

웹 표준에 맞는 HTML5 프로그래밍 강의 노트

제 22회차
jQuery 고급

■ 학습목표

- jQuery를 이용해서 이벤트와 애니메이션을 활용할 수 있다.
- jQuery와 Ajax을 사용할 수 있다.

■ 학습내용

- jQuery와 이벤트 및 애니메이션
- jQuery와 Ajax

1. jQuery와 이벤트 및 애니메이션

1) jQuery와 이벤트

- 이벤트 연결 함수

<jQuery 1.7 이전 버전>

- `bind()` : 현재 존재하는 문서 객체에만 이벤트를 연결
- `delegate()` : 현재 또는 미래에 존재하는 문서 객체에 이벤트를 연결
- `live()` : 현재 또는 미래에 존재하는 문서 객체에 이벤트를 연결
- `one()` : 한 번만 이벤트에 연결

<jQuery 1.7 이후 버전>

- `on()` : 선택된 엘리먼트와 자식 엘리먼트를 위한 하나 이상의 이벤트 핸들러를 첨부함
일관성 있는 API를 많이 가져올 수 있다는 장점이 있음

- `on()` 함수 사용법

```
$(selector).on(event, childSelector, data, function)
```

- 옵션

- `event(필수)` : 선택된 엘리먼트에 접근하기 위한 하나 이상의 이벤트나 네임스페이스를 지정
- `childSelector(선택)` : 이벤트 핸들러는 오직 특정한 자식 엘리먼트에 접근해야 한다고 지정
- `data(선택)` : 함수에 따라서 추가적인 데이터 지정
- `function(필수)` : 이벤트가 발생할 때 실행되는 함수 지정

1. jQuery와 이벤트 및 애니메이션

1) jQuery와 이벤트

- 이벤트 연결 함수를 활용한 실습 예제

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<script type="text/javascript" src="http://code.jquery.com/jquery-1.10.2.js"></script>
    <script type="text/javascript">
        $(document).ready(function(){
            $("h1").bind("click", function(){
                alert("클릭 이벤트 발생");
            });
        });
    </script>
<title>무제 문서</title>
</head>

<body>

    <h1> 이벤트 bind... </h1>

</body>
</html>
```

1. jQuery와 이벤트 및 애니메이션

1) jQuery와 이벤트

- 마우스 enter & leave event"라는 텍스트 위에 마우스를 위치하거나 해당 텍스트 밖으로 마우스를 이동 시 숫자가 카운팅 되는 실습을 활용한 예제

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<script type="text/javascript" src="http://code.jquery.com/jquery-1.10.2.js"></script>
<script type="text/javascript">
    var count = 0;
    $(document).ready(function(){
        $("h1").bind("mouseenter mouseleave", function(){
            count++;
            $("#result").text(count + " enter to leave ");
        });
    });
</script>
<title>ex_2</title>
</head>

<body>

    <h1> 마우스 enter & leave event </h1>

    <div>
        이벤트 : <span id="result"></span>
    </div>

</body>
</html>
```

1. jQuery와 이벤트 및 애니메이션

1) jQuery와 이벤트

- 이벤트 연결 해제 함수

〈jQuery 1.7 이전 버전〉

- `unbind()` : `bind()`를 통해 연결한 이벤트를 제거
- `die()` : `live()`를 통해 연결한 이벤트를 제거
- `undelegate()` : `delegate()`를 통해 연결한 이벤트를 제거

〈jQuery 1.7 이후 버전〉

- `off()` : `on()`를 통해 연결한 이벤트를 제거

일관성 있는 API를 많이 가져올 수 있다는 장점이 있음

- `Off()`함수 사용법

```
$(selector). off(event, selector, function(eventObj))
```

- 옵션

- `event(필수)` : 선택된 엘리먼트를 제거하기 위한 하나 이상의 이벤트 또는 네임스페이스를 지정
- `selector(선택)` : 이벤트 핸들러에 접근할 때 `on` 함수에 전달되는 `selector`와 동일해야 함
- `function(eventObj)(선택)` : 이벤트가 발생할 때 실행되는 함수를 지정

1. jQuery와 이벤트 및 애니메이션

2) jQuery와 애니메이션

- 애니메이션과 속성 값

- 움직임에 관련된 속성

단수 수치 값(Single Numeric Value)을 이용해서 움직임을 줄 수 있음



비수치형 속성 값들로는 애니메이션 효과를 줄 수 없음

- jQuery

- 기본 기능으로 움직임을 줄 수 있음
- 픽셀단위로 제어가능 할 수 있음
- em(이엠)과 %(퍼센트)같은 값 지원 가능

- 지속시간 (Duration)

- 시간(Durations)의 기본단위 : milliseconds(1000분의 1초)

- 높은 값을 줄 경우 : 느리게 움직임
- 작은 값을 줄 경우 : 빠르게 움직임
- 값을 설정하지 않을 경우 : 400 milliseconds가 세팅됨

- 문자열 사용 : 'fast'와 'slow'

- 'fast' 수치 값 : 200 milliseconds
- 'slow' 수치 값 : 600 milliseconds

- 콜백 함수 (Complete Function)

- 콜백 함수를 인자로 설정하면 움직임이 완료된 다음에 해당하는 함수가 실행됨
- 인자를 가질 수 없음
- this를 사용하면 움직임이 있었던 DOM 요소를 제어할 수 있음

1. jQuery와 이벤트 및 애니메이션

2) jQuery와 애니메이션

- 애니메이션 기본 사용법(움직임 적용)

- 마크업

```
<div id="clickme">  
    Click here  
</div>  

```

- 투명도, 좌표 값, 높이

```
$('#clickme').click(function() {  
    $('#book').animate({  
        opacity: 0.25,  
        left: '+=50',  
        height: 'toggle'  
    }, 5000, function() {  
        // Animation complete.  
    });  
});
```


1. jQuery와 이벤트 및 애니메이션

2) jQuery와 애니메이션

- 속성 값 변화에 대한 실습 예제

```
<!doctype html>
<html>
<head>
    <style>
        div{
            background-color:#BCA;
            width:100px;
            border:1px solid green;
        }
    </style>
    <script src="http://code.jquery.com/jquery-latest.js"></script>
    <meta charset="utf-8">
    <title>ex_3</title>
</head>

<body>

    <button id="go">&raquo;Run</button>

    <div id="block">Hello!</div>
    <script>
        $("#go").click(function(){
            $("#block").animate({
                width: "70%",
                opacity: 0.4,
                marginLeft: "0.6in",
                fontSize: "3em",
                borderWidth: "10px",
            }, 1500);
        });
    </script>

</body>
</html>
```

1. jQuery와 이벤트 및 애니메이션

2) jQuery와 애니메이션

- 좌표 이동에 대한 실습 예제

```

<!doctype html>
<html>
<head>
    <style>
        div {
            position:absolute;
            background-color:#abc;
            left:50px;
            width:90px;
            height:90px;
            margin:5px;
        }
    </style>
    <script src="http://code.jquery.com/jquery-latest.js"></script>

    <meta charset="utf-8">
    <title>ex_4</title>
</head>

<body>
    <button id="left">&laquo;</button> <button
id="right">&raquo;</button>
    <div class="block"></div>

    <script>
        $("#right").click(function(){
            $(".block").animate({"left": "+=50px"}, "slow");
        });

        $("#left").click(function(){
            $(".block").animate({"left": "-=50px"}, "slow");
        });
    </script>

</body>
</html>

```

1. jQuery와 이벤트 및 애니메이션

2) jQuery와 애니메이션

- 버튼 애니메이션에 대한 실습 예제

```
<!DOCTYPE html>
<html>
<head>
  <style>

      div {
        background-color:#bca;
        width:200px;
        height:1.1em;
        text-align:center;
        border:2px solid green;
        margin:3px;
        font-size:14px;
      }
      button {
        font-size:14px;
      }

  </style>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <meta charset="utf-8">
  <title>ex_4</title>
</head>
<body>

  <button id="go1">&raquo; Animate Block1</button>
  <button id="go2">&raquo; Animate Block2</button>
  <button id="go3">&raquo; Animate Both</button>

  <button id="go4">&raquo; Reset</button>
  <div id="block1">Block1</div>
  <div id="block2">Block2</div>
```

1. jQuery와 이벤트 및 애니메이션

2) jQuery와 애니메이션

- 버튼 애니메이션에 대한 실습 예제

```
<script>

    $( "#go1" ).click(function(){
        $( "#block1" ).animate( { width: "90%" }, { queue: false,
duration: 3000 })
                                .animate({ fontSize: "24px" }, 1500 )
                                .animate({ borderRightWidth: "15px" },
1500 );
    });

    $( "#go2" ).click(function(){
        $( "#block2" ).animate( { width: "90%" }, 1000 )
                                .animate({ fontSize: "24px" }, 1000 )
                                .animate({ borderLeftWidth: "15px" },
1000 );
    });

    $( "#go3" ).click(function(){
        $( "#go1" ).add( "#go2" ).click();
    });

    $( "#go4" ).click(function(){
        $( "div" ).css({ width: "", fontSize: "", borderWidth: "" });
    });

</script>

</body>
</html>
```

2. jQuery와 Ajax

1) Ajax란?

- Ajax(Asynchronous JavaScript and XML)의 약어
- XML을 이용하여 비 동기 통신을 처리하기 위한 기술
- HTTP방식으로 요청을 했을 때, XML파일 형식의 응답으로 결과를 보내줌

- 자바 스크립트 Ajax(XHR의 속성)

• readyState

- Ajax 개체의 상태를 나타내는 숫자
 - 처음 Ajax 개체를 생성할 경우 : 0
 - get() 메소드로 요청할 페이지 정보를 설정할 경우 : 1
 - send() 메소드로 요청을 보낼 경우 : 2
 - 서버에서 응답이 오기 시작할 경우 : 3
 - 서버 응답이 완료될 경우 : 4
- 읽기 전용

• status

- 서버로부터 받은 응답의 상태를 나타내는 숫자
 - 정상적으로 응답을 받은 경우 : 200
 - 페이지를 찾지 못한 경우 : 404
- 읽기 전용

• statusText

- 서버로부터 받은 응답의 상태를 나타내는 문자열
 - 정상적으로 응답을 받을 경우 : 'OK'
 - 파일을 찾지 못할 경우 : 'Not Found'
- 읽기 전용

2. jQuery와 Ajax

1) Ajax란?

- XHR의 메소드

• > open()

[] 부분은 있어도 되고 없어도 된다는 뜻

open(method, url[, async]);

비동기 통신 여부를 나타내며 true 또는 false로 지정함

- async인자를 지정하지 않으면 true를 기본값으로 사용함

요청할 페이지의 주소를 지정함

http 요청 방식을 나타내며 'get' 또는 'post' 방식을 사용함

Ajax 요청을 초기화하면서 요청 방식, 주소, 동기화 여부를 지정함

• > send()

Ajax 요청을 보냄

send(body);

요청과 함께 서버로 보낼 내용을 지정함

• > abort()

abort()

send 메소드로 보낸 요청을 취소함

2. jQuery와 Ajax

1) Ajax란?

- XHR의 메소드

- > getAllResponseHeaders()

getAllResponseHeaders();

- 응답을 받은 경우 응답의 모든 헤더 정보를 문자열로 돌려줌
→ 헤더 전체 내용을 가져옴

- > getResponseHeader()

getResponseHeader(header)

- 응답을 받은 경우 header 인자로 지정한 이름의 헤더 정보 값을 문자열로 돌려줌
→ 특정한 헤더를 가져옴

2. jQuery와 Ajax

2) jQuery로 Ajax 사용하기

- jQuery Ajax 을 활용한 실습 예제

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
    <script src="http://code.jquery.com/jquery-1.11.1.js"></script>
    <script type="text/javascript">
        $(function(){
                                $('button').click(function(){

                                $('#container').load('text.html');
                                });
        });
    </script>
<title> ex_6 </title>
</head>

<body>

    <button> 데이터 가져오기 </button>
    <div id="container"> 데이터 가져오기 전</div>

</body>
</html>
```

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Hello</title>
</head>

<body>

    <p> Hello </p>

</body>
</html>
```


2. jQuery와 Ajax

2) jQuery로 Ajax 사용하기

- 자바스크립트 파일 읽어오기를 활용한 실습 예제

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>ex_07</title>
    <script src="http://code.jquery.com/jquery-1.11.1.js"></script>
    <script type="text/javascript">
        $(function(){
            $.getScript("hello.js");
            $('#submit').click(function(){
                var msg =
call($('#username').val());
                $('#message').html(msg);
                return false;
            });
        });
    </script>
</head>
<body>
    <form>
        <label> 이름을 입력하세요 : </label>
        <input type="text" name="username" class="username"/><br/>
        <input type="button" id="submit" value="전송"/>
    </form>
    <div id="message"></div>
</body>
</html>
```

■ 정리하기

1. jQuery와 이벤트 및 애니메이션

- 이벤트 연결 함수 : **bind(), delegate(), live(), one()**
→ bind() 함수를 이용하면 엘리먼트에 이벤트를 쉽게 연결할 수 있음
- 이벤트 연결 해제 함수 : **unbind(), die(), undelegated()**
- 애니메이션 지속시간의 기본 단위 : **ms(1000분의 1초)**
→ 설정하지 않을 경우 : **400ms**

2. jQuery와 Ajax

- **Ajax** : xml을 이용하여 비동기 통신을 처리하기 위한 기술
- **open()** : Ajax 요청을 초기화하면서 요청 방식, 주소, 동기화 여부를 지정함
- **send()** : Ajax의 요청을 보냄