

jQuery 마스터하기

Javascript Framework

장민창






mcjang@hucloud.co.kr

jQuery 마스터하기

1. jQuery 시작하기
2. 선택팅
3. DOM Traversing
4. DOM 다루기
5. Event
6. Ajax
7. Form Element

jQuery

jQuery

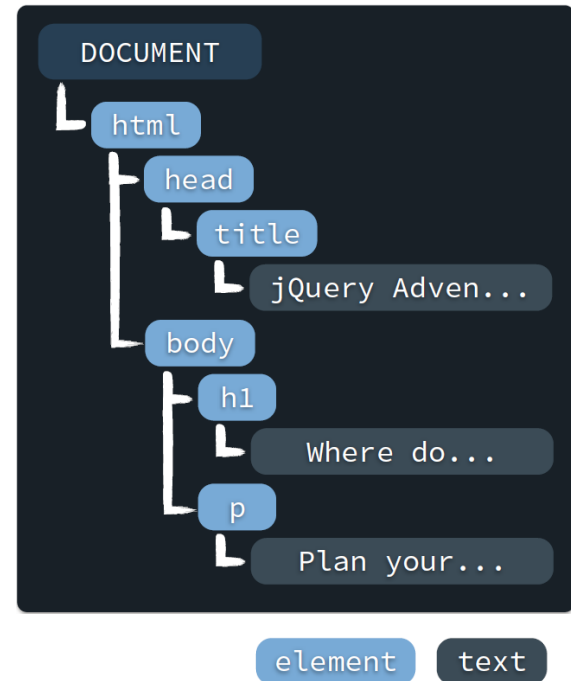
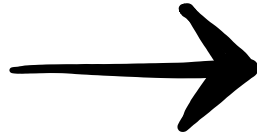
- Node(Element)를 간편하게 컨트롤 하기 위한 라이브러리
- jQuery는 아래 작업들을 편하게 만들어준다.
 -  HTML 문서에서 Element 찾기
 -  HTML Content 변경하기
 -  사용자의 Action 에 반응하기
 -  웹 페이지의 내용을 애니메이션 하기
 -  네트워크에서 새로운 콘텐츠 받아오기
- jQuery 로 처리할 수 있는 Element 들의 기능들.
 - CSS
 - Event
 - Animation
 - Create, Modify, Delete 등을 처리.
- Network
 - 비동기 통신을 간편하게 처리함.

jQuery 시작하기

jQuery 시작하기

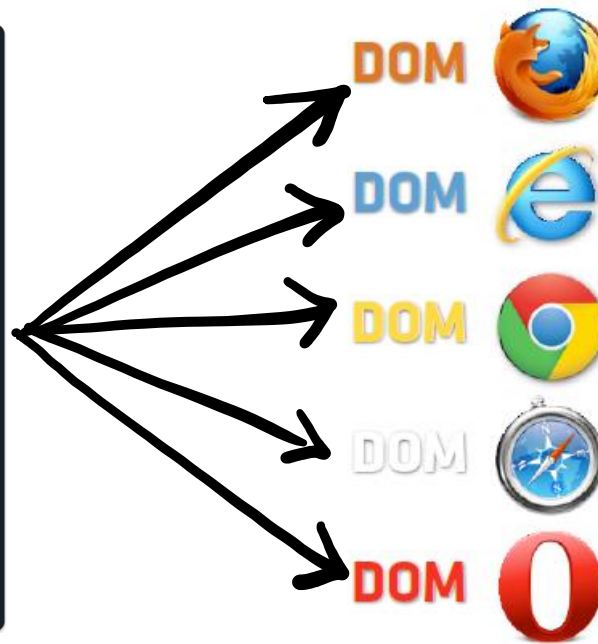
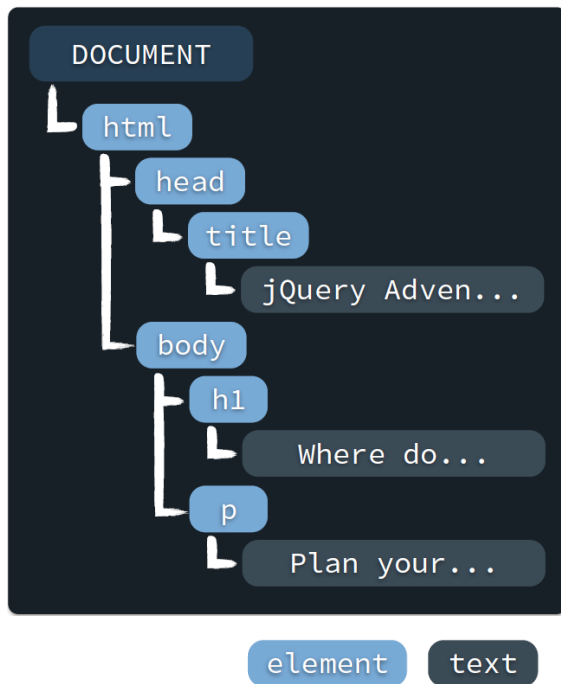
- jQuery를 시작하기 전..
- 사실 HTML 문서는 DOM(Document Object Model) 의 집합.

```
<!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>
```



jQuery 시작하기

- Javascript로 DOM 들을 컨트롤 하려면..
- Document.get..... 길고 복잡하다.
- 게다가..!



브라우저마다
처리하는 방법이 다르다!

jQuery 시작하기

- 이 덕에..
- 안그래도 복잡한 Javascript 에 Javascript Hack이 포함되기 시작함.
- 이런 상황을 안타깝게 여긴 John Resig

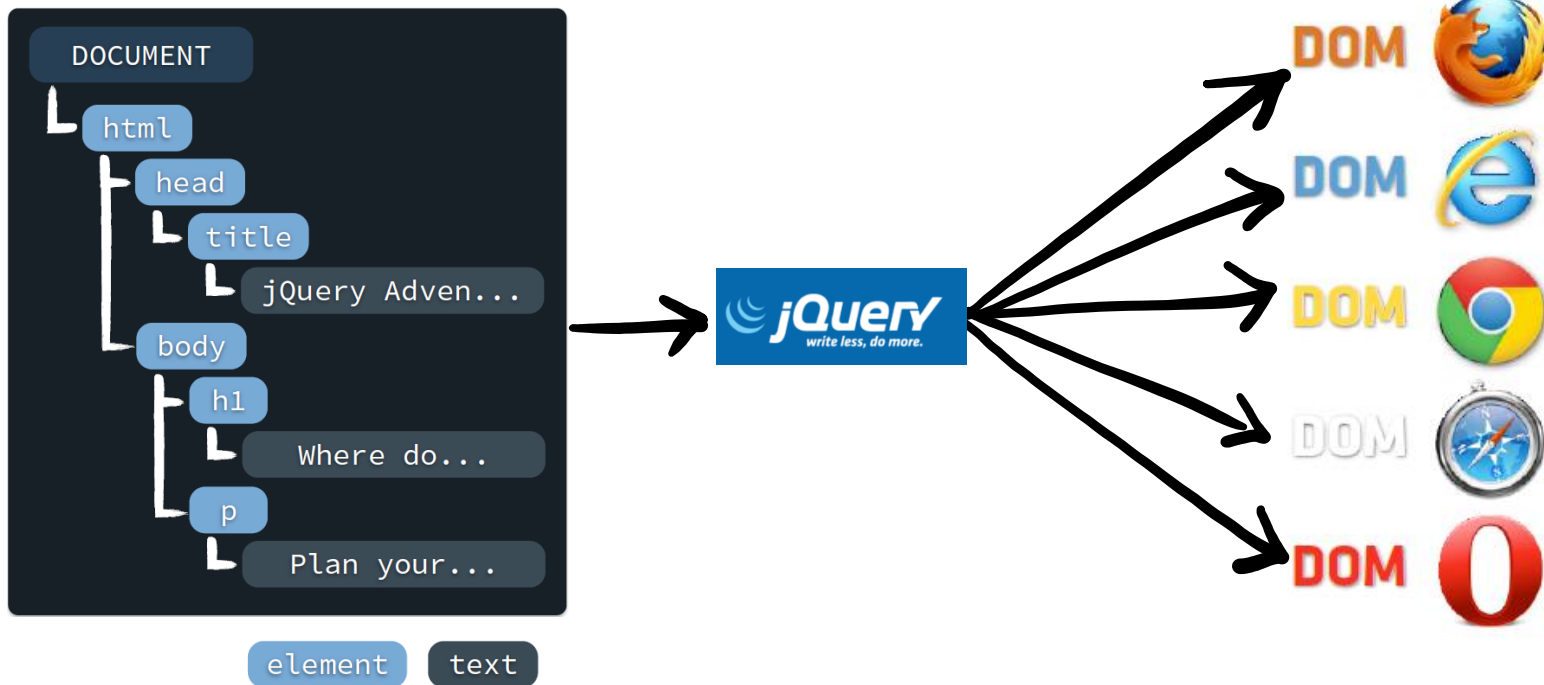


jQuery로 자바스크립트 개발자들을 구원하다!

https://ko.wikipedia.org/wiki/존_레식

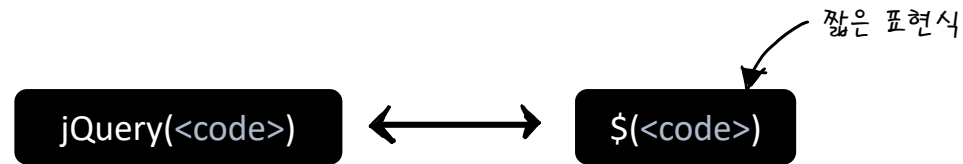
jQuery 시작하기

- jQuery 는 브라우저별로 적절한 처리를 수행해 준다.
 - 크로스 브라우징!



jQuery 시작하기

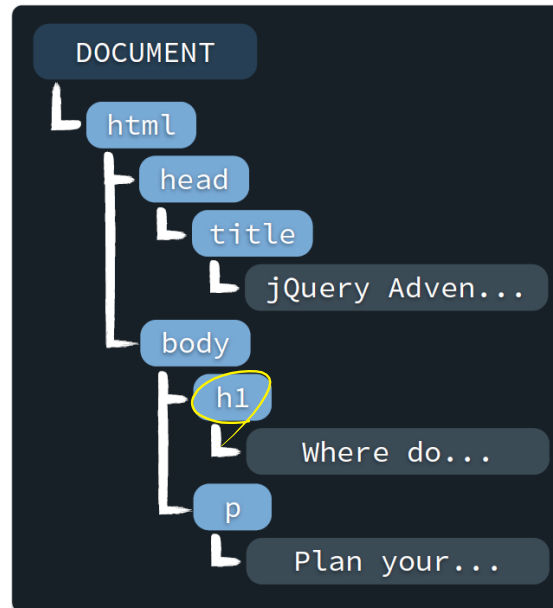
- jQuery



jQuery 시작하기

- jQuery 사용 준비하기
- H1 엘리먼트를 jQuery로 가져와보기

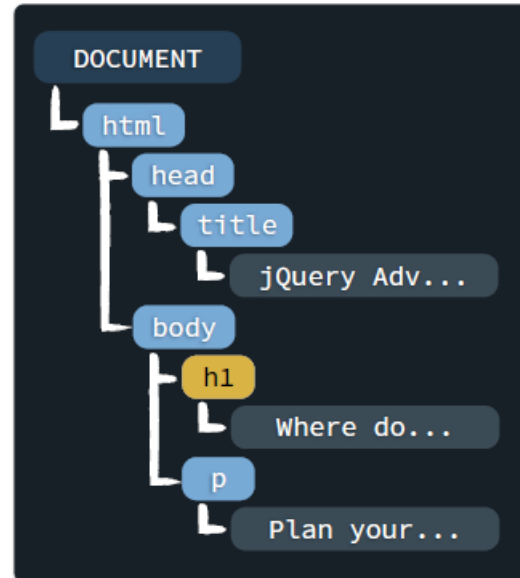
```
<!DOCTYPE html>
<html>
<head>
  <title>jQuery Adventures</title>
</head>
<body>
  <h1>Where do you want to go?</h1>
  <p>Plan your next adventure.</p>
</body>
</html>
```



jQuery 시작하기

- jQuery 사용 준비하기
- H1 엘리먼트를 jQuery로 가져와보기

```
$("h1");
```

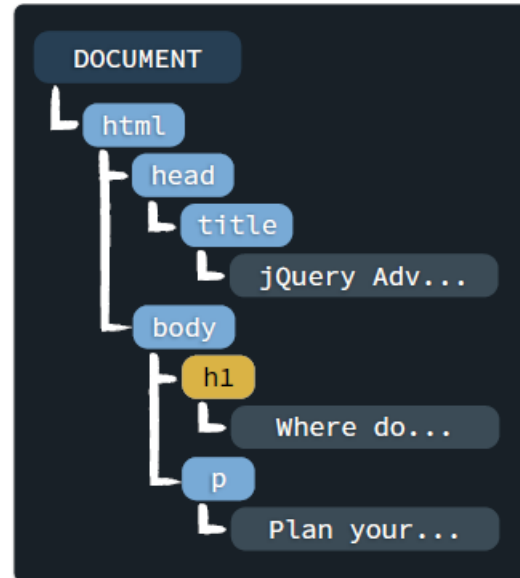


jQuery 시작하기

- jQuery 사용 준비하기
- 텍스트 가져오기

```
$("#h1").text();
```

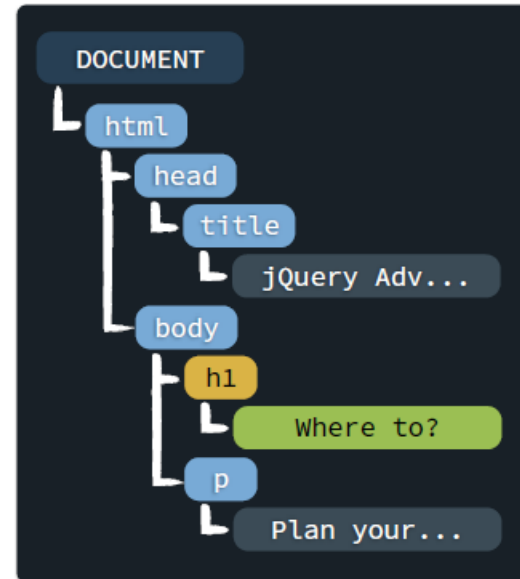
→ "Where do you want to go"



jQuery 시작하기

- jQuery 사용 준비하기
- 텍스트 셋팅하기

```
$("#h1").text("Where to?");
```



jQuery 시작하기

- jQuery 시작하기

```
jQuery(document).ready(function(){  
    <code>  
});
```

jQuery는 `.ready()` 이벤트를 사용합니다.
`Ready()` 이벤트가 실행되면, `html` 문서를 읽어
`dom` 요소를 모두 객체화 시키게 됩니다.

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

DOM 로딩이 끝난 직후, `h1` 엘리먼트의 텍스트를 변경한다.

jQuery 시작하기

- jQuery 사용 준비하기

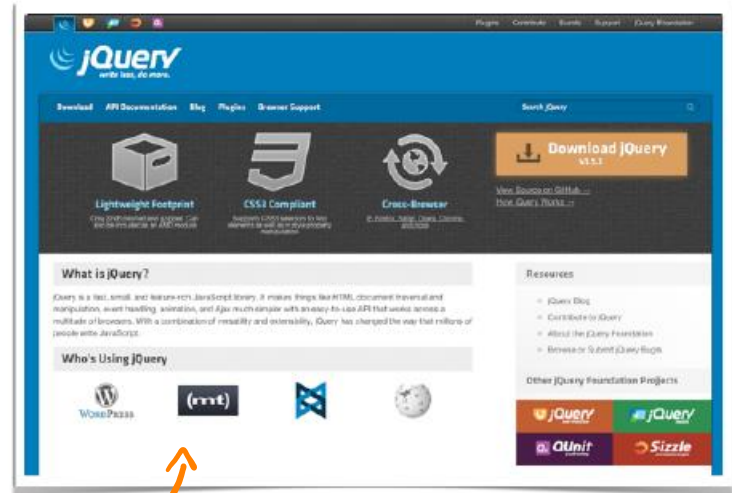
1 download jQuery

2 load it in your HTML document

```
<script src="jquery.min.js"></script>
```

3 start using it

```
<script src="application.js"></script>
```



<http://jquery.com>

선택팅

Selector

- 여러 엘리먼트 변경해보기

```
<h1>Where do you want to go?</h1>
<h2>Travel Destinations</h2>
<p>Plan your next adventure.</p>
<ul id="destinations">
  <li>Rome</li>
  <li>Paris</li>
  <li class='promo'>Rio</li>
</ul>
```

</i> 요소를 찾고 텍스트를 바꿔줍니다.

```
$("li");
```

</i> 요소를 가져옵니다.



Selector

- jQuery 셀렉터는 CSS와 동일하다.

CSS Selector

```
p { ... }
```

```
#container { ... }
```

```
.articles { ... }
```

jQuery Selector

```
$("#p");
```

```
$("#container");
```

```
$(".articles");
```



Selector

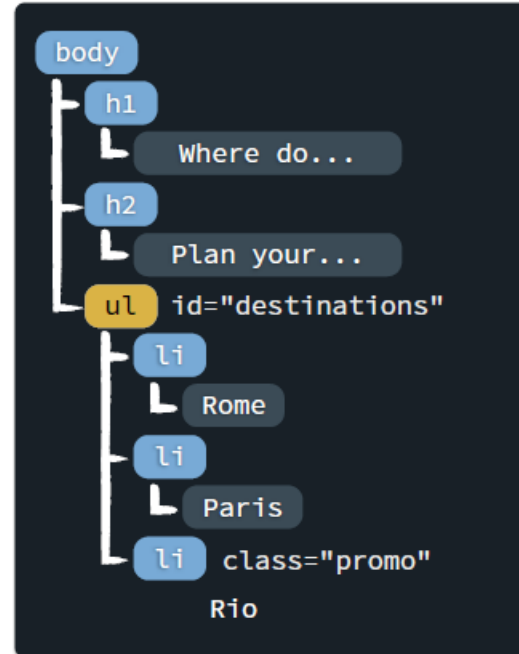
- 여러 엘리먼트 중 하나만 가져오기 (ID Selector)

```
<h1>Where do you want to go?</h1>
<h2>Travel Destinations</h2>
<p>Plan your next adventure.</p>
<ul id="destinations">
  <li>Rome</li>
  <li>Paris</li>
  <li class='promo'>Rio</li>
</ul>
```

id가 destinations 인 요소를 찾아 봅니다.

```
$("#destinations");
```

id가 destinations 인 요소를 가져옵니다.



Selector

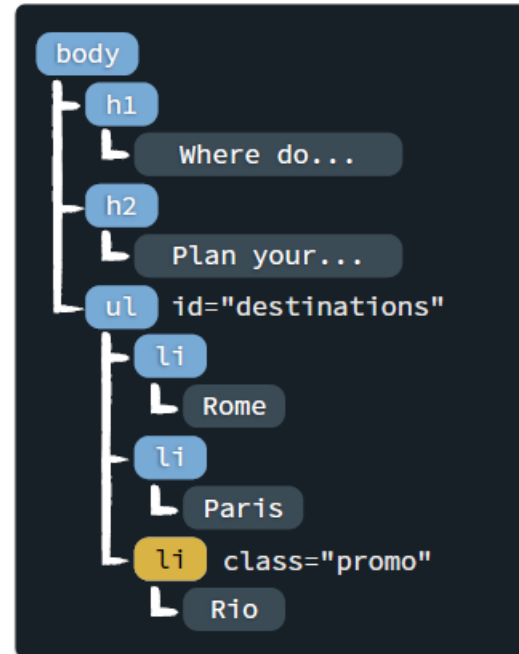
- 여러 엘리먼트 중 특정한 것만 가져오기 (Class Selector)

```
<h1>Where do you want to go?</h1>
<h2>Travel Destinations</h2>
<p>Plan your next adventure.</p>
<ul id="destinations">
  <li>Rome</li>
  <li>Paris</li>
  <li class='promo'>Rio</li>
</ul>
```

class가 promo 인 요소를 찾아 봅니다.

```
$(".promo");
```

class가 promo 인 요소를 가져옵니다.



하위 요소 선택팅

하위 요소 선택팅

- 자식 요소 가져오기

```
<h1>Where do you want to go?</h1>
<h2>Travel Destinations</h2>
<p>Plan your next adventure.</p>
<ul id="destinations">
  <li>Rome</li>
  <li>Paris</li>
  <li class='promo'>Rio</li>
</ul>
```

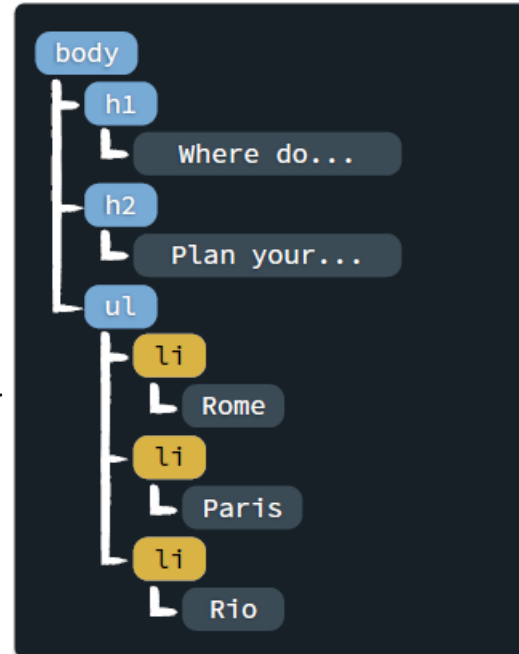
id가 destinations 하위의 li 요소를 찾아 봅니다.

스페이스로 하위(부모와 자식)요소를 구분함

```
$("#destinations li");
```

부모요소

자식요소



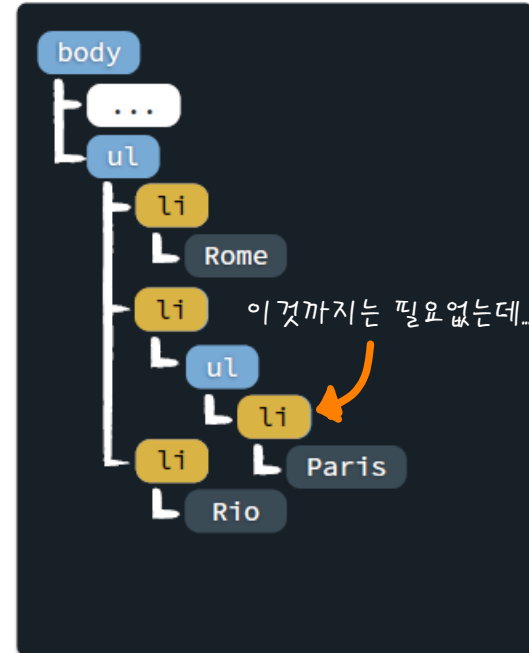
하위 요소 선택팅

- 자식 요소 가져오기

```
<h1>Where do you want to go?</h1>
<h2>Travel Destinations</h2>
<p>Plan your next adventure.</p>
<ul id="destinations">
  <li>Rome</li>
  <li>
    <ul id="france">
      <li>Paris</li>
    </ul>
  </li>
  <li class='promo'>Rio</li>
</ul>
```

ul 하위의 li 요소를 찾아봅니다.

```
$("#destinations li");
```



- 하위 요소 선택팅은 자식 요소 중 존재하는 것이 있다면, 모두 가져온다.
- Step(Depth)에 상관없이 모두 가져옴.

하위 요소 선택팅

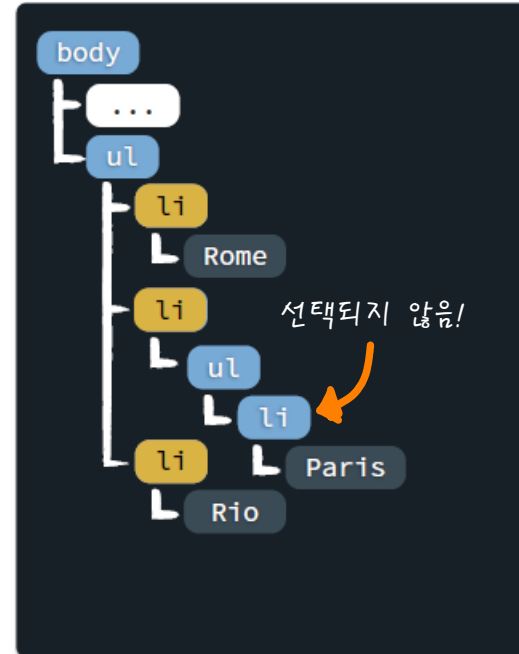
- 자식 요소 가져오기

```
<h1>Where do you want to go?</h1>
<h2>Travel Destinations</h2>
<p>Plan your next adventure.</p>
<ul id="destinations">
  <li>Rome</li>
  <li>
    <ul id="france">
      <li>Paris</li>
    </ul>
  </li>
  <li class='promo'>Rio</li>
</ul>
```

ul 바로 하위의 li 요소를 찾아봅니다.

```
$("#destinations > li");
```

> 는 부모요소의 직속 자식 요소만 검색한다.



다중 요소 셀렉팅

다중 요소 선택팅

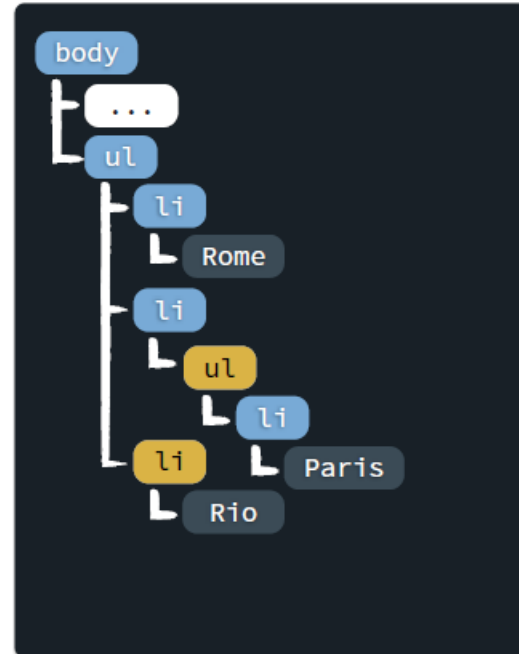
- 다중 요소 선택팅

```
<h1>Where do you want to go?</h1>
<h2>Travel Destinations</h2>
<p>Plan your next adventure.</p>
<ul id="destinations">
  <li>Rome</li>
  <li>
    <ul id="france">
      <li>Paris</li>
    </ul>
  </li>
  <li class='promo'>Rio</li>
</ul>
```

관계없는 요소 검색하기

```
$(".promo, #france");
```

, 는 여러 개의 요소를 동시에 검색할 수 있도록 합니다.



PSEUDO Classes

PSEUDO Classes

- 처음과 끝

```
$("#destinations li:first");
```

```
$("#destinations li:last");
```

*Filter
:first
:last*



PSEUDO Classes

- 처음과 끝

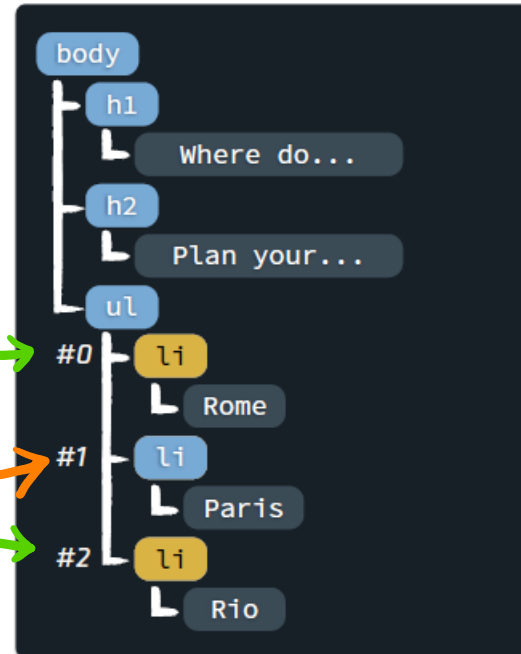
```
$("#destinations li:odd");
```

```
$("#destinations li:even");
```

Filter

:odd → 홀수번째 엘리먼트

:even → 짝수번째 엘리먼트




DOM Traversing

DOM Traversing

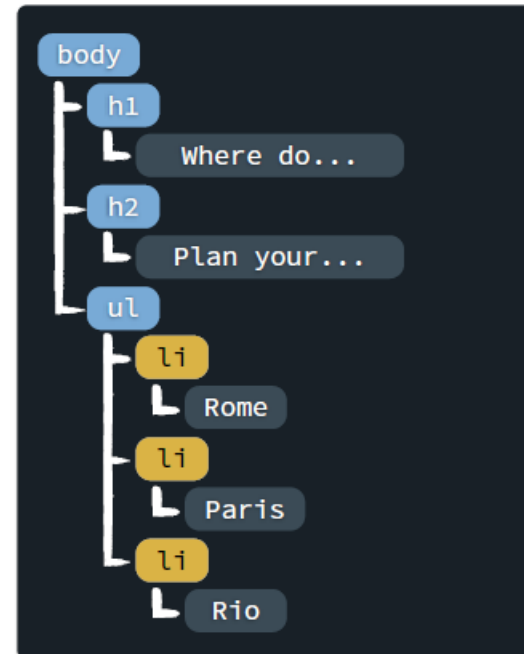
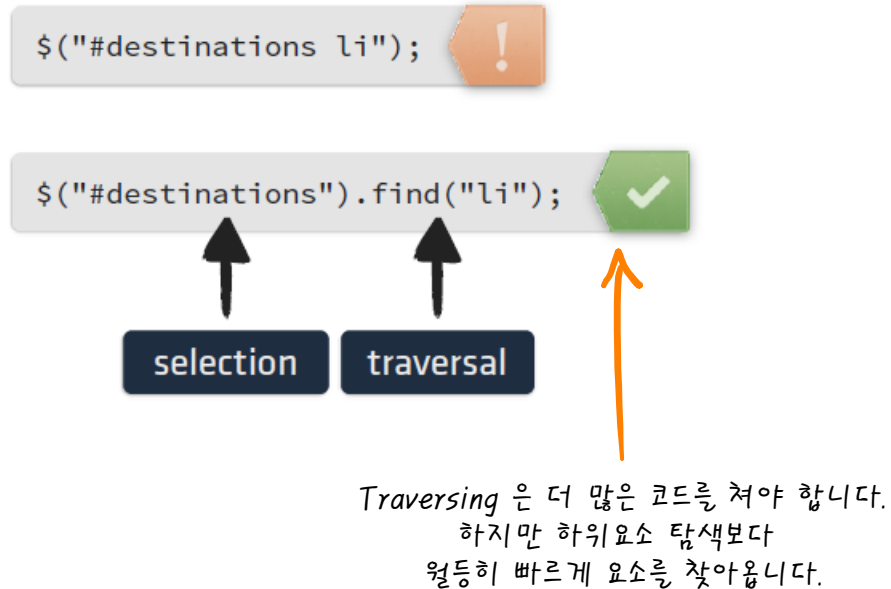
- HTML Document 탐색하기

```
<h1>Where do you want to go?</h1>
<h2>Travel Destinations</h2>
<p>Plan your next adventure.</p>
<ul id="destinations">
  <li>Rome</li>
  <li>Paris</li>
  <li class='promo'>Rio</li>
</ul>
```

 destination 의 li 요소를 하위요소 탐색이외의 방법으로 찾아볼 수 없을까요?

DOM Traversing

- HTML Document 탐색하기



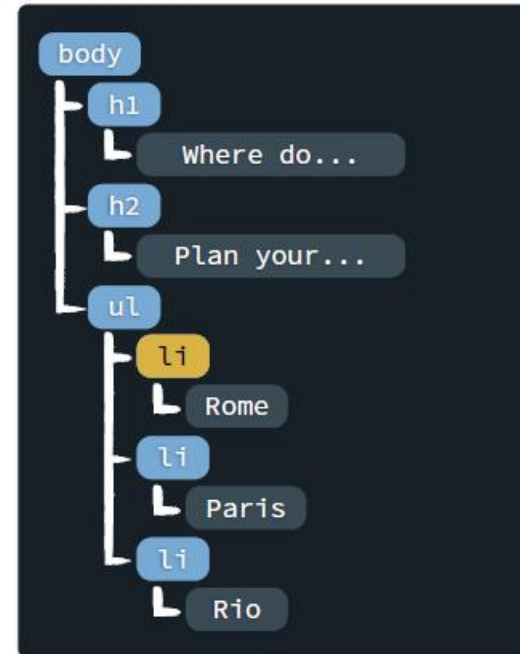
DOM Traversing

- HTML Document 탐색하기

```
$("#li:first");
```



```
$("#li").first();
```



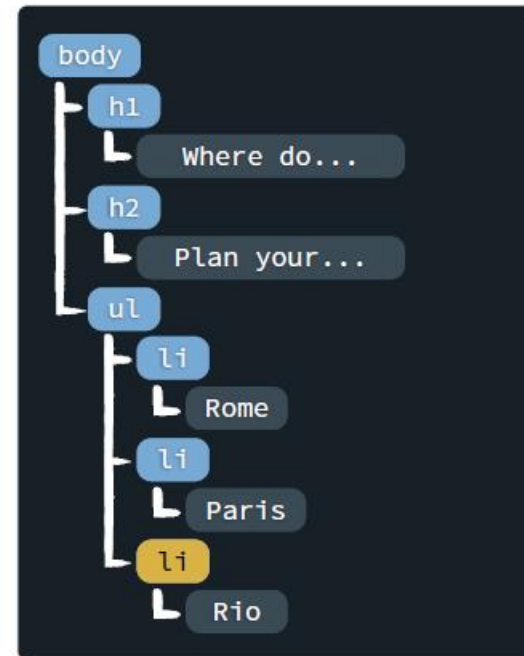
DOM Traversing

- HTML Document 탐색하기

```
$("#li:last");
```



```
$("#li").last();
```



DOM Traversing

- HTML Document 탐색하기

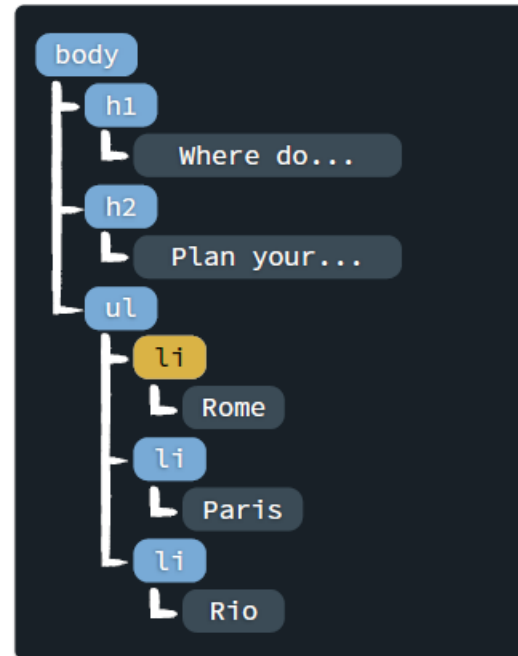
```
<h1>Where do you want to go?</h1>
<h2>Travel Destinations</h2>
<p>Plan your next adventure.</p>
<ul id="destinations">
  <li>Rome</li>
  <li>Paris</li>
  <li class='promo'>Rio</li>
</ul>
```

처음과 끝이 아닌 중간 요소를 찾는 방법은 없을까요?

DOM Traversing

- HTML Document 탐색하기

```
$("li").first();
```

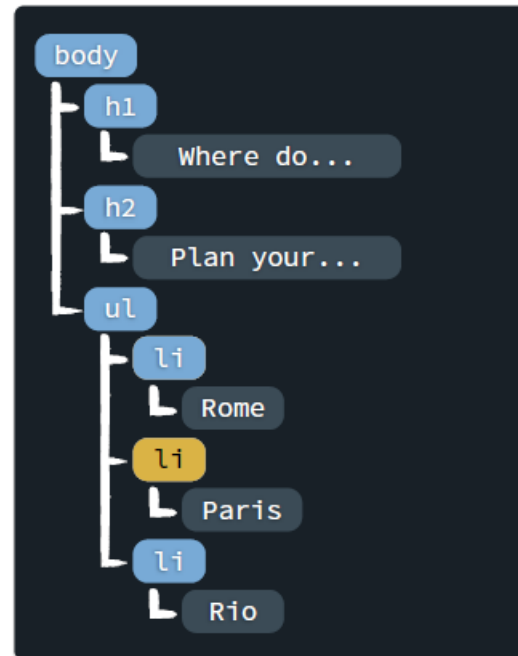


DOM Traversing

- HTML Document 탐색하기

```
$("#li").first();
```

```
$("#li").first().next();
```



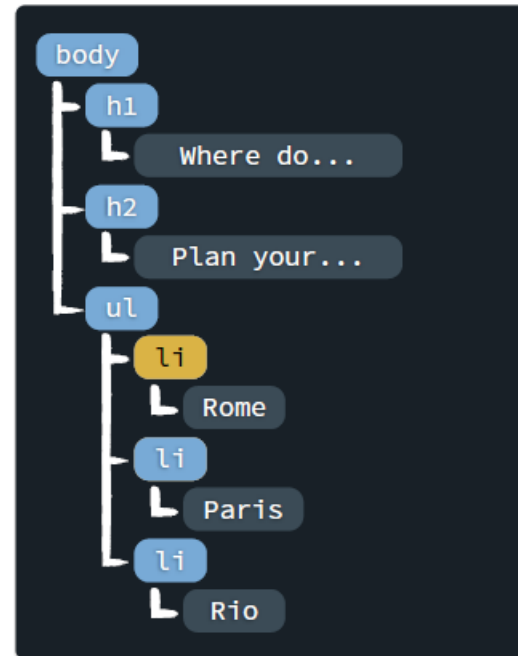
DOM Traversing

- HTML Document 탐색하기

```
$("#li").first();
```

```
$("#li").first().next();
```

```
$("#li").first().next().prev();
```



DOM Traversing

- HTML Document 탐색하기

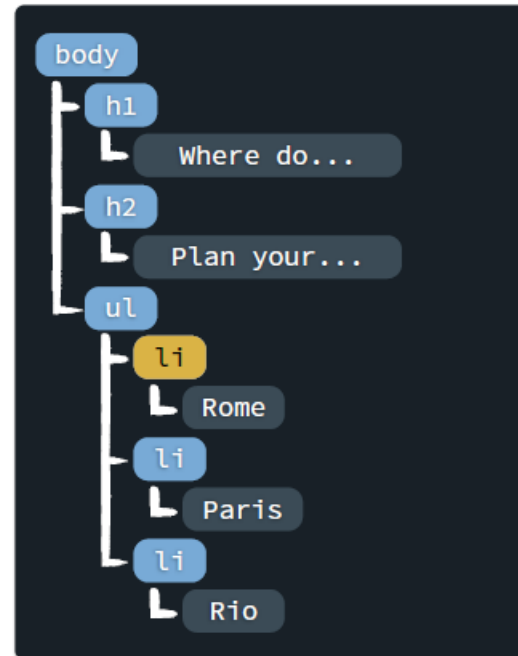
```
<h1>Where do you want to go?</h1>
<h2>Travel Destinations</h2>
<p>Plan your next adventure.</p>
<ul id="destinations">
  <li>Rome</li>
  <li>Paris</li>
  <li class='promo'>Rio</li>
</ul>
```

이때까지는 상위에서 하위로 탐색했는데
하위에서 상위로 탐색할 수 없을까요?
.promo 에서 #destination 을 찾도록 말이죠.

DOM Traversing

- HTML Document 탐색하기

```
$("li").first();
```



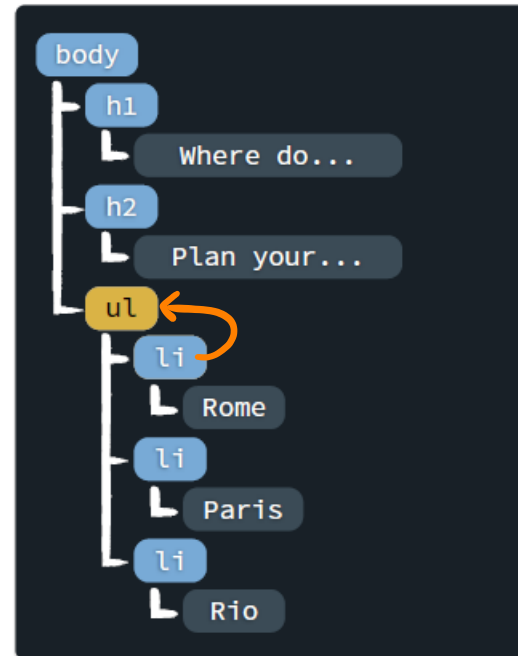
DOM Traversing

- HTML Document 탐색하기

```
$("#li").first();
```

```
$("#li").first().parent();
```

*.parent()*는 체이닝을 통해 여러 번 연결해 사용할 수 있습니다.
.parent().parent()...



DOM Traversing

- HTML Document 탐색하기

```
<h1>Where do you want to go?</h1>
<h2>Travel Destinations</h2>
<p>Plan your next adventure.</p>
<ul id="destinations">
  <li>Rome</li>
  <ul id="france">
    <li>Paris</li>
  </ul>
  <li class='promo'>Rio</li>
</ul>
```

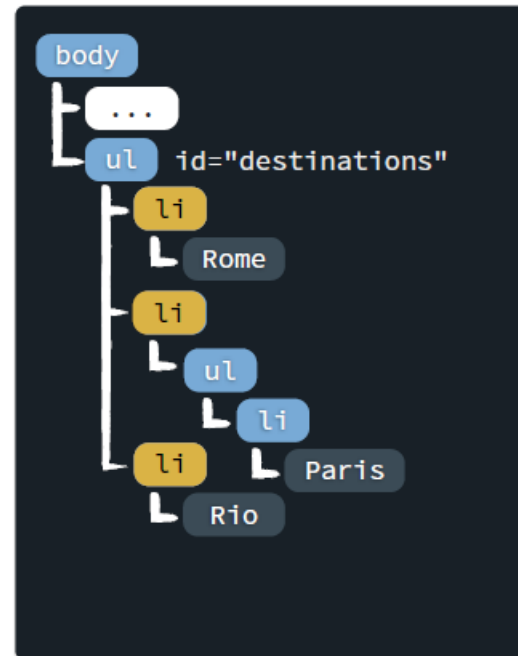
#destination 아래에 있는 여러 li 요소 중에서
바로 하위에 있는 li 만 가져오는 방법이 있을까요?

DOM Traversing

- HTML Document 탐색하기

```
$("#destinations").children("li");
```

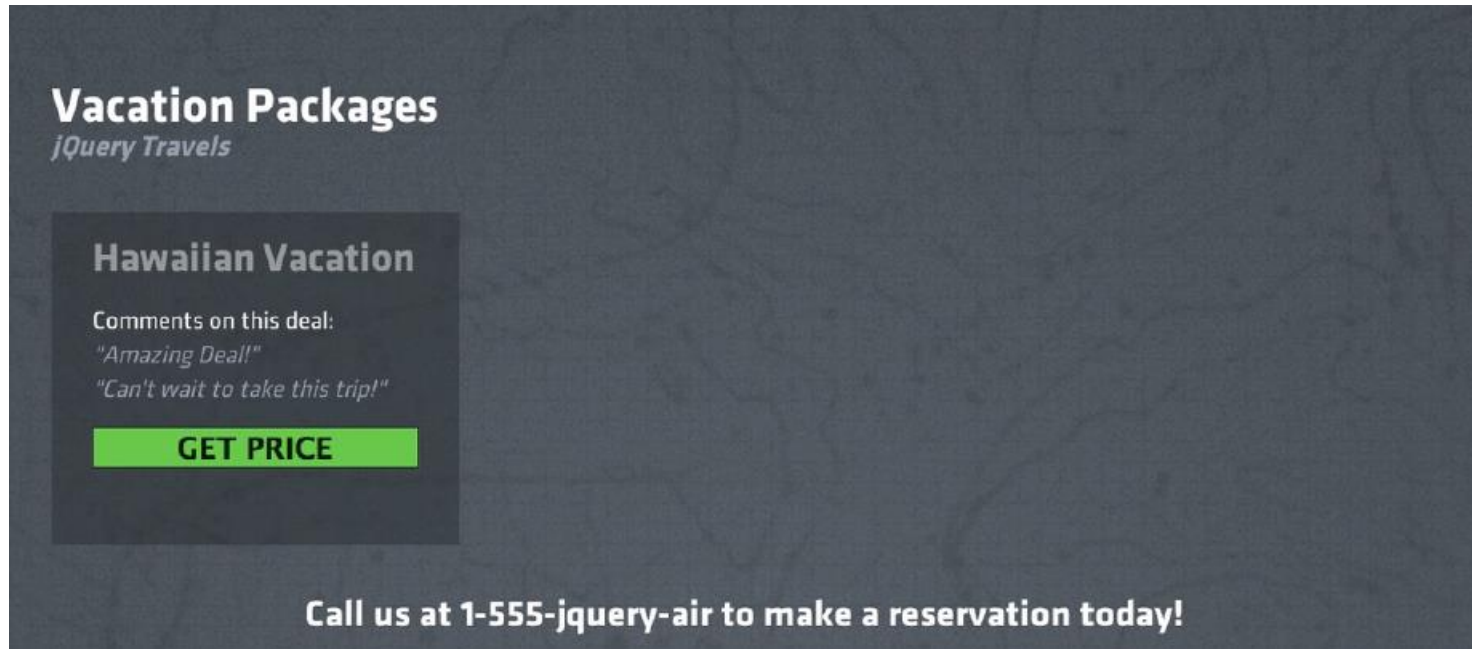
.children() 은 *find()*라는 조금 다릅니다.
.children() 은 직속 자식요소만 가져옵니다.
.parent() 처럼 체이닝으로 연결할 수 있습니다.



DOM 다루기

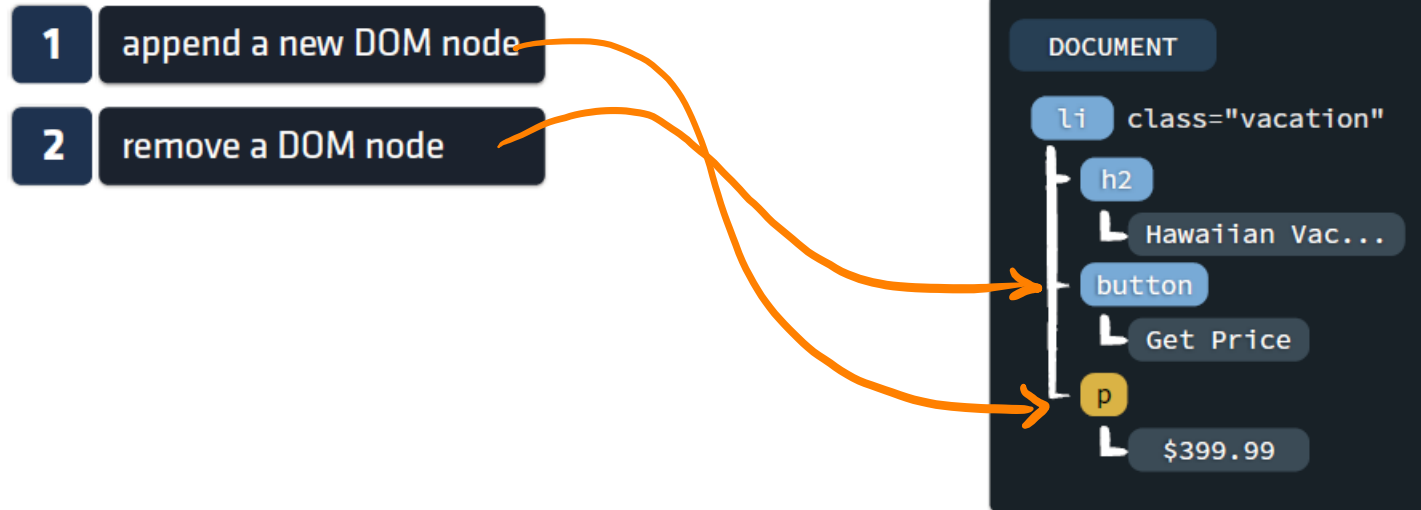
DOM 다루기

- DOM 다루기



DOM 다루기

- DOM 다루기



DOM 다루기

- DOM 다루기

```
$(document).ready(function() {  
  // create a <p> node with the price  
});
```

```
var price = "From $399.99";
```

```
var price = "<p>From $399.99</p>";
```

```
var price = $('<p>From $399.99</p>');
```

DOCUMENT

li class="vacation"

h2

Hawaiian Vac...

button

Get Price

p

\$399.99

요소를 동적으로 추가하기 위해서는 반드시 Element가 필요합니다.

DOM 다루기

- DOM 다루기

```
$(document).ready(function() {  
    var price = $('<p>From $399.99</p>');  
});
```

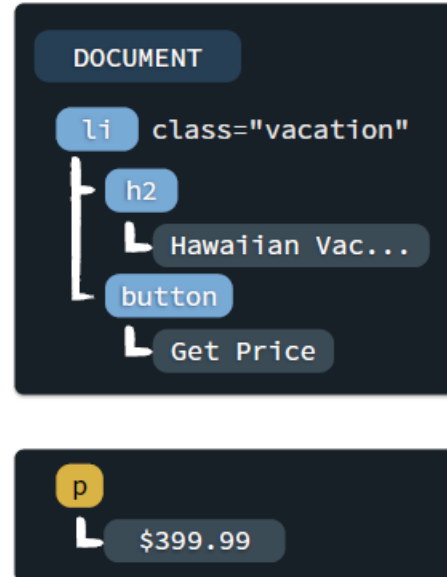
`.append(<element>)`

`.prepend(<element>)`

`.after(<element>)`

`.before(<element>)`

요소를 추가하기 위해서 하나씩 사용해보며, 결과를 알아봅니다!

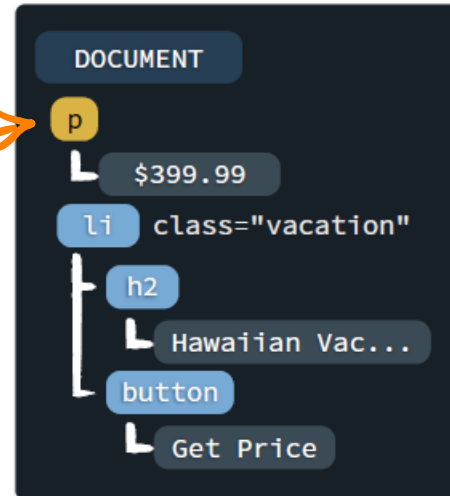


DOM 다루기

- DOM 다루기 : BEFORE

```
$(document).ready(function() {  
    var price = $('<p>From $399.99</p>');  
    $('.vacation').before(price);  
});
```

.vacation 요소의 이전에 *price* 요소를 추가합니다.

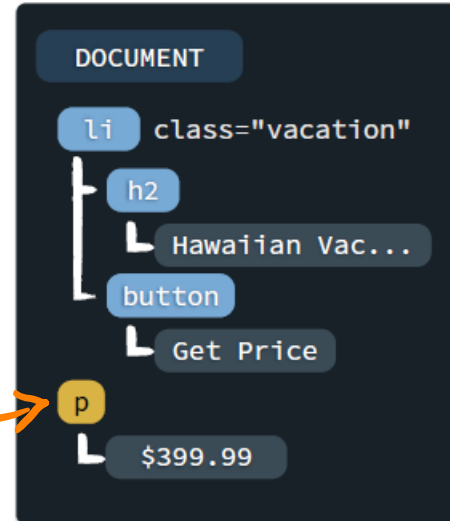


DOM 다루기

- DOM 다루기 : AFTER

```
$(document).ready(function() {  
  var price = $('<p>From $399.99</p>');  
  $('<.vacation>').after(price);  
});
```

*.vacation요소의 이후에 price 요소를 추가합니다.
우리가 만드려는 목표와는 다르므로 땡!*

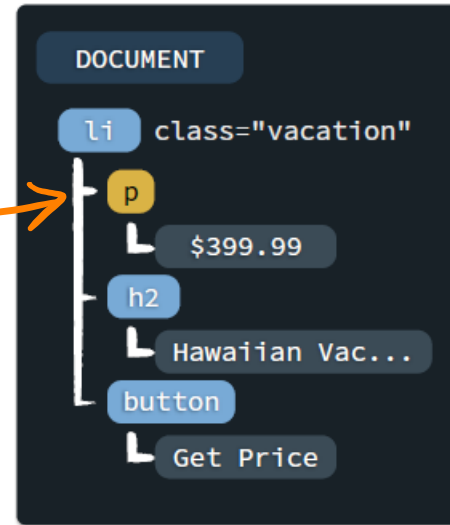


DOM 다루기

- DOM 다루기 : PREPEND

```
$(document).ready(function() {  
    var price = $('<p>From $399.99</p>');  
    $('<.vacation>').prepend(price);  
});
```

.vacation요소의 안쪽 가장 첫 부분에 요소를 추가합니다.
우리가 만드려는 목표와는 다르므로 땡!

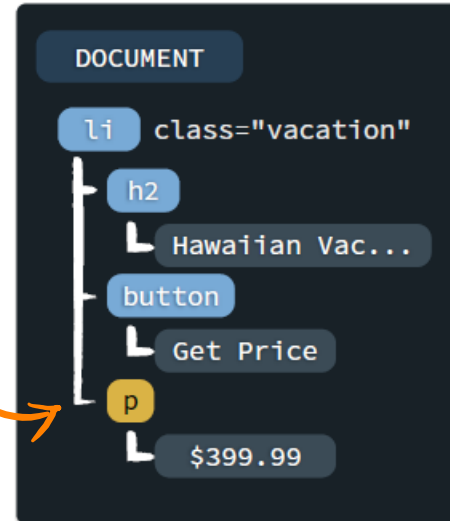


DOM 다루기

- DOM 다루기 : APPEND

```
$(document).ready(function() {  
    var price = $('<p>From $399.99</p>');  
    $('.vacation').append(price);  
});
```

.vacation요소의 안쪽 가장 마지막 부분에 요소를 추가합니다.
이제 button 요소를 지워야 합니다!

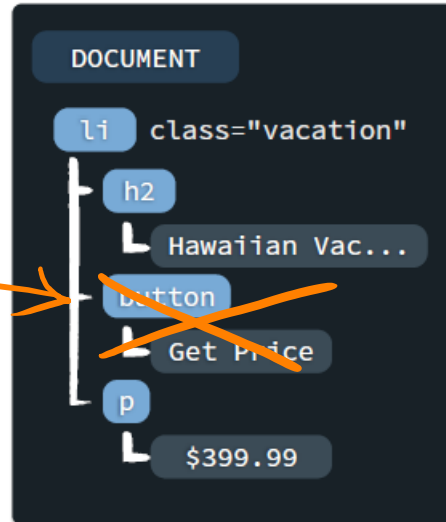


DOM 다루기

- DOM 다루기 : APPEND

```
$(document).ready(function() {  
  var price = $('<p>From $399.99</p>');  
  $('.vacation').append(price);  
  $('button').remove();  
});
```


button 요소를 삭제합니다.



이벤트와 함께 DOM 다루기

이벤트와 함께 DOM 다루기

- 클릭이벤트 만들어보기



Vacation Packages
jQuery Travels

Hawaiian Vacation

Comments on this deal:
"Amazing Deal!"
"Can't wait to take this trip!"

GET PRICE

GET PRICE 버튼을 클릭하면,
이 버튼은 삭제하고
From \$399.99 를 추가하려합니다.

Call us at 1-555-jquery-air to make a reservation today!

이벤트와 함께 DOM 다루기

- 클릭이벤트 만들어보기

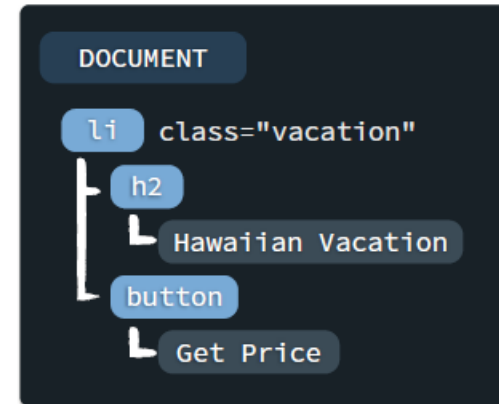
HTML DOM 이 준비되면 실행됩니다.

```
$(document).ready(function() {  
  // runs when the DOM is ready  
});
```

모든 button 객체를 대상으로 클릭이벤트가 발생한다면..

```
$('button').on('click', function() {  
  // runs when any button is clicked  
});
```

Button 요소가 클릭되면 실행됩니다.



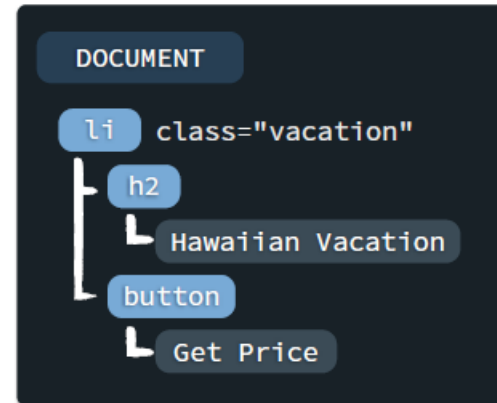
jQuery event method

```
.on(<event>, <event handler>)
```

이벤트와 함께 DOM 다루기

- 클릭이벤트 만들어보기

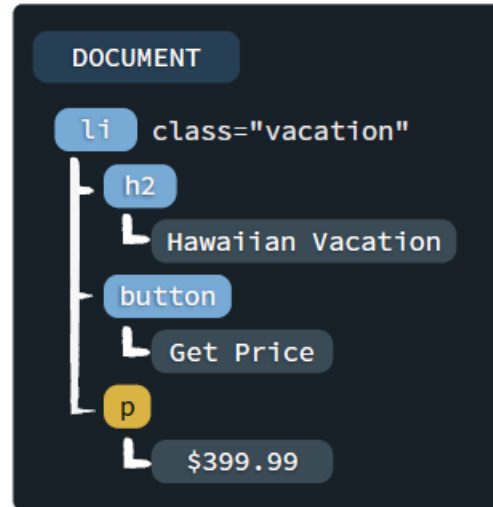
```
$(document).ready(function() {  
    $('button').on('click', function() {  
        // run this function on click  
    });  
});
```



이벤트와 함께 DOM 다루기

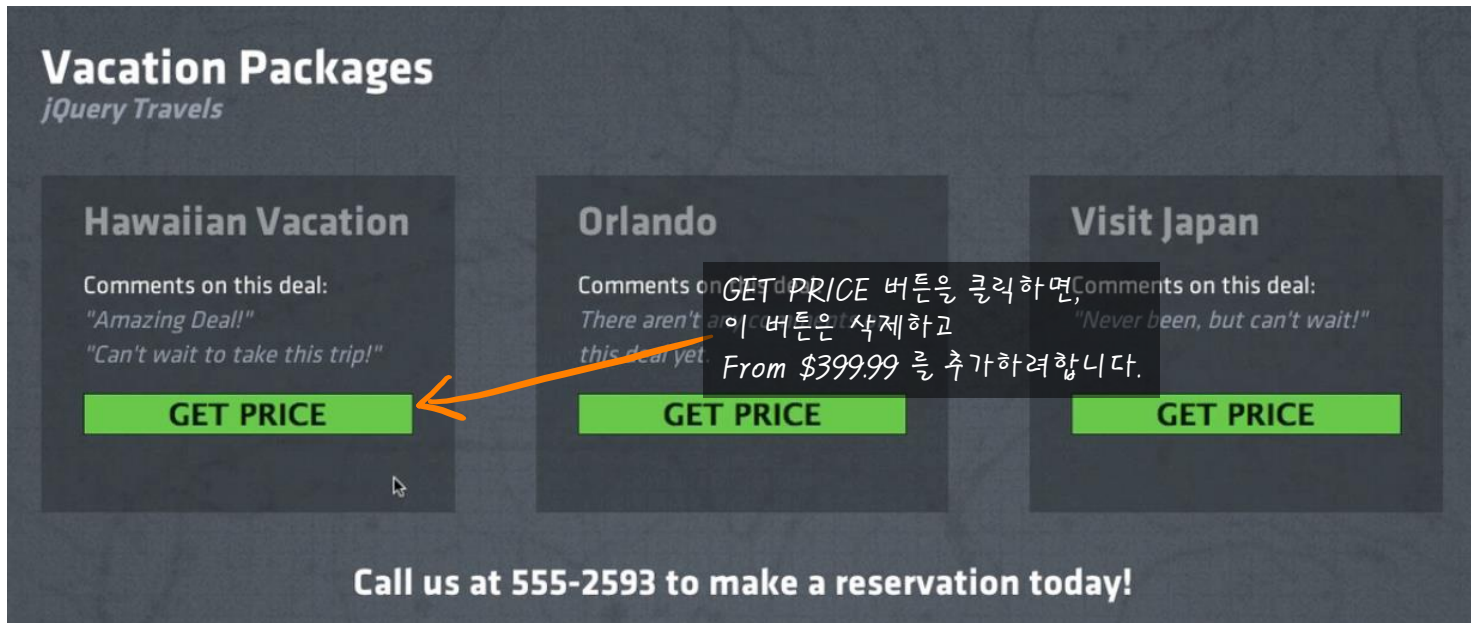
- 클릭이벤트 만들어보기

```
$(document).ready(function() {  
    $('button').on('click', function() {  
        var price = $('<p>From $399.99</p>');  
        $('.vacation').append(price);  
        $('button').remove();  
    });  
});
```



이벤트와 함께 DOM 다루기

- 클릭이벤트 만들어보기



이벤트와 함께 DOM 다루기

- 클릭이벤트 만들어보기

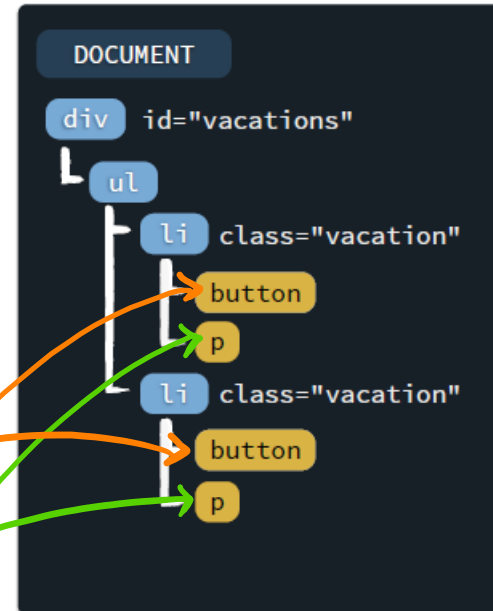
```
$(document).ready(function() {  
  $('button').on('click', function() {  
    var price = $('<p>From $399.99</p>');  
    $('.vacation').append(price);  
    $('button').remove();  
  });  
});
```

X?

클릭된 버튼만 지우고 싶은데...

버튼이 클릭되면, 삭제됩니다.

동시에 vacation 클래스에
p 엘리먼트가 추가됩니다.



이벤트와 함께 DOM 다루기

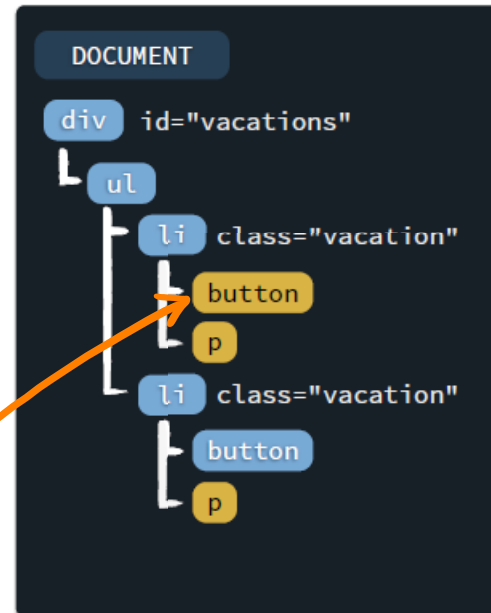
- 클릭이벤트 만들어보기

```
$(document).ready(function() {  
    $('button').on('click', function() {  
        var price = $('<p>From $399.99</p>');  
        $('.vacation').append(price);  
        $('button').remove();  
    });  
});
```

```
this.remove();
```

```
$(this).remove();
```

클릭한 버튼요소만 가져와 삭제합니다.

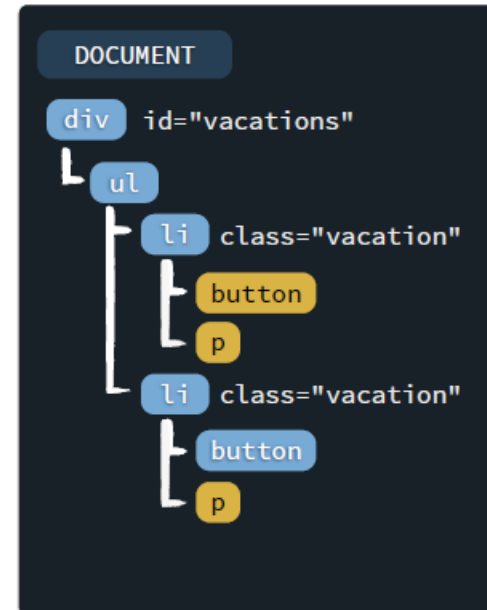


이벤트와 함께 DOM 다루기

- 클릭이벤트 만들어보기

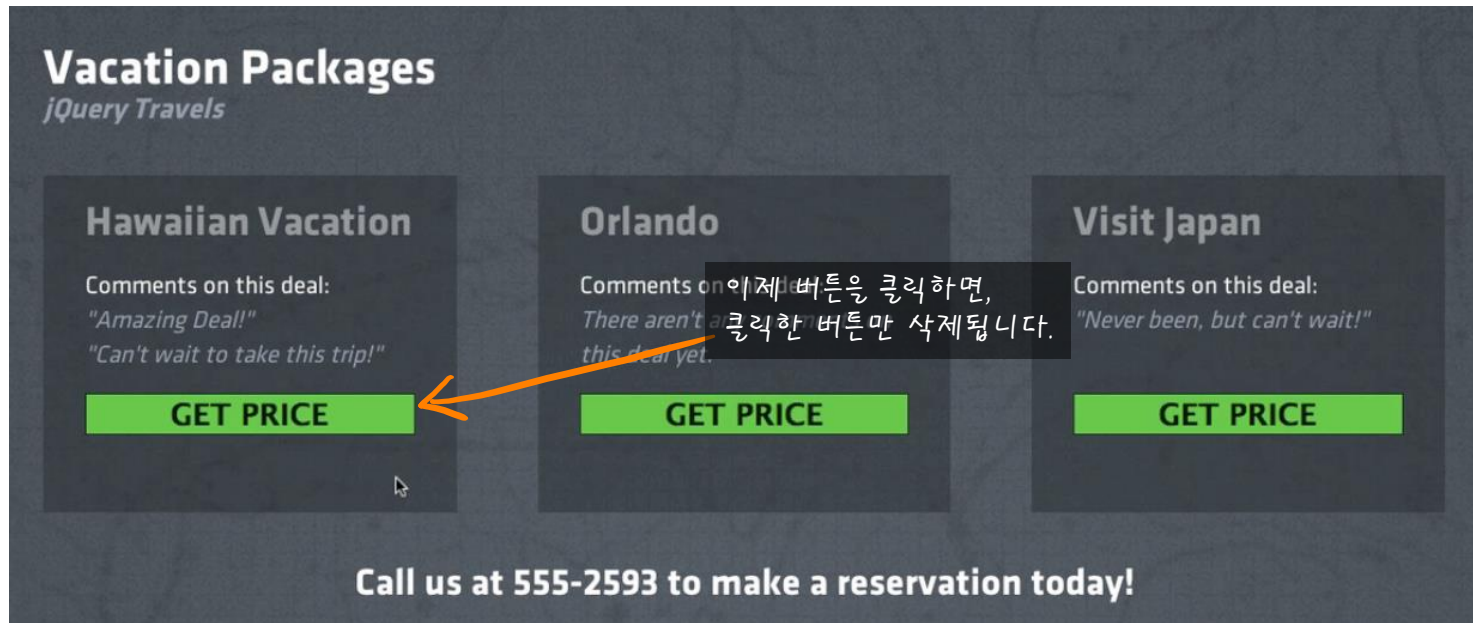
```
$(document).ready(function() {  
    $('button').on('click', function() {  
        var price = $('<p>From $399.99</p>');  
        $('.vacation').append(price);  
        $(this).remove();  
    });  
});
```

클릭한 버튼요소만 가져와 삭제합니다.



이벤트와 함께 DOM 다루기

- 클릭이벤트 만들어보기

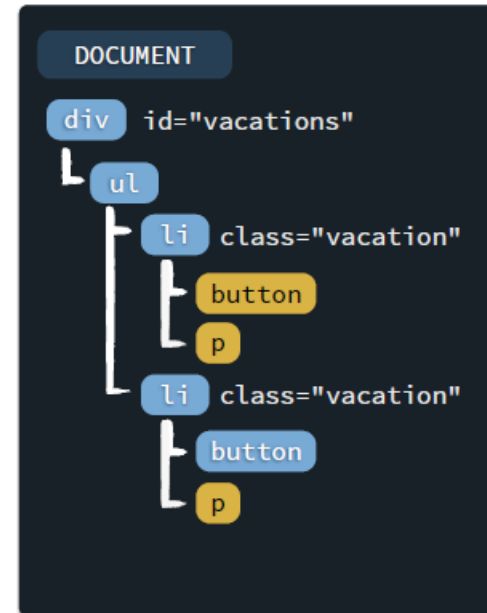


이벤트와 함께 DOM 다루기

- 클릭이벤트 만들어보기

```
$(document).ready(function() {  
    $('button').on('click', function() {  
        var price = $('<p>From $399.99</p>');  
        $('.vacation').append(price);  
        $(this).remove();  
    });  
});
```

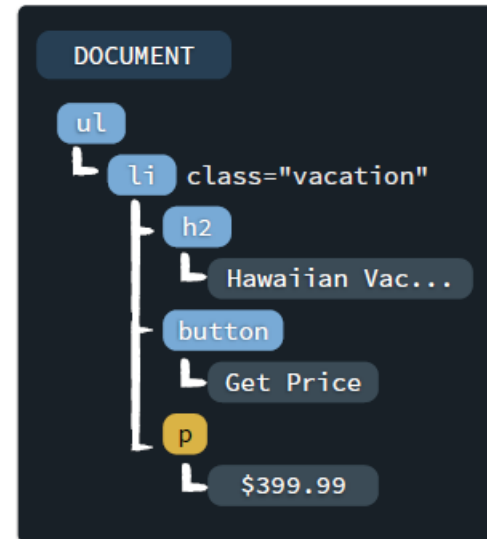
아직 존재하는 문제점은
클릭하지 않은 버튼 영역에도
p 엘리먼트가 추가된다는 것입니다.



이벤트와 함께 DOM 다루기

- 클릭이벤트 만들어보기

```
$(document).ready(function() {  
    $('button').on('click', function() {  
        var price = $('<p>From $399.99</p>');  
        $(this).after(price);  
        $(this).remove();  
    });  
});
```



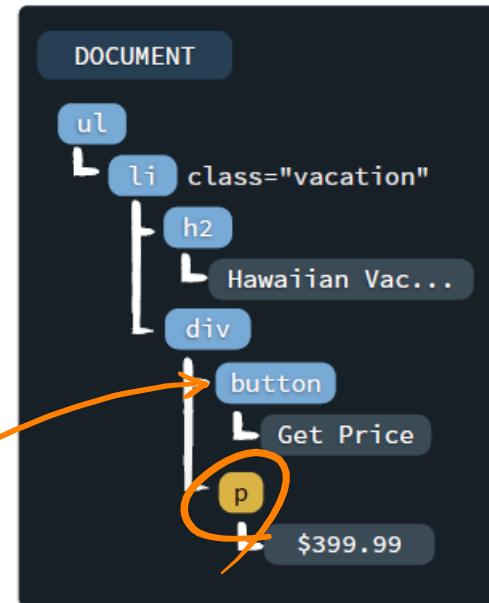
이벤트와 함께 DOM 다루기

- 클릭이벤트 만들어보기

```
$(document).ready(function() {  
  $('button').on('click', function() {  
    var price = $('<p>From $399.99</p>');  
    $(this).after(price);  
    $(this).remove();  
  });  
});
```

만약, 버튼이 div 아래로 이동한다면,
p 엘리먼트는 div 안쪽으로 추가됩니다.

p 엘리먼트를 li 안에 추가하는 방법은 무엇일까요?



이벤트와 함께 DOM 다루기

- 클릭이벤트 만들어보기

```
$(this).after(price);
```



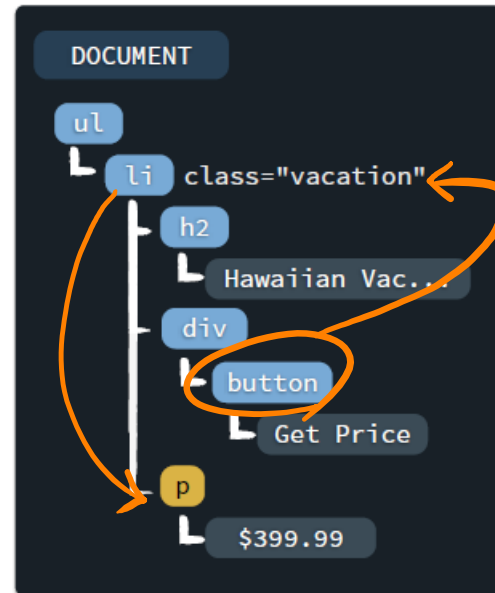
```
$(this).parent().parent().append(price);
```



```
$(this).parents('.vacation').append(price);
```



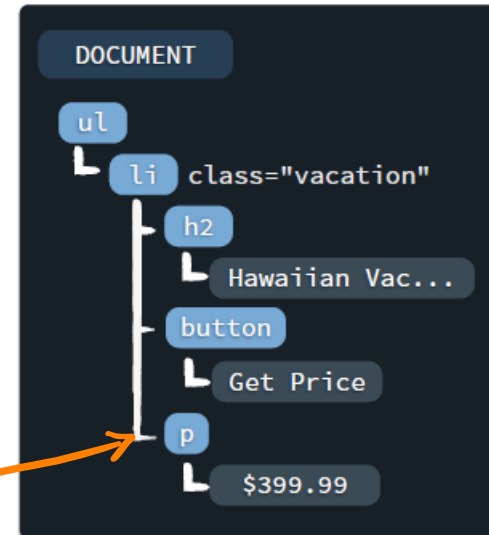
```
$(this).closest('.vacation').append(price);
```



이벤트와 함께 DOM 다루기

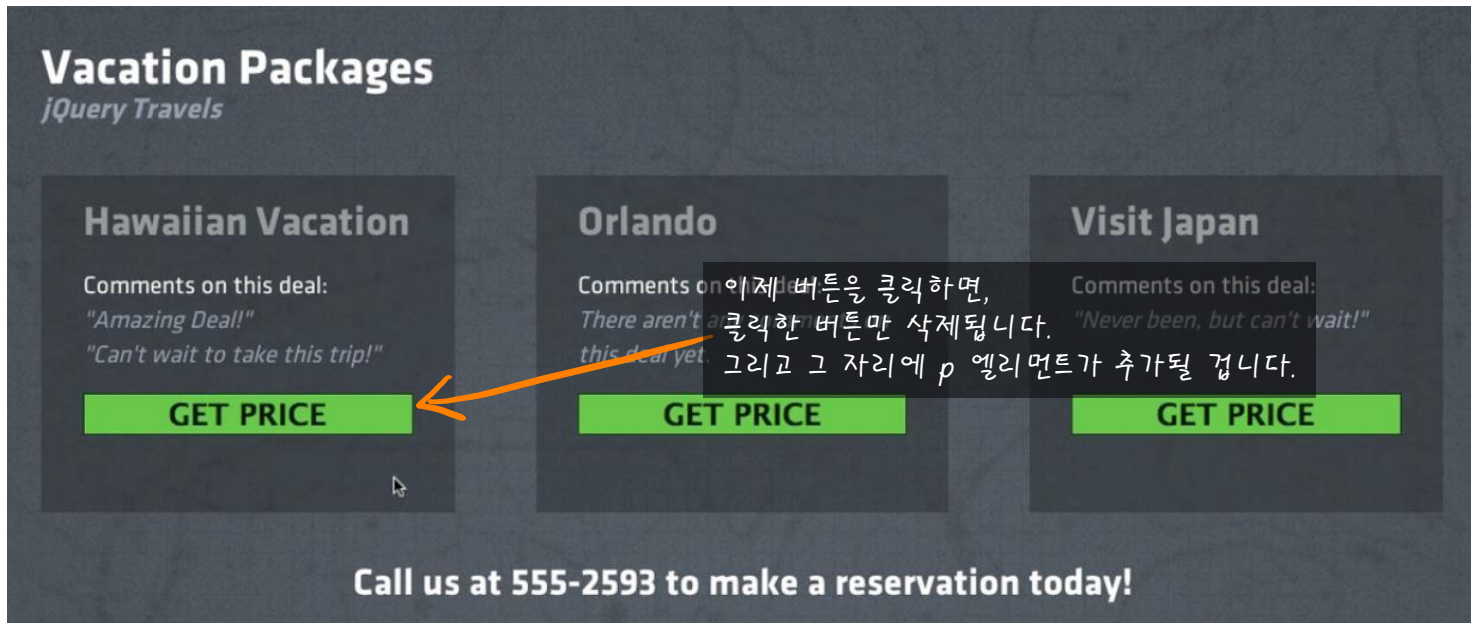
- 클릭이벤트 만들어보기

```
$(document).ready(function() {  
  $('button').on('click', function() {  
    var price = $('<p>From $399.99</p>');  
    $(this).closest('.vacation').append(price);  
    $(this).remove();  
  });  
});
```



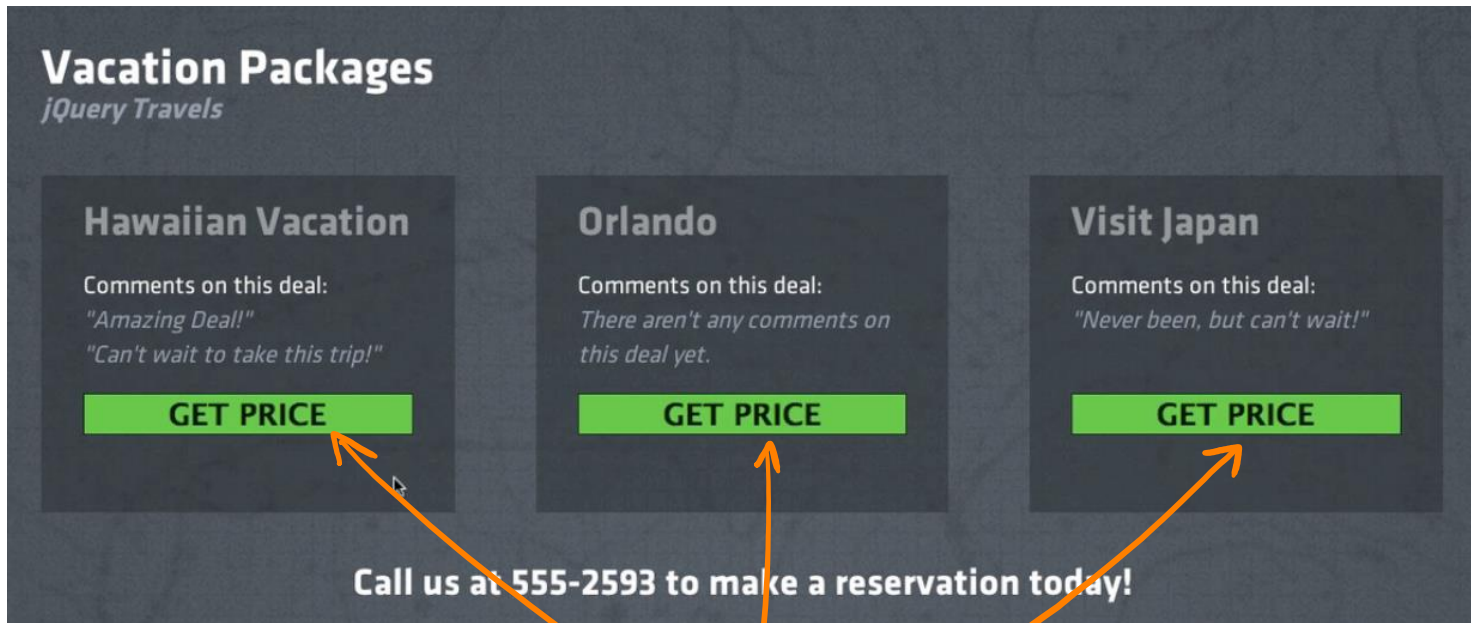
이벤트와 함께 DOM 다루기

- 클릭이벤트 만들어보기



이벤트와 함께 DOM 다루기

- 클릭이벤트 만들어보기



휴가 패키지 마다 서로 다른 금액을 책정하고 싶은 때,
어떻게 해야 할까요?

이벤트와 함께 DOM 다루기

- 클릭이벤트 만들어보기

```
<li class="vacation onsale" data-price='399.99'>  
  <h3>Hawaiian Vacation</h3>  
  <button>Get Price</button>  
  <ul class='comments'>  
    <li>Amazing deal!</li>  
    <li>Want to go!</li>  
  </ul>  
</li>
```

HTML5에 새로 추가된 속성입니다.
기본 속성이외에 다른 속성을 추가하고 싶을 때 사용합니다.
data-key= 'value' 형식으로 사용합니다.

```
$('.vacation').first().data('price');  
"399.99"
```

jQuery Object Methods

.data(<name>)

.data(<name>, <value>)

이벤트와 함께 DOM 다루기

- 클릭이벤트 만들어보기

```
$(document).ready(function() {  
  $('button').on('click', function() {  
    var price = $('<p>From $399.99</p>');  
    $(this).closest('.vacation').append(price);  
    $(this).remove();  
  });  
});
```

```
var amount = $(this).closest('.vacation').data('price');  
var price = $('<p>From $'+amount+'</p>');
```

이벤트와 함께 DOM 다루기

- 클릭이벤트 만들어보기

```
$(document).ready(function() {  
  $('button').on('click', function() {  
    var amount = $(this).closest('.vacation').data('price');  
    var price = $('<p>From $'+amount+'</p>');  
    $(this).closest('.vacation').append(price);  
    $(this).remove();  
  });  
});
```

이벤트와 함께 DOM 다루기

- 클릭이벤트 만들어보기

```
$(document).ready(function() {  
  $('button').on('click', function() {  
    var amount = $(this).closest('.vacation').data('price');  
    var price = $('<p>From $'+amount+'</p>');  
    $(this).closest('.vacation').append(price);  
    $(this).remove();  
  });  
});
```

같은 연산이 두 번 반복되고 있습니다.
요소의 검색(탐색)은 생각보다 복잡한 과정을 거치는 작업입니다.
이렇게 계속 반복되다 보면, 성능에 문제가 발생할 수 있습니다.
또한, 이렇게 여러 번 사용하는 것은 관리측면에서 좋지 못합니다.

이벤트와 함께 DOM 다루기

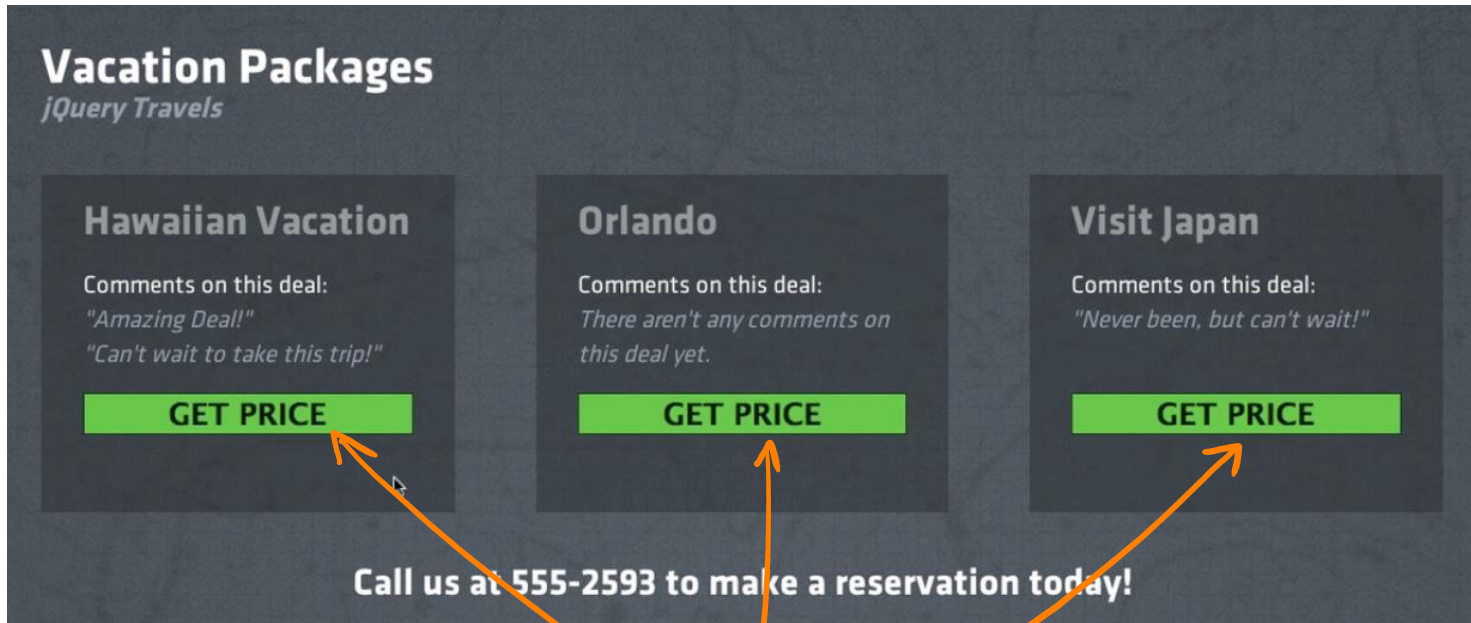
- 클릭이벤트 만들어보기

```
$(document).ready(function() {  
  $('button').on('click', function() {  
    var vacation = $(this).closest('.vacation');  
    var amount = vacation.data('price');  
    var price = $('<p>From $'+amount+'</p>');  
    vacation.append(price);  
    $(this).remove();  
  });  
});
```

요소를 가져오는 과정은 한번이면 충분합니다.

이벤트와 함께 DOM 다루기

- 클릭이벤트 만들어보기



이제 각 휴가 패키지는 서로 다른 금액을 보여주게 되었습니다.

Event가 적용되는 대상

Event가 적용되는 대상

- jQuery는 Event 를 효율적으로 제어할 수 있도록 해줌.
- Event 제어의 대상이 되는 객체는 `$(document).ready();` 가 호출되기 전에 객체화 된 Element 들.
- jQuery를 통해 동적으로 추가된(`$(document).ready()` 이후 추가된) Element 들에 대해서는 Event를 할당할 수 없게 됨.

Event가 적용되는 대상

- 클릭이벤트 만들어보기

Button 이 새로 추가된 객체라면, click 이벤트는 동작하지 않습니다.

```
$(document).ready(function() {  
  $('button').on('click', function() {  
    ...  
  });  
});
```

```
$('.vacation button').on('click', function() {});
```

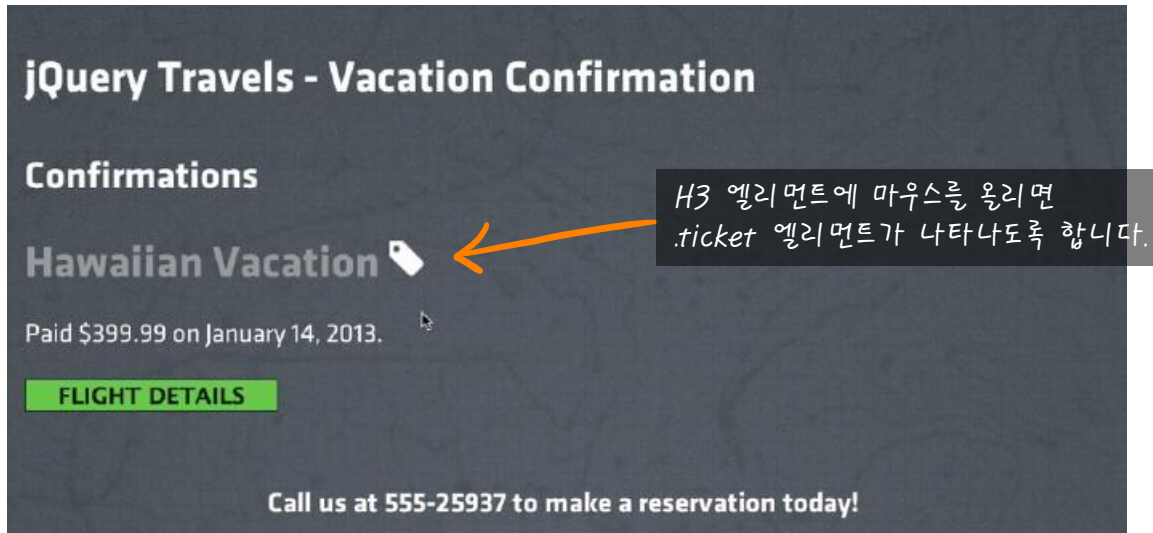
```
$('.vacation').on('click', 'button', function() {});
```

새롭게 추가된 객체는 조금 다른 on 이벤트를 사용합니다.
.vacation 하위의 button 객체에 click 이벤트를 할당합니다.
새롭게 추가된 객체라도 click 이벤트를 할당할 수 있습니다.
이 방법은 모든 객체에 접근할 수 있습니다.

Mouse Event

이벤트와 함께 DOM 다루기

- 마우스 이벤트 만들어보기



이벤트와 함께 DOM 다루기

- 마우스 이벤트 만들어보기

```
$(document).ready(function() {  
  $('.confirmation').on('click', 'button', function() {  
    $(this).closest('.confirmation').find('.ticket').slideDown();  
  });  
  $('.confirmation').on('?', 'h3', function() {  
    $(this).closest('.confirmation').find('.ticket').slideDown();  
  });  
});
```



마우스를 요소에 올렸을 때

click

focusin

mousedown

mousemove

mouseover

mouseenter



dblclick

focusout

mouseup

mouseout

mouseleave

이벤트와 함께 DOM 다루기

- 마우스 이벤트 만들어보기

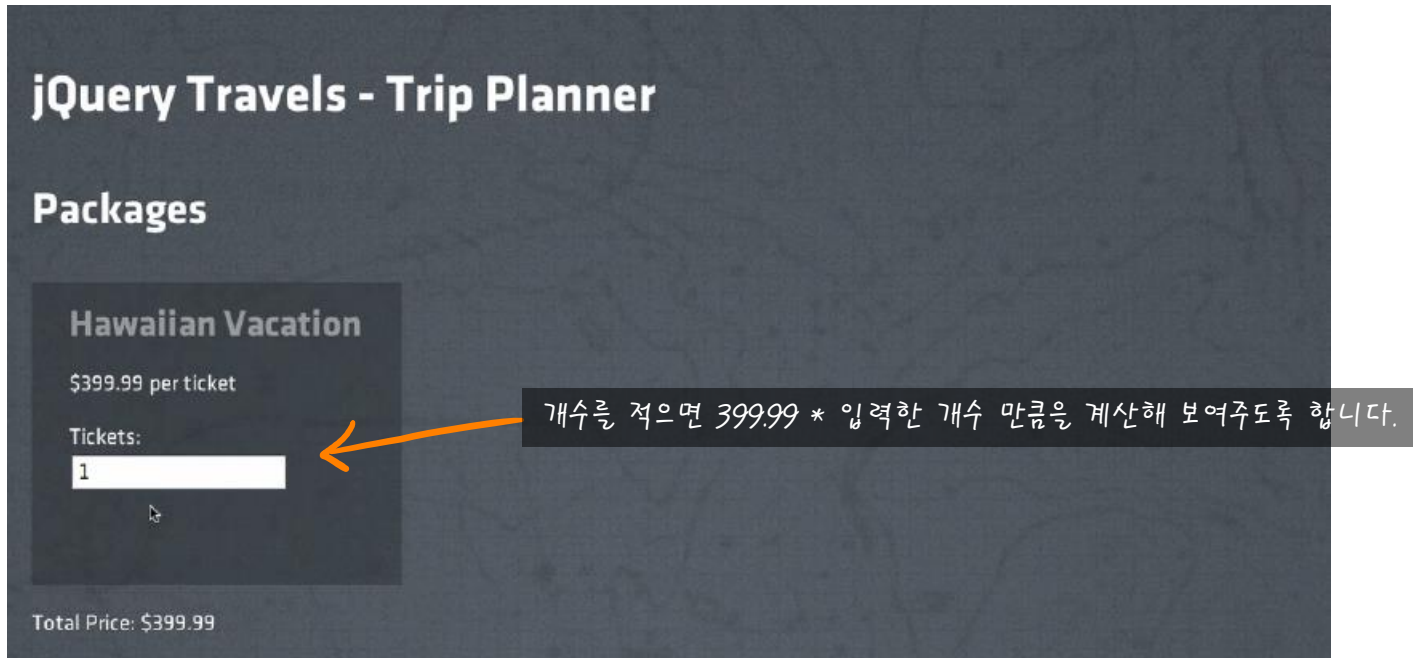
```
$(document).ready(function() {  
  $('.confirmation').on('click', 'button', function() {  
    $(this).closest('.confirmation').find('.ticket').slideDown();  
  });  
  $('.confirmation').on('mouseenter', 'h3', function() {  
    $(this).closest('.confirmation').find('.ticket').slideDown();  
  });  
});
```

이제 마우스를 올리면, *ticket* 이 슬라이드 되며 아래로 내려옵니다.

Keyboard Event

이벤트와 함께 DOM 다루기

- 키보드 이벤트 만들어보기



The screenshot displays a web application titled "jQuery Travels - Trip Planner". Under the "Packages" section, there is a "Hawaiian Vacation" package. The price is listed as "\$399.99 per ticket". Below this, there is a "Tickets:" label and a text input field containing the number "1". An orange arrow points from a text box to this input field. The text box contains the Korean text: "개수를 적으면 399.99 * 입력한 개수 만큼은 계산해 보여주도록 합니다." (When you enter the number, it will calculate and show the amount based on the entered number multiplied by 399.99). At the bottom left of the package section, it says "Total Price: \$399.99".

jQuery Travels - Trip Planner

Packages

Hawaiian Vacation

\$399.99 per ticket

Tickets:

1

Total Price: \$399.99

개수를 적으면 399.99 * 입력한 개수 만큼은 계산해 보여주도록 합니다.

이벤트와 함께 DOM 다루기

- 키보드 이벤트 만들어보기

만약 이 값이 변경된다면...

```
<div class="vacation" data-price='399.99'>
  <h3>Hawaiian Vacation</h3>
  <p>$399.99 per ticket</p>
  <p>
    Tickets:
    <input type='number' class='quantity' value='1' />
  </p>
</div>
<p>Total Price: $<span id='total'>399.99</span></p>
```

이 값을 갱신하도록 합니다.

이벤트와 함께 DOM 다루기

- 키보드 이벤트 만들어보기

```
$(document).ready(function() {  
  $('.vacation').on('?', '.quantity', function() {  
  });  
});
```

Keyboard Events

keypress

keydown

keyup



Form Events

blur

select

change

focus

submit

키를 눌렀다 떼는 순간 이벤트가 발생합니다.

이벤트와 함께 DOM 다루기

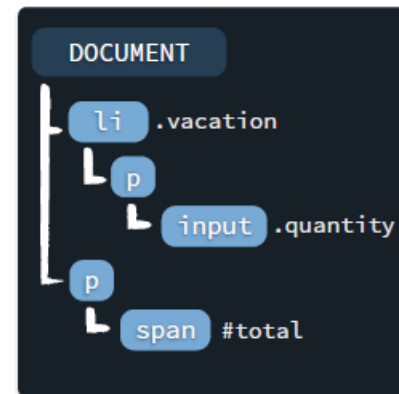
- 키보드 이벤트 만들어보기

```
$(document).ready(function() {  
  $('.vacation').on('keyup', '.quantity', function() {  
    // Get the price for this vacation  
    // Get the quantity entered  
    // Set the total to price * quantity  
  });  
});
```

```
var price = $(this).closest('.vacation').data('price');  
↳ '399.99'
```

Returns price as a string

```
var price = +$(this).closest('.vacation').data('price');  
↳ 399.99
```



이벤트와 함께 DOM 다루기

- 키보드 이벤트 만들어보기

```
$(document).ready(function() {  
  $('.vacation').on('keyup', '.quantity', function() {  
    var price = +$(this).closest('.vacation').data('price');  
    // Get the quantity entered  
    // Set the total to price * quantity  
  });  
});
```

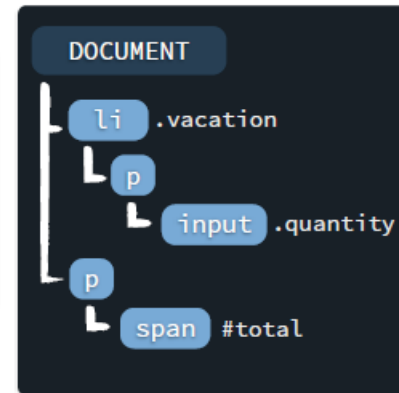
```
var quantity = this.val();
```



```
var quantity = $(this).val();  
↳ '2'
```



```
var quantity = +$(this).val();  
↳ 2
```

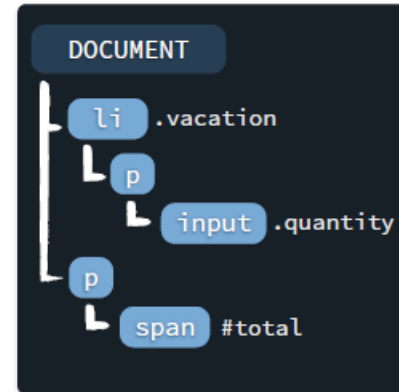


이벤트와 함께 DOM 다루기

- 키보드 이벤트 만들어보기

```
$(document).ready(function() {  
  $('.vacation').on('keyup', '.quantity', function() {  
    var price = +$(this).closest('.vacation').data('price');  
    var quantity = +$(this).val();  
    // Set the total to price * quantity  
  });  
});
```

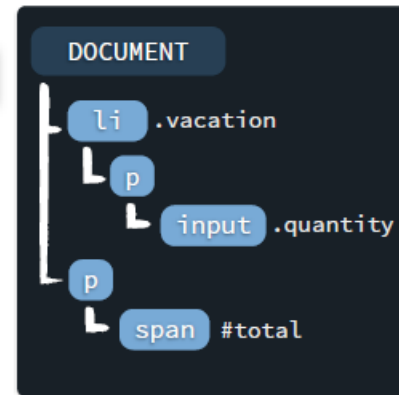
```
$('#total').text(price * quantity);
```



이벤트와 함께 DOM 다루기

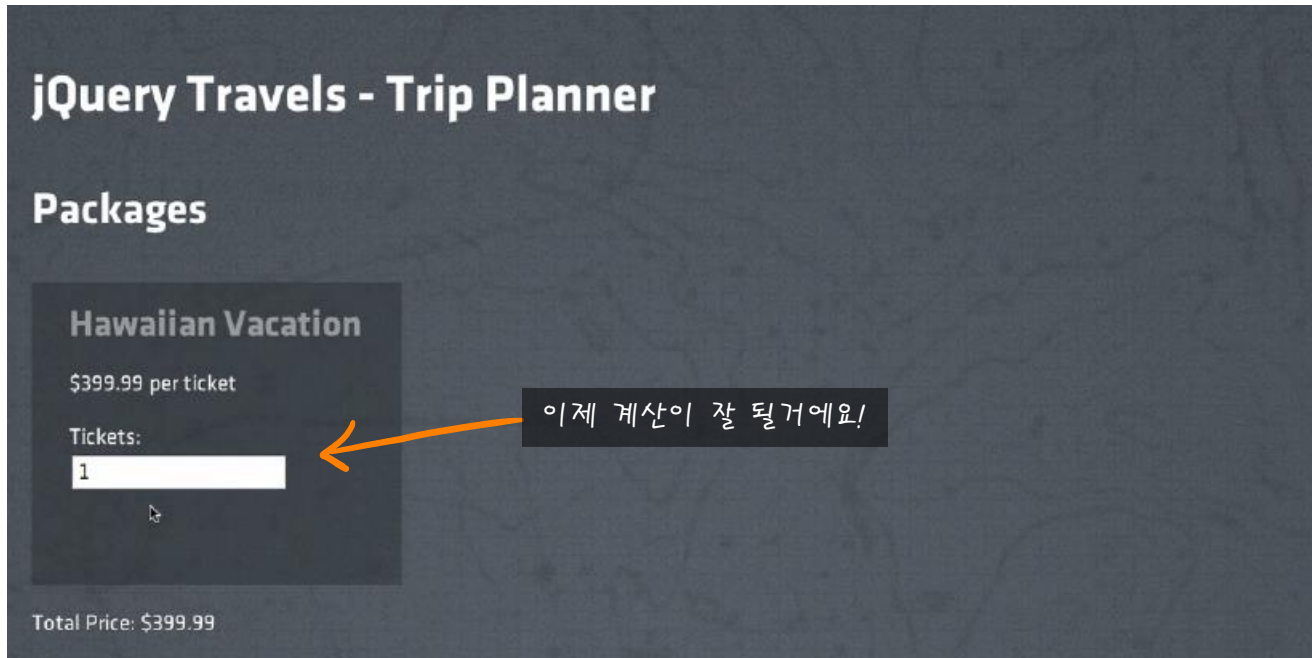
- 키보드 이벤트 만들어보기

```
$(document).ready(function() {  
  $('.vacation').on('keyup', '.quantity', function() {  
    var price = +$(this).closest('.vacation').data('price');  
    var quantity = +$(this).val();  
    $('#total').text(price * quantity);  
  });  
});
```



이벤트와 함께 DOM 다루기

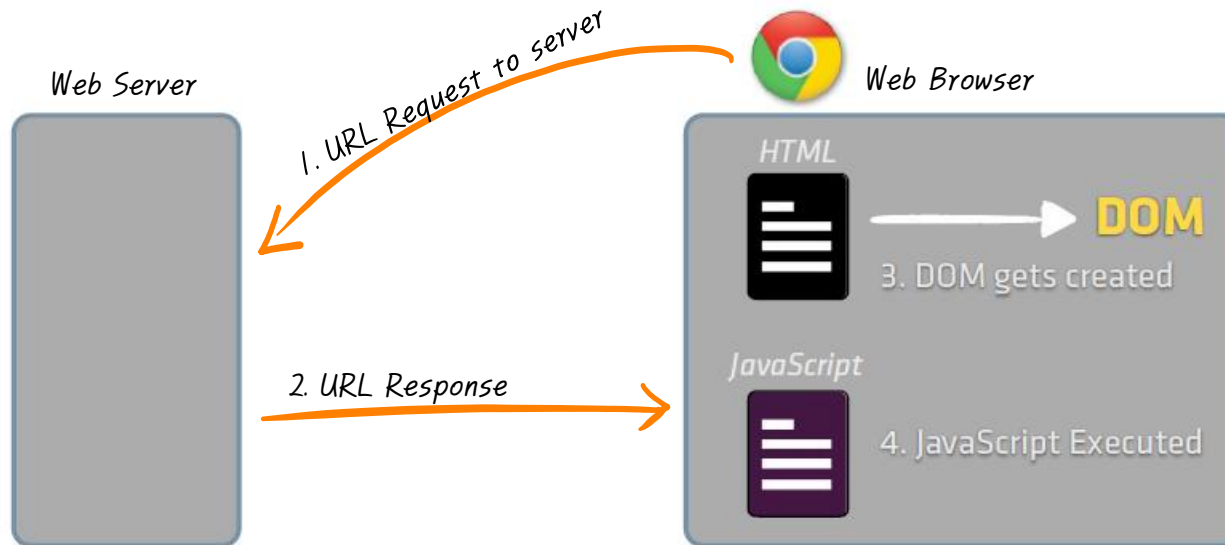
- 키보드 이벤트 만들어보기



AJAX

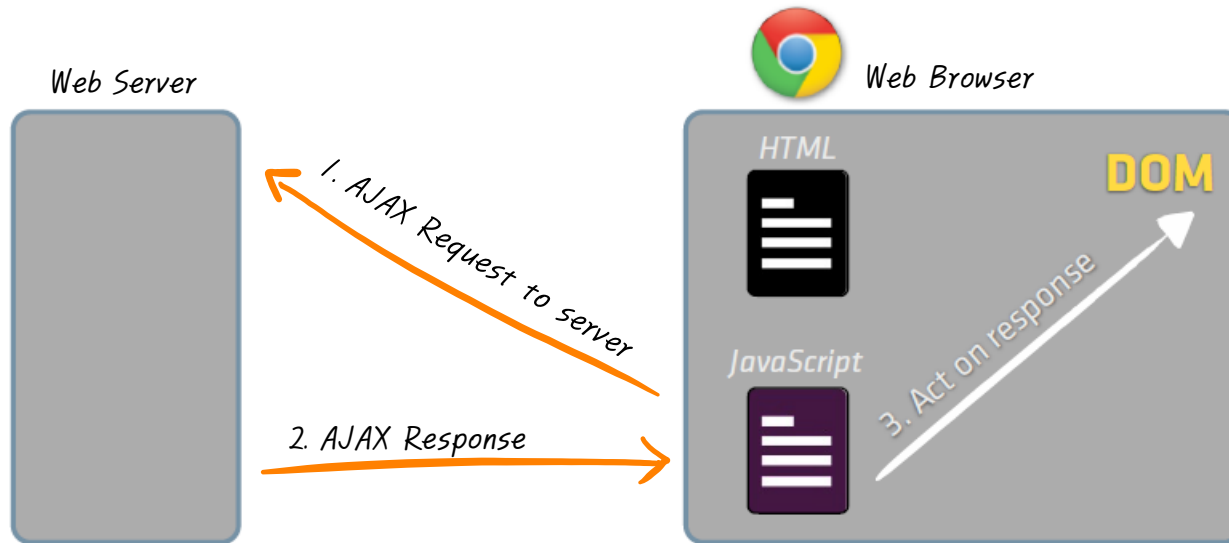
AJAX

- 일반적인 HTTP 요청



AJAX

- AJAX 요청



첫 번째 AJAX Application

첫 번째 AJAX Application

- AJAX 요청

```
$('.confirmation').on('click', 'button', function(){  
    $.ajax('http://example.org/confirmation.html', {  
        success: function(response) {  
            $('.ticket').html(response).slideDown();  
        }  
    });  
});
```

Ajax 요청을 보낼 주소를 적습니다.
이 주소는 아래의 웹페이지를 호출합니다.

Ajax Method

```
$.ajax(url[, settings])
```

호출이 성공적으로 이루어졌을때 실행됩니다.

응답의 Body 가 response로 전달됩니다.

<http://example.org/confirmation.html>

```
<div> ...  
  <strong>Boarding Pass: </strong>  
  <a href='#' class='view-boarding-pass'>View Boarding Pass</a>  
  <img src='ticket.png' alt='Your boarding pass' class='boarding-pass' />  
</div>
```

첫 번째 AJAX Application

- AJAX 요청


절대주소를 사용할 수도 있지만,
지금처럼 상대주소를 사용할 수도 있습니다.

```
$('.confirmation').on('click', 'button', function(){  
  $.ajax('confirmation.html', {  
    success: function(response) {  
      $('.ticket').html(response).slideDown();  
    }  
  });  
});
```

첫 번째 AJAX Application

- Short Cut 표현식

```
$.ajax('confirmation.html', {  
  success: function(response) {  
    $('.ticket').html(response).slideDown();  
  }  
});
```



```
$.get('confirmation.html', function(response) {  
  $('.ticket').html(response).slideDown();  
});
```


\$.get() 과 \$.post() 로 사용할 수 있습니다!

파라미터 전달하기

파라미터 전달하기

- 파라미터 전달하기

여기에 파라미터를 보내려면...



```
$.ajax('confirmation.html', {  
  success: function(response) {  
    $('.ticket').html(response).slideDown();  
  }  
});
```

```
confirmation.html?confNum=1234
```

이렇게 해야할까요?

뭐... 틀리진 않았어요..

파라미터 전달하기

- 파라미터 전달하기

```
$.ajax('confirmation.html?confNum=1234', {  
  success: function(response) {  
    $('.ticket').html(response).slideDown();  
  }  
});
```



훨씬 더 구조적이에요. 알아보기도 편합니다. 게다가, 여러 파라미터를 보내기에도 수월해요.

```
$.ajax('confirmation.html', {  
  success: function(response) {  
    $('.ticket').html(response).slideDown();  
  },  
  data: { "confNum": 1234 }  
});
```

파라미터 전달하기

- 파라미터 전달하기

```
<div class='ticket' data-confNum='1234'>
```

jQuery 요소를 파라미터로 보낼수도 있습니다.

```
$.ajax('confirmation.html', {  
  success: function(response) {  
    $('.ticket').html(response).slideDown();  
  },  
  data: { "confNum": $(".ticket").data("confNum")  
  }  
});
```

URL로 보내는 것과 같아지는거죠!

```
confirmation.html?confNum=1234
```

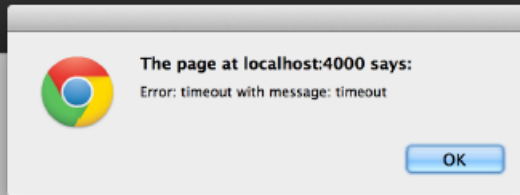

AJAX Options

AJAX Options

- Error 처리하기

```
$('.confirmation').on('click', 'button', function(){  
    $.ajax('confirmation.html', {  
        success: function(response) {  
            $('.ticket').html(response).slideDown();  
        },  
        error: function(request, errorType, errorMessage) {  
            alert('Error: ' + errorType + ' with message: ' + errorMessage);  
        }  
    });  
});
```

서버에서 *timeout*, *abort*,
server error 가 발생하면
이 메소드가 실행됩니다.



에러가 발생하면 이렇게 *Alert* 이 실행됩니다.

AJAX Options

- Timeout 설정하기

```
$('.confirmation').on('click', 'button', function(){  
    $.ajax('confirmation.html', {  
        success: function(response) { ... },  
        error: function(request, errorType, errorMessage) { ... },  
        timeout: 3000  
    });  
});
```

3000 은 3초입니다.
1초는 1000으로 표현합니다.
이런 초를 *Milliseconds*라고 해요.

AJAX Options

- Ajax 전후 처리하기

```
$('.confirmation').on('click', 'button', function(){
  $.ajax('confirmation.html', {
    success: function(response) { ... },
    error: function(request, errorType, errorMessage) { ... },
    timeout: 3000,
    beforeSend: function() {
      $('.confirmation').addClass('is-loading');
    },
    complete: function() {
      $('.confirmation').removeClass('is-loading');
    }
  });
});
```

← Ajax Request를 보내기 전에 실행합니다.

← Ajax Request를 보내고 응답을 받으면 실행합니다. 성공, 실패를 모두 응답으로 간주합니다.

Animation

Animation

- CSS로 DOM Animation 하기



Animation

- CSS로 DOM Animation 하기

```
$(document).ready(function() {  
    $(".box").css("opacity", 0.25);  
});
```

화면이 로드 되는 순간
투명도가 25%로 조정됩니다.

투명도가 서서히 줄어든게 만드려면...

```
<body class="boxAnimate">  
    <div class="box inline">  
    </div>  
    <div class="box inline">  
    </div>  
    <div class="box inline">  
    </div>  
    <div class="box inline">  
    </div>  
    <div class="box inline">  
    </div>  
    <div class="box inline">  
    </div>  
    <div class="box inline">  
    </div>  
</body>
```

Animation

- CSS로 DOM Animation 하기

```
$(document).ready(function() {  
  $(".box").animate({  
    "opacity": 0.25,  
  });  
});
```

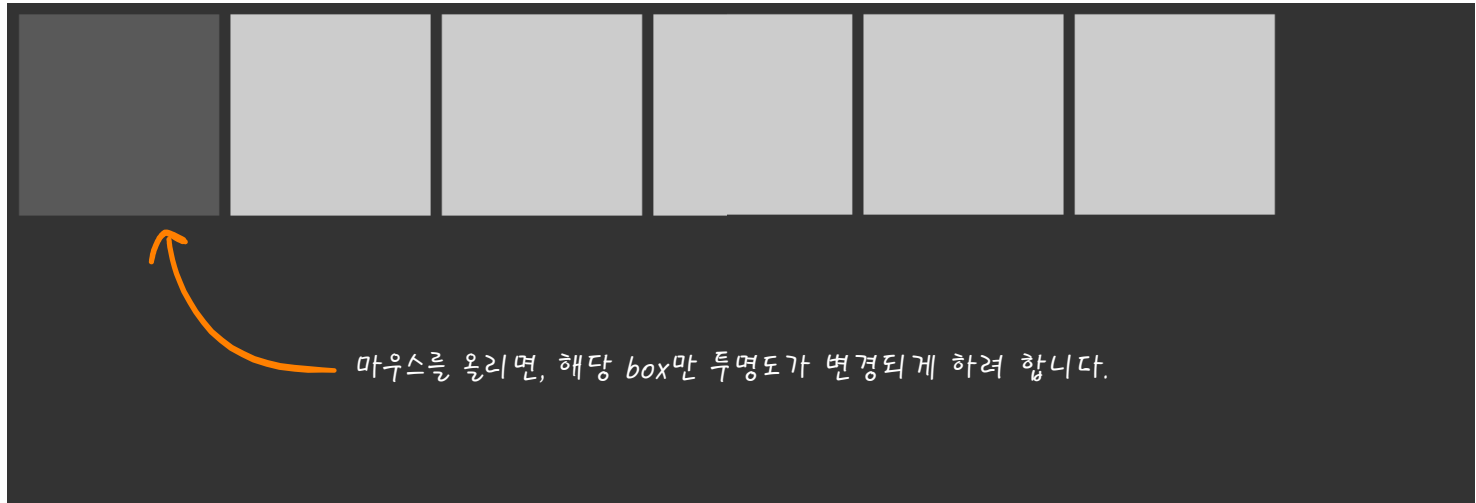
CSS가 아닌 *animate* 가 필요합니다.

지정된 CSS로 서서히 변화됩니다.

```
<body class="boxAnimate">  
  <div class="box inline">  
  </div>  
  <div class="box inline">  
  </div>  
  <div class="box inline">  
  </div>  
  <div class="box inline">  
  </div>  
  <div class="box inline">  
  </div>  
  <div class="box inline">  
  </div>  
</body>
```


Animation

- CSS로 DOM Animation 하기 – 이벤트와 Animation



Animation

- CSS로 DOM Animation 하기

```
$(document).ready(function() {  
    $(".box").on("mouseover", function() {  
    });  
});
```

```
$(".box").animate({  
    "opacity": 0.25,  
});
```

```
<body class="boxAnimate">  
    <div class="box inline">  
    </div>  
    <div class="box inline">  
    </div>  
    <div class="box inline">  
    </div>  
    <div class="box inline">  
    </div>  
    <div class="box inline">  
    </div>  
    <div class="box inline">  
    </div>  
</body>
```

Animation

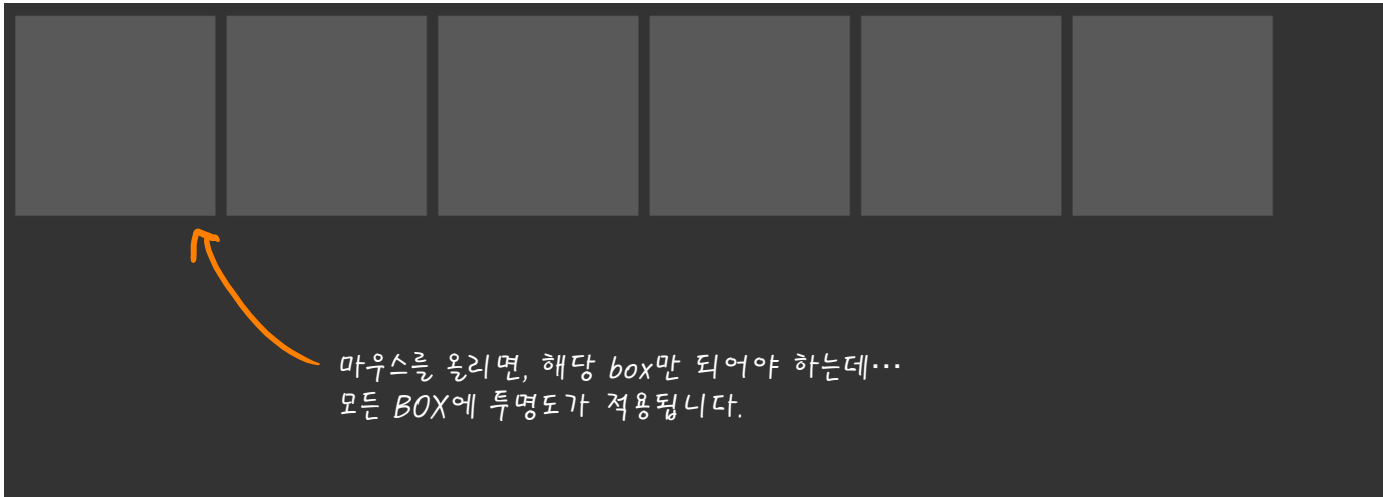
- CSS로 DOM Animation 하기

```
$(document).ready(function() {  
    $(".box").on("mouseover", function() {  
        $(".box").animate({  
            "opacity": 0.25,  
        });  
    });  
});
```

```
<body class="boxAnimate">  
    <div class="box inline">  
    </div>  
    <div class="box inline">  
    </div>  
    <div class="box inline">  
    </div>  
    <div class="box inline">  
    </div>  
    <div class="box inline">  
    </div>  
    <div class="box inline">  
    </div>  
</body>
```

Animation


- CSS로 DOM Animation 하기



Animation

- CSS로 DOM Animation 하기

```
$(document).ready(function() {  
    $(".box").on("mouseover", function() {  
        $(this).animate({  
            "opacity": 0.25,  
        });  
    });  
});
```

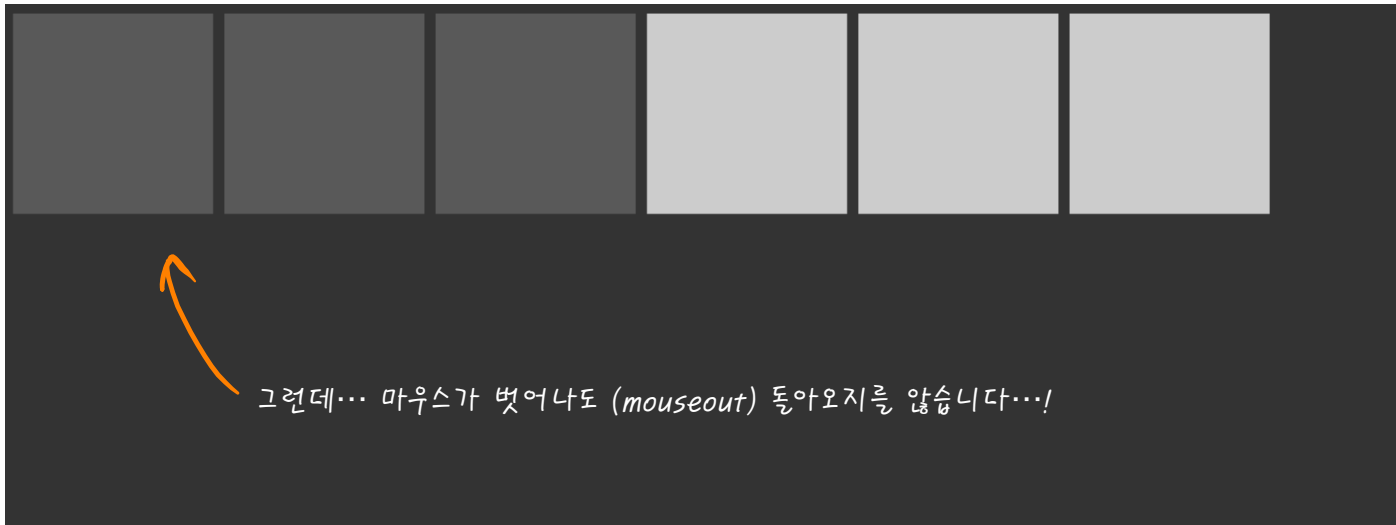


*This*를 선택하면 마우스이벤트가 발생한
Box 만 애니메이션이 적용됩니다!

```
<body class="boxAnimate">  
    <div class="box inline">  
    </div>  
    <div class="box inline">  
    </div>  
    <div class="box inline">  
    </div>  
    <div class="box inline">  
    </div>  
    <div class="box inline">  
    </div>  
    <div class="box inline">  
    </div>  
</body>
```

Animation

- CSS로 DOM Animation 하기



Animation

- CSS로 DOM Animation 하기

마우스가 올라가면 이벤트가 발생합니다!

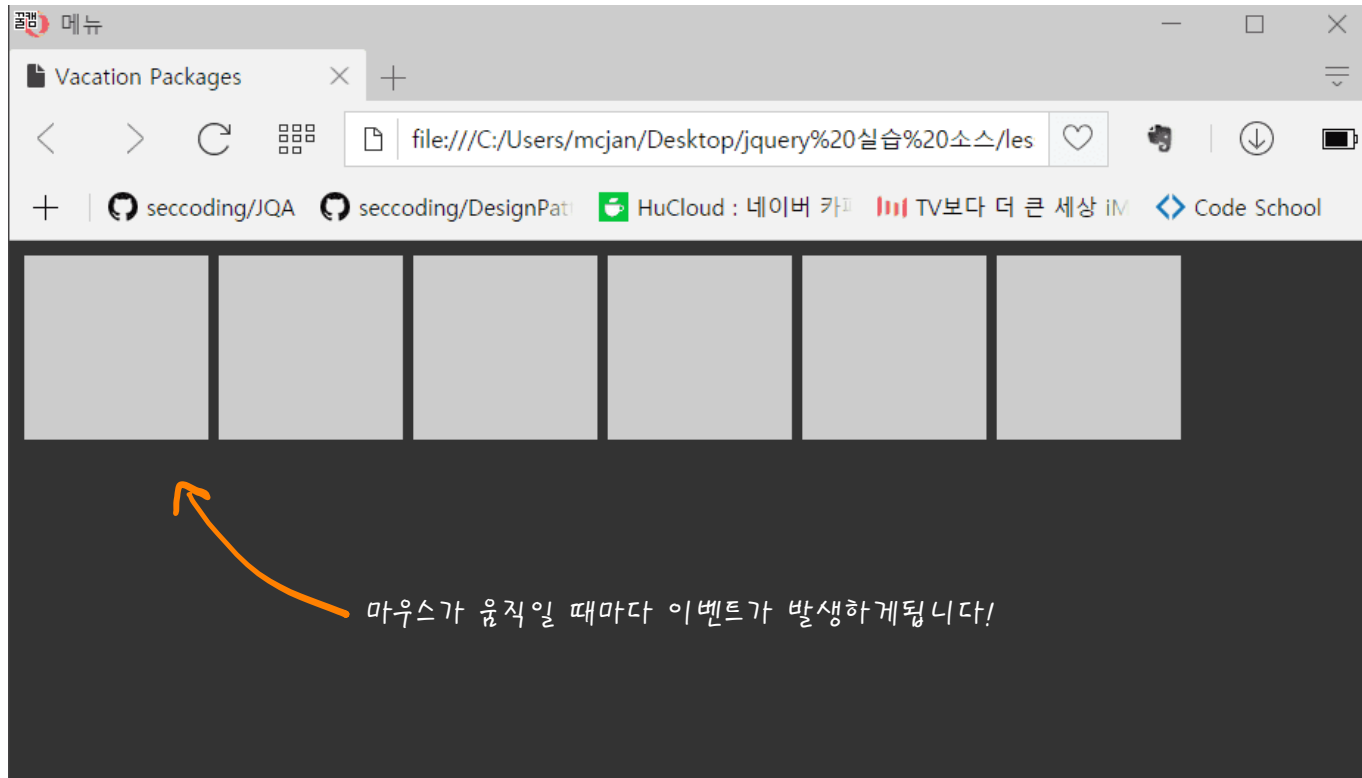
```
$(document).ready(function() {  
    $(".box").on("mouseover", function() {  
        $(this).animate({  
            "opacity": 0.25,  
        });  
    });  
  
    $(".box").on("mouseout", function() {  
        $(this).animate({  
            "opacity": 1.0,  
        });  
    });  
});
```

마우스가 벗어나면 이벤트가 발생합니다!

```
<body class="boxAnimate">  
    <div class="box inline">  
    </div>  
    <div class="box inline">  
    </div>  
    <div class="box inline">  
    </div>  
    <div class="box inline">  
    </div>  
    <div class="box inline">  
    </div>  
    <div class="box inline">  
    </div>  
</body>
```

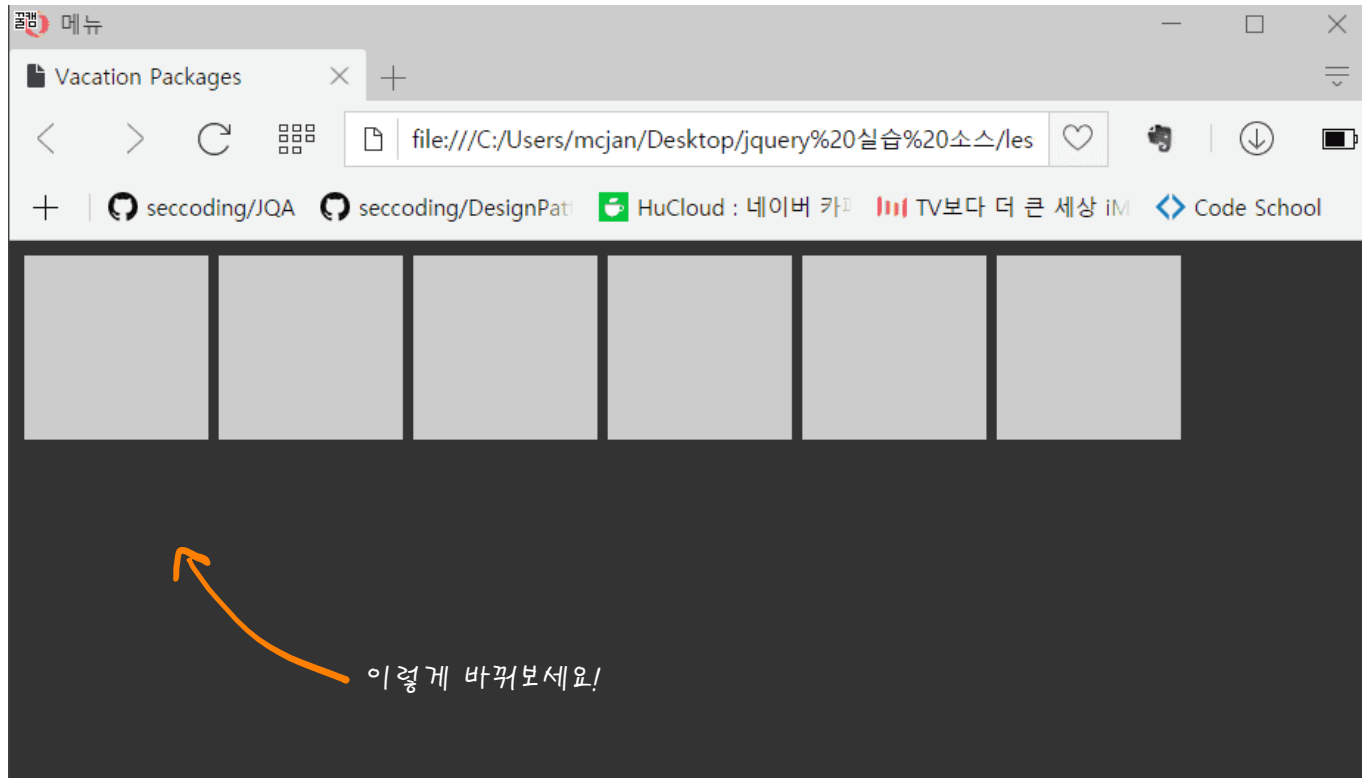
Animation

- CSS로 DOM Animation 하기



Animation

- CSS로 DOM Animation 하기



Form Element

Form Element

■ Input / Textarea 태그 공통

```
var email = $("#email").val();
```

← Email input 에 입력된 값을 가져옵니다.

```
$("#email").val("Email 입니다.");
```

← Email input 에 값을 설정합니다.
입력 값 초기화에 유용하겠지요!

```
<label for="email">이메일</label>
<hr/>
<input type="text" id="email" placeholder="Email을 입력하세요." /><br/>
<br/>
```

Form Element

■ Select Element

```
<label for="jobs">직업</label>
<hr/>
<select id="jobs">
  <option value="0">선택하세요.</option>
  <option value="1">회사원</option>
  <option value="2">학생</option>
  <option value="3">교수</option>
</select><br/>
```

```
var jobs = $("#jobs").val();

$("#jobs").val("2");

$("#jobs").on("change", function() {
  alert($("#jobs option:selected").text());
});
```

Jobs select의 선택된 값을 가져옵니다.

Jobs select에 값을 강제로 선택합니다.

Jobs의 선택 항목이 바뀌었을 때,
옵션의 text 값을 가져와 경고창으로 보여줍니다.

Form Element

■ Radio / CheckBox 공통

```
<span class="title">연령대</span><hr/>
<label for="age10">10대</label>
<input type="radio" name="age" id="age10" value="10">
<label for="age20">20대</label>
<input type="radio" name="age" id="age20" value="20">
<label for="age30">30대</label>
<input type="radio" name="age" id="age30" value="30">
<label for="age40">40대</label>
<input type="radio" name="age" id="age40" value="40">
<label for="age50">50대</label>
<input type="radio" name="age" id="age50" value="50"><br/>
```

Radio / Checkbox 가 선택된 개수를 가져옵니다.
선택했는지, 안했는지 판단할 수 있습니다.

```
console.log($(".input[type=radio][name=age]:checked").length);
console.log($(".input[type=checkbox][name=hobby]:checked").length);

$(".input[type=radio][name=age]").eq(1).prop("checked", true);
$(".input[type=checkbox][name=hobby]").eq(1).prop("checked", true);
```

Radio / Checkbox 를 강제로 선택하게 합니다. "모두선택" 에서 사용할 수 있습니다.

Form Element

■ Radio / CheckBox 공통

```
$("#input[type=checkbox][name=hobby]").on("click", function() {  
    var checked = $(this).prop("checked");  
    console.log(checked);  
    if(checked) {  
        $("#input[type=checkbox][name=hobby]").prop("checked", true);  
    }  
    else {  
        $("#input[type=checkbox][name=hobby]").prop("checked", false);  
    }  
});
```

체크박스를 클릭했을 때,
체크가 된 경우 모든 체크박스를 체크합니다.
반대의 경우 모든 체크박스의 체크를 해제합니다.

Form Element

■ Form Submit

```
<input type="button" id="done" value="완료" />
```

```
$("#done").on("click", function() {  
    // Validation Check...  
    var form = $("#signUpForm");  
    $(form).attr("action", "/doSignUp");  
  
    $(form).attr({  
        "method": "post"  
        "enctype": "multipart/formdata"  
    });  
  
    $(form).submit();  
});
```

Attr 은 이용해서 속성을 정의할 수 있습니다.

Attr 은 객체 리터럴로 정의할 수도 있습니다.

```
var action = $(form).attr("action");  
console.log(action);
```

Attr 에 속성의 이름을 작성하면, 값을 읽어 올 수 있습니다.

감사합니다.

jQuery 마스터하기

장민창

mcjang@hucloud.co.kr