



30 입력 양식 플러그인

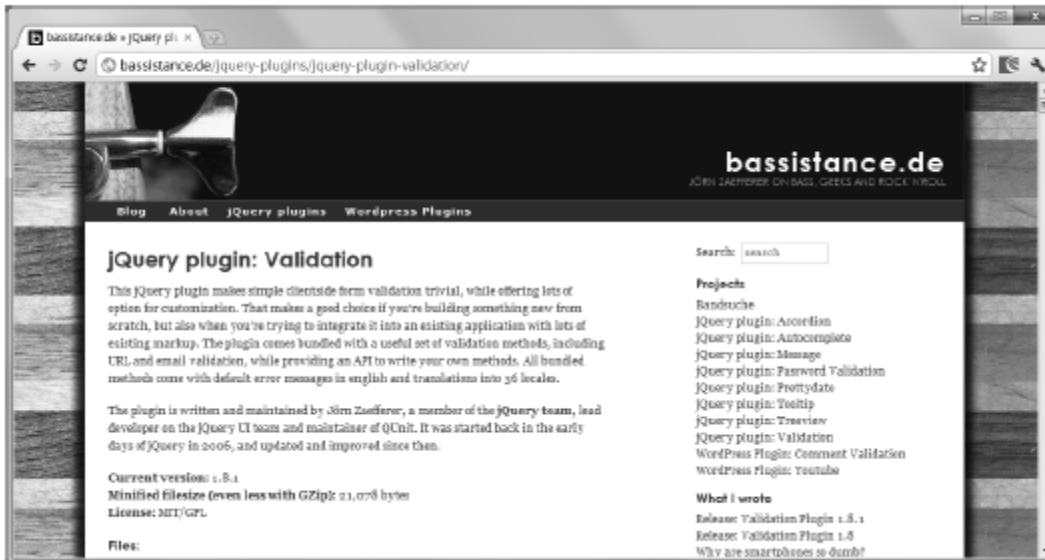
모던 웹을 위한 Javascript jQuery 입문

30.1 jQuery Validation 플러그인

❖ Validation 플러그인

- ASP.NET MVC 기본 내장 플러그인
- 입력 양식의 유효성을 검사할 때 반드시 사용하는 플러그인
- Validation 플러그인의 공식 사이트
 - <http://bassistance.de/jquery-plugins/jquery-plugin-validation/>

그림 30-1 jQuery Validation 플러그인의 공식 사이트



30.1 jQuery Validation 플러그인

❖ head 태그 구성

코드 30-1 head 태그 구성

```
<head>
  <style>

  </style>
  <script src="http://code.jquery.com/jquery-1.7.js">/script>
  <script src="jquery.validate.js">/script>
  <script>

  </script>
</head>
```



❖ Body 태그 구성

코드 30-2 body 태그 구성

```
<body>
  <formid="my_form">
    <table>
      <tr>
        <td><label>이름</label></td>
        <td><input type="text" name="name" /></td>
      </tr>
      <tr>
        <td><label>이메일</label></td>
        <td><input type="text" name="email" /></td>
      </tr>
      <tr>
        <td><label>홈페이지</label></td>
        <td><input type="text" name="homepage" /></td>
      </tr>
    </table>
    <input type="submit" value="SEND"/>
  </form>
</body>
```



30.1 jQuery Validation 플러그인

❖ Validation 플러그인 사용

- Validation 플러그인의 핵심 메서드는 validate()
- validate() 메서드는 form 태그에 적용

코드 30-3 validate() 메서드

```
<script>
  $(document).ready(function () {
    $('#my_form').validate();
  });
</script>
```



❖ 데이터 유효성 검사

- validate() 메서드를 적용한 form 태그 안의 input 태그
 - 자동으로 유효성 검사가 이루어짐
 - input 태그의 class 속성에 Validation 플러그인이 지정한 키워드 입력

코드 30-4 class 속성을 사용한 유효성 검사

```
<input type="text" name="name" class="required" minlength="2" />  
<input type="text" name="email" class="required email" />  
<input type="text" name="homepage" class="required url" />
```

그림 30-3 유효성 검사

이름	<input type="text"/>	This field is required.
이메일	<input type="text" value="rintianta"/>	Please enter a valid email address.
홈페이지	<input type="text" value="Antar"/>	Please enter a valid URL.
<input type="button" value="SEND"/>		



30.1 jQuery Validation 플러그인

❖ 옵션 객체를 사용한 유효성 검사

코드 30-5 옵션 객체를 사용한 유효성 검사

```
<script>
    $(document).ready(function () {
        $('#my_form').validate({
            rules: {

            },

            messages: {

            }
        });
    });
</script>
```



❖ 예제 코드 30-6]

- rules 속성과 messages 속성은 객체 할당
- 내부에는 각 input 태그의 name 속성 넣음
- rules 속성 - 어떠한 유효성 검사를 수행할지 지정하는 부분
- message 속성은 유효성 검사에 통과하지 못했을 때 출력하는 메시지



30.1 jQuery Validation 플러그인

❖ 예제 코드 30-6]

코드 30-6 Validate 플러그인의 옵션

```
$('#my_form').validate({
  rules: {
    name: {
      required: true,
      minlength: 2
    },
    email: 'required email',
    homepage: {
      required: true,
      url: true
    }
  },
  messages: {
    name: {
      required: '이름을 입력해주세요.',
      minlength: '{0}글자 이상 입력해주세요.'
    },
    email: {
      required: '이메일을 입력해주세요.',
      email: '이메일 형식에 맞게 입력해주세요.'
    },
    homepage: {
      required: '홈페이지 주소를 입력해주세요.',
      url: 'URL 형식에 맞게 입력해주세요.'
    }
  }
});
```

그림 30-4 실행 결과

이름	<input type="text"/>	이름을 입력해주세요.
이메일	<input type="text" value="RintlanTia"/>	이메일 형식에 맞게 입력해주세요.
홈페이지	<input type="text" value="Antar"/>	URL 형식에 맞게 입력해주세요.
<input type="button" value="SEND"/>		



30.1 jQuery Validation 플러그인

❖ Validation 플러그인으로 자주 사용하는 유효성 검사 종류

표 30-1 Validation 플러그인의 유효성 검사 종류

유효성 검사 종류	설명
required	필수 입력 사항
email	이메일 형식
url	URL 주소 형식
date	날짜 형식
dateISO	ISO 날짜 형식
number	숫자 형식
digits	정수 숫자 형식
creditcard	신용 카드 형식
equalTo	값이 같아야 하는 다른 대상을 지정
maxlength	최대 문자 길이
minlength	최소 문자 길이
rangelength	문자 길이
range	숫자 범위
max	최대 숫자
min	최소 숫자

참고 더 많은 유효성 검사의 종류는 <http://docs.jquery.com/Plugins/Validation>을 확인하세요.



30.1 jQuery Validation 플러그인

❖ 예제코드 30-7]

- extend() 메서드 사용 - Validation 플러그인 기본 메시지 바꿀 수 있음

코드 30-7 Validate 플러그인의 기본 메시지 변경

```
<script>
$(document).ready(function () {
    // 메시지를 정의합니다.
    $.extend(jQuery.validator.messages, {
        required: '필수 입력 사항입니다.',
        minlength: '{0}글자 이상 입력해주세요.',
        email: '이메일 형식에 맞게 입력해주세요.',
        url: 'URL 형식에 맞게 입력해주세요.'
    });

    // Validate 플러그인을 적용합니다.
    $('#my_form').validate({
        rules: {
            name: {
                required: true,
                minlength: 2
            },
            email: 'required email',
            homepage: {
                required: true,
                url: true
            }
        }
    });
});
</script>
```



30.1 jQuery Validation 플러그인

❖ validate() 메서드의 옵션

표 30-2 Validation 플러그인의 옵션

옵션 이름	설명
errorClass	오류 라벨의 클래스를 지정합니다.
errorPlacement	오류가 발생했을 때 위치를 지정합니다.
submitHandler	입력 양식의 제출이 성공했을 때 실행됩니다.
success	유효성 검사를 통과했을 때 사용됩니다.

코드 30-8 Validate 플러그인의 이벤트

```
// Validate 플러그인을 적용합니다.
$('#my_form').validate({
  errorClass: 'rintiantta',
  rules: {
    /* 생략 */
  },
  submitHandler: function (form) {

  },
  success: function(label){

  },
  errorPlacement: function (error, element) {

  }
});
```



❖ errorClass 속성

- errorClass 속성을 rintiantta로 설정

코드 30-9 Validate 플러그인의 errorClass 사용

```
<style>
  label.rintiantta { color: Red; }
</style>
```

참고 errorClass 속성을 별도로 지정하지 않으면 errorClass 속성은 error로 설정됩니다.

❖ submitHandler 이벤트

- 모든 유효성 검사를 통과한 상태
- submit 버튼을 눌렀을 때 실행
- submitHandler 이벤트 핸들러 안에서 this 키워드는 form 객체가 아님
 - submitHendler 이벤트 핸들러의 첫 번째 매개 변수가 form 객체

코드 30-10 Validate 플러그인 이벤트의 매개 변수

```
submitHandler: function (form) {
  var serialize = $(form).serialize();
  alert(serialize);
}
```



❖ Password Validation 플러그인

- Validation 플러그인의 파생 플러그인
- Password Validation 플러그인을 사용하려면?
 - Validation 플러그인 필요
- Password Validation 플러그인의 공식 사이트
 - <http://bassistance.de/jquery-plugins/jqueryplugin-password-validation/>

그림 30-5 jQuery Password Validation 플러그인 공식 사이트

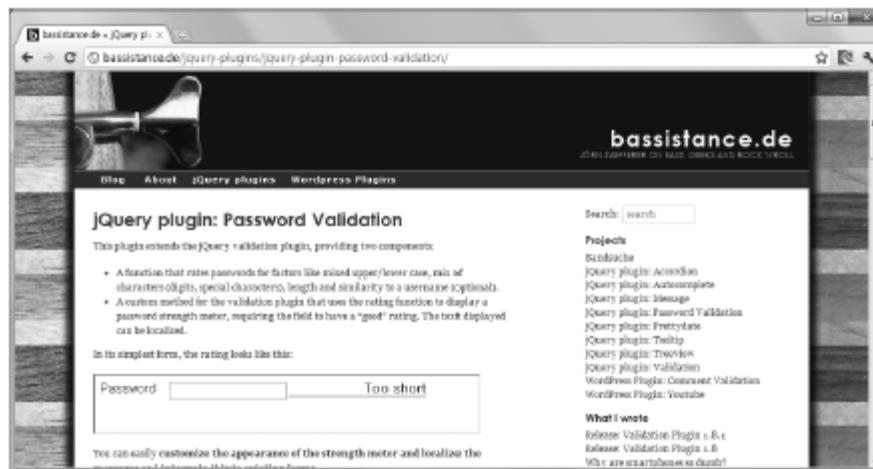


그림 30-6 폴더 구성



❖ head 태그를 구성

- Password Validation 플러그인을 Validation 플러그인 뒤에 추가해야!

코드 30-11 head 태그 구성

```
<head>
  <link rel="stylesheet" href="jquery.validate.password.css" />
  <script src="http://code.jquery.com/jquery-1.7.js"></script>
  <script src="jquery.validate.js"></script>
  <script src="jquery.validate.password.js"></script>
  <script>
    $(document).ready(function () {

      });
  </script>
</head>
```



❖ Body 태그를 구성

- input[type=password] 다음 div.password-meter 태그 부분이 Password Validation 플러그인의 핵심

코드 30-12 body 태그 구성

```
<body>
<form id="my_form">
  <table>
    <tr>
      <td><label>아이디</label></td>
      <td><input type="text" name="name" /></td>
    </tr>
    <tr id="password_row">
      <td><label>비밀번호</label></td>
      <td><input type="password" name="password" /></td>
    </tr>
  </table>
</form>
</body>
```



❖ Body 태그를 구성 (2)

```
        <td>
            <div class="password-meter">
                <div class="password-meter-message">X</div>
                <div class="password-meter-bg">
                    <div class="password-meter-bar">X</div>
                </div>
            </div>
        </td>
    </tr>
    <tr>
        <td>Xlabel비밀번호 확인</label>X</td>
        <td>Xinput type="password" name="pass_check" />X</td>
        <td>X</td>
    </tr>
</table>

</form>
</body>
```



❖ 예제 코드 30-13]

- Password Validation 플러그인을 사용하려면 스타일을 추가 설정해야
- 스타일을 지정하지 않고 실행하면 라벨이 공간을 차지해 입력 양식이 찌그러짐

코드 30-13 style 태그 구성

```
<style>
    #password_rowlabel.error { display:none !important; }
</style>
```



30.2 jQuery Password Validation 플러그인

❖ 예제 코드 30-15]

- required 속성에 true를 입력하듯 password 속성에 true 입력

코드 30-15 Password Validation 플러그인의 사용

```
<script>
    $(document).ready(function () {
        $('#my_form').validate({
            rules: {
                name: {
                    required: true,
                    minlength: 2
                },
                password: { password: true },
                pass_check: { equalTo: "#password" }
            },
            errorPlacement: function (label, element) {
                label.prependTo(element.parent().next());
            }
        });
    });
</script>
```

그림 30-7 실행 결과

아이디	<input type="text" value="RintlanTta"/>	
비밀번호	<input type="password" value="*****"/>	Good
비밀번호 확인	<input type="password" value="****"/>	Please enter the same value again.
<input type="button" value="SEND"/>		



❖ jQuery Uploidy 플러그인

- 한 번에 여러 파일을 쉽게 업로드할 수 있게하는 jQuery 플러그인
- 플래시를 사용하므로 플래시를 지원하지 않는 모바일 장치에서 사용 불가
- Uploadify 플러그인의 공식 홈페이지
 - <http://www.uploadify.com/>

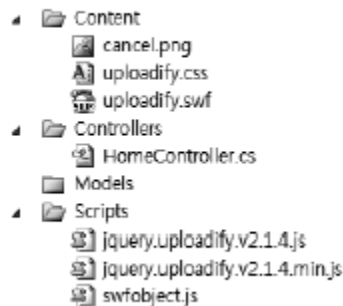
그림 30-8 jQuery Uploadify 플러그인 공식 사이트



❖ UploDify 플러그인

- 안정적인 2.1.4 버전 사용
- 파일을 업로드해야 하므로 서버가 필요함
- ASP.NET MVC3 프로젝트를 생성한 후에 다운받은 UploDify 플러그인의 압축을 풀어 그림 30-9처럼 프로젝트 구성

그림 30-9 프로젝트 구성



❖ HomeController.cs

- HomeController에 사용자가 업로드한 파일을 받아 저장할 수 있는 액션 UploadFile을 만듦

코드 30-16 HomeController.cs: 액션 UploadFile 생성

```
public class HomeController : Controller
{
    public ActionResult Index()
    {
        return View();
    }

    public ActionResult UploadFile()
    {
        // 파일을 저장합니다.
        return Content("OK");
    }
}
```



30.3 jQuery Uploidy 플러그인

❖ 예제 코드 30-17]

- ASP.NET MVC3에서 파일 저장하는 코드
- Path 클래스를 사용할 때 using 구문 자동 추가

코드 30-17 HomeController.cs: 파일 업로드

```
public ActionResult UploadFile()
{
    foreach (string file in Request.Files)
    {
        var fileBase = Request.Files[file];
        if (fileBase.ContentLength != 0)
        {
            string savedFilename = Path.Combine(AppDomain.CurrentDomain.BaseDirectory,
                Path.GetFileName(fileBase.FileName));
            fileBase.SaveAs(savedFilename);
        }
    }
    return Content("OK");
}
```



30.3 jQuery Uploadify 플러그인

❖ 예제 코드 30-18]

■ head 태그 구성

코드 30-18 Index.cshtml: head 태그 구성

```
<head>
  <title>Index</title>
  <link rel="Stylesheet" href="../../Content/uploadify.css" />
  <script src="http://code.jquery.com/jquery-1.7.js"></script>
  <script src="../../Scripts/jquery.uploadify.v2.1.4.js"></script>
  <script src="../../Scripts/swfobject.js"></script>
  <script>
    $(document).ready(function () {

      });
  </script>
</head>
```



30.3 jQuery Uploadify 플러그인

❖ 예제 코드 30-19]

■ Uploadify 플러그인

- type 속성이 file인 input 태그에 적용하는 플러그인

코드 30-19 Index.cshtml: body 태그 구성

```
<body>
  <form id="my_form">
    <input type="file" name="file" id="uploadify" />
    <input type="submit" />
  </form>
</body>
```



30.3 jQuery Uploadify 플러그인

❖ Uploadify 플러그인의 핵심 메서드는 uploadify()

- uploadify() 메서드를 사용함과 동시에 이벤트 연결

코드 30-20 Index.cshtml: Uploadify 플러그인 적용

```
<script>
    $(document).ready(function () {
        $('#uploadify').uploadify();

        $('#my_form').submit(function (event) {

            event.preventDefault();

        });
    });
</script>
```



30.3 jQuery UploDify 플러그인

❖ Uploadify 플러그인의 옵션과 메서드

표 30-3 Uploadify 플러그인의 옵션

옵션 이름	설명
uploader	uploadify.swf 파일 경로를 지정합니다.
ucancelimg	cancel.png 파일 경로를 지정합니다.
script	업로드 대상 웹 서비스(액션)를 지정합니다.
sizeLimit	업로드할 파일의 최대 한도를 지정합니다.
multi	여러 파일을 동시에 업로드할지 지정합니다.
auto	파일을 자동으로 업로드할지 지정합니다.

표 30-4 Uploadify 플러그인의 메서드

메서드 이름	설명
uploadifyCancel()	현재 업로드를 취소합니다.
uploadifyClearQueue()	대기 목록의 모든 업로드를 제거합니다.
uploadifySettings()	업로드 옵션을 변경합니다.
uploadifyUpload()	파일을 업로드합니다.



30.3 jQuery Uploidy 플러그인

❖ 예제 코드 30-21]

- Uploadify 플러그인의 옵션과 메서드를 사용해 파일 업로드
- auto 속성을 false로
 - submit 버튼 눌러야 파일이 업로드

코드 30-21 Index.cshtml: Uploidy 플러그인의 옵션과 메서드

```
$(document).ready(function () {  
    $('#uploidy').uploadify({  
        cancelImg: '/Content/cancel.png',  
        uploader: '/Content/uploadify.swf',  
        script: '/Home/UploadFile',  
  
        sizeLimit: 2097152,  
  
        multi: true,  
        auto: false,  
    });  
  
    $('#my_form').submit(function (event) {  
        $('#uploidy').uploadifyUpload();  
        event.preventDefault();  
    });  
});
```

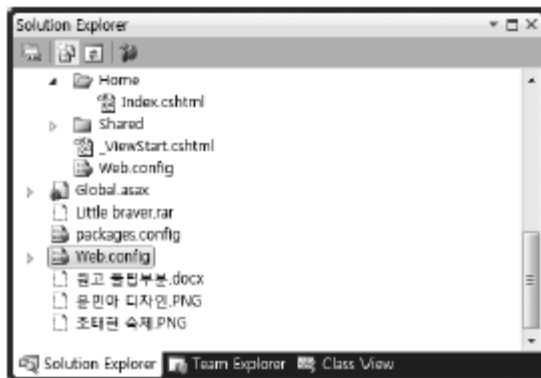


❖ 예제코드 실행 결과

그림 30-10 실행 결과



그림 30-11 업로드된 파일 보기





Thank You !

모던 웹을 위한 Javascript jQuery 입문