



웹 프로그래밍

11. jQuery 실습

`$(function () {})`



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

=

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

each()

- 단위(unit) 별로 각각 처리하기 메소드
 - 하나의 unit은 주로 동일 class 속성을 가진 그룹

```
<body>
  <ul>
    <li class="animal">cat</li>
    <li class="animal">dog</li>
    <li class="animal">turtle</li>
    <li class="animal">tiger</li>
  </ul>
</body>
```

```
$(function(){
  $('.animal').each(function(){
    var name = $(this).text();
    alert(name);
  });
});
```



페이지가 로드 됨과 동시에
4 개의 알림창이 차례대로 생성
 태그 내의 텍스트가 차례대로
알림창에 출력됨.

<dl>

- 정의: a **D**escription **L**ist
- <dt> 태그 및 <dd> 태그와 함께 사용

```
<dl>  
  <dt>Coffee</dt>  
  <dd>Black hot drink</dd>  
  <dt>Milk</dt>  
  <dd>White cold drink</dd>  
</dl>
```



```
Coffee  
  Black hot drink  
Milk  
  White cold drink
```

실습 #1: Accordion



- jquery.html에 다음 추가

```
<div class="lab">
  <dl class="accordion">
    <dt>Item 1</dt>
    <dd>some explanation here. some explanation here.</dd>
    <dt>Item 2</dt>
    <dd>some explanation here. some explanation here.</dd>
    <dt>Item 3</dt>
    <dd>some explanation here. some explanation here. </dd>
  </dl>
  <dl class="accordion">
    <dt>Item 1</dt>
    <dd>some explanation here. some explanation here.</dd>
    <dt>Item 2</dt>
    <dd>some explanation here. some explanation here.</dd>
    <dt>Item 3</dt>
    <dd>some explanation here. some explanation here. </dd>
  </dl>
</div>
```





```
dl{  
    width:420px;  
    margin:0 auto 20px;  
}  
dt{  
    font-size:16px;  
    color:#fff;  
    background:#31374c;  
    padding:8px 12px;  
    border-radius:3px;  
    margin:0 0 2px;  
}  
dd{  
    font-size:14px;  
    color:#4d4d4d;  
    background:#fff;  
    padding:10px 20px;  
    margin:0 0 2px;  
}
```



- `<dl class="accordion">` 별로
 - 모든 `<dd>`를 닫는다
 - `find()` 메소드 활용
 - `<dt>`를 클릭하면
 - 모든 `<dd>`를 닫고
 - 클릭된 `<dt>` 바로 다음에 있는 `<dd>`를 연다
 - `next()` 메소드 활용
 - 마우스 포인터 변경
 - 클릭하지 않은 `<dt>`에 마우스를 가져가면
 - 포인터 모양 변경
 - 포인터 모양: `css("cursor","pointer");`
 - 일반 모양: `css("cursor", "default");`



```
$("dd").hide() vs var alldd = dl.find("dd");  
alldd.hide();
```

Accordion 확장



- <dt>가 열려있을 때와, 닫혀 있을 때 외관 차이
 - jquery.css 파일에 클래스 추가
 - <dt>에 대해서 닫혔을 때, class=closed css 속성 추가
 - <dd>에 대해서 닫혔을 때, class=closed css 속성 추가

```
dt.closed {  
    background: #9396a2;  
    cursor: pointer;  
}  
  
dd.closed {  
    display: none;  
}
```

- jquery 코드를 활용하여 <dt>와 <dd>에 class=closed 속성 추가

```
$("dt").addClass("closed");  
$("dd").addClass("closed");
```


jquery.js - 확장



● 함수 활용

● closeAll()

- 모든 <dt>와 <dd>가 closed 상태
- class=closed 속성 추가
- addClass("closed");

● open(dt, dd)

- 선택된 <dt>와 <dd>만 열려있는 상태 (원래 상태)
- class=closed 속성 제거
- removeClass("closed")



```
$(".accordion2").each(function (){  
    var dl = $(this);  
    var allDt = dl.find('dt');  
    var allDd = dl.find("dd");  
    function closeAll(){  
        allDd.addClass("closed");  
        allDt.addClass("closed");  
    }  
    function open(dt,dd){  
        dt.removeClass("closed");  
        dd.removeClass("closed");  
    }  
});
```

함수를 이용하여 나머지 코드 완성!

```
});
```

실습 #2: 슬라이드 쇼



- 동작 구조
 - 3초마다 이미지 변경
 - 맨 앞의 img 요소를 맨 뒤로 이동시키고 fadeOut
 - 두 번째 img 요소를 fadeIn
- 주요 기술
 - eq 메소드
 - 인덱스 번호를 지정하여 jQuery 오브젝트 내의 요소를 간추린 메소드
 - setInterval에서 주기적 처리

jquery.html



```
<div class="lab">
  <h1>slideshow</h1>
  <div class="slideshow">
    
    
    
    
    
  </div>
  <div class="slideshow">
    
    
    
    
    
  </div>
  <div class="slideshow">
    
    
    
    
    
  </div>
</div>
```



```
.slideshow{
    width:210px;
    height:150px;
    margin:10px 20px;
    position:relative;
    display:inline-block;
}

.slideshow img{
    position:absolute;
    top:0;
    left:0;
    border:5px solid #dfdfff;
}

.slideshow img.alt{
    display:none;
}
```



```
var interval = 4000;
$('.slideshow').each(function(){
    var container = $(this);
    function switchImg(){

    }
    setInterval(switchImg, interval);
});
```

슬라이드 쇼 확장



- 이미지 위에 마우스를 가져가면 자동 전환 중지
 - 만약 이미지에 링크가 걸려 있다면, 해당 링크를 클릭할 수 있도록 이미지 변환 중지 기능 필요
 - 해당 이미지 전환만 중지
 - 나머지 이미지들은 계속해서 자동 전환
- 추가 기능
 - 타이머 변수 사용
 - clearInterval 활용
 - hover() 메소드 활용
 - hover(handler1, handler2);
 - mouseover 일 때 동작 (handler1)과 mouseout 일 때 동작(handler2)를 모두 지정해 주어야 함