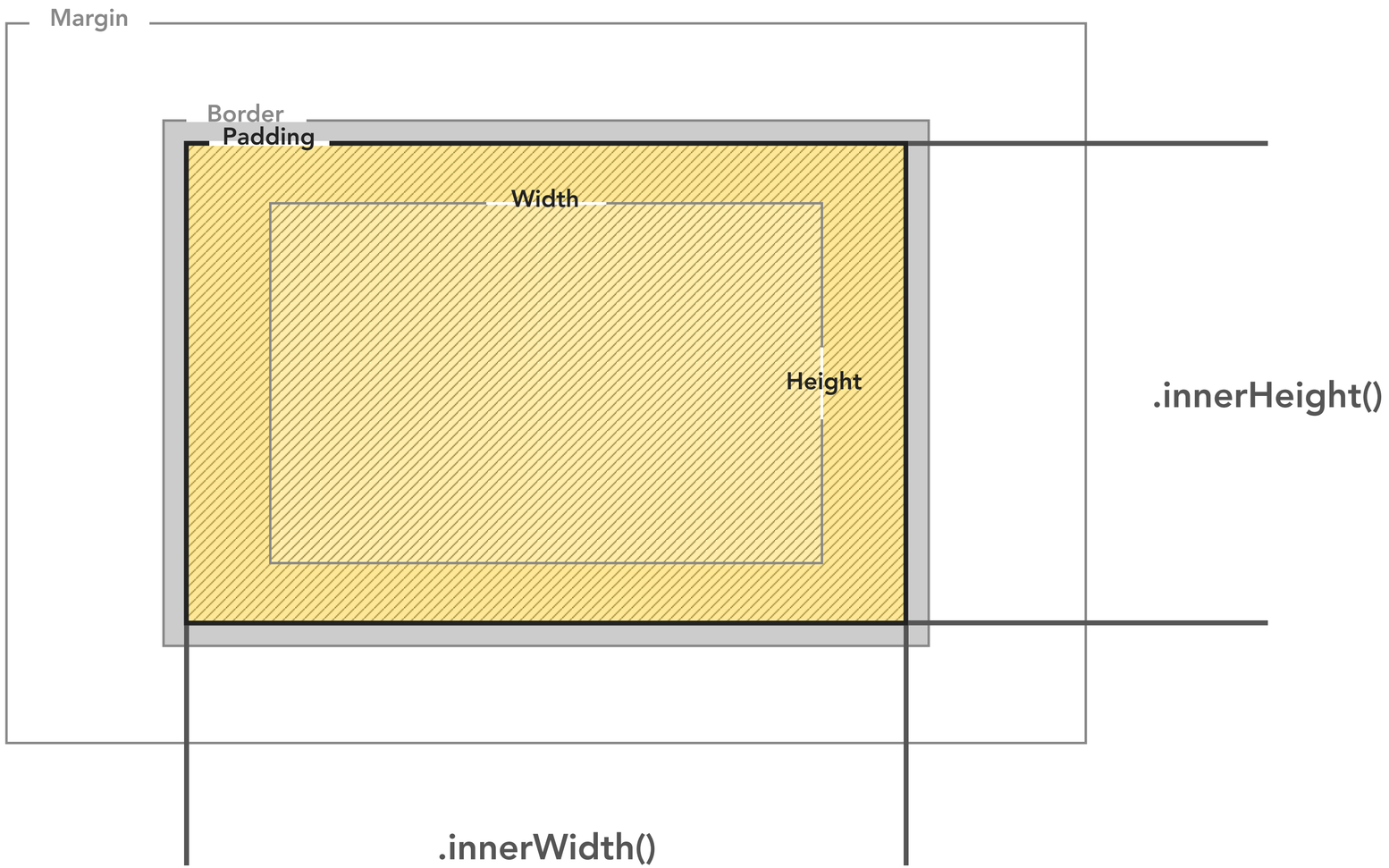
Style Properties - Dimension



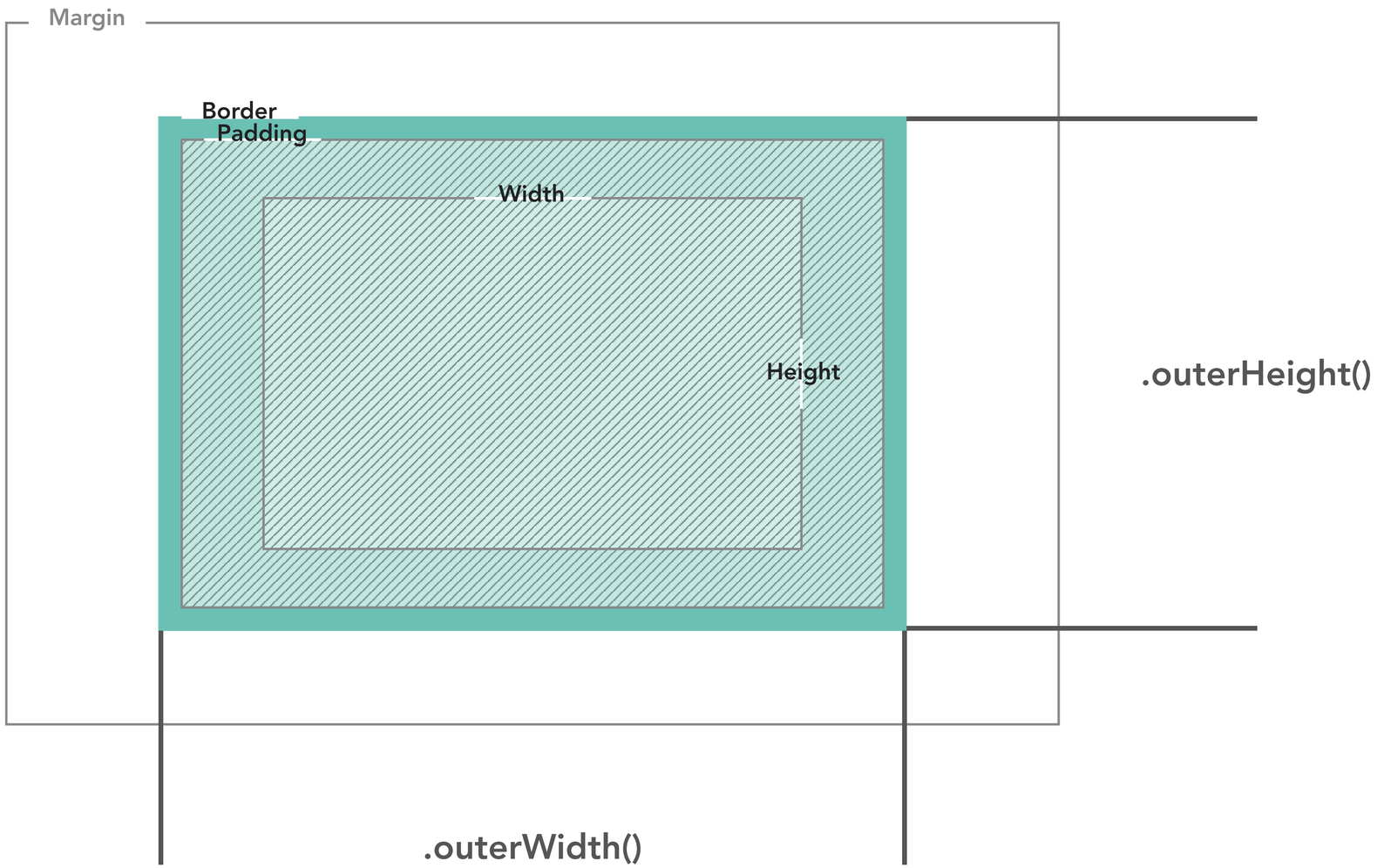
Style Properties - Dimension



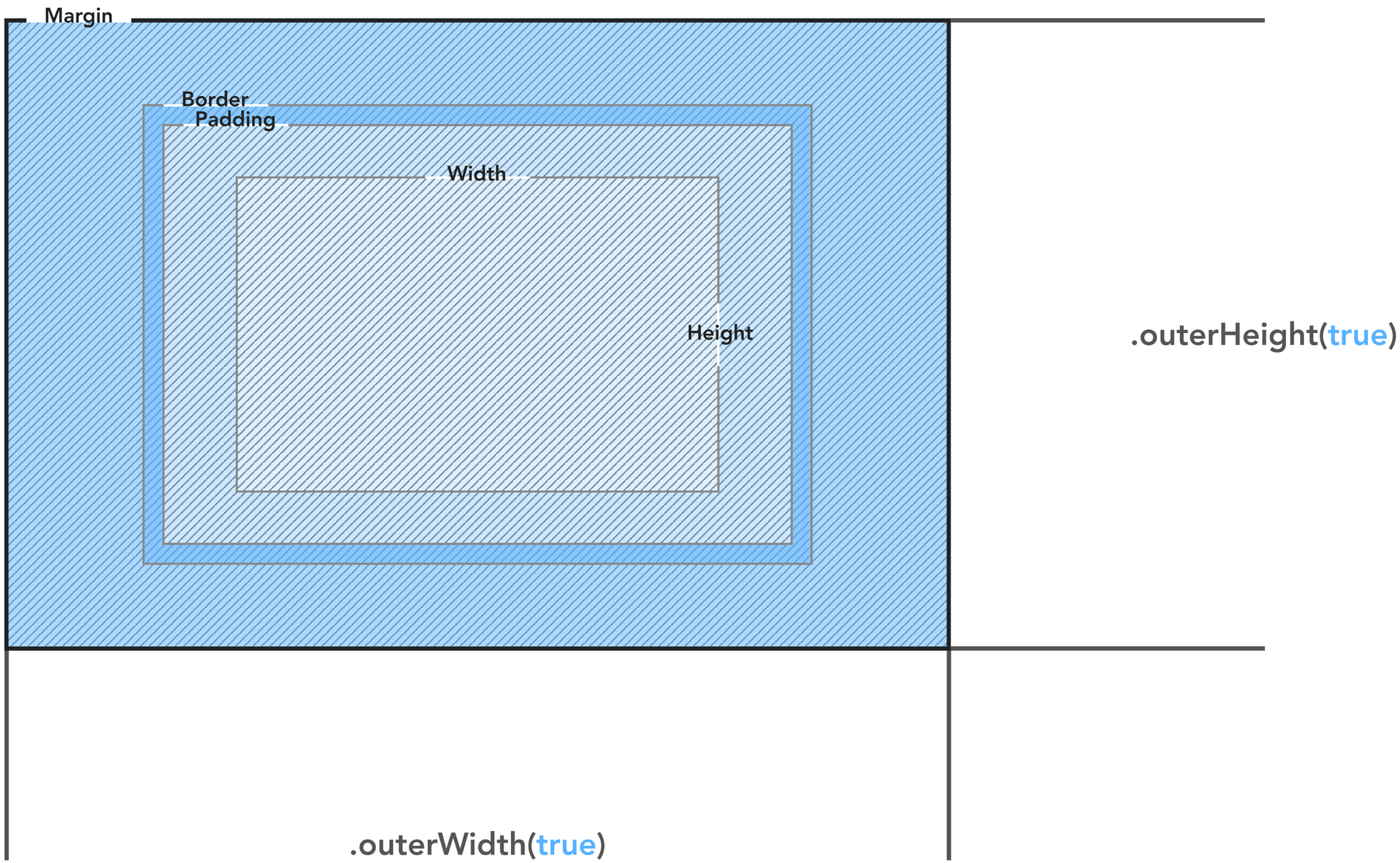
Style Properties - Dimension



Style Properties - Dimension



Style Properties - Dimension



```
$(this).width();
$(this).width(100);
```

```
$(this).innerWidth();
$(this).innerWidth(100);
```

```
$(this).outerWidth();
$(this).outerWidth(200);
$(this).outerWidth(true);
$(this).outerWidth(300, true);
```

```
$(this).height();
$(this).height(100);
```

```
$(this).innerHeight();
$(this).innerHeight(100);
```

```
$(this).outerHeight();
$(this).outerHeight(200);
$(this).outerHeight(true);
$(this).outerHeight(300, true);
```

```
$(this).position();
$(this).position().top;
$(this).position().left;
```

```
$(this).offset();
$(this).offset().top;
$(this).offset().left;
```

```
$(this).scrollTop();
$(this).scrollTop(300);
```

```
$(this).scrollLeft();
$(this).scrollLeft(300);
```

Style Properties - Window scroll

```
$(window).scrollTop();
$(window).scrollLeft();
```

Quest 1.

my-name 아이디를 가진 span 노드를 생성하고 '제 이름은 본인이름입니다.' 의 내용을 추가하여 hello 아이디를 가진 div 노드의 첫 번째 노드로 추가합니다.

Quest 2.

(1) images 배열의 원소 수 만큼 banner-item 클래스를 가지는 div 노드 객체를 생성합니다.

(2) 배열의 원소를 src 속성으로 가지는 img 노드를 생성하고, (1) 에서 생성한 div 노드에 추가합니다.

(3) img 노드를 포함하고 있는 div 노드를 banner-container 클래스를 가지는 div 노드에 추가합니다.

Quest 3.

p.move-top 노드를 body 의 가장 위로 이동시킵니다.

이동 후 p.move-top 에 move 라는 클래스를 추가하고, 노드 안의 내용을 '위로 이동했습니다.' 로 변경합니다.

Quest 4.

p.move-bottom 노드를 body 의 가장 아래로 이동시킵니다.

이동 후 p.move-bottom 에 move 라는 클래스를 추가하고, 노드 안의 내용을 '아래로 이동했습니다.'

로 변경합니다.

Quest 5.

p.error 노드를 삭제합니다.

Quest 6.

p.modify 노드의 내용을 ‘수정되었습니다.’ 로 수정합니다.

Quest 7.

p.modify 노드의 내용을 비우고 노드를 삭제합니다.

Quest 8.

ul.list-1 의 첫번 째 li 노드의 뒤에 반복문을 이용하여 ‘리스트 2’, ‘리스트 3’, ‘리스트 4’ 의 내용을 가진 li 노드를 생성하여 추가합니다.

Quest 9.

ul.list-2 의 첫번 째 li 노드의 앞에 반복문을 이용하여 ‘리스트 1’, ‘리스트 2’, ‘리스트 3’, ‘리스트 4’ 의 내용을 가진 li 노드를 생성하여 추가합니다.

Quest 10.

button.btn-1 를 클릭하여 div.box-1 의 width 를 조건에 따라 100px씩 증감시키는 함수를 만듭니다.

(1) div.box-1 의 width 는 100px 부터 시작됩니다.

(2) div.box-1 의 width 가 500px 이상이 될 때까지 클릭 1회 당 100px씩 증가합니다.

(3) div.box-1 의 width 가 500px 이상이 되면 0px 이 될 때까지 1회 당 100px 씩 감소합니다.

(4) **div.box-1 의 width 가 0px 이 되면 (2)의 조건과 (3) 조건을 반복합니다.**

jQuery Events

DOM, 브라우져, 윈도우, 마우스, 키보드, 폼 등의 동작 제어한다.

Event 연결

Mouse Events

Keyboard Events

Form Events

Window Events

Browser Events

Event 속성

Event 연결

.bind() .unbind() ~~.live()~~ ~~.die()~~

.delegate() .undelegate()

.on() .off() .one()

.trigger() .triggerHandler()

.proxy()

Event 연결

```
.bind()
```

.bind()

Event가 발생하면 실행될 함수를 지정한다.(이미 생성된 DOM 기준)

Event 연결

```
.bind()
.bind(eventType [, eventData], handler)
.bind(eventType [, eventData] [, preventBubble])
.bind(events)
```

Event 연결

```
$('#menu-1').bind('click', function(){
});
$('#menu-1').bind('mouseenter mouseleave', function(){
});
```

```
.unbind()
```

.unbind()

bind()를 해제한다.

```
.unbind()
.unbind(eventType [, handler])
.unbind(eventType, false)
.unbind(event)
.unbind()
```

```
$('#menu-1').unbind();
$('#menu-1').unbind('click');
$('#menu-1').unbind('click', handler);
```

```
.delegate()
```

.delegate()

동적으로 생성된 DOM 객체들의 이벤트를 연결한다. (이벤트 버블 방지)

```
.delegate()

.delegate(selector, eventType, handler)
.delegate(selector, eventType, eventData, handler)
.delegate(selector, events)
```

Event 연결

```
$('body').delegate('a', 'click', function(){
});
```

```
.undelegate()
```

```
.undelegate()
```

delegate()를 제거한다.

```
.undelegate()

.undelegate()
.undelegate(selector, eventType)
.undelegate(selector, eventType, handler)
.undelegate(selector, events)
.undelegate(namespace)
```

```
$('body').undelegate('a', 'click');
```

Event 연결

```
.on()
```

.on()

동적으로 생성된 DOM 객체들의 이벤트를 연결한다. (1.7 이후 버전)

Event 연결

```
.on()
.on(events [, selector] [, data], handler)
.on(events [, selector] [, data])
```

Event 연결

```
$('a').on('click', function(){
});
$('body').on('click', 'a', function(){
});
```

```
.off()
```

```
.off()
```

on()을 제거한다.

```
.off()
.off(events [, selector] [, handler])
.off(events [, selector])
.off(event)
.off()
```

```
$('a').off('click');
$('body').off('click', 'a');
$('body').off('click', 'a', handler);
```

```
.one()
```

.one()

Event가 발생하면 실행될 함수를 지정한다. 하지만 딱 한번만 이벤트가 실행된다.

```
.one()
.one(events [, data], handler)
.one(events [, selector] [, data], handler)
.one(events [, selector] [, data])
```

Event 연결

```
$('#menu-1').one('click', function(){
});
```

```
.trigger()
```

```
.trigger()
```

선택된 요소의 이벤트를 발생시킨다.

```
.trigger()

.trigger(eventType [, extraParameters])
.trigger(event [, extraParameters])
```

```
$('#menu-1').trigger('click');
```

```
.triggerHandler()
```

.triggerHandler()

.trigger()와 비슷하게 동작한다.

예외사항

- form 전송은 동작하지 않는다.
- 선택된 요소 중 첫번째 요소에만 동작한다.
- 브라우저 기본동작, 버블링은 일어나지 않는다.

```
.triggerHandler()
.triggerHandler(eventType [, extraParameters])
.triggerHandler(event [, extraParameters])
```

```
$('#menu-1').triggerHandler('focus');
```

Mouse Events

Mouse Events

.click() .dblclick()

.mousedown() .mouseup() .mouseover() .mouseseenter()

.mousemove() . mouseout() . mouseleave()

.hover() . toggle()

Mouse Events

```
.click()
```

```
.click()
```

마우스 클릭시 이벤트를 발생시킨다.

Mouse Events

```
.click()
.click(handler)
.click([eventData], handler)
.click()
```

Mouse Events

```
$('#menu-1').click(function(){
});
$('#menu-1').on('click', function(){
});
```

Mouse Events

```
.dblclick()
```

```
.dblclick()
```

마우스 더블클릭시 이벤트를 발생시킨다.

Mouse Events

```
.dblclick()
.dblclick(handler)
.dblclick([eventData], handler)
.dblclick()
```

Mouse Events

```
$('#menu-1').dblclick(function(){
});
$('#menu-1').on('dblclick', function(){
});
```

Mouse Events

```
.mousedown()
```

```
.mousedown()
```

마우스 버튼을 누른 순간 이벤트를 발생시킨다.

Mouse Events

```
.mousedown()
.mousedown(handler)
.mousedown([eventData], handler)
.mousedown()
```

Mouse Events

```
$('#menu-1').mousedown(function(){
});
$('#menu-1').on('mousedown', function(){
});
```

Mouse Events

```
.mouseup()
```

```
.mouseup()
```

마우스 버튼을 놓는 순간 이벤트를 발생시킨다.

Mouse Events

```
.mouseup()
.mouseup(handler)
.mouseup([eventData], handler)
.mouseup()
```

Mouse Events

```
$('#menu-1').mouseup(function(){
});
$('#menu-1').on('mouseup', function(){
});
```

Mouse Events

```
.mouseover()
```

```
.mouseover()
```

대상에 마우스 커서가 위치했을 때 이벤트를 발생시킨다.

Mouse Events

```
.mouseover()

.mouseover(handler)
.mouseover([eventData], handler)
.mouseover()
```

Mouse Events

```
$('#menu-1').mouseover(function(){
});
$('#menu-1').on('mouseover', function(){
});
```

Mouse Events

```
.mouseenter()
```

.mouseenter()

대상에 마우스가 위치했을 때 이벤트를 발생시킨다.

Mouse Events

```
.mouseenter()
.mouseenter(handler)
.mouseenter([eventData], handler)
.mouseenter()
```

Mouse Events

```
$('#menu-1').mouseenter(function(){
});
$('#menu-1').on('mouseenter', function(){
});
```

Mouse Events

```
.mousemove()
```

.mousemove()

대상에서 마우스 커서가 움직였을 때 이벤트를 발생시킨다.

Mouse Events

```
.mousemove()

.mousemove(handler)
.mousemove([eventData], handler)
.mousemove()
```

Mouse Events

```
$('#menu-1').mousemove(function(){
});
$('#menu-1').on('mousemove', function(){
});
```

Mouse Events

```
.mouseout()
```

```
.mouseout()
```

대상에서 마우스가 커서가 벗어 났을때 이벤트를 발생시킨다.

Mouse Events

```
.mouseout()
.mouseout(handler)
.mouseout([eventData], handler)
.mouseout()
```

Mouse Events

```
$('#menu-1').mouseout(function(){
});
$('#menu-1').on('mouseout', function(){
});
```

Mouse Events

```
.mouseleave()
```

.mouseleave()

대상에서 마우스가 벗어났을 때 이벤트를 발생시킨다.

Mouse Events

```
.mouseleave()
.mouseleave(handler)
.mouseleave([eventData], handler)
.mouseleave()
```

Mouse Events

```
$('#menu-1').mouseleave(function(){
});
$('#menu-1').on('mouseleave', function(){
});
```

Mouse Events

```
.hover()
```

.hover()

대상에 마우스가 요소에 오버되었을때 첫번째 함수가 실행되고,

마우스가 요소를 떠날때 두번째 함수가 실행된다.

Mouse Events

```
.hover()
.hover(handlerIn, handlerOut)
```

Mouse Events

```
$('#menu-1').hover(function(){
});
$('#menu-1').on('mouseseenter mouseleave', function(){
});
```

Keyboard Events

Keyboard Events

.keypress()

.keydown()

.keyup()

Keyboard Events

```
.keypress()
```

.keypress()

키 입력할 때 이벤트를 발생시킨다.(enter, tab 등의 특수키 제외)

Keyboard Events

```
.keypress()
.keypress(handler)
.keypress([eventData], handler)
.keypress()
```

Keyboard Events

```
$('input').keypress(function(){
});
$('input').on('keypress', function(){
});
```

Keyboard Events

```
.keydown()
```

.keydown()

키 입력을 누르고 있을 때 이벤트를 발생시킨다. 모든 키에 대해 적용

Keyboard Events

```
.keydown()
.keydown(handler)
.keydown([eventData], handler)
.keydown()
```

Keyboard Events

```
$('input').keydown(function(){
});
$('input').on('keydown', function(){
});
```

Keyboard Events

```
.keyup()
```

```
.keyup()
```

키 입력 후 이벤트를 발생시킨다.

Keyboard Events

```
.keyup()
.keyup(handler)
.keyup([eventData], handler)
.keyup()
```

Keyboard Events

```
$('input').keyup(function(){
});
$('input').on('keyup', function(){
});
```

Form Events

Form Events

.select() .change()

.focus() .focusin() .blur() .focusout()

.submit()

Form Events

```
.select()
```

.select()

개체를 선택 했을 때 이벤트를 발생시킨다.

Form Events

```
.select()
.select(handler)
.select([eventData], handler)
.select()
```

Form Events

```
$('input').select(function(){
});
$('input').on('select', function(){
});
```

Form Events

```
.change()
```

.change()

input, textarea, select 객체의 요소값 변경할 때 이벤트를 발생시킨다.

Form Events

```
.change()
.change(handler)
.change([eventData], handler)
.change()
```

Form Events

```
$('input').change(function(){
});
$('input').on('change', function(){
});
```

Form Events

```
.focus()
```

```
.focus()
```

해당 요소에 포커스를 받았을 때 이벤트를 발생시킨다.

Form Events

```
.focus()
.focus(handler)
.focus([eventData], handler)
.focus()
```

Form Events

```
$('input').focus(function(){
});
$('input').on('focus', function(){
});
```

```
.focusin()
```

.focusin()

focus() 의 업그레이드(1.4버전)

```
.focusin()
.focusin(handler)
.focusin([eventData], handler)
.focusin()
```

Form Events

```
$('input').focusin(function(){
});
$('input').on('focusin', function(){
});
```

Form Events

```
.blur()
```

```
.blur()
```

해당 요소에 포커스를 잃었을 때 이벤트를 발생시킨다.

Form Events

```
.blur()

.blur(handler)
.blur([eventData], handler)
.blur()
```

Form Events

```
$('input').blur(function(){
});
$('input').on('blur', function(){
});
```

```
.focusout()
```

```
.focusout()
```

blur()의 업그레이드(1.4버전)

Form Events

```
.focusout()
.focusout(handler)
.focusout([eventData], handler)
.focusout()
```

Form Events

```
$('input').focusout(function(){
});
$('input').on('focusout', function(){
});
```

Form Events

```
.submit()
```

.submit()

폼 전송(submit)이 일어날 때 이벤트를 발생시킨다.

Form Events

```
.submit()
.submit(handler)
.submit([eventData], handler)
.submit()
```

Form Events

```
$('input').submit(function(){
});
$('input').on('submit', function(){
});
```

Window Events

Window Events

.resize()

.scroll()

~~.error()~~

Window Events

```
.resize()
```

```
.resize()
```

브라우저 윈도우 창 크기가 변경되었을 때 이벤트를 발생시킨다.

Window Events

```
.resize()
.resize(handler)
.resize([eventData], handler)
.resize()
```

Window Events

```
$(window).resize(function(){
});
$(window).on('resize', function(){
});
```

Window Events

```
.scroll()
```

.scroll()

DOM이 스크롤 되었을 때 이벤트를 발생시킨다.

Window Events

```
.scroll()

.scroll(handler)
.scroll([eventData], handler)
.scroll()
```

Window Events

```
$(window).scroll(function(){
});
$(window).on('scroll', function(){
});
```

Browser Events

Browser Events

~~.load()~~ ~~.unload()~~

.ready() .holdReady()

```
.ready()
```

.ready()

DOM이 모두 준비 되었을 때 이벤트를 발생시킨다.

Browser Events

```
.ready()
.ready(handler)
```

Browser Events

```
$(document).ready(function(){
});
```

```
.holdReady()
```

```
.holdReady()
```

DOM이 로드되는 것을 지연시켜준다.

```
.holdReady()

jQuery.holdReady(hold)
```

```
$.holdReady(true);

...

$.holdReady(false);
```

Event 속성

Event 속성

event.target event.currentTarget event.relatedTarget

event.result event.data event.namespace

event.pageX event.pageY event.timeStamp event.type event.which

event.preventDefault() event.isDefaultPrevented()

event.stopPropagation() event.stopImmediatePropagation()

event.isPropagationStopped() event.isImmediatePropagationStopped()

Event 속성

event.target event.currentTarget event.relatedTarget

event.result event.data event.namespace

event.pageX event.pageY event.timeStamp event.type event.which

event.preventDefault() event.isDefaultPrevented()

event.stopPropagation() event.stopImmediatePropagation()

event.isPropagationStopped() event.isImmediatePropagationStopped()

Event 속성

```
event.target
```

```
event.target
```

이벤트가 시작되는 DOM 요소.

Event 속성

```
$('#menu-1').on('click', function(event){
    $(event.target);
});
```

Event 속성

```
event.pageX
event.pageY
```

```
event.pageX
event.pageY
```

문서의 왼쪽 가장자리를 기준으로 마우스 위치의 X좌표, Y좌표 값.

Event 속성

```
$('#menu-1').on('mousedown', function(event){
    event.pageX;
    event.pageY;
});
```

```
event.preventDefault();
event.stopPropagation();
```

객체 요소가 가지는 기본 이벤트를 해제한다.

이벤트의 버블링을 제거한다.

Event 속성

```
$('#menu-1').on('click', function(event){
    event.preventDefault();
    event.stopPropagation();
});
```

jQuery Effects

Animation

.stop()

.animate()

```
.stop()
```

```
.stop()
```

애니메이션을 정지시킨다.

```
$('img').stop();
$('img').stop(true);
$('img').stop(true, true);
```

Animation

```
.animate()
```

```
.animate()
```

선택된 객체에 애니메이션 효과를 준다.

Animation

```
$('img').animate({'left' : '100px'}, 1000);
```

Image Slider

Image Slider

wrap

Image Slider

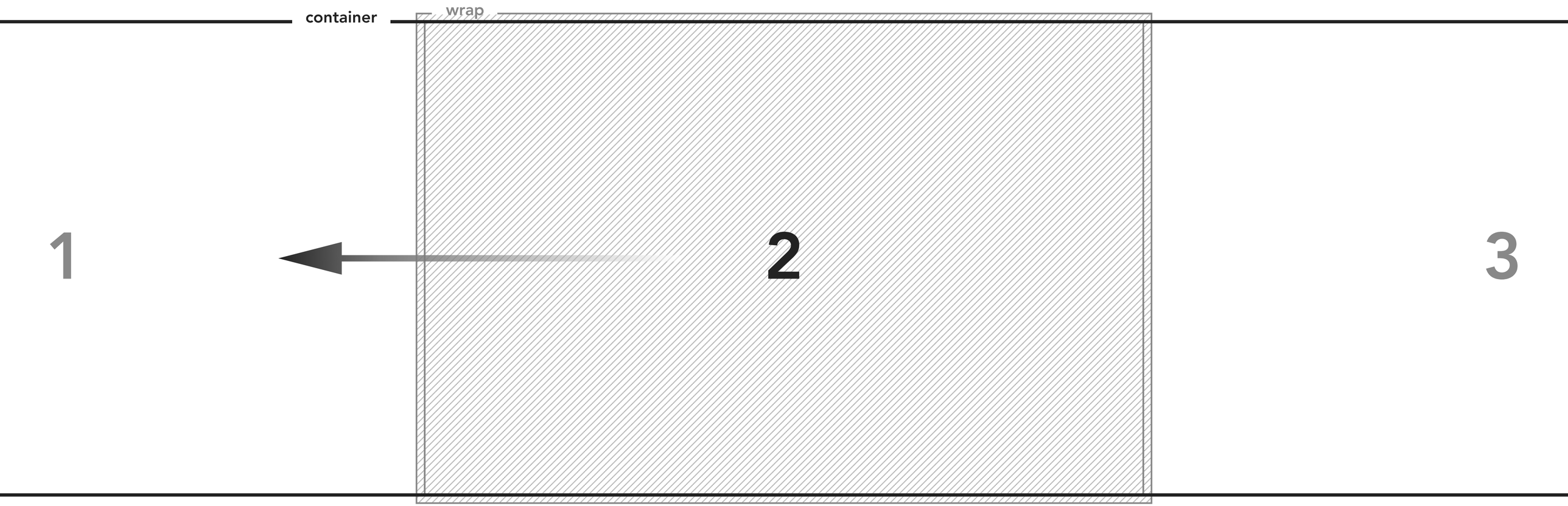



Image Slider