

17

플래시 폼

웹에서 이루어지는 사용자와 서버측 애플리케이션 사이의 상호작용은 사용자 입력 폼을 통해 가장 빈번하게 이루어진다. 얼핏 본다면 폼이 그다지 대단한 것이 아니라고 생각할 수도 있지만 사실 온라인 통신서 가장 중요한 부분을 차지하고 있으며(채팅이나 게시판), 데이터의 일관성 및 개인화, 그리고 요즘 각광받고 있는 전자상거래에서도 폼은 반드시 필요하다.

플래시에서 폼을 지원한 것은 플래시 4부터이다. 그 이후로 서버와 데이터를 주고받는 데도 플래시를 많이 사용하게 되었다. 이 장에서는 사용자 입력을 감지하고 플래시에서 어떤 내용을 출력하는 것부터 서버와 데이터를 주고받는 것에 이르기까지 플래시에서 폼을 사용하는 기본적인 방법을 알아본다. XML 처리와 같은 더 복잡한 내용은 '3부. 레퍼런스'에서 다룬다.

플래시 폼 데이터 주기

자세한 내용을 다루기 전에 일반적인 폼 제출 과정을 전체적으로 훑어보면 다음과 같다.

1. 플래시에서 사용자가 입력한 데이터를 받는다.
2. 데이터를 웹 서버에 보낼 준비를 한다(변수를 수집하고 확인한다).
3. HTTP(또는 HTTPS)를 통해 데이터를 웹 서버로 보낸다.
4. 웹 서버에서는 그 데이터를 받아서 서버측에서 데이터를 처리하는 애플리케이션(필 스크립트, PHP 스크립트, 콜드 퓨전, ASP 등)에 전달한다.
5. 데이터를 처리하는 애플리케이션에서는 넘겨받은 데이터를 파싱하여 처리한다.
6. 처리 결과를 웹 서버로 전달하면 웹 서버에서는 그 결과를 다시 플래시로 보낸다.
7. 플래시에서 상황에 따라 그 결과를 출력할 수도 있다.

따라서 플래시 폼이 제대로 동작하려면 다음과 같은 것들이 있어야 한다.

- 프론트엔드(사용자가 실제로 접하는 부분)
- 폼의 내용을 서버측 스크립트 또는 애플리케이션에 보내기 위한 플래시 스크립트
- 서버측 스크립트 또는 애플리케이션
- 서버에서 받은 데이터를 처리할 수 있는 플래시 스크립트

이제 각 구성요소를 하나씩 자세히 살펴보자.

클라이언트측 사용자 입력

사용자가 폼에 어떤 내용을 입력할 때는 보통 텍스트 필드에 내용을 입력하고 나서 폼의 보내기 버튼을 누른다. 내장된 폼 구성요소로는 텍스트 필드만 제공되므로 보내기 버튼을 비롯한 다른 요소들은 모두 플래시 개발자가 직접 만들어야 한다(만드는 방법은 잠시 후에 배울 것이다). 텍스트 필드에 대한 내용은 '18장. 온스크린 텍스트 필드'의 '사용자 입력 텍스트 필드'에 나와있다.

플래시의 액션스크립트를 이용하면 폼에서 서버로 보낼 데이터를 미리 확인하거나 수정할 수도 있다. 웹 서버로 사용자가 입력한 내용을 보내기 전에 그 내용을 미리 확인하면 웹 서버의 데이터 처리 애플리케이션에서 일어날 수 있는 문제를 미연에 방지할 수 있다. 일반적인 확인 과정에서는 모든 필수 필드에 내용을 기입했는지, 데이터의 형식이 제대로 갖춰졌는지와 같은 내용을 검사한다. 예를 들어 이메일 주소를 보낼 때는 아이디 뒤에 @ 기호가 들어가고 그 뒤에 도메인 이름이 제대로 들어있는지 미리 확인할 수도 있다.

서버로 데이터 전송하기

데이터를 확인하고 나면 그 데이터를 웹 서버로 보내기만 하면 된다. 액션스크립트에는 폼을 통해 받은 데이터를 웹 서버로 보내는 필요한 몇 가지 도구가 포함되어 있다.

- loadVariables() 함수(이 장에서 배움).
- XML 클래스의 send()와 sendAndLoad() 메소드(3부 참조).
- 전역 함수인 getURL() 함수(3부 참조).

웹 서버에서는 플래시로부터 받은 데이터를 서버측 애플리케이션으로 전달한다. 이러한 애플리케이션은 보통 미들웨어 데이터베이스(알레어의 콜드 퓨전이나 마이크로소프트의 ASP 등)나 CGI 스크립트(펄이나 PHP 스크립트 또는 자바 서블릿 등)로 만든다.

웹 클라이언트/서버 데이터 주기를 알아보기 전에 먼저 웹 서버와 데이터 처리 애플리케이션의 차이를 짚고 넘어가자. 종종 서버와 애플리케이션을 따로 구분하지 않는 경우도 있다. 클라이언트에서 데이터 처리 애플리케이션에 데이터를 보낼 때는 항상 HTTP 요청을 사용하기 때문에 웹 서버가 이 작업에 관여할 수밖에 없다. 하지만 플래시 폼을 개발할 때는 웹 서버와 데이터 처리 애플리케이션 사이의 데이터 전송을 생각해 보아야 한다. 데이터가 플래시에서 서버로 전달될 때는 URL에 데이터를 첨가하여 보낼 수도 있고(GET을 사용하는 경우) 변수의 이름과 값을 스트림 형태로 보낼 수도 있다(POST를 사용하는 경우). 웹 브라우저와 달리 플래시에서는 웹 서버에서 오류가 발생했을 때 서버로부터 받은 HTTP 오류 메시지를 출력하지 않는다. 예를 들어 웹 서버에서 CGI 스크립트를 찾지 못하는 경우에는 “404 Not

Found” 메시지를 보내지만, 플래시에서는 그러한 메시지를 화면에 표시하지 않는다. 마찬가지로 CGI 스크립트의 사용권한이 제대로 설정되지 않았을 경우에 나오는 메시지도 플래시에서는 볼 수 없다. 플래시를 사용할 때 클라이언트/서버 문제를 해결해야 하는 경우에는 스크립트를 실행시켜 웹 서버의 HTTP 오류 로그를 확인하는 것이 좋다. 이렇게 로그를 직접 확인하면 플래시만으로는 알 수 없는 서버에서 발생한 오류를 알아낼 수 있다.

데이터 처리 애플리케이션

데이터 처리 애플리케이션에서는 데이터를 받으면 우선 그 데이터를 파싱(데이터를 해석하여 필요하다면 여러 조각으로 잘게 나누는 작업)해야 한다. 데이터를 파싱하고 나면 서버 애플리케이션에서 여러 가지 방법으로 그 데이터를 조작할 수 있다. 일반적으로 데이터를 처리할 때는 나중에 쉽게 사용할 수 있도록 데이터를 데이터베이스나 일반 텍스트 파일로 저장하는 경우가 많다.

데이터 처리가 끝난 뒤에 어떤 결과가 나오면, 데이터 처리 애플리케이션에서는 그 결과를 플래시로 전달한다. 그 결과는(“정보를 제공해 주셔서 감사합니다”와 같은) 간단한 확인 메시지일 수도 있고 데이터베이스에서 뽑아낸 레코드의 목록 또는 제품 가격표가 될 수도 있다.

애플리케이션에서 생성된 결과가 웹 서버에 전달되면 웹 서버에서는 그 데이터를 다시 플래시로 보내고, 플래시에서는 그 데이터를 저장하거나 화면에 표시할 수 있다.



데이터 처리 애플리케이션을 개발할 때는 애플리케이션에서 데이터를 처리한 결과의 MIME 유형을 'application/x-www-url-form-encoded' 으로 설정해야 한다. 이 MIME 유형을 제대로 설정하지 않으면 그 결과가 플래시에 전달되더라도 플래시에서는 그 결과를 사용할 수 없을 가능성이 높다.

플래시에서 데이터를 받아 해석하는 과정

플래시에서 데이터를 받아 해석하는 과정은 플래시 폼이 작동하는 주기의 거의 마지막 부분에 해당한다. 플래시 폼의 주기를 끝마치려면 우선 서버측 애플리케이션

션에서 데이터를 처리하고 그 결과를 보내올 때까지 기다려야 한다. 주가지수를 확인하는 플래시 무비를 만든다고 생각해 보자. 사용자가 주가 확인 심벌을 선택하고 주식 가격 정보를 가져오는 버튼을 누르고 나서 가격을 화면에 표시하려면, 우선 주가를 확인하는 애플리케이션에서 가격을 조사하고 그 값을 리턴할 때까지 기다려야 한다. 무비에서 데이터를 받을 때까지 기다리는 동안에는 화면에 로딩 중이라는 메시지를 표시해야 한다. 가격을 모두 받고 나면 무비가 다시 시작된다.

loadVariables()를 호출했을 때 서버로부터 받은 데이터는 미리 정해 둔 대상 클립 또는 레벨에 저장된다. 그 데이터를 받고 나면 플래시 폼 주기가 완료되며 플래시 폼을 통해 받은 정보를 마음대로 사용할 수 있다. 지금까지 배운 내용을 모두 모아서 간단한 사용자 입력 폼을 만들어 보자.

플래시 입력 폼 만들기

우리가 만들 예제는 앞에서 언급했던 플래시 폼의 필수 구성요소가 모두 포함되어 있긴 하지만 최대한 간단하게 구성하였다. 이 예제에서는 플래시에서 echo.pl이라는 필 스크립트로 텍스트 필드 변수를 보내는 방법과 플래시에서 echo.pl로부터 받은 데이터를 다루는 방법을 알아본다. 이 예제 파일은 온라인 코드 창고에서 구할 수 있다. 이제 본격적으로 폼을 만들어 보자.

프론트엔드 만들기

플래시에는 HTML과는 달리 폼을 만들 수 있는 기능이 내장되어 있지 않다. HTML에서는 <SELECT>, <OPTION>, <INPUT TYPE="RADIO"> 태그만 사용하면 손쉽게 풀다운 메뉴와 라디오 버튼을 만들 수 있지만, 플래시에서는 그러한 도구를 프로그래머가 직접 만들어야 한다. 플래시에 내장된 폼 관련 도구는 사용자 입력 텍스트 필드(HTML의 <INPUT TYPE="TEXT"> 또는 <INPUT TYPE="TEXTAREA">에 해당함)뿐이다.



폼 위젯은 플래시 저작 도구에 바로 들어있지는 않지만, 라디오 버튼, 체크박스, 풀다운 메뉴는 내장된 스마트 클립을 이용하여 만들 수 있다. 폼 위젯 스마트 클립은 Window → Common Libraries → Smart Clips를 선택하여 찾을 수 있다.

우리가 만들 폼에는 사용자 입력 텍스트 필드와 보내기 버튼이 들어있다. 이 두 요소를 무비 클립에 넣어서 서버로 보낼 변수를 확인하기 쉽게 만들자. 우선 다음과 같이 새로운 문서를 만들고 그 안에 formClip 무비 클립을 만든다.

1. 새로운 플래시 문서를 시작한다.
2. Insert → New Symbol을 선택한다. Symbol Properties 대화상자가 나타난다.
3. Name 상자에 formClip이라고 입력한다.
4. OK를 클릭한다.
5. 라이브러리에서 formClip의 인스턴스를 선택하여 메인 스테이지로 끌어다 놓는다.

이제 formClip에 사용자 입력 텍스트 필드를 추가해 보자.

1. 라이브러리에서 formClip 심벌을 더블클릭하여 그 클립을 편집한다.
2. 텍스트 도구를 선택한다.
3. 클립 캔버스 위에 사용자가 한 줄 정도의 텍스트를 입력할 수 있을만한 크기로 직사각형을 그린다.
4. Window → Panels → Text Options를 선택한다.
5. Text Type에서 Input Text를 선택한다.
6. Line Display에서 Single Line을 선택한다.
7. Variable 상자에 input이라고 입력한다.
8. Border/Bg를 선택한다.

이제 input 텍스트 필드의 기본값을 설정하여 사용자가 아무런 값도 입력하지 않았을 때는 기본값을 서버로 전송하도록 만들자.

1. formClip 타임라인에서 1번 프레임을 선택한다.
2. 액션 패널에 input = “”; 라는 코드를 입력한다.

마지막으로 다음과 같은 작업을 마치면 formClip에 보내기 버튼을 추가할 수 있다.

1. Window → Common Libraries → Buttons를 선택한다.
2. Buttons.fla 라이브러리에서 Push Bar 버튼을 선택하여 formClip 스테이지로 끌어다 놓는다.

서버에 데이터 보내기

플래시에서 데이터를 서버 애플리케이션으로 보내는 방법에는 loadVariables(), getURL(), loadMovie(), XML.load(), XML.sendAndLoad(), XMLSocket.send() 메소드 등이 있다. 여기에서는 loadVariables() 메소드를 사용하자. 다른 메소드에 대한 설명은 3부에 나와 있다.

조금 전에 formClip 무비 클립에 사용자 입력 텍스트 필드와 보내기 버튼을 추가했다. 다음과 같이 하면 보내기 버튼을 눌렀을 때 formClip의 변수를 echo.pl로 전달할 수 있다.

1. 라이브러리에서 formClip 심벌을 더블클릭한다(이렇게 하면 클립 편집 모드로 들어간다).
2. Push Bar 버튼 인스턴스를 선택한다.
3. Window → Actions를 선택한다.
4. 액션 패널에 아래 코드를 입력한다.

```
on (release, keyPress "<Enter>") {
    loadVariables ("http://www.yourserver.com/cgi-bin/echo.pl",
                  "_root.response",
                  "GET");

    _root.response.gotoAndStop("loading");
}
```

보내기 버튼에서는 loadVariables() 함수를 이용하여 formClip에서 echo.pl로 변수를 보낸다. 그리고 나서 response 클립에서 값을 로딩하고 있다는 메시지를 출

력한다. response 클립은 잠시 후에 알아보기로 하고 일단 loadVariables()를 호출하는 방법을 살펴보자.

loadVariables()의 첫 번째 매개변수에는 서버에 있는 echo.pl의 위치를 적어준다(cgi-bin이라는 디렉토리에서 CGI 프로그램을 돌리도록 되어 있는 서버가 꽤 많이 있기 때문에 이 예제에서도 그러한 형태의 주소를 사용하였다). 서버측 프로그램의 URL을 적을 때는 서버의 도메인 이름과 디렉토리 이름을 모두 포함시켜 정확한 위치를 입력해야 한다. loadVariables() 함수가 실행되면 formClip에 있는 모든 변수가 그 스크립트로 전송된다.

loadVariables()의 두 번째 매개변수는 echo.pl에서 처리 결과를 보낼 때 필요한 클립의 경로이다. echo.pl에서 돌려받은 값은 “_root._response”라는 변수에 저장된다.

세 번째 매개변수에는 formClip의 변수를 서버에 전달할 때 사용하는 HTTP 전송 방법을 지정해 준다. 이 경우에는 GET 메소드를 사용한다. 3부의 loadVariables() 부분에도 나와 있지만, 액션스크립트에서는 GET과 POST를 모두 지원한다.

필 스크립트, echo.pl

사용자가 무비의 formClip에서 보내기 버튼을 클릭하면 플래시는 HTTP GET 요청을 전송한다. 이 요청에 의해 echo.pl이라는 필 스크립트가 실행된다. 폼이 제대로 작동하려면 echo.pl 파일을 웹 서버의 CGI를 실행시킬 수 있는 디렉토리에 저장해야 하며, 서버의 관리자가 다음과 같은 설정을 해주어야 한다.

- 스크립트를 실행할 수 있어야 한다(파일 퍼미션을 755로 설정해야 한다).
- 유닉스의 경우에는 스크립트에 필 인터프리터의 경로가 제대로 적혀있어야 한다.¹⁾

1) 역자주: [예제 17-1]의 첫 번째 줄에 있는 주석 부분은 유닉스에서 스크립트를 실행하는 데 필요한 쉘(즉 필 인터프리터)의 경로를 나타낸다. 만약 독자가 사용하는 웹 서버에서 필 인터프리터의 위치가 다르다면 이 값을 고쳐줘야 한다. 대부분의 경우에 /usr/local/bin/perl 또는 /usr/bin/perl 로 설정되어 있다. 이 값을 확인하려면 유닉스 명령 프롬프트에서 ‘which perl’ 을 입력하면 된다.

[예제 17-1]에 echo.pl의 소스 코드가 수록되어 있다. 필에서 # 문자는 주석을 의미한다.

[예제 17-1] echo.pl 소스 코드

```
#!/usr/local/bin/perl
#-----
# 이름:      간단한 플래시 에코 프로그램
# 버전:      1.2.0
# 제작자:    Derek Clayton  derek_clayton@iceinc.com
# 설명:      플래시에서 GET이나 POST로 전송한 이름과 값을 다시 돌려보낸다.
#-----
# 메인 프로그램 부분
#-----
use CGI;          # 쉽게 파싱할 수 있도록 CGI.pm(필의 CGI 모듈) 이용
$query = new CGI;  # 질의 객체
$echoString = "output=";  # 출력 문자열 초기화
&getInput;         # 플래시에서 보내온 입력사항을 받아들임
&writeResponse;    # 플래시에 응답을 보냄
exit;              # 스크립트를 빠져나간다.
#-----
sub getInput {
    # 각 키별로 값을 받아서 에코 문자열에 추가한다.
    foreach $key ($query->param) {
        $value = $query->param($key);
        $echoString .= "$key:$value\n";
    }
    # 응답을 보내기 전에 불필요한 줄바꿈 문자(\n)를 제거한다.
    chomp($echoString);
}

sub writeResponse {
    # 플래시에서 사용할 수 있는 유형으로 설정한다.
    print "Content-type: application/x-www-urlform-encoded\n\n";
    # 에코 문자열을 출력한다.
    print $echoString;
}
```

echo.pl 스크립트에서는 다음과 같이 세 가지 작업을 처리한다.

- 플래시에서 보낸 데이터를 받아서 그 데이터를 일련의 변수 이름과 값으로 파싱한다.
- 변수의 이름과 값을 모두 모아서 플래시로 전송할 문자열을 만든다. 문자열의 형식은 다음과 같다.

```
output=name1:value1\name2:value2\n...\namen:valuen
```

- 플래시로 문자열을 보낸다.

플래시에서 echo.pl에서 보내온 문자열을 받으면, 자동으로 그 문자열을 일련의 URL-인코딩 변수(3부의 loadVariables() 부분 참조)로 해석한다. 이렇게 하면 output은 response 클립의 타임라인 변수가 된다. 플래시 무비에서 output 값을 조사하면 어떤 변수 이름과 값을 echo.pl에 보냈는지 알 수 있다.

물론 echo.pl은 일반적인 용도로 사용하기 위한 애플리케이션이 아니라 이 책에서 개념을 설명하기 위해 만든 간단한 프로그램일 뿐이다. 하지만 이런 개념을 확장시키면 매우 강력한 기능을 활용할 수 있다. 제대로 된 웹 시스템의 예를 보고 싶다면, 온라인 코드 창고의 'Server Communication' 부분에서 구할 수 있는 일반 파일을 이용한 데이터베이스 샘플을 참조하기 바란다.

서버로부터 결과 받아오기

formClip의 변수를 echo.pl로 보낼 때 echo.pl의 리턴 값을 response라는 무비 클립에 저장하도록 지정했다.

```
loadVariables ("http://www.yourserver.com/cgi-bin/echo.pl",
              "_root.response",
              "GET");
```

이제 response 클립을 만들 차례이다. 이 클립에는 대기, 로딩, 로딩 끝의 세 가지 상태가 있다. '대기(idle)' 상태에서 response 클립은 사용자가 볼 수 없는 상태가 되며 로딩을 시작할 때까지 데이터를 기다릴 뿐이다. '로딩/loading)' 상태에서는 데이터가 서버에 전송되었으며 서버로부터 결과를 받을 때까지 기다리고 있다는 것을 사용자에게 알려야 한다. '로딩 끝(done loading)' 상태는 response 클립에서 서버의 응답을 받은 상태이며 그 결과를 텍스트 필드를 통해 사용자가 볼 수 있도록

화면에 출력한다. 이와 같은 response 클립의 세 가지 상태를 통해 타임라인 구조를 구축할 수 있다. 각 상태는 idle, loading, doneLoading이라는 레이블이 있는 키프레임으로 표현한다. 다음과 같은 규칙에 따라 프레임이 화면에 표시된다.

- 무비를 로딩하고 나면 response에서는 idle 프레임을 화면에 표시한다.
- 변수를 보내고 나면 보내기 버튼에서는 response를 loading 프레임으로 이동시킨다.
- 변수를 받고 나면 response의 데이터 이벤트 핸들러에서 response 클립 자체를 doneLoading 프레임으로 이동시킨다.

response 클립은 다음과 같이 만들 수 있다.

1. Insert → New Symbol 선택. Symbol Properties 대화상자가 나타난다.
2. Name 상자에 responseClip이라고 입력한다.
3. OK를 클릭한다.
4. 라이브러리에서 responseClip의 인스턴스를 선택하여 메인 스테이지로 드래그한다.
5. responseClip 인스턴스의 이름을 response로 변경한다.
6. 라이브러리에서 responseClip 심벌을 더블클릭한다(이렇게 하면 이 심벌을 편집하는 상태가 된다).
7. 네 개의 타임라인 레이어를 만든다. 각 레이어의 이름은 (위에서 아래 순으로) scripts, labels, loading, outputField로 지정한다.
8. 각 레이어에 세 개의 키프레임을 만든다.
9. 1, 2, 3번 프레임의 labels 레이어에 각각 idle, loading, doneLoading이라는 레이블을 추가한다.
10. 1번 프레임의 scripts 레이어에 stop();이라는 코드를 추가한다.
11. 2번 프레임의 loading 레이어에 “loading, please wait”라는 텍스트를 입력한다.

12. 텍스트 도구를 선택한다.
13. 3번 프레임의 outputField 레이어에 텍스트 상자를 그린다.
14. Window → Panels → Text Options를 선택한다.
15. Text Type에서 Dynamic Text를 선택한다.
16. Line Display에서 Multiline을 선택한다.
17. Variable 상자에 outputField라고 입력한다.

이제 서버에서 플래시에 데이터를 전송하는 작업이 끝났을 때 실행될 데이터 이벤트 핸들러를 추가하자.

1. 메인 스테이지에서 response 인스턴스를 클릭한다.
2. Window → Actions를 선택한다.
3. 액션 패널에 다음과 같은 코드를 입력한다.

```
onClipEvent (data) {  
    this.gotoAndStop("doneLoading");  
    outputField = output;  
}
```

echo.pl에서 보낸 내용을 모두 전송받고 나면 response의 데이터 이벤트 핸들러가 자동으로 실행된다. 데이터 이벤트 핸들러에서는 response의 플레이헤드를 doneLoading으로 이동시키고 나서 echo.pl로부터 받은 output 변수 값을 outputField로 출력시킨다. 데이터 이벤트 핸들러를 사용하기 때문에 변수의 값을 제대로 전송받지 못한 경우에는 output 변수 값이 outputField 텍스트 필드를 통해 화면에 표시되지 않는다.

드디어 폼을 테스트할 준비가 끝났다. 무비를 .swf 형식으로 저장하고 실행시켜 input 텍스트 필드에 텍스트를 입력하고 보내기 버튼을 클릭해 보자. 폼이 제대로 동작하지 않으면 가장 먼저 서버측 스크립트가 제대로 작동하는지 알아보는 것이 좋다(서버의 오류 로그 확인). 뭔가 이상하다면 제대로 온라인 코드 창고의 예제를 받아서 확인해 보는 것도 좋은 방법이다.

연습 문제: 입력 필드의 값을 모두 삭제하는 리셋 버튼을 폼에 추가해 보자. formClip에 새로운 버튼을 하나 추가하고 그 버튼에 다음과 같은 코드를 추가하면 된다.

```
on (release) {
    input = "";
}
```

하나의 폼에 두 개 이상의 입력 필드를 추가해 보자. 앞에 나온 펄 스크립트에서는 변수의 개수에 상관없이 모든 변수의 이름과 그 값을 되돌려 주는데, 이렇게 받은 변수 이름과 값을 각 변수별로 분리하여 서로 다른 필드에 출력하는 법을 생각해 보자.

앞으로 배울 내용

이 장에서 배운 내용은 플래시에서 비교적 고급 기술에 해당하는 내용이다. 플래시 폼을 이용하면 사용자 설정이나 게시판, 카탈로그용 제품 정보를 기록하고 보여주는 것과 같은 다양한 작업을 처리할 수 있다. 온라인 코드 창고에 있는 일반 과제에 데이터를 저장하는 펄 데이터베이스를 이용하면 그러한 애플리케이션을 더 쉽게 만들 수 있다. 다음 장에서는 텍스트 필드에 대한 자세한 내용과 텍스트 필드를 플래시 폼에서 응용하는 방법에 대해 알아보자.