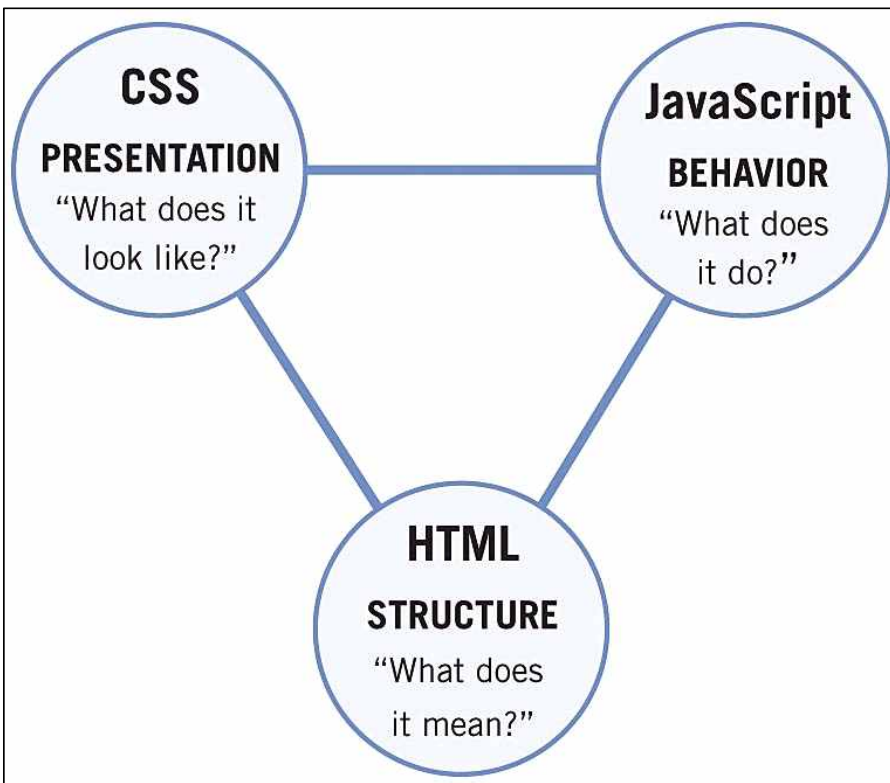
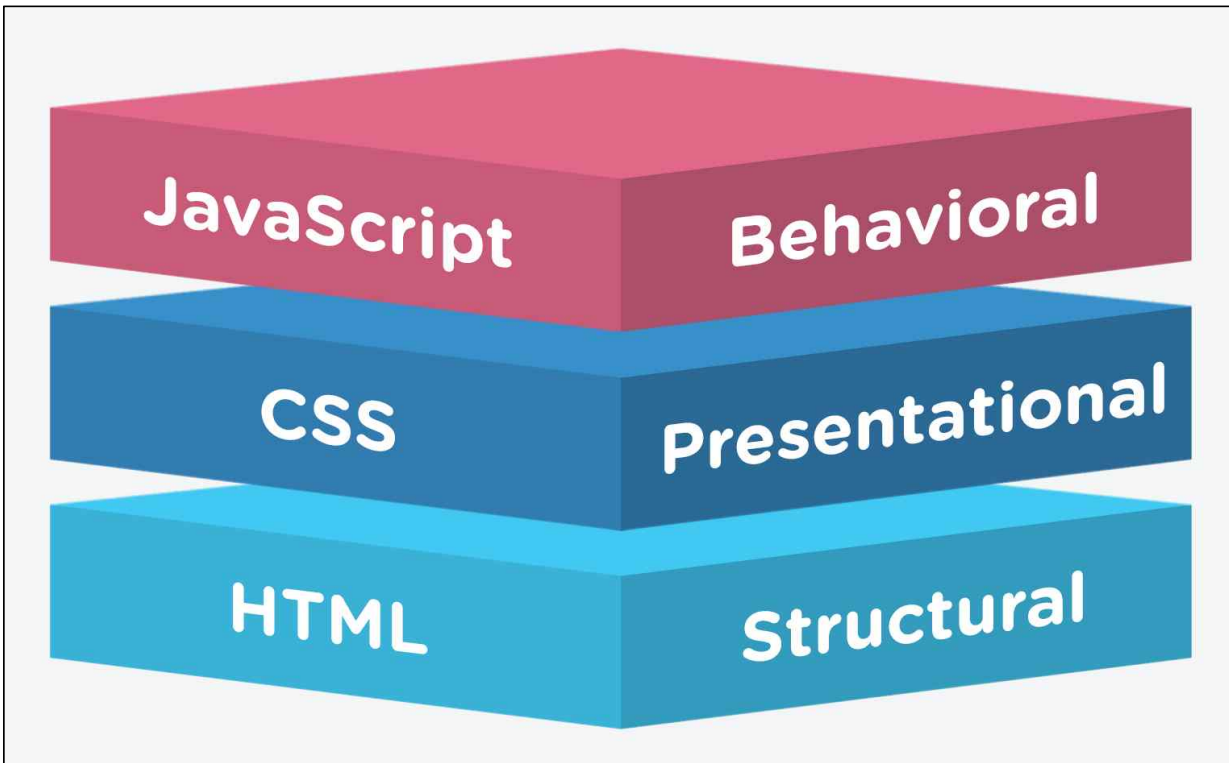


■ 웹개발 플랫폼



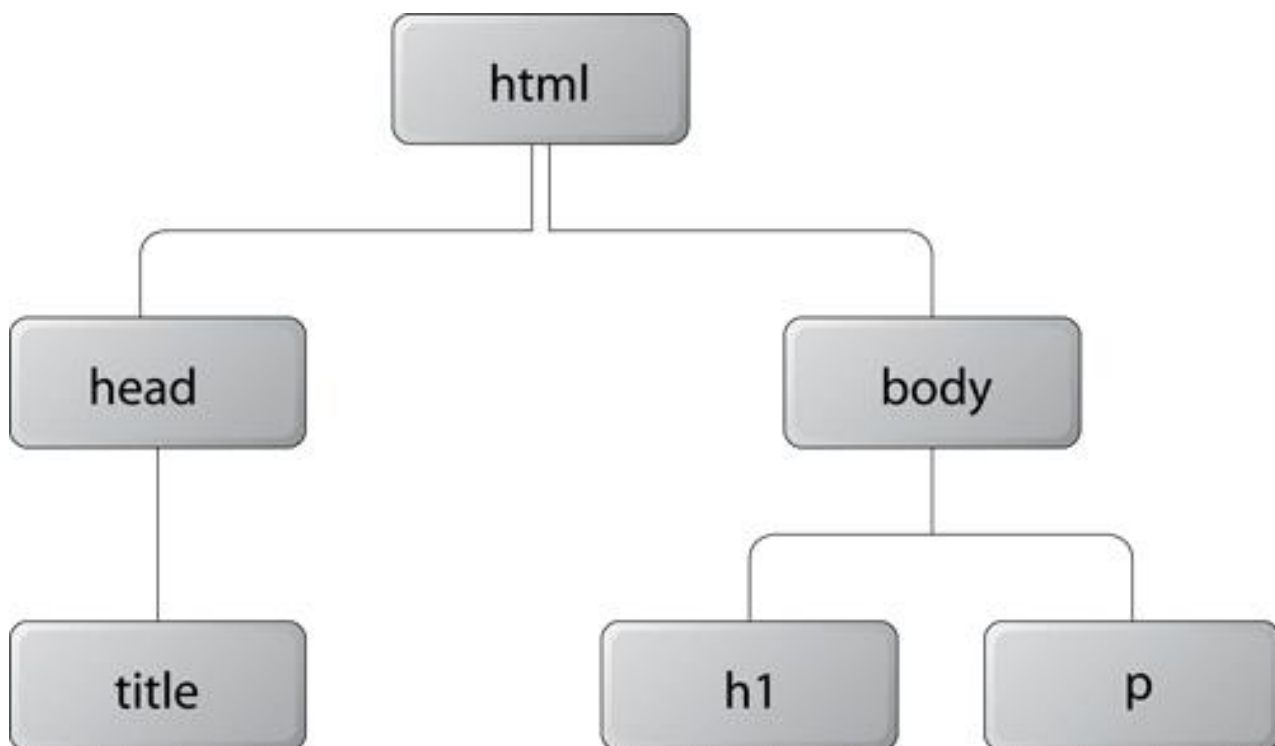
- 1) 구조와 데이터는 HTML
- 2) 표현(디자인, 레이아웃) 및 동작은 CSS
- 3) 동작 및 제어는 Javascript(및 jQuery 등 라이브러리 or 프레임워크)

4) HTML5(즉, CSS3)이후 부터는 동작을 CSS3가 담당

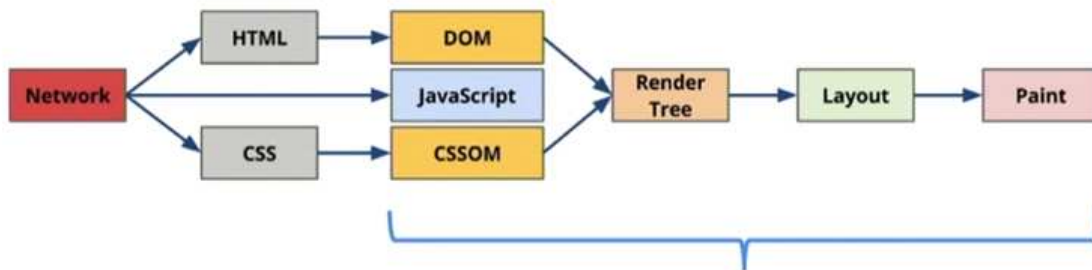
- transform / transition / animation으로

■ DOM Tree

```
<!DOCTYPE html>
<html lang="ko">
  <head>
    <title>Widgets</title>
  </head>
  <body>
    <h1>Widgets</h1>
    <p>Welcome to Widgets, the number one company
    in the world for selling widgets!</p>
  </body>
</html>
```

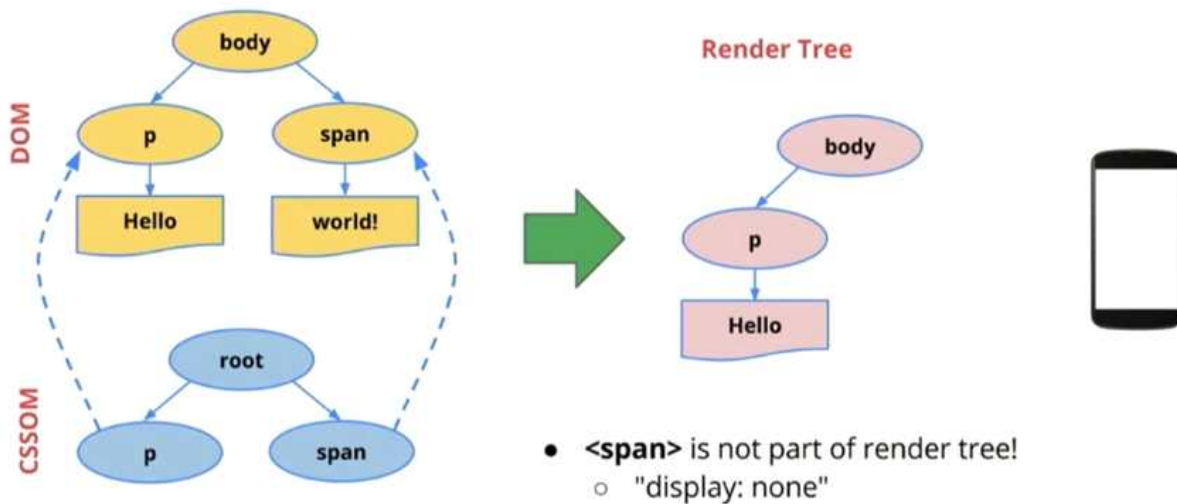


■ 브라우저에서 표현되는 방식



In-app performance: CPU + Render

DOM + CSSOM = **Render Tree(s)**



■ javascript

1. 자바스크립트란?

- 1) HTML에서 동작이 가능하게 하는 프로그래밍 언어
- 2) 객체기반의 스크립트 프로그래밍 언어
- 3) 넷스케이프사에서 자사의 웹브라우저인 Netscape 브라우저에서 동작하는 언어로 LiveScript 개발
- 4) 썬마이크로시스템즈와 넷스케이프가 공동작업시 Javascript로 명칭 변경
- 5) 자바와 자바스크립트는 이름은 비슷하나 전혀 다른 언어
- 6) 마이크로소프트는 Jscript라는 언어를 개발

1. 자바스크립트의 특징

- 1) JavaScript는 컴파일할 필요가 없는 인터프리터 언어
- 2) JavaScript는 웹브라우저의 자원을 활용하고 제어
- 3) <script>요소를 만나면 자바스크립트가 작동
- 4) 현재 표준은 ECMAScript

3. 자바스크립트의 자료형

- 1) 숫자형 : 정수/실수 구별이 없음
- 2) 문자열형 : 문자형은 없고 문자열형만(내부적으로 String)
- 3) 논리형 : true / false
- 4) null : (아무것도 아닌 값)
- 5) undefined : 정의되지 않은 값
- 6) 자바스크립트에서 false값은 'false / null / undefined / '' / 0 / NaN

7) 자바스크립트의 키워드

abstract	case	continue	extends	for
import	long	private	static	throw
var	boolean	catch	default	false
function	in	native	protected	super
throw	void	break	char	do
final	goto	instanceof	new	public
switch	transient	while	byte	class
double	finally	if	int	null
return	synchronized	true	with	case
const	else	float	implements	interface
package	short	this	try	

- 키워드를 전부 알고 있을 필요는 없음
- 키워드로 변수를 지정할 수 없음

5. 변수의 선언

```
var 변수명
```

- 자료형이 없음
- 그 어떤 자료형도 다 변수로 지정 가능(객체, 함수, 배열도 모두)

6. 기본연산

- 산수의 기본연산과 동일
- + 연산자는 문자열간의 합침도 있음

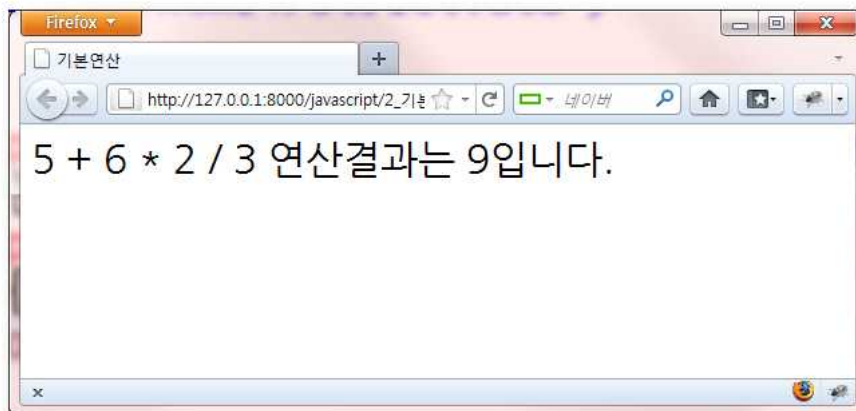
연산자	역 할	사용법
+	두 숫자를 더함, 문자열을 합침	6+12
-	한 숫자에서 다른 하나를 뺌	7-4
*	두 숫자를 곱함	50*13
/	한 숫자를 다른 숫자로 나눔	14/5

```

<script type="text/javascript">
    var number = 5 + 6 * 2 / 3;

    document.write("5 + 6 * 2 / 3 연산결과는 "+number+"입니다.");
</script>

```

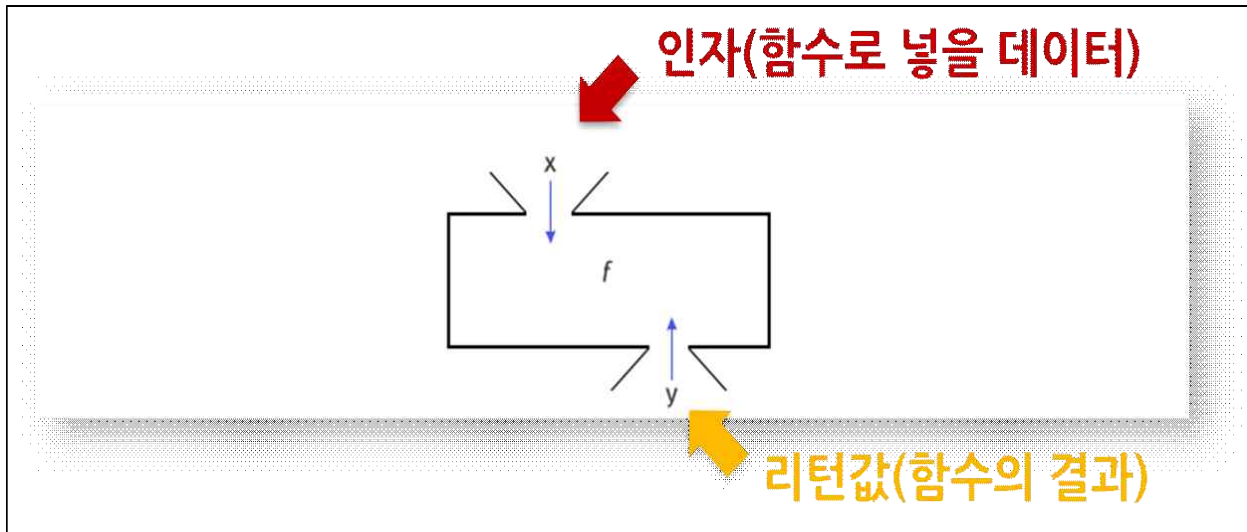


7. 연산자의 우선순위

순 위	연 산 자
1	() , [] , .
2	! ++ --
3	* / %
4	+ -
5	> >= <= <
6	== !=
7	&&
8	
9	= += -= *= /=

- ☐ ++ / -- : 증감연산자
- ☐ % : 나머지 연산자
- ☐ == : (데이터형까지 동일한)항등연산자
- ☐ && / || : and / or 연산자
- ☐ += / -=
- ☐ typeof : 어떤 자료형인지 알아내는 연산자

8. 함수



- ☐ 재사용성 때문에 함수를 사용
- ☐ 함수와 메서드로 나뉘

```
function 함수명() { ..... }
```

변수 / 함수는 언제 생성되나요?

- 전역범위의 함수 : script 시작시
- 전역범위의 var 변수 : script시작시 생성(값 대입은 X)
- 전역범위의 변수 : 코드를 만났을때 전역변수로 생성
- 함수내의 함수 : 함수 진행시
- 함수내의 속성 : 함수 진행시

9. 자바스크립트 내장 함수

■ 원래 존재하는 함수들

함수명	설 명
eval(문자열)	문자열을 자바스크립트 코드로 작동시킴
isNaN(데이터)	데이터가 NaN인지 확인
parseInt(문자열)	문자열을 정수로 변환
alert(문자열)	확인버튼이 있는 다이얼로그창을 띄움
setInterval(함수,시간)	함수를 시간후에 계속 호출
setTimeout(함수,시간)	함수를 시간후에 한번만 호출

10. 자바스크립트 내장 객체

객체명	설 명
String	문자열 객체 / indexOf()등의 메서드를 가짐
Date	현재 날짜 / 시간등의 정보를 가짐
Math	수학관련 속성과 메서드를 가짐
Array	자바스크립트의 배열은 크기 변경이 가능함

11. 난수 발생

■ 랜덤한 정수를 발생

■ Math.random() : 난수가 실수로 발생됨

■ floor()로 실수 버림 / (버림 함수)

```
Math.floor(Math.random()*갯수)+시작수
```

■ 제어문

1) 분기문

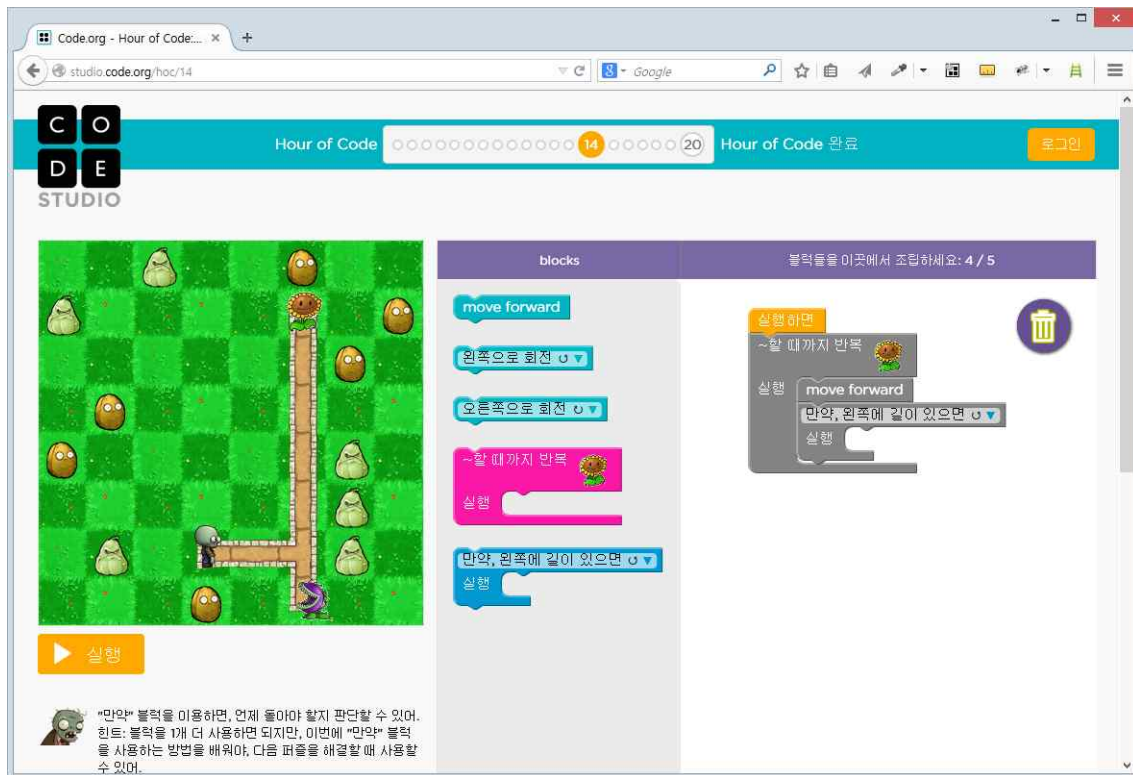


그림) code.org에서 분기문 공부

2) 반복문

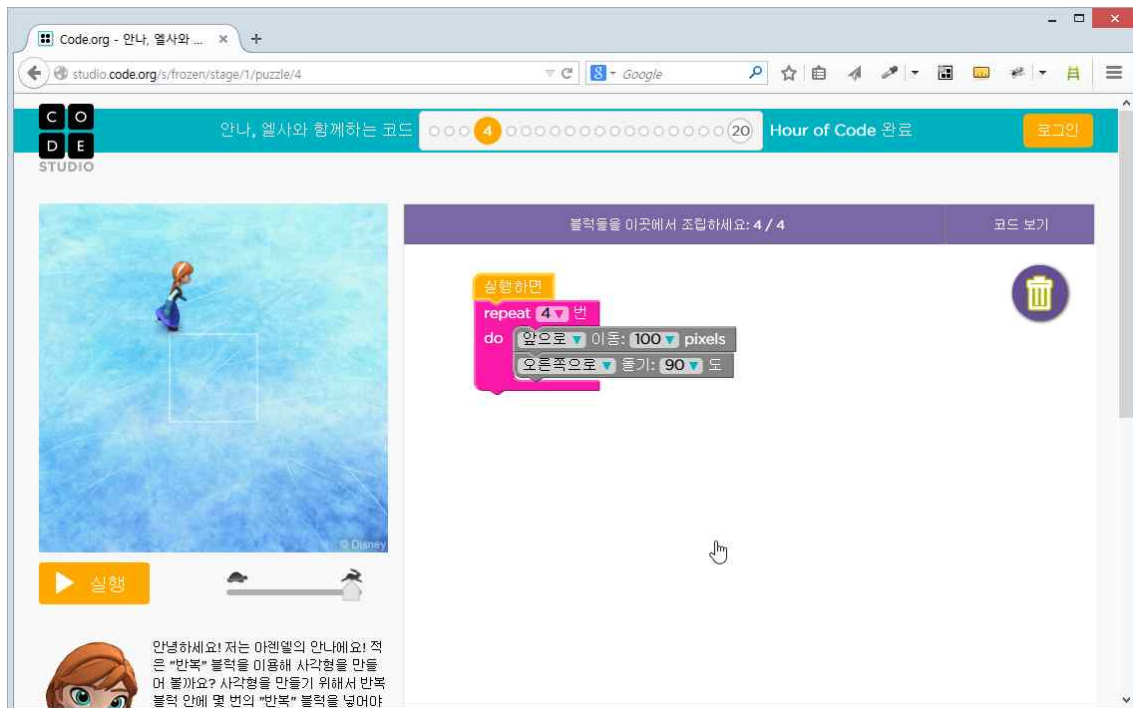
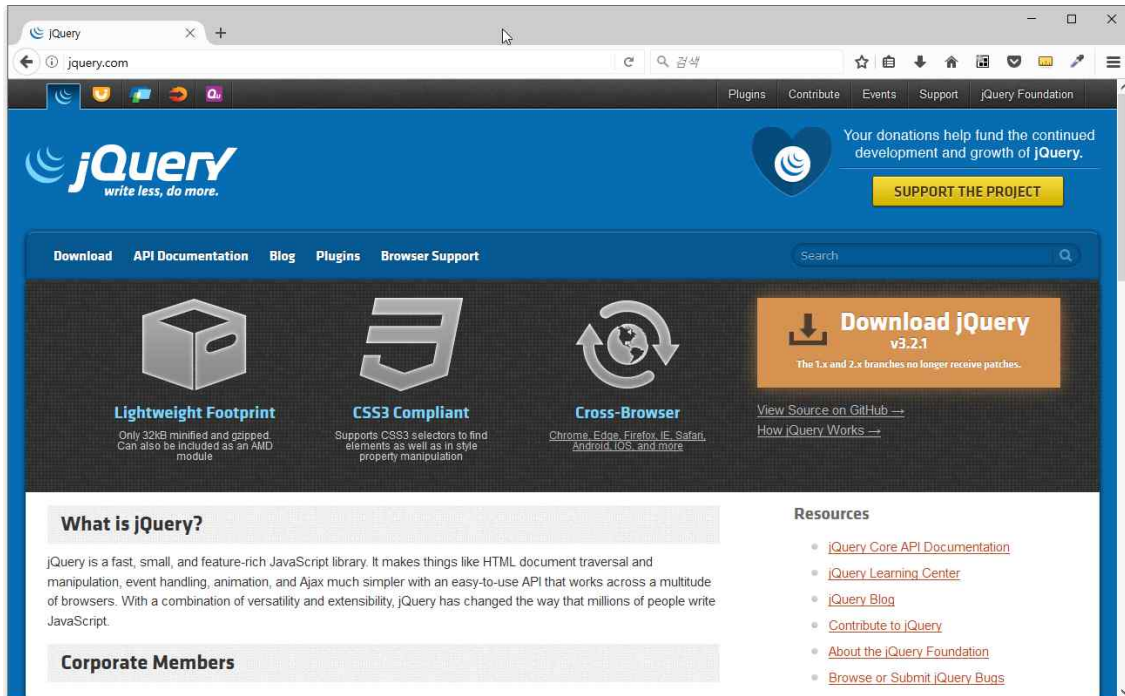


그림) code.org에서 반복문 공부




■ jquery



1) <http://jquery.com>

2) jQuery의 장점

jQuery makes it easy to:

	find	요소의 선택
	change	콘텐츠 / CSS의 변경
	listen	유저 이벤트 감지
	animate	요소의 움직임
	talk	쉬운 ajax를 이용한 실시간 데이터 처리

3) 자바스크립트 객체와 jQuery객체의 관계

- 자바스크립트가 빗자루 / 쓰레받기로 청소하는 거라면,



- jquery는 청소기로 청소하는 것





- 훨씬 쉽고 편리함 / 다양한 기능이 많음

■ h1요소를 찾아 글자 바꾸기

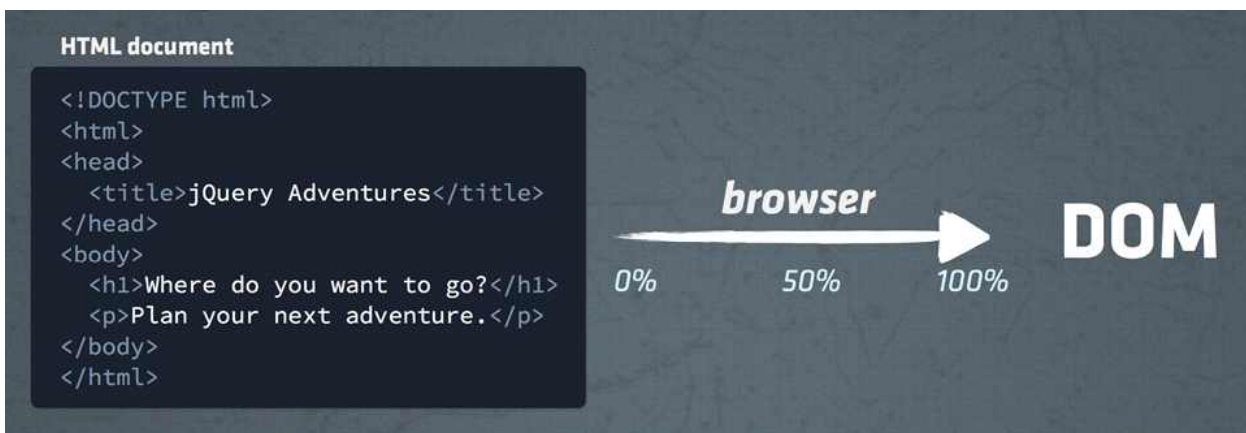
HTML document

```
<!DOCTYPE html>
<html>
<head>
  <title>jQuery Adventures</title>
</head>
<body>
  <h1>Where do you want to go?</h1>
  <p>Plan your next adventure.</p>
</body>
</html>
```

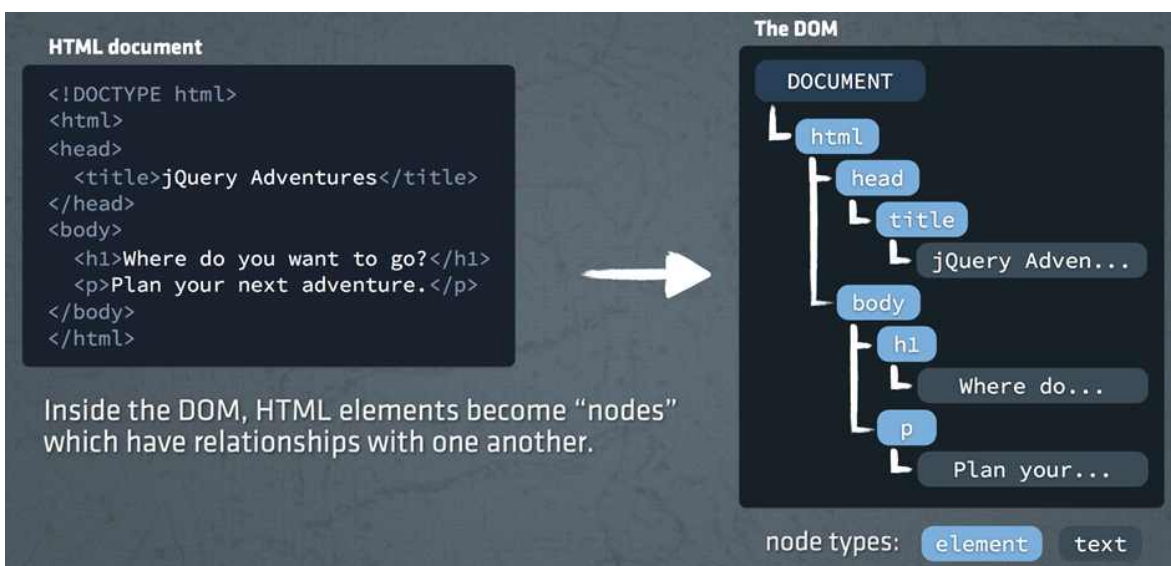
 **find it**
How can we search through our html?

 We need to understand how our browser organizes the HTML it receives.

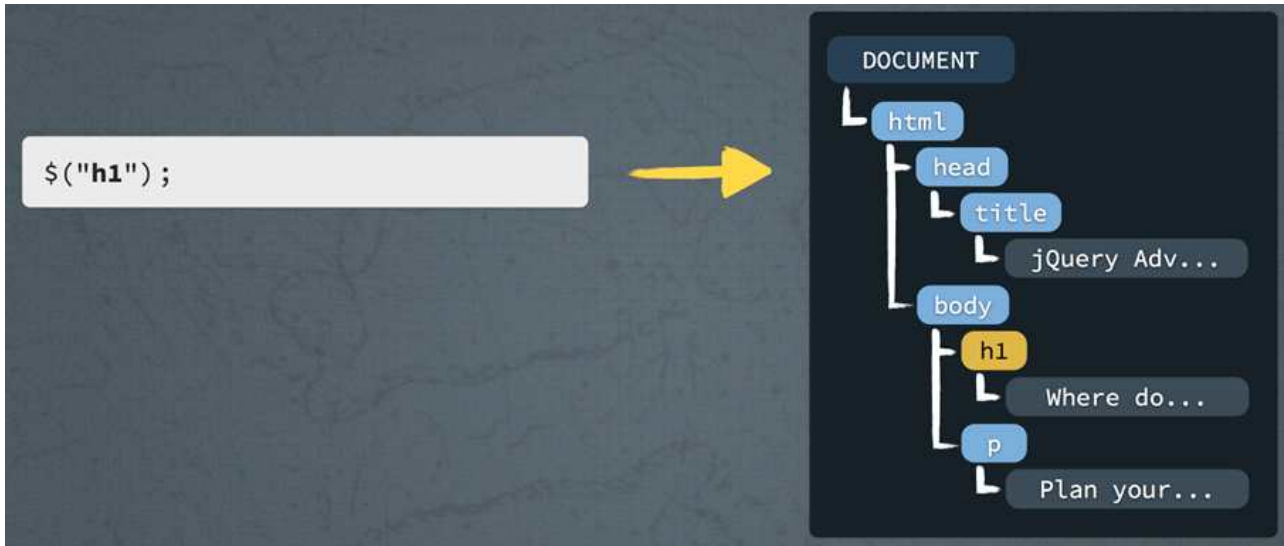
■ DOM은 로딩되는데 시간이 걸림



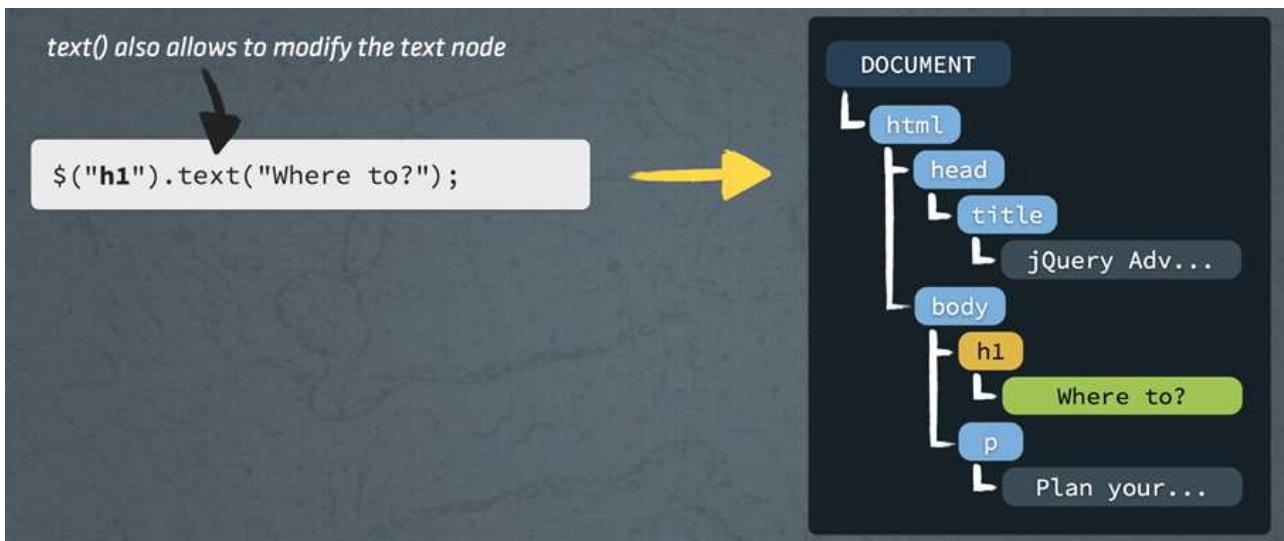
■ DOM 트리 생성 결과



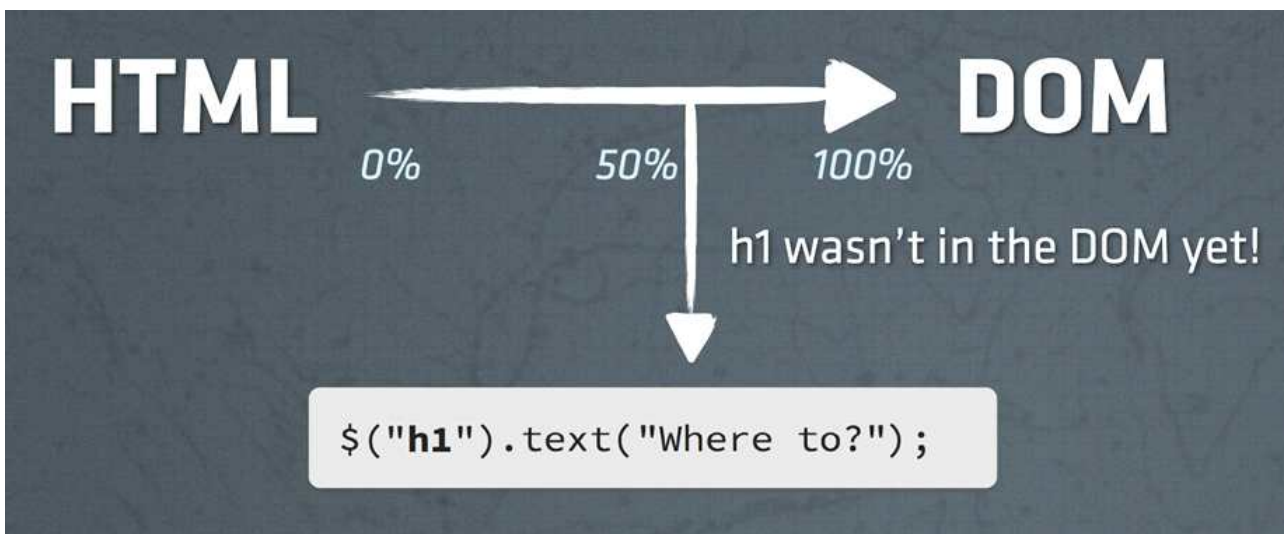
■ CSS의 선택자로 선택



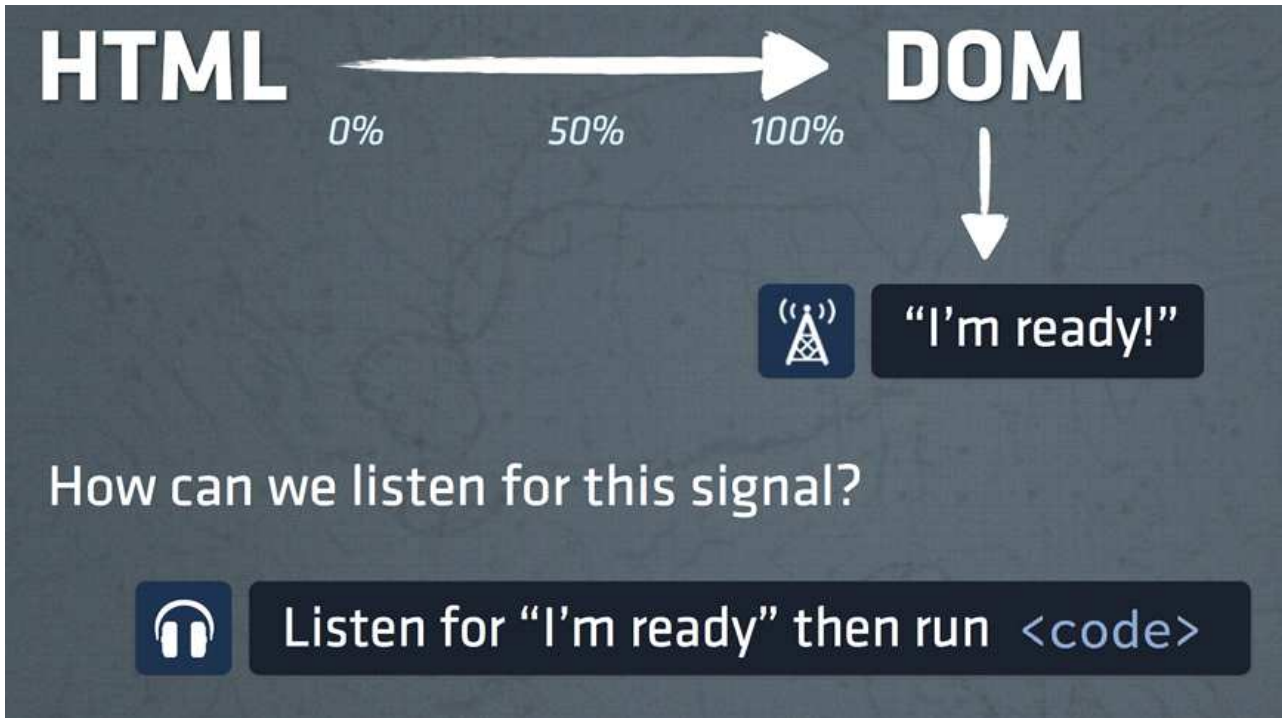
■ 선택된 h1요소에 글자 변경



■ 만약 DOM이 로딩되기 전이라면?



■ 브라우저의 ready 이벤트(DOM 로딩 완료!)



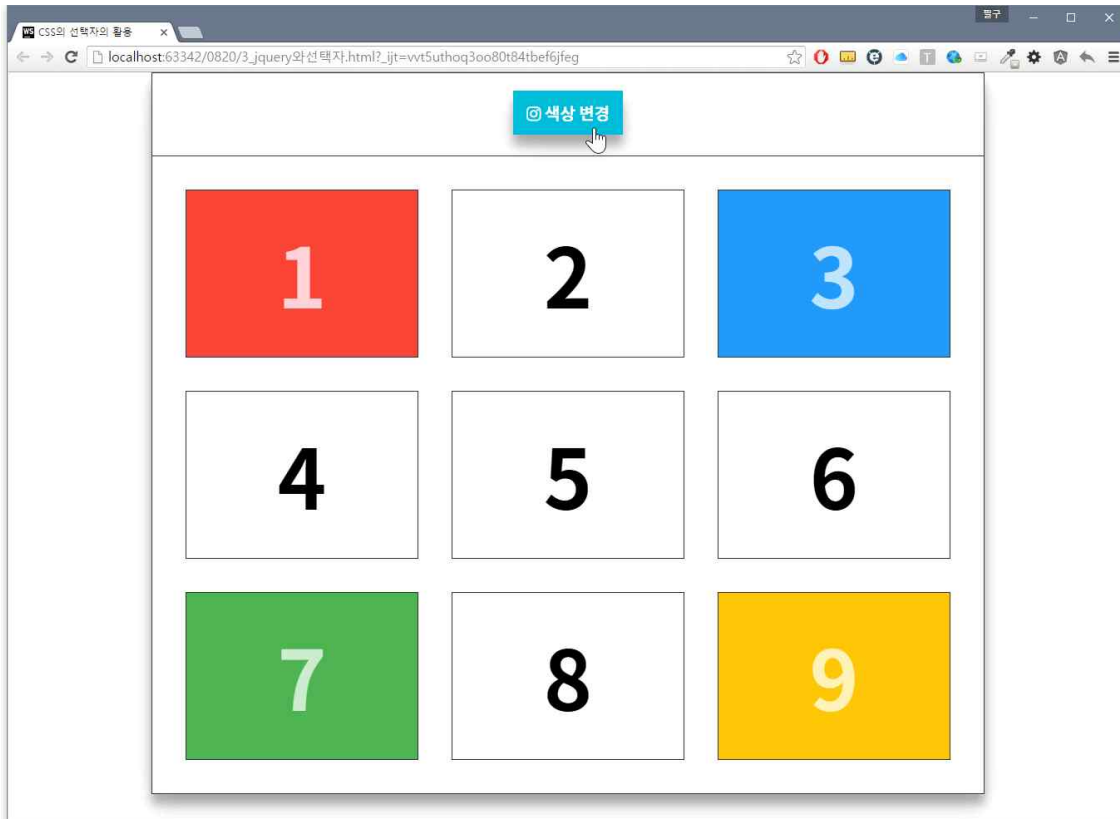
■ 완성된 코드

```
jQuery(document).ready(function(){
    $("h1").text("Where to?");
});
```

■ jQuery(document).ready() 함수를 손쉽게 사용

```
$(function(){
    $("h1").text("Where to?");
});
```

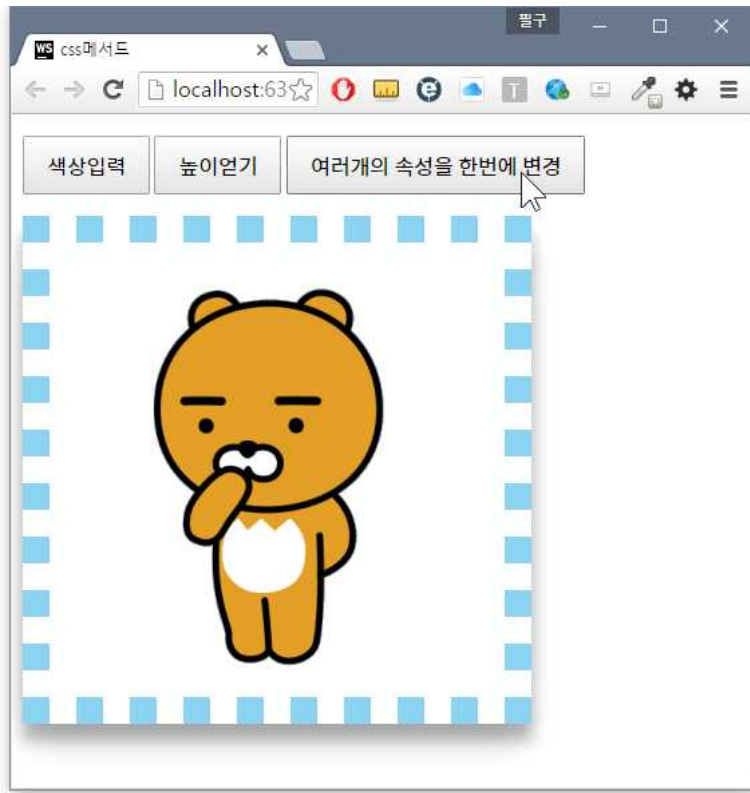
■ 선택자 예제



```
$("#button").click(function() {  
    $("li:eq(0)").css({color:"#FFCDD2",background:"#F44336"});  
    $("li:eq(2)").css({color:"#BBDEFB",background:"#2196F3"});  
    $("li:eq(6)").css({color:"#C8E6C9",background:"#4CAF50"});  
    $("li:eq(8)").css({color:"#FFECB3",background:"#FFC107"});  
});
```

■ CSS 메서드

- 1) jquery는 setter / getter가 하나의 메서드로
- 2) css("속성명") : getter
- 3) css("속성명", "속성값") : setter
- 4) css(속성객체) : setter



```

$("button:first").click(function(){
    $("#box").css("background","darkturquoise");
});

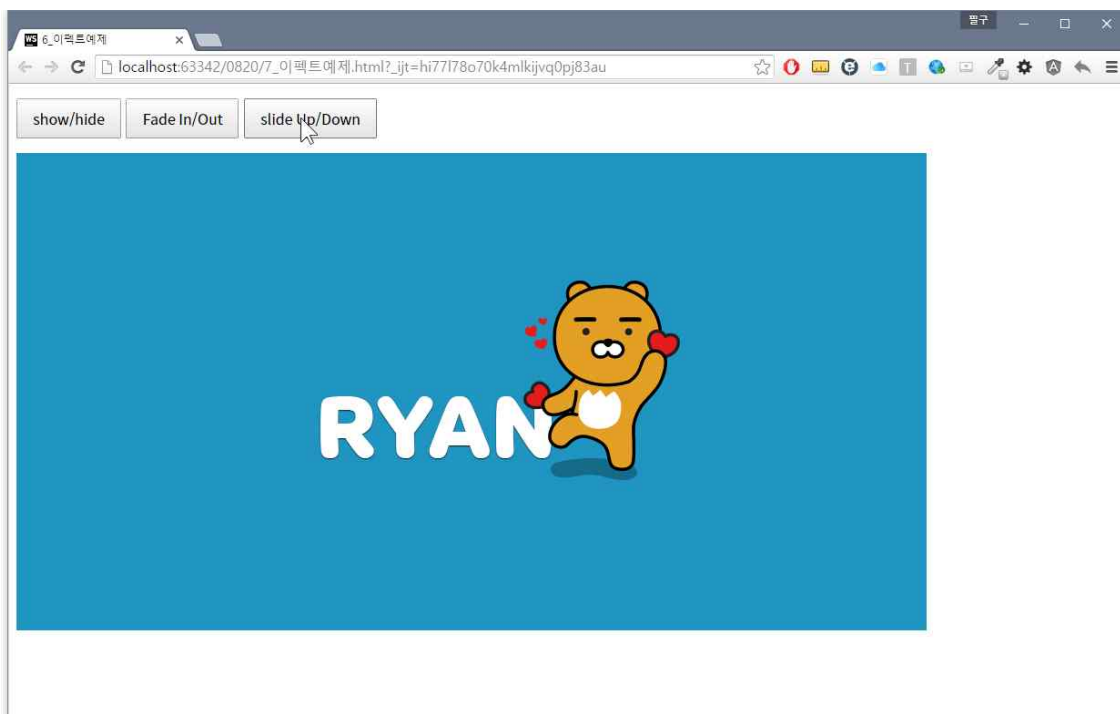
$("button:eq(1)").click(function(){
    var height = $("#box").css("height");
    alert(height);
});

$("button:last").click(function(){
    var obj = {"width":340,"height":340,
        "background":"url(img/k7.png)",
        "border":"20px dotted skyblue",
        "box-shadow":"0 12px 15px 0 rgba(0, 0, 0, 0.24), 0 17px 50px 0 rgba(0, 0, 0, 0.19)"};
    $("#box").css(obj);
});

```

■ jQuery의 이펙트

- 1) 쉽고 편리하게 요소를 숨기거나 보여줄 수 있음
- 2) 가벼운 애니메이션을 손쉽게 사용
- 4) 편리하게 show() / hide()
- 5) 이펙트 : show(시간) / hide(시간)
- 6) slideUp() / slideDown()
- 7) fadeIn() / fadeOut()
- 8) toggle 가능



```
$("#button:eq(0)").click(function(){
    //alert("test");
    //show() / hide()
    //toggle() : show/hide반복
    $("#box").toggle(200);
}); //1번째버튼 click end

$("#button:eq(1)").click(function(){
    //alert("test");
    //fadeIn(밀리세컨드) / fadeOut(밀리세컨드)
    //fadeToggle(시간)
    $("#box").fadeToggle(400);
```

```
});//2번째버튼 click end
```

```
$("#button:eq(2)").click(function(){
```

```
    //alert("test");
```

```
    //slideUp() / slideDown()
```

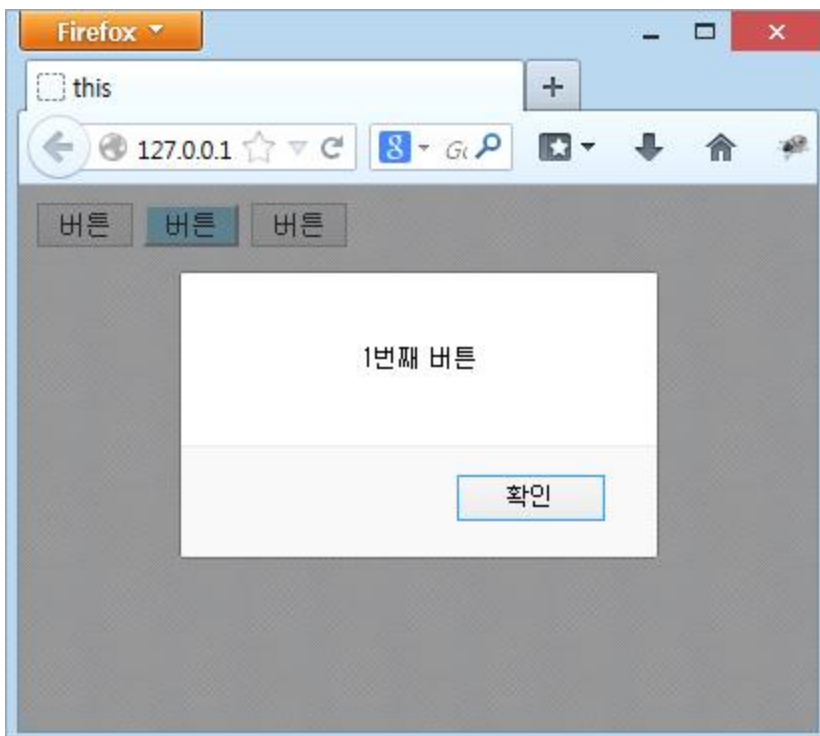
```
    //slideToggle()
```

```
    $("#box").slideToggle(200);
```

```
});//3번째버튼 click end
```

■ this / index()

- 1) 함수에서 this는 그 현재 요소를 나타냄
- 2) this는 자바스크립트 객체
- 3) index()는 부모기준으로 자신의 index를 정수로 알려줌



```

$("button").click(function() {
    $(this).css("background","skyblue");
    var index = $(this).index();
    alert(index+"번째 버튼");
})//button click end

```

■ 글자관련 메서드

- 1) 요소의 글자관련 메서드는 text()
- 2) 요소안에 마크업을 하고 싶으면 html()
- 3) getter / setter를 하나의 메서드로
- 4) text(데이터) : setter
- 5) text() : 인자 없으면 getter



```

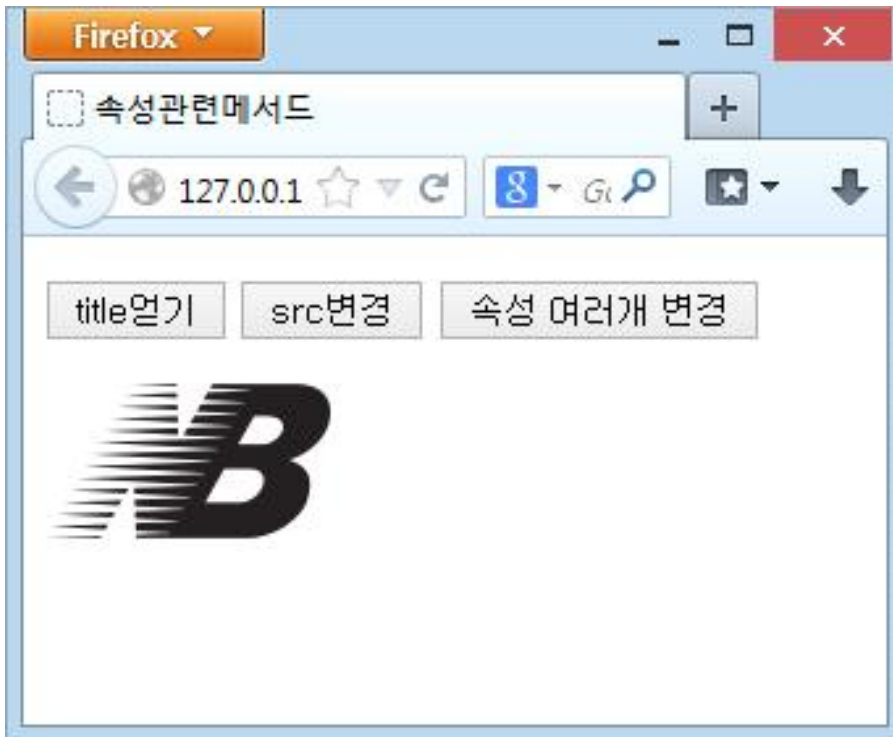
$("button:eq(0)").click(function() {
    $("#txtBox").text("클릭했군요!");
});

$("button:last").click(function() {
    var msg = $("#txtBox").text();
    alert(msg);
});

```

■ 속성관련 메서드 : attr()

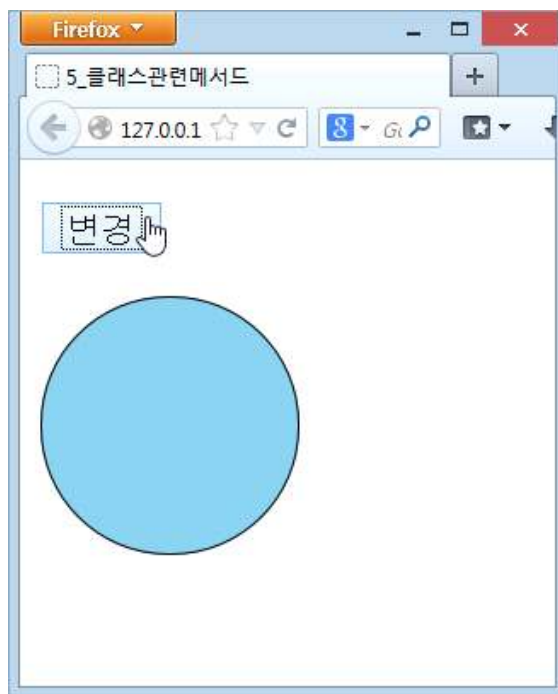
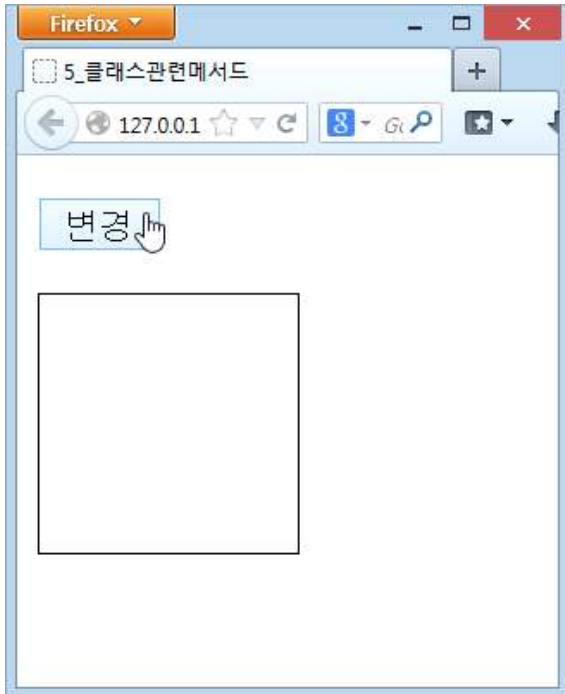
- 1) 속성은 영어로 attribute
- 2) 줄여서 메서드 명이 attr()
- 3) setter / getter 가능
- 4) 속성은 명/값으로 구성되기 때문에 css()메서드와 흡사함



```
$('#getBtn').click(function() {  
    var title = $("img").attr("title");  
    alert(title);  
});  
  
$("#setBtn").click(function() {  
    $("img").attr("src", "img/puma.png");  
});  
  
$("#setAllBtn").click(function() {  
    $("img").attr({  
        width: 400,  
        title: "하하하",  
        src: "img/nike.png"  
    });  
});
```

■ 클래스 관련 메서드

- 1) `addClass()` : 클래스 추가
- 2) `removeClass()` : 클래스 제거
- 3) `hasClass()` : 클래스 가지고 있는 지 확인
- 4) `toggleClass()` : `toggle()`과 흡사함



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
  <title>5_클래스관련메서드</title>
  <style type="text/css">
    button {
      cursor: pointer;
    }
    .box {
      width:100px;
      height:100px;
      border:1px solid #000;
    }

    .round {
      background:skyblue;
    }
  </style>
</head>
<body>
  <button>변경</button>
  <div class="box"></div>
</body>
</html>
```

```

        border-radius:50px;
    }
</style>
<script type="text/javascript" src="js/jquery.js"></script>
<script type="text/javascript">
    $(function() {

        //$(".box").toggleClass("round");

        var box = $(".box");

        if(box.hasClass("round")) {
            //둥근 상태
            //alert("둥글어요!")
            box.removeClass("round");

        }else {
            //둥글지 않은 상태
            //round클래스추가
            box.addClass("round");

        }

    });
</script>
</head>
<body>
    <p>
        <button>변경</button>
    </p>
    <div class="box"></div>
</body>
</html>

```

■ 이벤트 바인딩 / 언바인딩

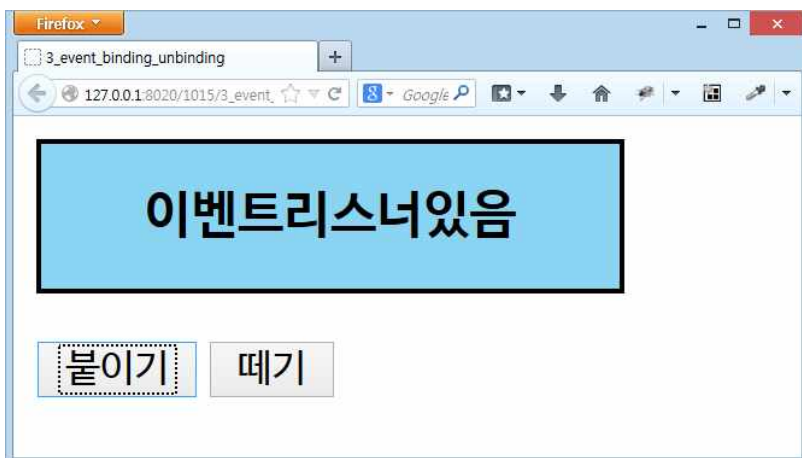
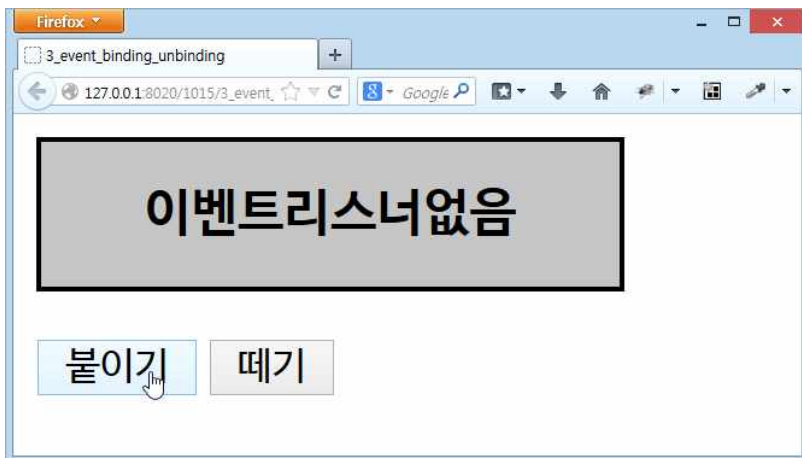
- 1) click은 이벤트리스너를 바인딩하는 메서드
- 2) 원래 bind() 메서드를 이용했음

```
jquery객체.bind(이벤트명, 콜백함수)
```

- 3) 편리성 때문에 click() / keydown() 등의 메서드가 추가됨
- 4) 이벤트리스너는 중복하여 계속 binding됨
- 5) 바인딩된 이벤트리스너를 제거하기 위해서는 unbind()

```
jquery객체.unbind(이벤트명)
```

```
jquery객체.unbind(이벤트명, 콜백함수)
```



■ input관련 메서드

1) input요소는 text() 가 아님

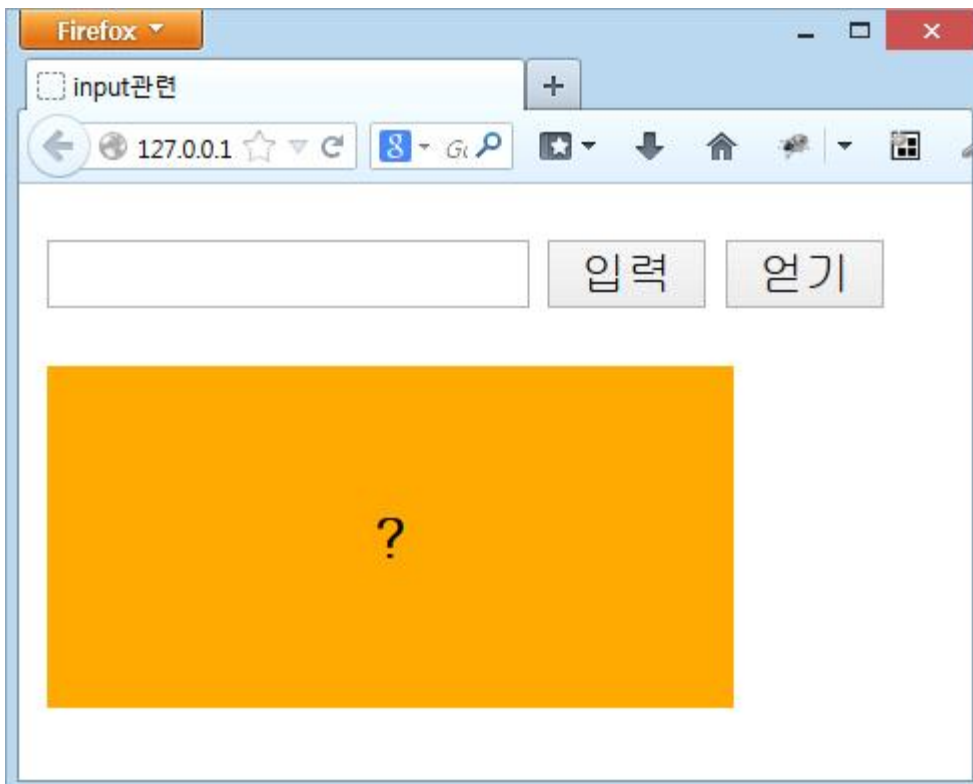
```
<input type="text" id="txt" value="id"/>  
<!-- input의 글자는 value속성 -->  
<input type="text"></input>  
<!-- input은 자식요소가 없음 -->
```

입력

얻기

2) val() 메서드를 사용

3) getter / setter 하나의 메서드로



```

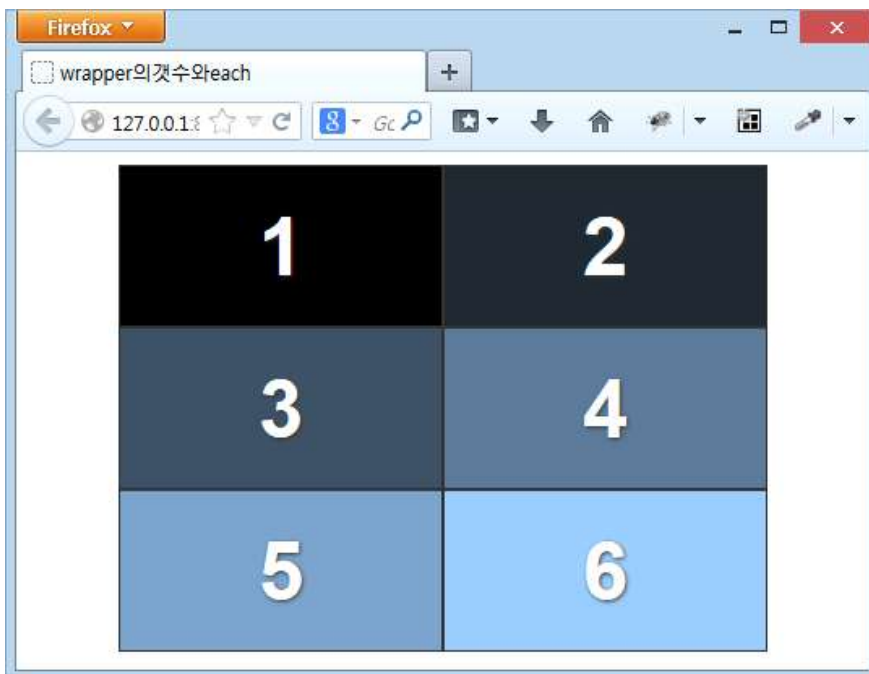
$("button:eq(0)").click(function(){
    $("#txt").val("하하하하");
});

$("button:eq(1)").click(function() {
    var msg = $("#txt").val();
    $("#box").text(msg);
});

```

■ Wrapper 반복 each() 메서드

- 1) jquery객체는 요소가 여러개일수 있음
- 2) 그 요소들마다 각각 접근하여 할 일 수행가능
- 3) 내부적으로 for문이 작동됨
- 4) for문보다 쉽게 사용 가능

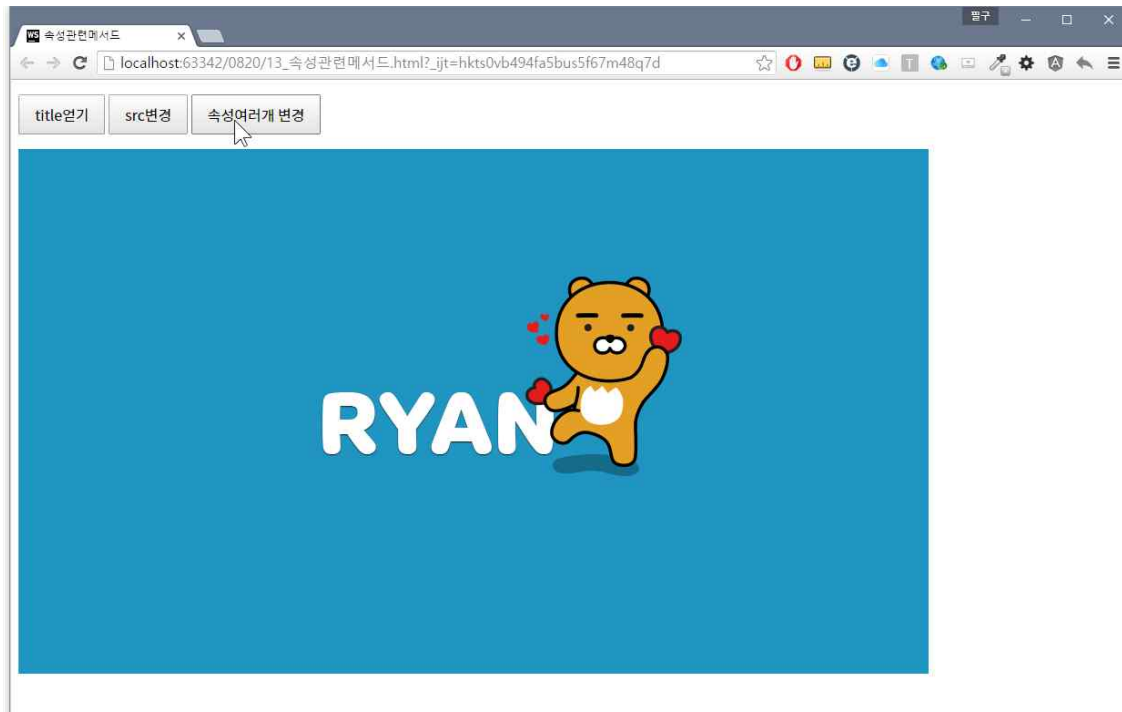


```

var i = 0;
$(".box").each(function() {
    var color = "rgb(" + i*30 + "," + i*40 + "," + i*50 + ")";
    $(this).css("background",color);
    ++i;
});

```

■ 속성관련 예제



```
<!DOCTYPE html>
<html lang="ko">
  <head>
    <meta charset="utf-8"/>
    <title>속성관련메서드</title>
    <link href="https://maxcdn.bootstrapcdn.com/font-awesome/4.6.3/css/font-awesome.min.css"
      rel="stylesheet" integrity="sha384-T8Gy5hrqNKT+hzMcLp0118YTQ06cYprQmhrYwIiQ/3axmI1hQomh7Ud2hP0y8SP1"
      crossorigin="anonymous">
    <link href="http://fonts.googleapis.com/earlyaccess/notosanskr.css" />
    <style>
      body {
        font-family: 'Noto Sans KR', sans-serif;
      }
      button {
        font:16px 'Noto Sans KR', sans-serif;
        padding:8px 16px;
      }
    </style>
```

```

</head>
<body>
    <p>
        <button>title얻기</button>
        <button>src변경</button>
    <button>속성여러개 변경</button>
    </p>
    

    <script type="text/javascript" src="js/jquery.js"></script>

    <script type="text/javascript">

        //text(), html(), val() <-- 컨텐츠

        //css() <== css속성

        //click() <-- 이벤트처리

        //index() <- 인덱스값

        //요소의 속성관련 메서드
        //속성(attribute)
        //attr() <=== html속성

        // getter : attr(속성명)
        // 1개만 세팅 : attr(속성명,속성값)
        // 여러속성을 세팅 : attr(객체)

        $(function() {

            $("button:eq(0)").click(function(){

                //img요소의 title속성을
                //얻어오기
                var title = $("img").attr("title");

                alert("title : "+title);
            });
        });
    </script>

```

```

}); //1번째버튼 클릭 끝

//src변경 버튼을 누르면
//poster3으로
$("button:eq(1)").click(function(){

    //img요소를 선택후
    //src속성(!)을 'img/poster3.jpg'
    //로 변경
    $("img").attr("src","img/b_muzi.png");

}); //2번째버튼 클릭 끝

//3번째버튼을 클릭하면

$("button:eq(2)").click(function(){
    //img요소의 width속성이 600
    //title속성이 드래곤길들이기2
    //alt속성이 드래곤길들이기2
    //src속성이 img/poster2.jpg로
    $("img").attr({
        width:800,
        title:'어피치',
        alt:"어피치",
        src:"img/b_apeach.png"
    });

}); //3번째버튼 클릭 end!

```

```

}); //function() end

```

```

</script>

```

```

</body>

```

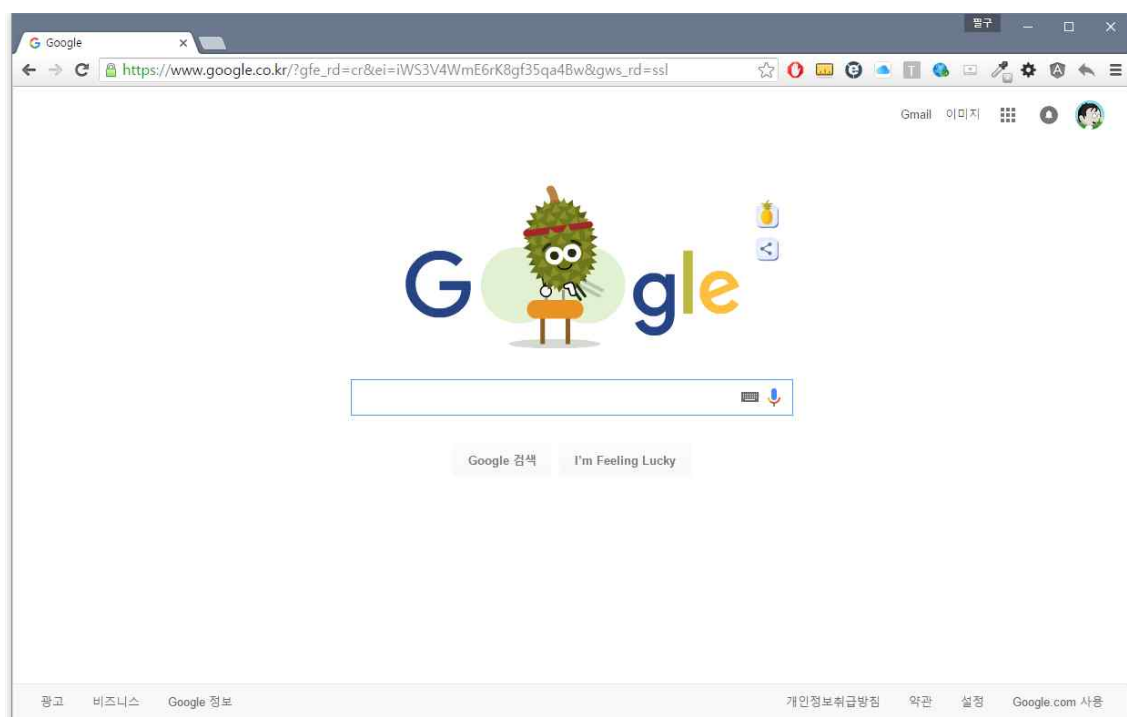
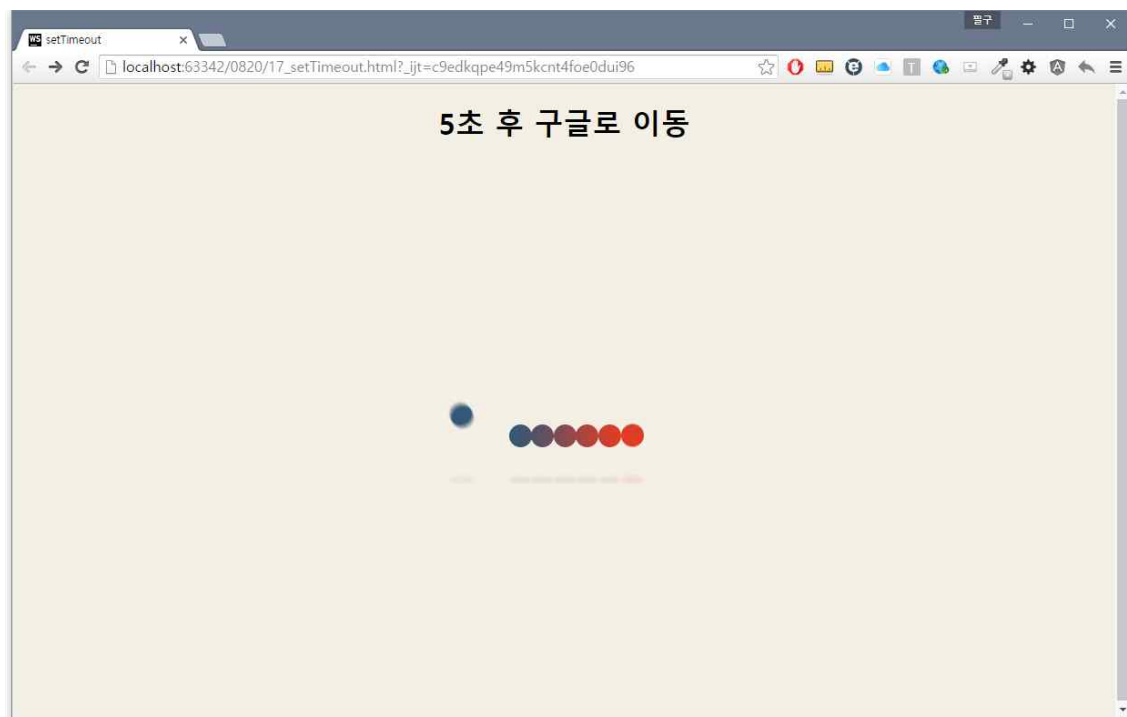
```

</html>

```

■ 타이머메서드

함수명	설명
setInterval(함수,시간)	함수를 시간후에 계속 호출
clearInterval(멈출id)	setInterval() 반복을 중단
setTimeout(함수,시간)	함수를 시간후에 한번만 호출
clearTimeout(멈출id)	setTimeout()을 멈춤



```

<!DOCTYPE html>
<html lang="ko">
  <head>
    <meta charset="utf-8"/>
    <title>setTimeout</title>
    <style>
      body {
        text-align: center;
        background:#EDE9DF;
      }
    </style>
  </head>
  <body>
    <h1>5초 후 구글로 이동</h1>
    
    <script type="text/javascript">

      //타이머 관련 함수(자바스크립트)

      //setTimeout(함수,시간(천분의1초))
      //시간후에 함수 딱 한번만 호출

      setTimeout(function(){
        //alert("5초후!");

        location.href="http://google.com";

      },5000);

      //setInterval(함수,시간(천분의1초))
      //시간마다 함수를 계속 호출

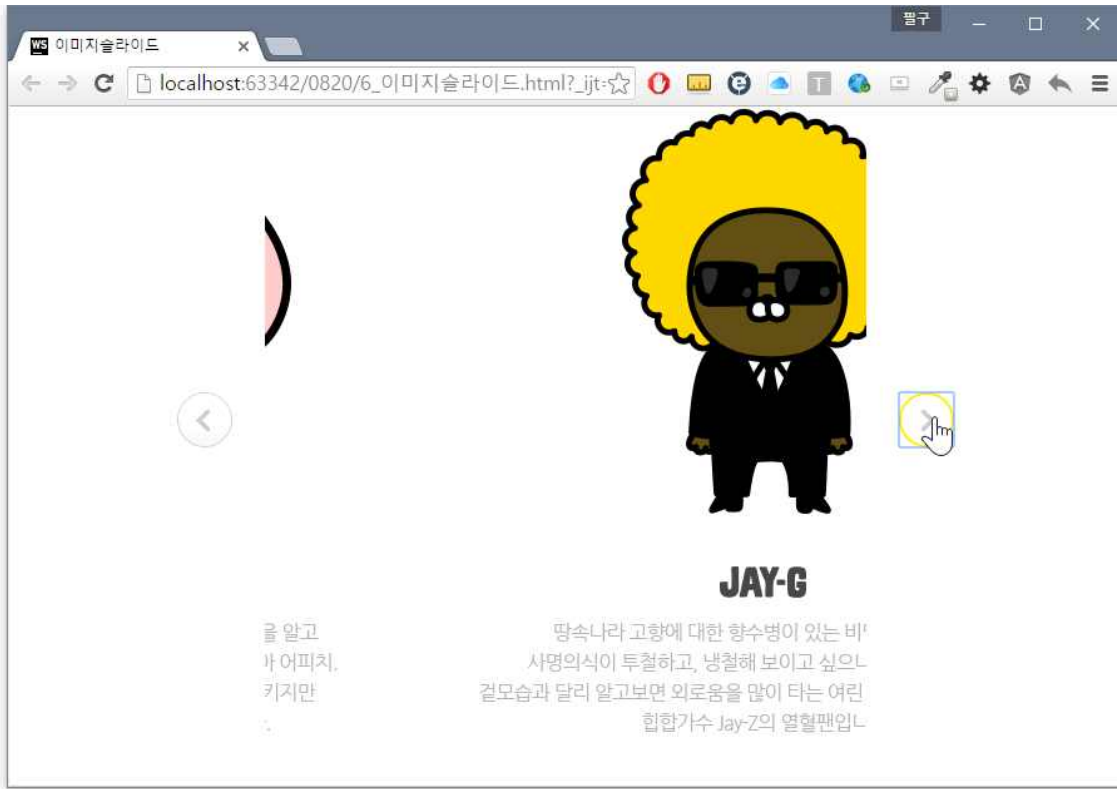
      //메서드는 객체.xxx() <== 메서드

    </script>
  </body>
</html>

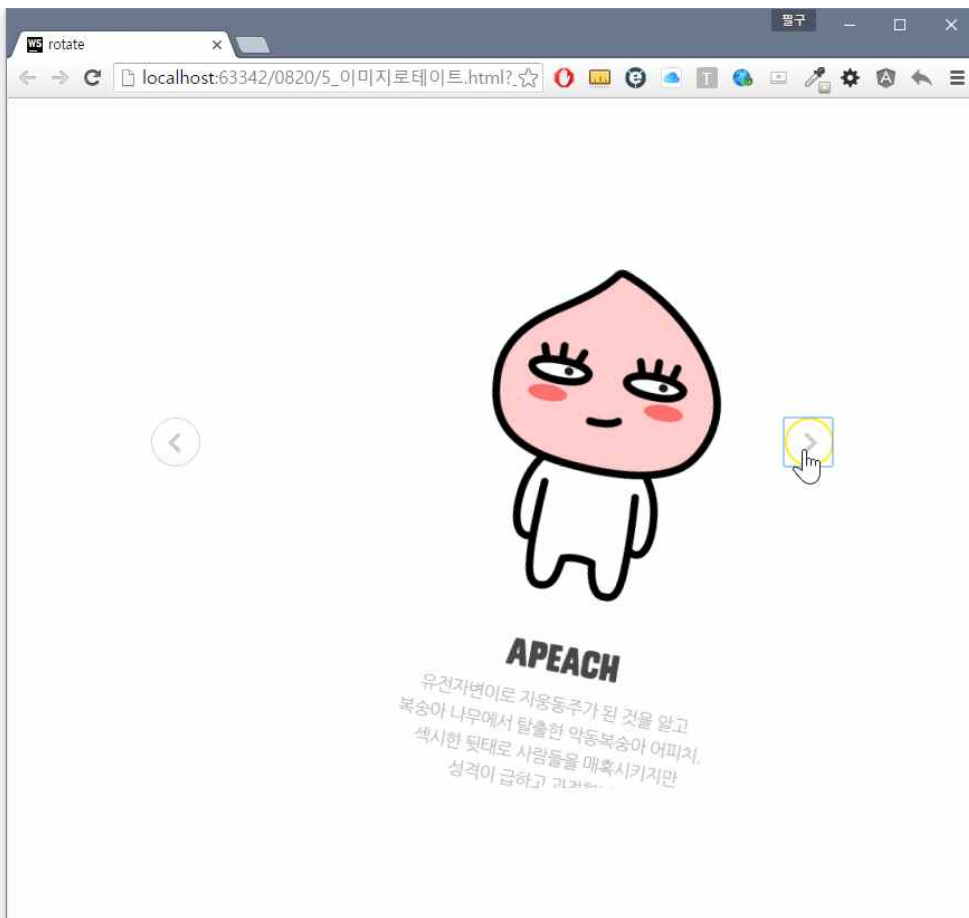
```

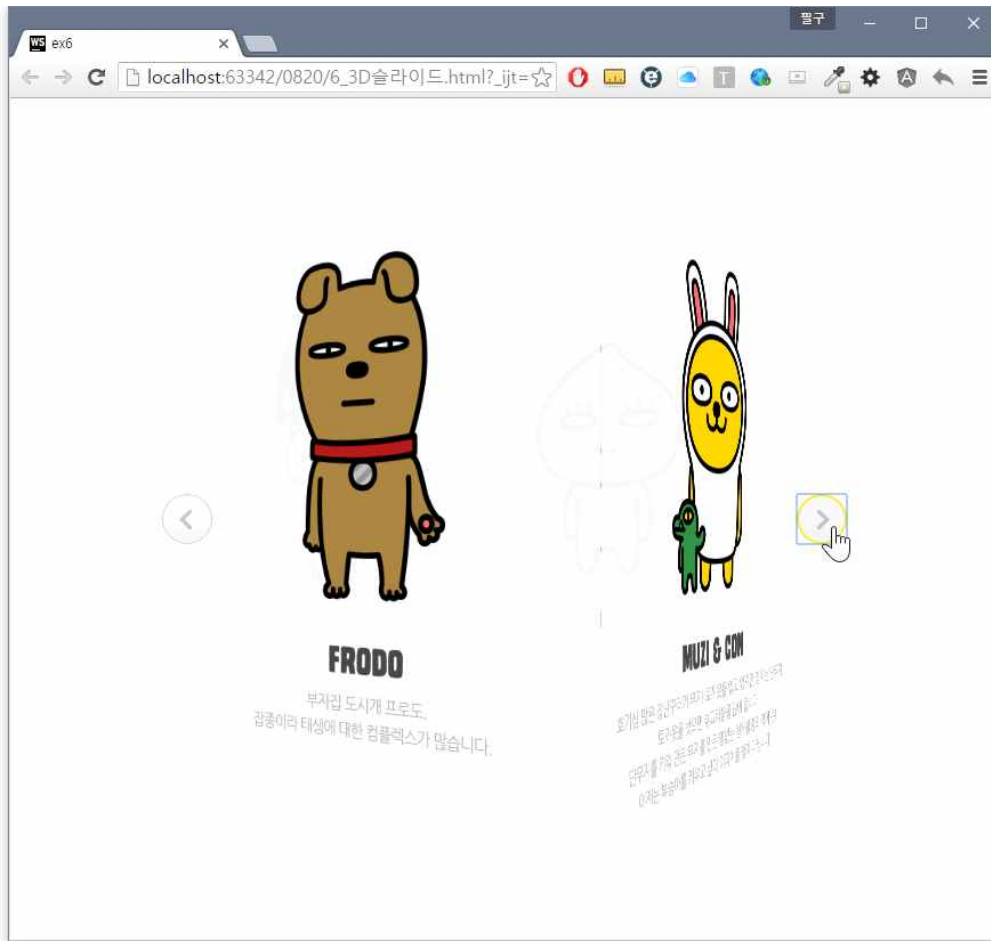
■ 다양한 이미지 슬라이드

1) 기본 이미지 슬라이드



2) 회전하는 슬라이드

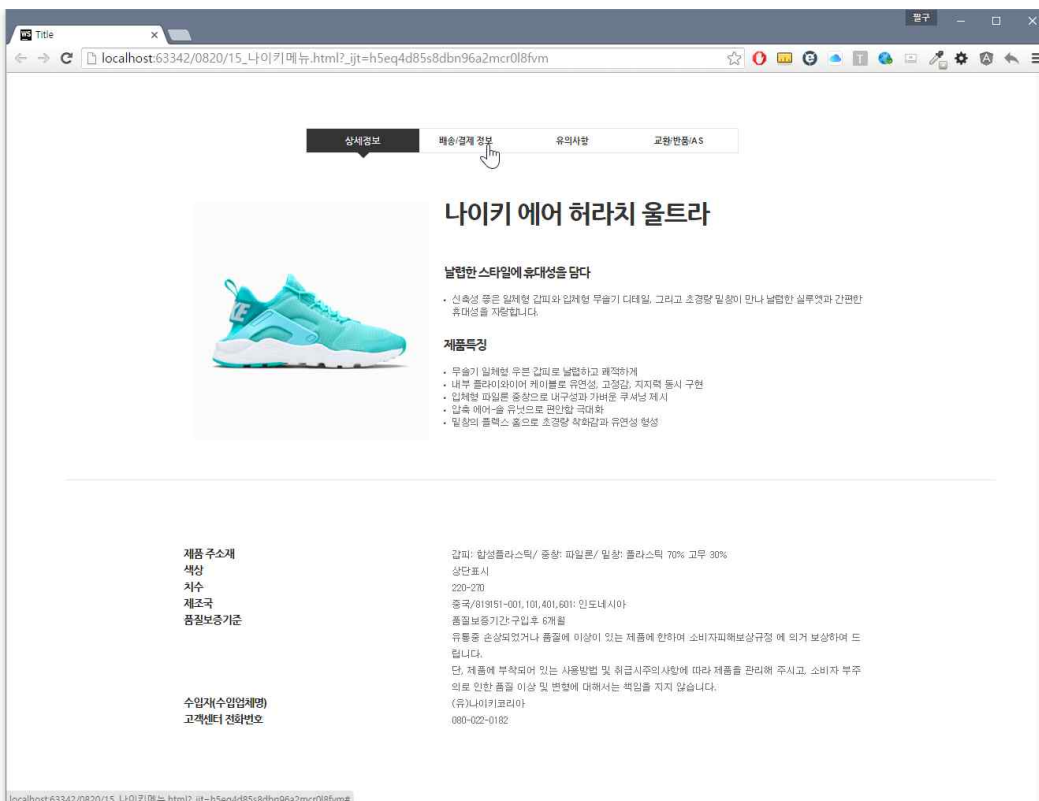




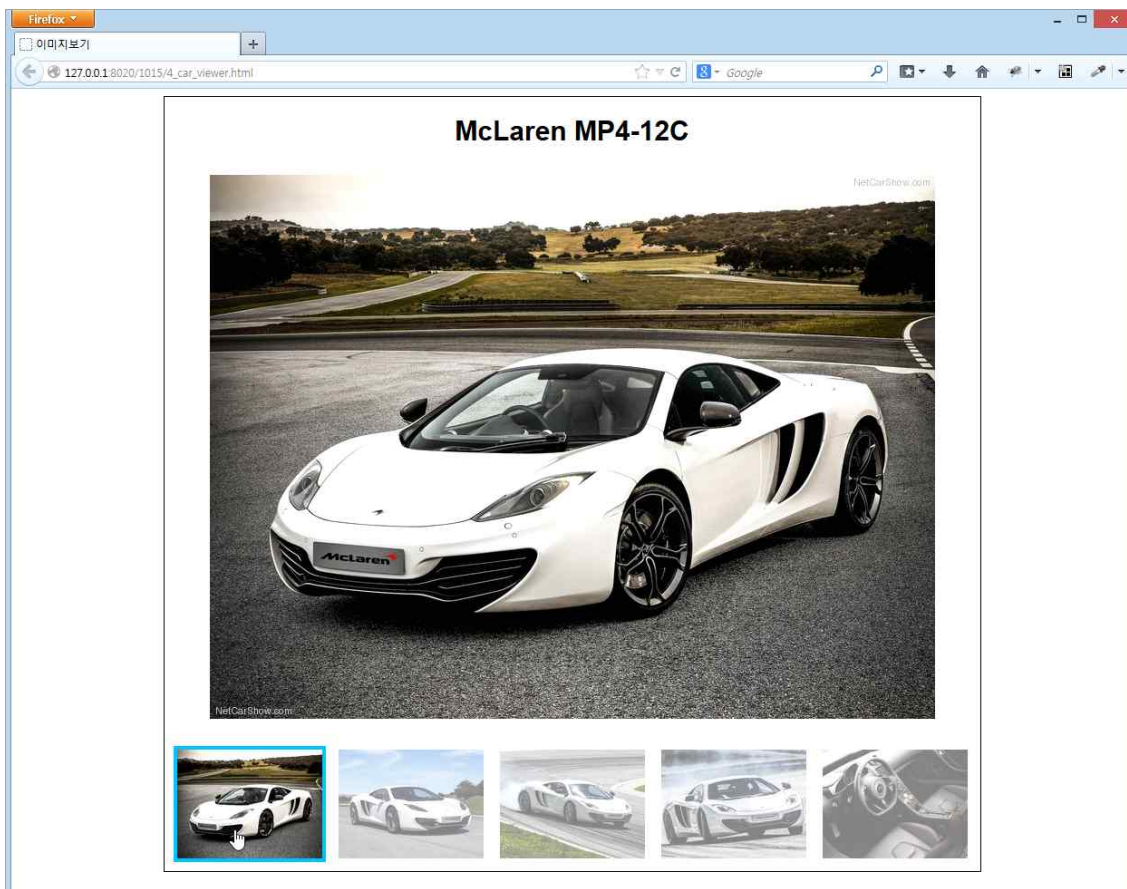
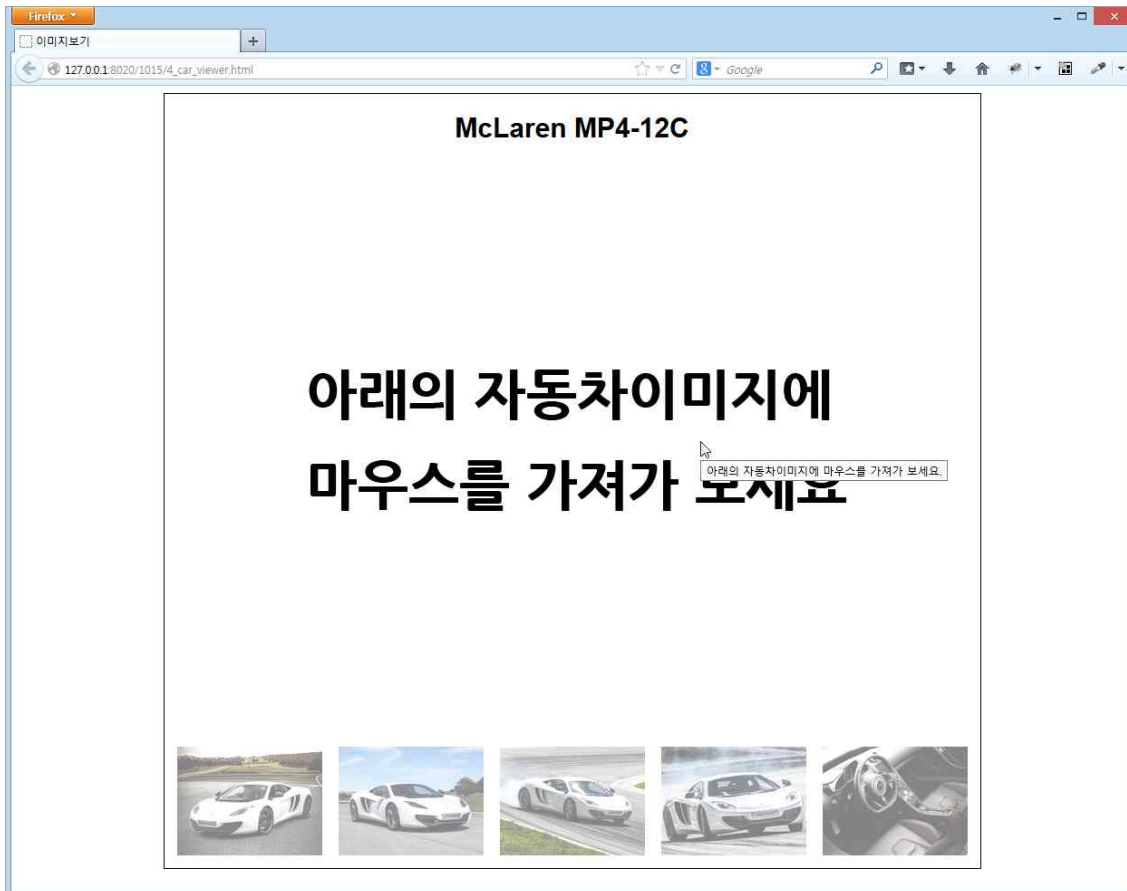
3) 무한 슬라이드등 다양한 방법이 있음

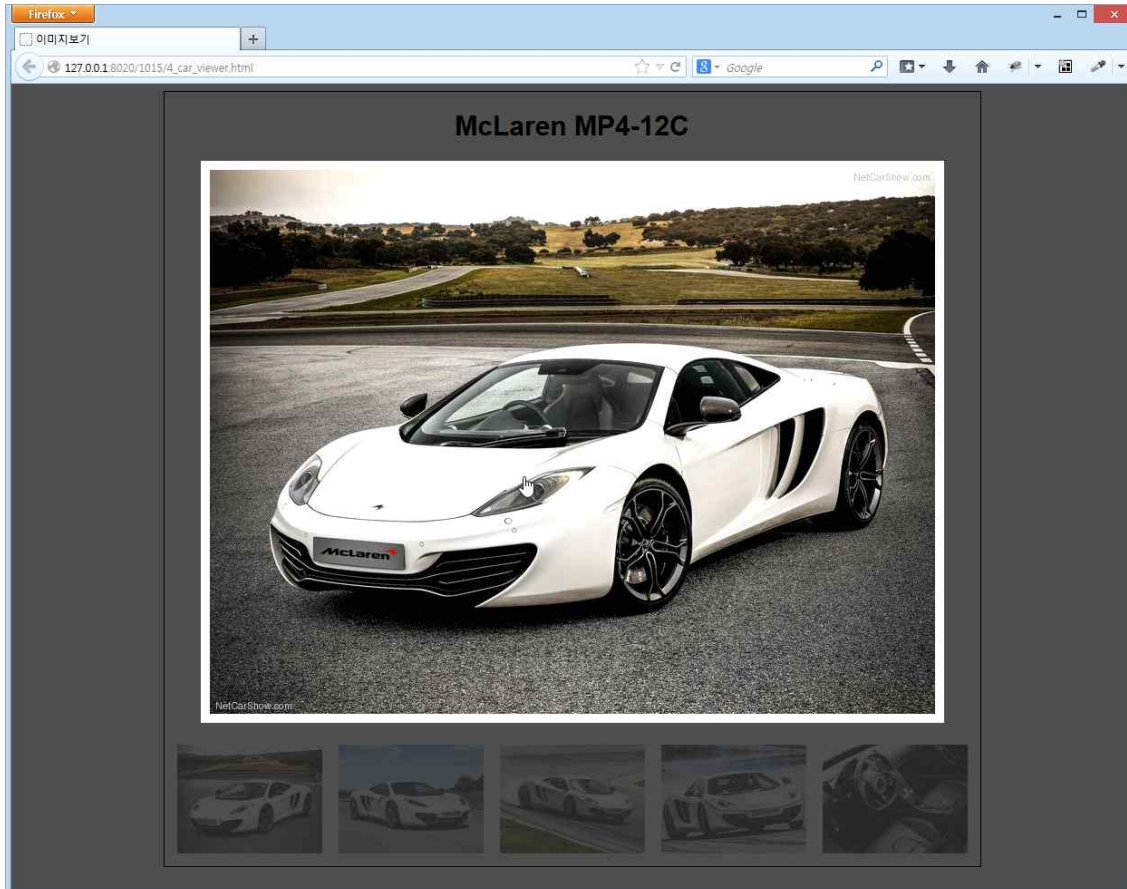
■ 템메뉴 만들기





■ 속성관련 예제





- 1) 아래쪽 작은 이미지에 마우스오버시 위의 이미지 변경
- 2) 위 이미지 클릭시 반투명한 뒷배경과 함께 이미지가 부각됨

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ko" lang="ko">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<title>이미지보기</title>
<style type="text/css">
    #wrap {
        width:900px;
        margin:auto;
        border:1px solid #000;
    }

    #wrap>h1 {
        text-align:center;
        font:bold 30px Arial, Helvetica, sans-serif;
    }
</style>
</head>
<body>
    <div id="wrap">
        <h1>McLaren MP4-12C</h1>
        <img alt="Main image of McLaren MP4-12C" data-bbox="172 121 638 371" />
        <div id="thumbnails">
            <img alt="Thumbnail 1" data-bbox="155 381 248 431" />
            <img alt="Thumbnail 2" data-bbox="258 381 351 431" />
            <img alt="Thumbnail 3" data-bbox="361 381 454 431" />
            <img alt="Thumbnail 4" data-bbox="464 381 557 431" />
            <img alt="Thumbnail 5" data-bbox="567 381 660 431" />
        </div>
    </div>
</body>
</html>
```

```
#imgBox {
    text-align:center;
    margin:20px 0;
}

#imgBox>img {
    position:relative;
    z-index:1;
    border:10px solid #fff;
}

#imgList {
    margin:10px 0;
    padding:0;
    list-style: none;
}

#imgList>li {
    float: left;
    margin-left:10px;
}

#imgList>li>img {
    width:160px;
    height:120px;
    display: block;
    border:4px solid transparent;
    opacity: .5;
    cursor: pointer;

    transition: .5s ease;
}

#imgList>li>img:hover {
    border-color:deepskyblue;
    opacity: 1;
}
```

```

/*clear fix */
#imgList:after {
    content: "";
    display: block;
    clear: both;
}
#bg {
    width: 100%;
    height: 100%;
    position: fixed;
    left: 0;
    top: 0;
    background: #000;
    opacity: .7;
    display: none;
}

</style>
<script type="text/javascript" src="js/jquery.js"></script>
<script type="text/javascript">

    $(function() {

        $("#imgList>li").mouseover(function() {

            var index = $(this).index();

            $("#imgBox>img").attr({
                src: "img/mclaren" + (index + 1) + ".jpg",
                alt: "마우스를 클릭하세요",
                title: "마우스를 클릭하세요"
            })
            .css("cursor", "pointer")
            .unbind("click")
            .click(function() {

                $("#bg").fadeToggle(500);
            })
        })
    })

```

```

});

});

});
</script>
</head>
<body>
<div id="wrap">
    <h1>McLaren MP4-12C</h1>
    <div id="imgBox">
        
    </div>
    <ul id="imgList">
        <li>
            
        </li>
        <li>
            
        </li>
        <li>
            
        </li>
        <li>
            
        </li>
        <li>
            
        </li>
    </ul>
    <div id="bg"></div>
</div>
</body>
</html>

```