



Codingapple

# Event Bubbling

JS

## Project4 : Tab 만들기

Products

Services

Info

한국말로 tab 기능을 설명하자면  
product 버튼을 누르면 거기에 맞는 product 박스가 보여지고, 나머지는 사라짐.

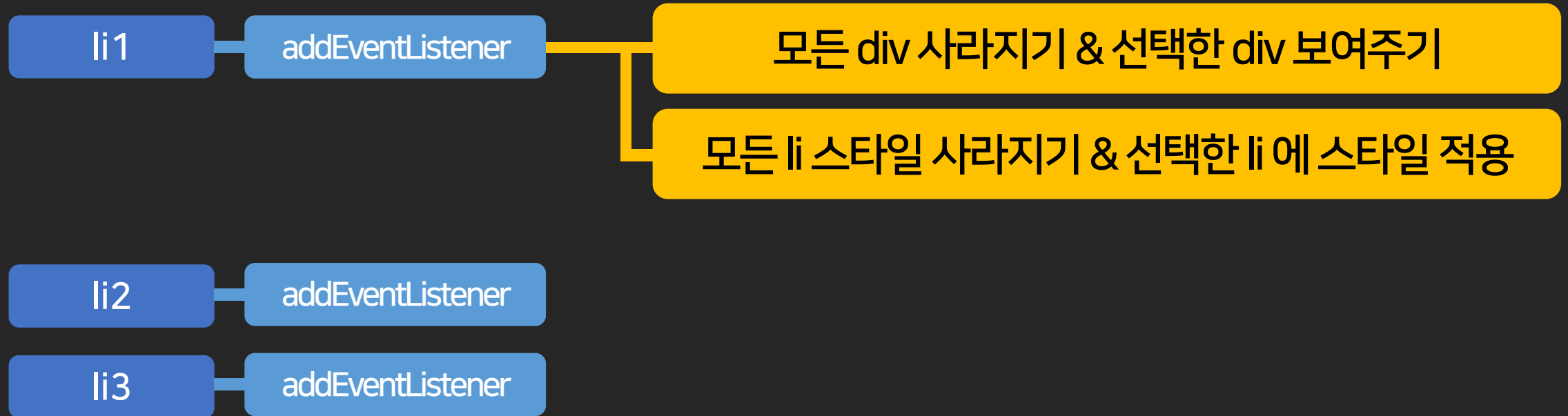
# tab 기능 만들기

준비물 :

1. li 태그로 버튼
2. 밑에 div 태그로 내용
3. 오픈되어있는 li 디자인 하기 (css class를 추가)
2. li 요소를 쉽게 잡기 위해 data로 이름쓰기 (data-tab = "tab1")
3. div 요소를 쉽게 잡기 위해 id로 이름쓰기 (id="tab1")



# tab 기능



오늘의 HTML상식 : HTML에다가 data attribute를 넣으면 좋은 점

```
< li data-tab = "tab1" >
```

```
< li data-tab = "tab2" >
```

```
document.dataset.tab;
```

- HTML 요소 쉽게 찾기 가능
- HTML에 중요한 정보 읽기/쓰기 가능

## Javascript selector 2 (여러 element 선택할 때)

```
document.getElementsByClassName('tab-content') ;
```

```
document.querySelectorAll('div.tab-content') ;
```



select 결과는 **array**와 비슷하게 쓸 수 있는 Nodelist에 담김

## tab 기능 cheet sheet 4-1. 클릭시 모든 div는 사라지기

```
var tabs = document.querySelectorAll('div.tab-content');
```

```
tabs[0].style.display = 'none';
```

```
tabs[1].style.display = 'none';
```

```
tabs[2].style.display = 'none';
```

```
.....
```



반복적인거 보면 반드시 흥분해야함



# Javascript for

같은 코드를 여러 번 반복하고 싶으면 교양있게 for 문을 씁니다.

```
for (초기값; 동작조건; 반복할 때 마다 값의 변동) {
```

```
    //반복할 코드
```

```
}
```

# Javascript for

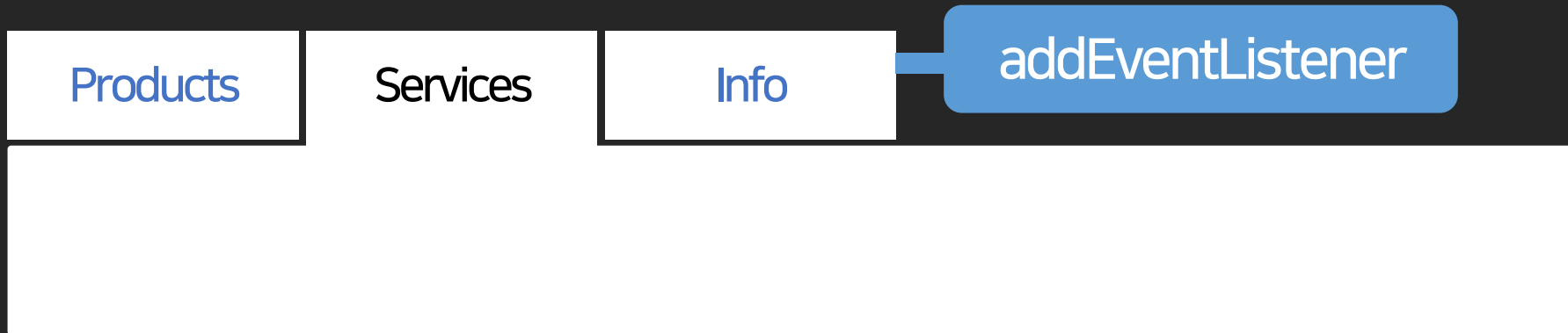
```
for ( var i = 0 ; i < 4 ; i ++ ) {  
    console.log( i );  
}
```

“ 0에 1을 더해가며 3이 될때까지만  
밑에 코드를 계속 반복하셈 ”


콘솔창엔 무엇이 출력될까요?

# tab 기능 만들기

그럼 이제 모든 list 버튼에 이벤트를 달아주면 되겠죠?



# 여러 **element**에 이벤트 달기

1. 그냥 모든 element를 잡고 하드코딩으로 박아넣거나
  2. **for** 문을 사용해 하드코딩을 좀 줄여주던가
  3. 하위요소를 **bubbling**을 이용해 잡거나 (중급자용)
- 

```
버튼1.addEventListener('click', function(){  
    //element1 클릭 시 실행할 코드  
});  
  
버튼2.addEventListener('click', function(){  
    //element2 클릭 시 실행할 코드  
});
```

## 여러 **element**에 이벤트 달기 (하수버전)

button

addEventListener

각 버튼마다 event핸들러를 달아줌

button

addEventListener

각 버튼마다 event핸들러를 달아줌

button

addEventListener

각 버튼마다 event핸들러를 달아줌

= 코드 30줄

## 여러 **element**에 이벤트 달기 (중수버전)

button

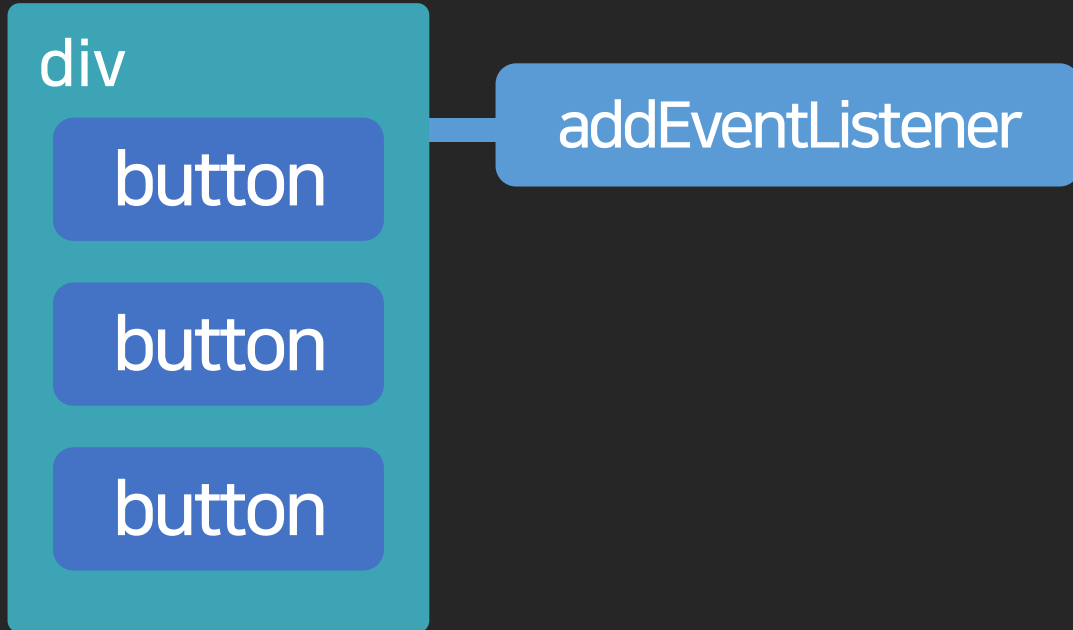
addEventListener

for 반복문으로 3번 돌리기

for 반복문으로 묶음

= 코드 12줄

## 여러 **element**에 이벤트 달기 (고수버전)



이벤트는 위 아래로 퍼진다고 했으니

1. **parent 요소**에
2. event 핸들러를 달아서
3. 이벤트 bubbling을 이용하면 되겠군

= 코드 12줄 + 자원 절약

# closure : 외부함수의 변수를 참조하는 함수

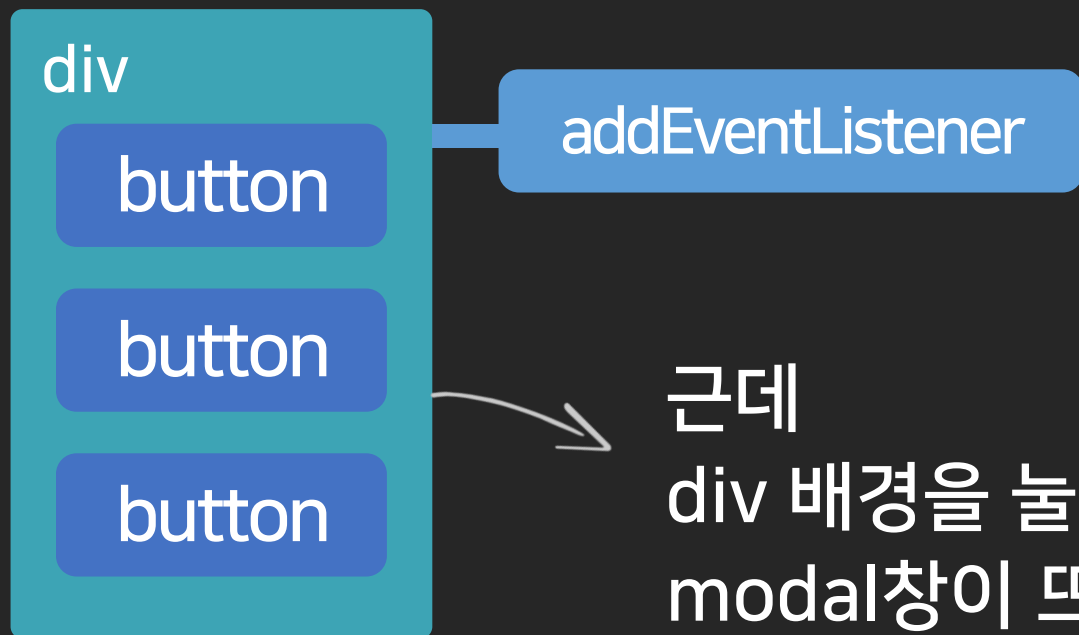
<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Closures>

<http://chanlee.github.io/2013/12/10/understand-javascript-closure/>

closure 함수들은.. 함수 밖에서 정의된 변수를 이용할 때 외부 변수의 값을 저장해서 새로 변수테이블을 만들어서 사용하는게 아니라 '참조'만 합니다. 참조만..

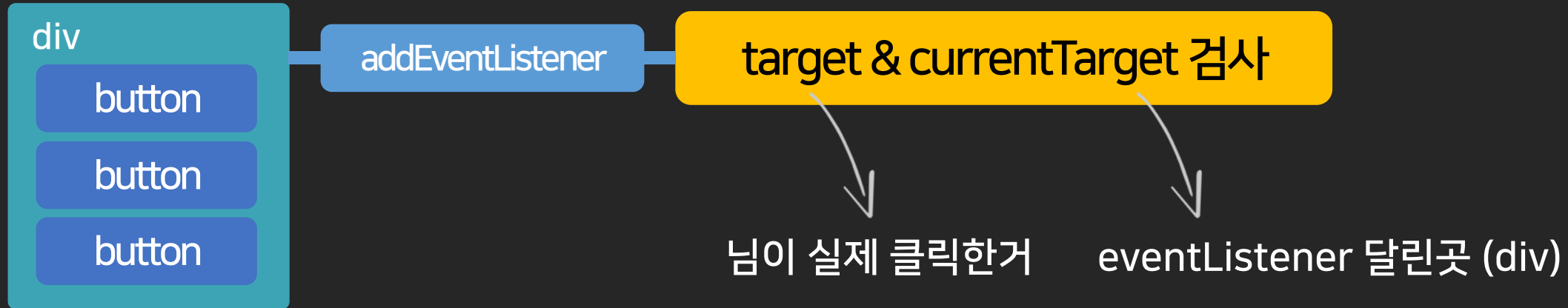


## 여러 **element**에 이벤트 달기 (고수버전)



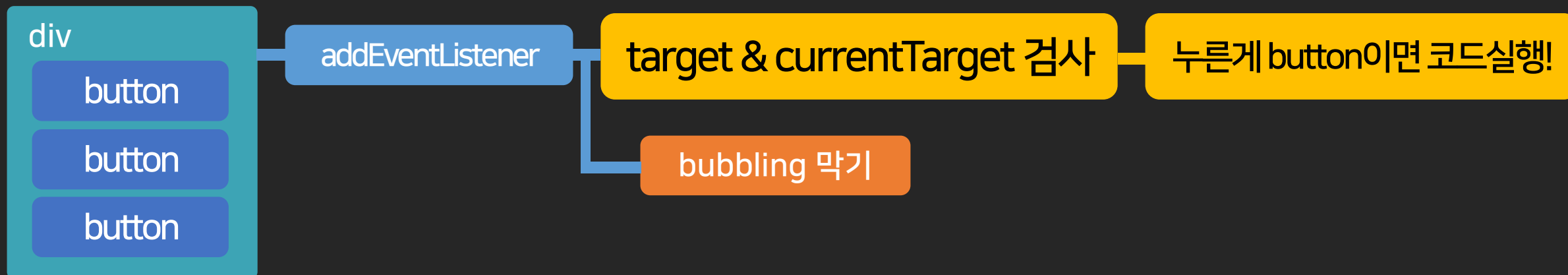
근데  
div 배경을 눌러도  
modal창이 뜨네요? ㅜㅜ

# 여러 **element**에 이벤트 달기

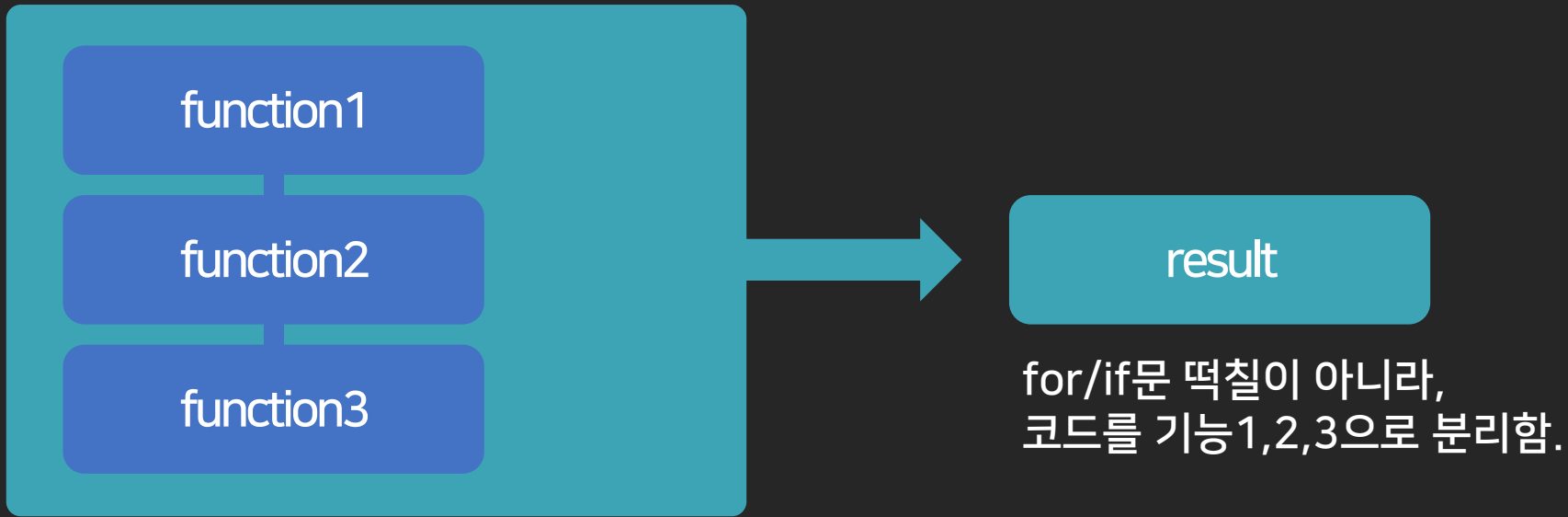


```
div.addEventListener('click', function(e){  
    console.log(e.target);  
    console.log(e.currentTarget);  
});
```

# 여러 **element**에 이벤트 달기



# 자바스크립트가 쥔 잘하는 functional Programming



## <기능들을 함수로 분할하여 개발>

1. 함수들은 input/output이 일정하고, 다른 함수에 의존적이지 않아야 좋다고 함.
2. 읽기도 쉽고 큰 프로젝트 만들기도 쉬움
3. unit testing이 쉬움

# Javascript **selector3** (부모/자식 element 선택할 때)

## 부모 선택하기

element.**parentElement**

element.**closest**('.list')

## 자식 선택하기

element.**children**

element.**querySelector**("")

(querySelector 연달아서 사용가능)

# Javascript **selector4** (형제 element 선택할 때)

## 형제 선택하기

element.**nextElementSibling**

element.**previousElementSibling**

## 처리속도향상 코딩 팁

```
document.querySelector('div').어쩌구~  
document.querySelector('div').저쩌구~
```

\*DOM 탐색은 매우 오래걸리는 작업\*

NOT GOOD

```
var element = document.querySelector('div');  
element.어쩌구~  
element.저쩌구~
```

GOOD

## (ES6) 익명함수 대신쓰는 arrow functions

```
var num = [1,2,3,4];
```

```
var num2 = num.map(function(a){  
    return a+1  
});
```

```
var num2 = num.map( a => a + 1);
```





# jQuery 맛보기

- 자바스크립트를 짧게 작성하기 위해
- HTML요소를 쉽게 잡기 위해
- CSS요소를 쉽게 추가하기 위해
- 애니메이션을 넣기 위해
- Ajax요청을 쉽게 하기 위해
- IE7+ 환경에서도 자바스크립트를 쉽게 쓰기 위해

기본 형태 :

`$( 'element' ).action( )`

# jQuery 설치

서버에서 jquery파일 전송하기 vs CDN 이용하기

# jQuery 권장 템플릿

```
$(document).ready( function( ){  
    //여기에 모든 자바스크립트 코드 작성  
});
```

하지만 꼭 써야하는건 아님

# jQuery 로 HTML 요소를 잡아보자

`$('tag')`

`$('#id')`

`$('.class')`

`$('[attribute]')` .. 혹은 여러 개 섞어서

이외에도 하위요소잡기 :first-child, 속성잡기 [target='\_blank'] 등  
간편하게 HTML element를 잡을 수 있음.

# jQuery 로 요소를 잡았으면 이벤트를 걸자

.click()

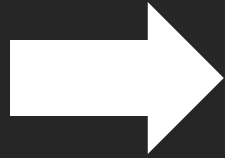
.keydown()

.change()

.scroll()

.resize()

.on()



```
$('#button').click(function(){
```

```
//버튼을 클릭 시 할일
```

```
});
```

# jQuery 로 display 애니메이션을 넣어보자

요소.함수이름()

```
$('#button').click(function(){  
    $('#div').fadeOut();  
});
```

.hide()

.show()

.toggle()

.slideDown()

.slideUp()

.slideToggle()

.fadeIn()

.fadeOut()

.fadeToggle()

# jQuery 로 애니메이션을 넣어보자

```
요소.animate({ 바꿀 css 속성 나열하기 })
```

```
$('#button').click(function(){  
    $('#div').animate({  
        marginLeft: '15px',  
        top: '+=30px' // 기존 값에 + or - 할 수 있음  
    });  
});
```



# jQuery 로 HTML 태그 세부 속성을 잡아보자

```
<a id="link" href="/app.html" > 이동하기 </a>
```

```
$('#link').text() //내부 text
```

```
$('#link').html() //내부 html 전부
```

```
$('#link').val() //form value
```

```
$('#link').attr('href') //attribute (속성) value
```

# jQuery 로 HTML 태그 세부 속성을 변경하려면

```
<a id="link" href="/app.html" > 이동하기 </a>
```

```
$('#link').text( '이동하지말기' ) //내부 text
```

```
$('#link').html('이동하지말기' ) //내부 html 전부
```

```
$('#link').val( 30 ) //form value
```

```
$('#link').attr( 'href', '/app2.html' ) //attribute (속성) value
```

# jQuery 로 HTML element를 추가해보자

`$('#id').append( '<div></div>' )` ★

`$('#id').prepend( '<div></div>' )`

`$('#id').after( '<div></div>' )`

`$('#id').before( '<div></div>' )`

`.before( )`

`.prepend( )`

`.append( )`

`.after( )`

`$('#id')`

# HTML 템플릿

```
var text = '<div>내용</div>'
```

자바스크립트로 동적인 HTML을 생성하기위해선  
string 자료형안에 HTML 내용을 그대로 입력합니다.

```
var text = '<div>' + variable + '</div>'
```

string 중간에 변수 등을 넣고 싶다면,  
+ 연산자를 이용합니다.

## jQuery 로 CSS class를 추가해보자

`$('#id').addClass( 'black' )`

`$('#id').removeClass( 'black' )`

`$('#id').toggleClass( 'black' )` ★

`$('#id').css( 'color', '#000' )`

`<div`

`>`

`class="black"`

# jQuery 로 화면 width/height 조정

`$('#id').width( )` //요소 크기출력

`$('#id').height( )`

`$('#id').width( 500 )` //요소 크기변경

`$( 'document' ).width( )` //body 태그 폭 구하기

`$( 'window' ).width( )` //viewport 폭 구하기

## jQuery 기타 편리한 함수들

`$('#id').parent()`

`$('#id').children()` ★

`$('#id').siblings()` //형제들 선택

`$('#id').filter()` ★ //요소 재선택

`$('#id').not( '.red-color' )` //요소제외

JS

## Project5 : localStorage에 상품 저장하기 (Add to cart)



### Springfield Shirt

This shirt is from Massachusetts, and we didn't even mean to promote any Springfield bolt-action rifles.



Add to Cart

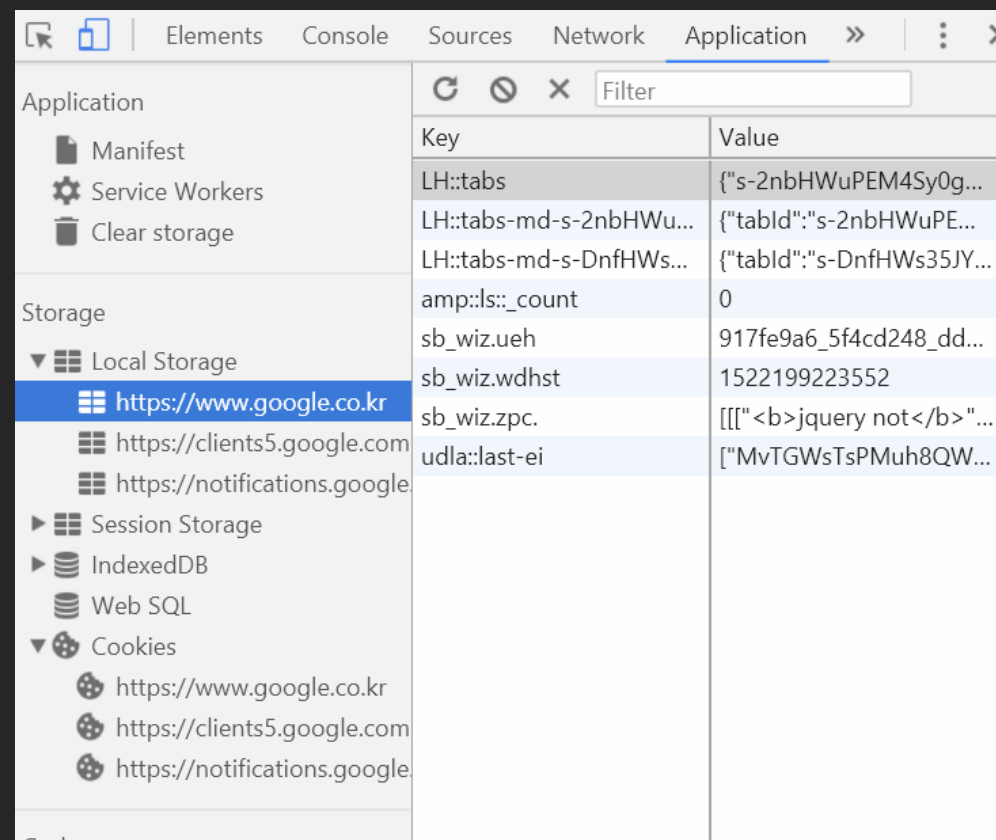
My Cart



정보를 '영구저장'하려면 서버로 전송해야하지만  
덜 중요한 정보들은 **브라우저 저장공간**을 사용하면 편합니다.

1. localStorage
2. sessionStorage
3. Cookie

←  
각각 5MB 정도  
object형태로  
텍스트 데이터를 저장가능  
(IE8+)



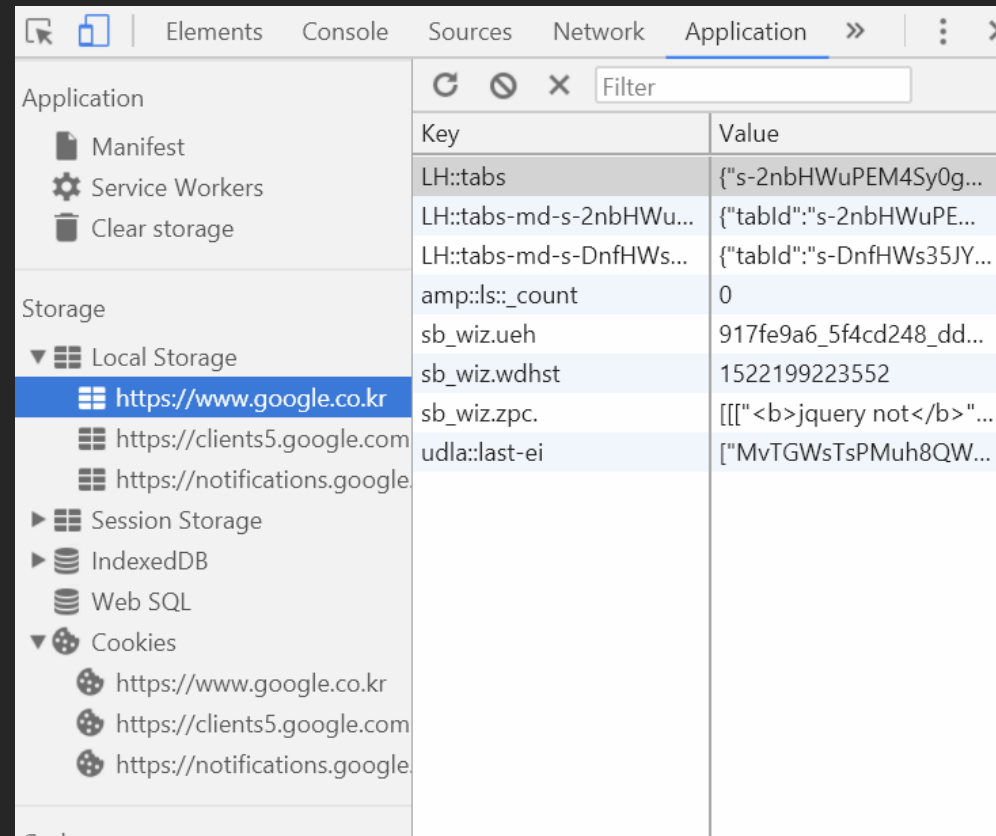
# localStorage, sessionStorage

브라우저 저장공간을 사용하면 만들 수 있는 것들 :

1. 장바구니 추가기능
2. 뒤로가기시 스크롤 위치 기억하기
3. 최근 본 상품
4. form의 내용 기억하기
5. 사용자의 개인 설정 저장 등

단점 :

1. localStorage 데이터가 출입할때 다른 자바스크립트 코드는 작동 중지상태
2. 문자열만 저장가능
3. LocalStorage 기능 끄고 웹서핑하는 사람도 있긴 함



## Syntax

### localStorage에 정보를 저장하고 빼는 법

#### 저장

```
localStorage.setItem('name', 'Smith');
```

#### 출력

```
localStorage.getItem('name');
```

```
localStorage['name']
```

```
localStorage.name
```

#### 삭제

```
localStorage.removeItem('name');
```

상품의 id, 상품명, 사이즈, 수량, 가격 등의 정보를 저장하면  
장바구니 추가 기능이 완성됩니다.

근데 단점은.. localStorage에는 '문자열' 밖에 저장이 안됩니다.

localStorage에는 '문자열' 밖에 저장이 안됩니다.

{Object}, [Array] 등을 저장하고 싶다면?

JSON 함수들을 이용합니다.

Object -> 문자열      `JSON.stringify(object)`

문자열 -> Object      `JSON.parse(object)`

# LocalStorage에 Object 저장/출력법

저장

```
var data = { firstname: 'John', lastname: 'Park'}  
localStorage.setItem('name', JSON.stringify(data));
```

object를 문자형태로 바꿔주는 함수

출력

```
JSON.parse(localStorage.data)
```

문자를 object로 바꿔줌

Add to cart 버튼을 누르면  
주문정보를 localStorage에 저장해봅시다.

Object 이름(key) : product1

Object에 들어갈 것들

1. ID
2. product name
3. quantity
4. size
5. price



## Springfield Shirt

This shirt is from Massachusetts, and we didn't even mean to promote any Springfield bolt-action rifles.



Add to Cart

My Cart

Add cart 버튼

clickevent 발동할 때

local storage에  
product1로 추가





1. 동적으로 Object 자료 생성 후
2. JSON화 하기

```
{  
  ID : 1,  
  name : 'Springfield',  
  size : 'small',  
  number : 2  
}
```



# JS

## Project6 : Shopping cart list 만들기

<input type="checkbox"/>	Image	ID	Product Name ▾	Model	Category	Price	Quantity	Status	Action
<input type="checkbox"/>		30	Canon EOS 5D	Product 3	Desktops Cameras	400.0000 80.0000	7	Enabled	
<input type="checkbox"/>		32	iPod Touch	Product 5	Cameras MP3 Players	100.0000	999	Enabled	

or 서버에서 보내준 쇼핑리스트 데이터로 동적으로 list 생성하기

Application

- Manifest
- Service Workers
- Clear storage

Storage

- Local Storage
  - http://127.0.0.1
- Session Storage
- IndexedDB
- Web SQL
- Cookies

Cache

- Cache Storage
- Application Cache

Frames

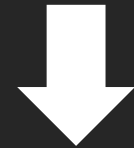
- top

Key	Value
product1	{"name": "springfield", "number": "1", "size": "Small"}

```
{name: "springfield", number: "1", size: "Small", ...}
```



localStorage안의 object를  
꺼내어 장바구니를 만들어보자



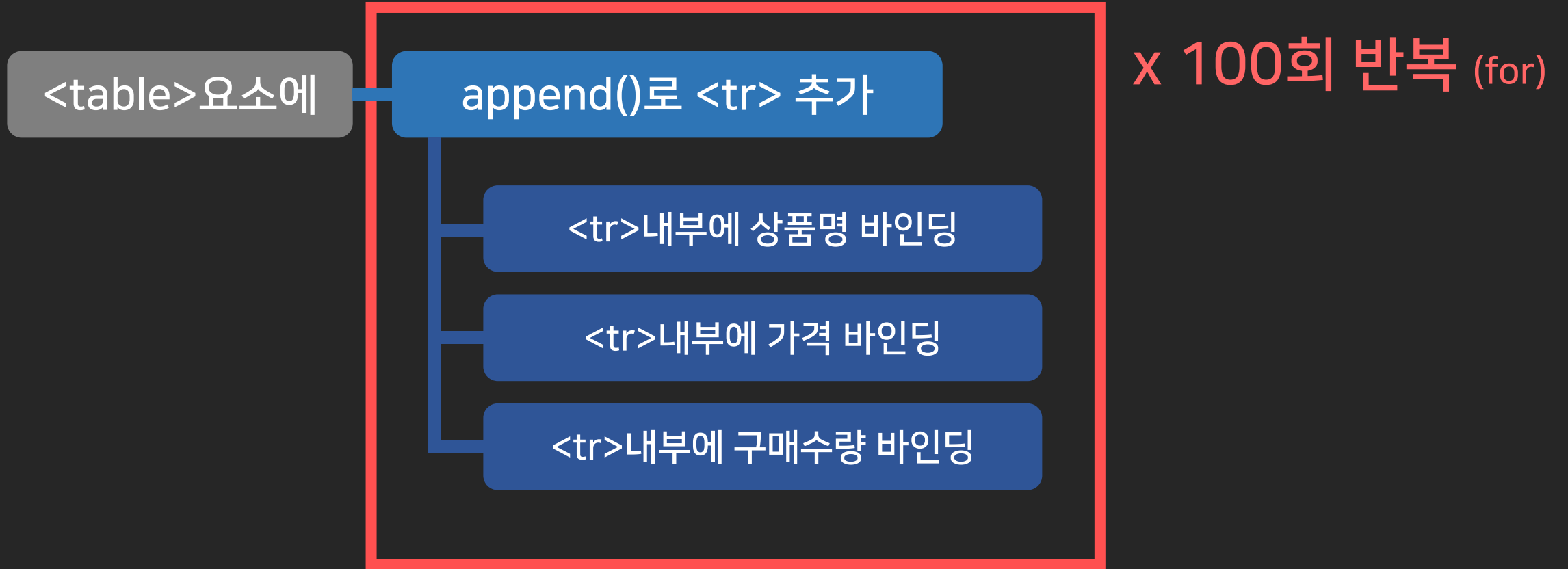
My Cart			
#	Name	Quantity	Size
1	springfield	1	Small

# shopping cart list 기능 만들기

Pseudo code :

1. 일단 HTML/CSS로 table 디자인 부터 해놓자 (dummy data 필요)
2. 데이터 넣을 상위 요소를 찾는다. div...tbody..같은거. (그리고.append()로 데이터를 넣으면 되겠네요)
3. 필요한 tr 개수만큼 for문을 돌린다.
4. for문 안에 tr을 tbody요소에.append() 하는 코드를 짠다.
5. 근데 그냥 append()하면 안되고 수량, 가격 데이터를 변동하면서 append() 시킨다.

jQuery pseudo 코드:



localStorage에서 데이터 꺼내는 방법은 아까 했었죠?

근데 product1, product2 .. product 50..  
이렇게 50개의 상품이 장바구니에 들어있으면  
어떻게 해야할까요

for 반복문을 돌리면 됩니다.

..근데 반복을 몇번 해야하는것?  
항상 50번인가요?

## Syntax

for문을 이용해 array 혹은 object를 **iterate** 할 수 있습니다.

### Array iteration

```
for (var i = 0; i < 어레이.length; i++) {  
    console.log(array[i]);  
}
```

array에 담긴 데이터 수를 알려주는 함수

### Object iteration

```
for (var key in 오브젝트) {  
    //만약 오브젝트.hasOwnProperty(key)  
    console.log(오브젝트[key]);  
}
```

## Syntax

forEach함수를 이용해도 array를 **iterate** 할 수 있습니다.

## Array iteration

```
어레이.forEach(function(a){  
    console.log(a)  
});
```

Array안의 요소를 a로 쪽 사용가능





# for / for ... in / forEach()

for은 Array iterate 할 때

for in은 object iterate 할 때

forEach는 Array iterate 할 때 (IE>9)

```
array.forEach(function( item ) {  
    console.log( item )  
});
```

```
for (var keys in object) {  
    console.log(keys)  
}
```

참고 : `querySelectorAll`를 `iterate` 하는 법

`querySelectorAll`, `getElementsByClassName` 등으로 요소를 잡으면 `[Array]`와 비슷한 `[NodeList]` 라는 것에 담겨옵니다.

`iterate` 하려면 전통적인 `for`문 혹은 `Array`로 바꾼 후 `forEach`문을 사용합니다.

<https://css-tricks.com/snippets/javascript/loop-queryselectorall-matches/>

# Javascript로 HTML 요소를 만들기 위해선...

```
var tableRow = '<tr><th scope="row">1번</th><td>1개  
</td><td>3000원</td><td>Small 사이즈</td></tr>';
```

(엔터키를 못넣어 끄 - 짝하기 때문에 ES6 문법도 좋은 선택)

시간 절약을 위해 append( )는 jQuery를 쓴다.

```
$('#div.content').append('컨텐츠를 추가합니다')
```

\* append()는 하위요소(children)에 내용을 추가하는 함수

(vanilla 자바스크립트는 넘나길어지는 것)

## JS Project7 : 이미지 slow scroll 애니메이션 개발 (jQuery)



New Nordstrom line  
What you've expected.

# 재료부터 만듭시다

뒷 신발 div만들고

앞 신발 div만들고

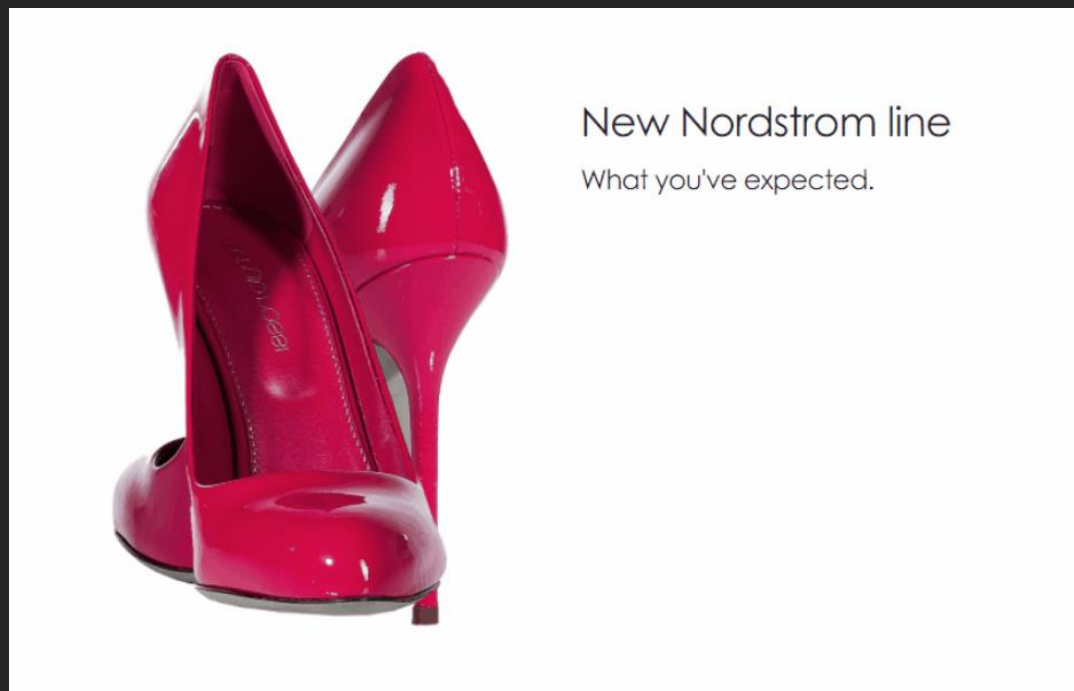
각각 background 속성으로 신발이미지 넣기.

그리고 background: position 속성으로 배경 위치를 조정하면 되겠네요

background-position : left 30px top ??px;

가장 쉬운 애니메이션부터 한번 만들어봅시다.

스크롤 바를 내릴 때마다  
앞신발은 위로 올라오는 애니메이션



스크롤 바를 내릴 때마다  
앞신발은 위로 올라오는 애니메이션

현재 window

Scroll event 발동할 때

신발 위치 CSS 바꾸기

(신발의 높이는  
현재 스크롤바 위치와  
반비례 해야합니다.)



현재 window

Scroll event 발동할 때

신발 위치 CSS 바꾸기

현재 문서에서 스크롤바가 몇 px 내려갔는지 알려주는 함수

`scrollTop( );`

요소의 CSS를 바꾸려면

`.css({'background-position': 'top 00px'})`

Skrollr.js

Scrollreveal.js

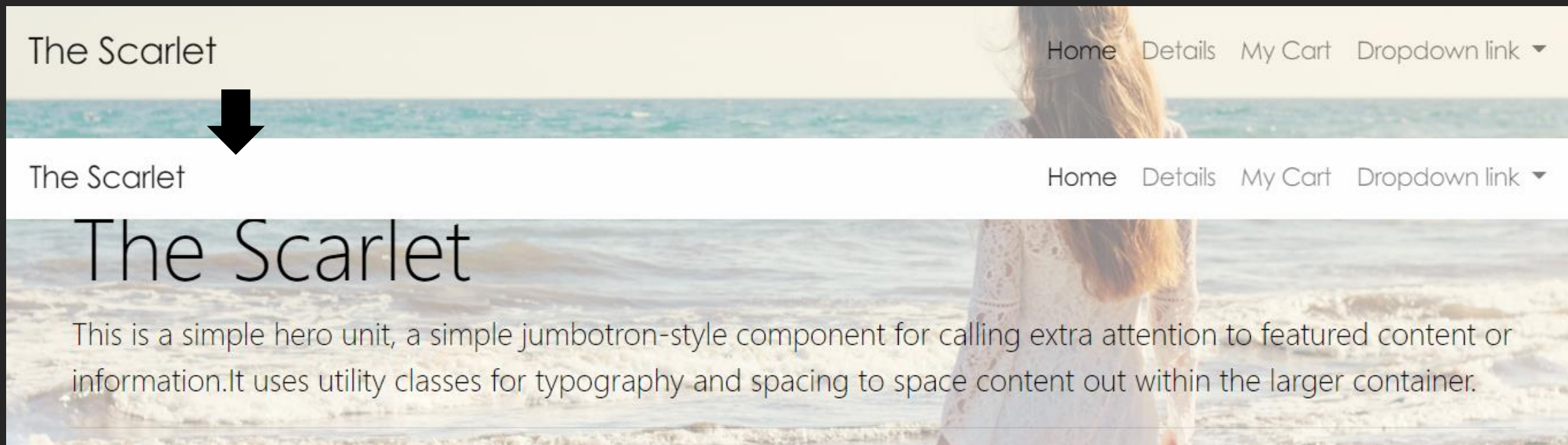
Greensock Animation API (GSAP) (IE6+)

## CSS animation vs JS animation

<https://www.heartinternet.uk/blog/should-you-use-css-or-javascript-for-web-animations/>

JS

## Project8 : Nav 메뉴 투명화 애니메이션 (transition/animate)



1. 맨 처음 로딩시엔 어떤 CSS를 적용?
2. 어떤 이벤트를 측정해야하나?
3. 이벤트 발동시엔 어떤 CSS를 적용해야함?
4. 이벤트 미발동시엔 어떤 CSS를 적용해야함?

## jQuery animate() 로 만드는 애니메이션

```
element.animate({fontSize: '30px'}, {duration: 3000});
```

animate함수로 CSS를 변동시킬 때 애니메이션을 줄 수 있음.

장점 : JS단에서 애니메이션 관리 가능

단점 : 버벅거림 (Velocity.js 쓰셈)

## CSS transition 으로 만드는 애니메이션 (IE10+)

```
.big-font { transition : 1s linear; }
```

장점 : animate 보다는 **부드러움**, JS와 별도로 처리됨.

단점 : CSS class를 하나 더 만들어야함, IE 호환성

스크롤 애니메이션 개발을 위한 두개의 큰 틀 :

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

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



## velocity.js

모든 jQuery animate함수처럼 이용가능,

쉬움,

부드러움,

용량7KB

# JS

## Project9 : 메인페이지 Product list 더보기 버튼 (AJAX이용)

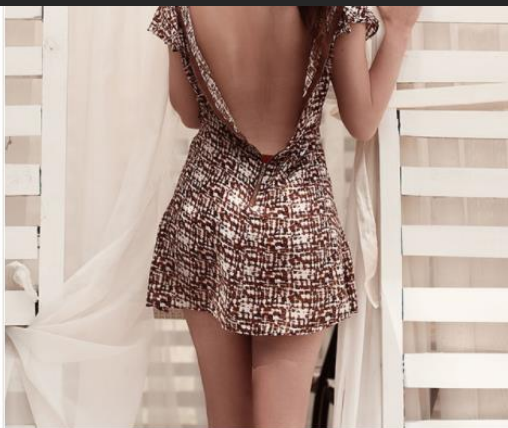


### Black Assassin

Hey look up! It's a plane! No, It's a bird! No, it's the assassin performing leap of faith.

from \$350

Details

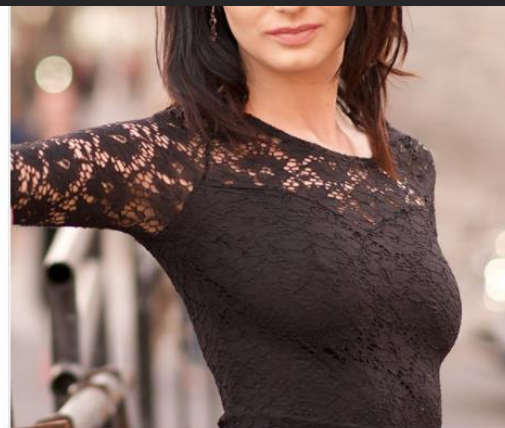


### No zipper dress

The price of this dress has been cut off since we lost the zipper on the back.

from \$59

Details



### Dear Ernest

We'll bring readers to the edge of their seats and keep them there until we sell this dress.

from \$299

Details



Show More



# jQuery AJAX

`$(element).load("")`

특정 element 위치에  
특정 파일(코드)를 삽입할 때 씀

`$.ajax()`

그냥 기본 AJAX함수

`$.get()`

GET요청만 처리하는 AJAX함수

(vanilla 자바스크립트는 XMLHttpRequest 함수를 씁니다. 이거나 \$.ajax나 아무 차이 없음)

## Syntax

# jQuery AJAX요청으로 JSON데이터 가져오는 법

```
$.ajax({  
    type : 'GET',  
    url : '/data.json',  
    dataType: 'json',  
  
}).done(function(data){  
    console.log(data)  
}).fail(function(jqXHR, textStatus, error){  
    console.log(error)  
});
```

## jQuery AJAX 기타 설정 부분

`data : {"id" : 'qwer1234'}` // ajax요청시 서버로 정보를 보내고 싶다면 이렇게 (특히 POST)

`dataType: 'json'` // 서버에서 수신될 정보의 expected type (없어도 가끔 잘됨)

`beforeSend: 함수(){}` // ajax 요청하기 전에 먼저 실행할 함수(로딩중~ 띄운다든지 그럴 때)

`$ajax( ).done( function(){} )` // 성공시 실행할 코드

`$ajax( ).fail( function(){} )` // 실패시 실행할 코드

`$ajax( ).always( function(){} )` // ajax 실패하든 성공하든 무조건 실행할 함수

서버에서 JSON 데이터를 보내왔는데 ..

```
{  
  "name":"John",  
  "age":31,  
  "pets":[  
    { "animal":"dog", "name":"Fido" },  
    { "animal":"cat", "name":"Felix" },  
    { "animal":"hamster", "name":"Lightning" }  
  ]  
}
```

이건 object 가 아닙니다. 데이터를 뽑아 쓸 수가 없어요.  
object로 바꿔줘야겠죠?

## 서버에서 데이터를 보내왔는데 ..

```
{
  "name":"John",
  "age":31,
  "pets":[
    { "animal":"dog", "name":"Fido" },
    { "animal":"cat", "name":"Felix" },
    { "animal":"hamster", "name":"Lightning" }
  ]
}
```

`var data = JSON.parse('object 처럼 생긴 텍스트');`

(근데 jQuery AJAX는 parse 할 필요 없음)

# AJAX 활용방안

회원가입시  
ID 중복체크

더보기 버튼  
더보기 스크롤

초기 수신 되는  
데이터 절약

쇼핑몰의 탭기능  
(위메프)

★ 구글 크롤러봇은  
AJAX데이터  
추적불가 ★

댓글,  
댓글의 vote기능



# AJAX 단점

★ 구글 크롤러봇은  
AJAX데이터  
추적불가 ★

GET요청 pagination  
이 더 좋을 수도

다른 url로 요청 불가

IE 7 +

AJAX탭 4개를 만들면  
서버의 요청 횟수도 4배

# JS

## Project10 : 총 금액 계산 기능 (즐거운 수학시간)

My Cart

#	Name	Quantity	Size	Price
1	blossom	<input type="text" value="0"/>	Medium	299
2	springfield	<input type="text" value="2"/>	Small	99

Total : 198

Order Now

기능 :

1. 맨 처음 로드시 table안의 총 금액을 계산해줌
2. 상품수량을 변경하면 table 안의 총 금액을 다시 계산해줌

준비물 :

- 상품 수량을 <input> 태그로 바꾸기
- 최종가격 표시할 text 부분 만들기

Pseudo code 를 생각하기 어려울 때 꿀팁 :

1. Happy User가 사용하는 경우만 일단 생각 (예외는 나중에)
2. 반복문이 어렵다면 반복하지말고 하나만 작성해볼것
3. Dummy data를 채워넣고 시작하면 생각이 잘날지도
4. 논리적으로! 단계별로! 생각하기

최종 값 = 0

최종 값 = 최종 값 + 상품1 갯수 X 상품1 가격

최종 값 = 최종 값 + 상품2 갯수 X 상품2 가격

최종 값 = 최종 값 + 상품3 갯수 X 상품3 가격

.....

최종 값 출력

## 2. input내의 상품수량을 변경시에 가격을 다시 계산하게 하려면?

- 함수화
- input에 ??이벤트 달기

# JS

## Project11 : Increment Button 만들기 (jQuery)

Common Projects Bball High White	<div><div>+</div><div>2</div><div>-</div></div>	\$549
Maison Margiela Future Sneakers White	<div><div>+</div><div>1</div><div>-</div></div>	\$870
Our Legacy Brushed Scarf Brown	<div><div>+</div><div>1</div><div>-</div></div>	\$349

원리는 간단합니다. button 누르면 input value가 +1 / -1

```
<td><+button><input><-button></td>
```

```
<td><+button><input><-button></td>
```

```
<td><+button><input><-button></td>
```

하지만 input과 button이 100개가 있다면?



버튼을 눌렀을 때 input 100개 중에 정확한 input을 찾아야함

```
<td><+button><input><-button><td>
```

```
<td><+button><input><-button><td>
```

```
<td><+button><input><-button><td>
```

Q. 그럼.. 전부 이름쓰거나 id 달아줘야하나요?

jQuery는 형제, 부모, 자식 element를 빠르게 찾아줍니다.

형제는 `siblings('input')`

부모는 `closest('')`

자식은 `children('')`

입력



변경



출력

h1 요소를 선택하고  
태그 안의 내용을 가져온다

내용을 'Hello'로 바꾼다

h1 요소 내부에  
박아넣는다

데이터



변경



출력

그냥 데이터

내용을 이리저리 바꾼다

그 자리에 박아넣는다

## Misc 카드결제?

### 방법 1.

- Paypal/Stripe 를 통한 Visa, Mastercard 결제 (수수료 3% + 원화결제 수수료 X%)
- JS코드 7줄만 추가하면 되니 초간단

### 방법 2.

- 국내카드는 PG사 가입 후 PG사 결제모듈 설치해야함 (심각하게 복잡)
- 설치 쉽게 도와주는 아임포트 [iamport.kr](http://iamport.kr)

**Misc** ES6 문법을 쓰려면 Node.js + Babel을 써야..

호환성 안좋은 ES6 문법을 '예전' 문법으로 바꿔주는 고마운 툴  
Node.js 설치하고 babel 설치하고 튜토리얼 대로 따라하면 됩니다.

또는 에디터 부가기능에 babel화 해주는 기능을 찾아서 써도 잘됨

## Misc 서버와 데이터베이스 몰라도되는 '블로그' & '게시판' 만들기

### <Bludit>

PHP설치된 호스팅 공간만 있으면 쉽게 사용가능합니다. (워드프레스보다 가벼움)

Markdown, HTML로 게시물 작성.

### <Xpress Engine>

한국스타일 게시판을 만들기 위한 무료 php 게시판 엔진. 복붙만 하면 설치 끝입니다.

admin 페이지에서 모든걸 관리 가능.

커뮤니티 사이트 만들기 정말 쉽쥬?

**Misc** more 자바스크립트 프로젝트형 강의

<https://javascript30.com> (영어주의)