

접근성을 해치지 않는 자바스크립트의 사용

신현석

2006년 5월

자바스크립트에 대한 잘못된 생각

자바스크립트 없이는 개발을 할 수 없다?

많은 개발자들이 자바스크립트가 없이는 개발을 할 수 없다고 생각한다. 그만큼 자바스크립트를 일상적으로 많이 사용하고, 자바스크립트에 대해서 잘못 오해하고 있는 부분이 많다는 것이다. 자바스크립트는 클라이언트 환경에서 작동하기 때문에 핵심로직에 사용해서는 안된다. 핵심적인 로직은 서버사이드 언어로 처리를 하고 자바스크립트는 그 옆에서 UI구성을 위해서 도와주기만 해야 한다. 바꿔 말하면 자바스크립트가 없어도 핵심 로직은 작동을 해야 한다. 먼저 자바스크립트없이 HTML과 서버사이드 언어만으로 완벽하게 작동 할 수 있게 개발을 끝낸 후 자바스크립트를 이용해서 좋은 UI나 유효성 검사 등을 하는 부분을 추가 하는 순서로 만들어야 한다. 만약 이를 어기고 자바스크립트를 기능의 일부로 사용하면 접근성이 떨어지는 것 뿐만 아니라 보안도 떨어지고 데이터의 무결성도 보장되지 않는다.

자바스크립트가 작동하지 않는 환경도 고려를 해 줘야 하는가?

자바스크립트가 작동하지 않는 환경이 일반적인 상황은 아니지만 스크립트를 사용할 수 없는 환경에 대한 고려는 반드시 필요하다. 실제로 브라우저에 따라서는 자바스크립트 버전이 다를 수 있고 개발 환경에서는 잘 작동하던 기능이 어떤 사용자 환경에서는 작동하지 않을 수도 있다. 그리고 사용자 중에서도 여러 이유로 스스로 자바스크립트 사용을 중지하고 사용하는 경우도 있다. 자바스크립트를 사용하지 않아도 핵심 기능을 구현할 수 있음에도 불구하고 불필요한 자바스크립트를 사용하여 접근성을 떨어뜨리는 것이 올바른 페이지 제작 방법은 아니다.

서버 부하를 줄이기 위해서 자바스크립트로만 유효성 검사를 한다.

자바스크립트를 이용해서 서버의 부하를 줄일 수 있다는 것은 많은 사람들이 알고 있을 것이다. 자바스크립트로 선처리를 행함으로써 네트워크나 서버의 처리를 감소시킨다는 것인데 이것을 잘못 이해하여 자바스크립트만으로 유효성을 체크하는 것이 좋다고 생각하는 사람들이 있다. 자바스크립트로 선처리를 한결과물을 그대로 믿으면 안된다. 사용자측으로 부터 넘어온 모든 값은 그 값의 유효성을 완전히 보장할 수 없기 때문에 서버측에서도 유효성 체크를 해 줘야만 한다. 이것은 서버의 부하를 높이는 것 이전에 자료의 무결성을 보장하는 중요한 과정이다. 자바스크립트를 거치지 않고 서버에 임의로 변조된 값을 보내는 일은 아주 쉬운 일이다.

자바스크립트를 사용하지 않으면 UI가 불편해 진다.

자바스크립트를 이용해서 RIA나 좋은 UI를 만들 수 있는 것은 사실이지만 웹환경을 이해하고 일반적인 웹 인터페이스를 이용해서 만드는 것이 더 좋은 UI를 제공할 수 있다. 사용자 친화적인 UI에 대한 고민을 해야지 자바스크립트를 사용했다고 좋은 UI가 나오는 것은 아니다. 실제로 가장 사용자가 이해하기 쉬운 UI는 사용자 OS나 브라우저에서 기본적으로 제공하는 컨트롤 인터페이스를 그대로 사용하는 것이고 이는 자바스크립트 없이도 충분히 가능하다.

자바스크립트는 사용자 환경에 내려보내지고 사용자의 브라우저에서 실행이 되기 때문에 이에 의존해서 프로그램을 만들게 되면 보안, 접근성에 문제가 생기기 쉽다. 이러한 자바스크립트의 특징을 정확히 이해하지 못한 잘못된 개발 방법이나 상식들 때문에 자바스크립트가 오용되고 있는 경우가 많다. 자바스크립트는 어디까지나 보조적인 수단이라는 것을 이해하고 핵심적인 기능을 해치지 않는 한도에서 사용하는 것이 무엇보다 중요하다.

자바스크립트 선언

```
<script language="Javascript">
//code
</script>
```

자바스크립트는 `<script>` 엘리먼트로 선언을 한다. 모든 `<script>` 엘리먼트는 `type`을 명시해 주어야 하고 자바스크립트의 `type`은 "text/javascript" 이다. 많은 경우 `language` 만을 선언해서 자바스크립트 버전을 명시하는데 반드시 `type`도 같이 명시를 해 주어야 한다.

```
<script type="text/javascript">
//code
</script>
```

<a>의 href 속성과 자바스크립트의 사용

`href`는 Hypertext REference의 약자이다. 다시 말해서 hypertext의 위치를 나타내는 uri를 그 값으로 갖는다. 하지만 많은 경우 이 `href`안에 "javascript:myFunction()"과 같이 잘못된 구문을 이용하는 것을 볼 수 있다. 이와 같이 `href`안에 잘못된 값이 들어가게 될 경우, 북마크나 새창, 새탭으로 열기 등의 `href` 관련된 브라우저의 기능들이 정상적으로 작동하지 않게 된다. 따라서 `href`안에는 항상 uri가 들어가도록 하고 자바스크립트 적용은 `onclick`과 같은 이벤트 속성을 이용해야 한다.

의미 없는 href 값을 사용한 경우

사용자의 링크 클릭이 링크와 관련이 있고 이를 자바스크립트를 이용해서 처리를 해야 하는 경우가 있다면 우선은 자바스크립트를 빼고도 페이지의 이동을 할 수 있게 `href`에 적절한 값을 넣어 주어야 한다. 이와 같은 경우로 탭메뉴를 들 수 있다. 탭메뉴의 경우 탭을 누르면 해당 탭과 관련있는 콘텐츠를 보여주는 식으로 작동하게 된다. javascript가 없다면 탭을 클릭했을 때 해당 콘텐츠로 이동을 하는 식으로 구현 되면 된다. `href` 안의 값으로 페이지 안에서의 해당 콘텐츠 앵커 주소를 넣는 것으로 간단히 구현 된다.

```
<a href="#notice-list"></a>
```

그리고 이 마크업을 기본으로 하여 이벤트 속성으로 원하는 기능을 넣어주면 된다.

```
<a href="#notice-list" onclick="showNoticeTab(); return false;"></a>
```

onclick으로 탭을 보여준 후 false를 리턴하여 해당 앵커로 이동하지 않도록 처리 한다. 자바스크립트가 작동을 할 때에는 클릭하면 정상적으로 탭으로 작동을 할 것이고, 그렇지 않을 경우에는 해당 콘텐츠로 이동을 하여 높은 접근성을 유지할 수 있다.

만약 자바스크립트가 링크와 관련이 없는 경우에는 <a>태그를 사용해서 스크립트를 적용하면 안 된다. 보통 특정 효과를 주는 것이 이러한 것에 해당하게 되는데 이 경우 자바스크립트가 작동을 하지 않아도 콘텐츠 이해에 크게 문제가 되지 않는 경우이다.

```

```

효과를 위한 자바스크립트는 단순히 onclick을 이용해서 적용을 하고 <a>를 사용하지 않는다. 그리고 사용자가 마우스 포인터를 올렸을 때 손모양으로 나오는 것은 스타일로 처리하면 된다.

팝업창을 열 때(window.open)

```

<a href="#"></a>
```

href에 #과 같은 의미 없는 값을 넣거나 onclick 안에 경로를 처리하는 경우가 있는데 팝업창은 링크이고 페이지가 별도로 존재 하기 때문에 <a>를 이용해서 기능을 구현하고 href엔는 해당 팝업의 경로를 넣어야 한다.

```
<a href="popup.html" onclick="window.open(this.href, 'popupName', 'width=300,height=200'); return false;"></a>
```

이럴 경우 사용자가 자신의 의도 대로 팝업창을 새창, 새탭 등으로 열 수 있고 심지어 즐겨 찾기도 할 수 있다.

자바스크립트를 이용한 페이지 이동

웹사이트를 이용하다 보면 폼에서 값을 입력하고 서밋을 하는 순간 "따다다닥" 하는 식으로 클릭을 여러번 한 것과 같은 소리가 나는 경우를 접하게 된다. 프로세스가 진행 되는 동안 여러페이지를 걸쳐서 일어나게 되는데 이 처리를 자바스크립트로 처리를 해서 이러한 현상이 발생을 하게 된다.

```
<script type="text/javascript">document.location.href="redirection.html";</script>
```

이렇게 출력을 하거나 <form>을 이용해서 값을 넘기는 경우가 이러한 경우이다.

```
<form name="login_form">
  <input type="hidden" name="user_id" value="myid" />
  <input type="hidden" name="user_pwd" value="mypassword" />
  <input type="hidden" name="redirect_url" value="http://mysite.com/login/" />
```

```

    <input type="hidden" name="somevalue" value="blahblah" />
    ...
</form>
<script type="text/javascript">
f = document.forms.login_form;
f.action = "http://login.oursite.com/login/";
...
f.submit();
</script>

```

심한 경우 아래와 같이 전혀 의미 없는 폼을 이용하기도 한다.

```

<form method="post" name="sg_form" action="http://www.qubi.com/" target="_top">
</form>
<script> sg_form.submit(); //3</script>

```

위와 같은 페이지들은 html 문법상 오류가 있는 페이지 들이고 이 때문에 작동이 안 될 수도 있다. <form>엘리먼트나 <script>엘리먼트는 상위에 <body>나 <head>엘리먼트가 있어야 하는데 위와 같은 경우 이러한 엘리먼트가 없기 때문에 html로 해석이 안되어 스크립트가 작동 되지 않거나 값이 넘어가지 않을 수도 있다. 그리고 <form>에 submit <input>이 없기 때문에 자바스크립트로 submit이 일어나지 않을 수도 있다.

이와 같이 페이지를 이동하거나 값을 넘길 필요가 있을 때 자바스크립트에 의존해서 이를 처리하게 되면 클라이언트의 환경에 따라서 동작이 실패할 수 있다. 따라서 이러한 처리는 자바스크립트에서 처리 하지 말고 서버 측에서 http헤더 정보를 이용해서 처리해야 한다.

이러한 중간과정에서의 처리를 서버측에서 모두 처리 하는 것이 가장 바람직 하지만 어쩔 수 없이 사용을 해야할 경우에는 - 그럴 경우가 많지는 않겠지만 기존의 호환성을 위해서 - DTD선언이나 <html> 루트 엘리먼트, <head>, <body>와 같이 필수 엘리먼트들이 존재하는 완결된 페이지를 사용 하도록 하고, 자바스크립트가 작동하지 않는 경우를 위해서 <form>에 submit버튼도 제공을 하고, 결과 메시지도 alert외에 일반 text와 <a>를 이용한 링크를 제공하도록 해야 한다.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=euc-kr" />
<title>Redirect</title>
</head>
<body>
<script type="text/javascript">
/*
some processes...
*/
alert('이래저래한 이유로 다시 돌아 갑니다. ');
document.location.href="redirection.html";
</script>
<p><a href="redirection.html">이래저래한 이유로 다시 돌아 갑니다.</a></p>
</body>
</html>

```

charset이 없는 경우 브라우저의 기본 설정으로 alert이 작동하기 때문에 브라우저의 기본 설정이 ko-kr이 아닌 경우 한글이 깨지게 된다. 문서의 mime-type도 text/html인지 확인해야 한다.

<form>에서의 자바스크립트 사용

일반적으로 <form>은 사용자가 입력한 값을 서버측에 전달하는 역할을 하고 그 전달은 <form>의 submit 기능을 통해서 이루어 진다. 이러한 폼을 구현 할 때에 일반적으로 자바스크립트를 사용하는 경우가 많다.

submit

폼은 그 자체적으로 값을 보내는 서밋기능을 제공하고 있다. <input type="submit" />이나 <input type="image" />이 서밋기능을 하는 컨트롤인데 폼의 값 유효성 체크를 하는 과정에서 이러한 폼의 자체적인 서밋기능을 이용하지 않고 자바스크립트로 폼 서밋을 하는 경우가 있다.

```
<script type="text/javascript">
function submitForm() {
    loginForm.submit();
}
</script>
<form id="loginForm" name="loginForm" action="">
    User Id <input type="text" name="loginId">
    User Password <input type="password" name="loginPassword"><br>
    
</form>
```

위와 같은 경우 폼에 서밋기능이 없기 때문에 브라우저에 따라서 script로 서밋이 안되는 경우도 있고 서밋 대신에 이미지가 들어가 있기 때문에 의미적으로도 맞지 않게 된다. 따라서 폼을 제작 할 때에는 반드시 서밋기능을 <input>을 이용해서 제공해야 한다.

```
<form id="loginForm" name="loginForm" action="">
    <p>
        <label for="loginId">User Id</label>
        <input type="text" id="loginId" name="loginId" /><br />
        <label for="loginPassword">User Password</label>
        <input type="password" id="loginPassword" name="loginPassword" />
    </p>
    <p>
        <input type="image" src="login.gif" alt="Login" />
    </p>
</form>
```

많은 경우 <html>제작 과정에서 서밋을 로 넣는 경우가 있는데 이 경우 이를 그냥 사용하면 안되고 적절한 <input>으로 바꾸어서 사용해야 한다.

validation

클라이언트 측에서 자바스크립트를 이용한 유효성 검증은 <form>의 서밋 이벤트를 캐치하는 방식으로 구현해야 하고 자바스크립트가 폼을 서밋하는 방식을 사용하면 안된다.

```
<script type="text/javascript">
function submitForm() {
    if (!loginForm.loginId.value) {
        alert("아이디를 넣어주세요.");
    }
}
```

```

        loginForm.loginId.focus();
    } else if (!loginForm.loginPassword.value) {
        alert("비밀번호를 넣어주세요.");
        loginForm.loginPassword.focus();
    } else {
        loginForm.submit();
    }
}
</script>
<form id="loginForm" name="loginForm" action="">
    아이디 <input type="text" name="loginId">
    비밀번호 <input type="password" name="loginPassword"><br>
    
</form>

```

위와 같은 경우 자바스크립트만을 이용해서 폼을 서밋하고 있기 때문에 자바스크립트가 없이 HTML만으로는 기능이 작동하지 않는다.

```

<script type="text/javascript">
function submitForm(formEl) {
    //TrimAll(formEl);

    var errorMessage = null;
    var objFocus = null;

    if (formEl.loginId.value.length == 0) {
        errorMessage = "아이디를 넣어주세요.";
        objFocus = formEl.loginId;
    } else if (formEl.loginPassword.value.length == 0) {
        errorMessage = "비밀번호를 넣어주세요.";
        objFocus = formEl.loginPassword;
    }

    if(errorMessage != null) {
        alert(errorMessage);
        objFocus.focus();
        return false;
    }
    return true;
}
</script>

<form id="loginForm" name="loginForm" action="" onsubmit="return submitForm(this)">
    <label for="loginId">아이디</label> <input type="text" id="loginId" name="loginId"
/>
    <label for="loginPassword">비밀번호</label> <input type="password" id="loginPassword"
name="loginPassword" /><br />
    <input type="image" src="login.gif" alt="Login" />
</form>

```

이와 같이 onsubmit 이벤트를 이용해서 폼의 유효성을 체크하고 그 결과를 true나 false로 보내줌으로써 폼이 스크립트에 따라서 서밋을 진행하거나 멈출 수 있고, onsubmit 이벤트가 발생하지 않아도 사용자는 폼을 사용할 수 있기 때문에 접근성이 높아지게 된다.

게시판 등에서 기능을 모두 자바스크립트로 하는 경우

게시판에서 페이지의 이동을 자바스크립트만으로 하는 것을 많이 볼 수 있다. 아래와 같이 사용자 인풋이 없는 빈 `<form>`을 하나 만들고 이것과 자바스크립트를 이용해서 글을 읽거나 페이지를 이동하는 것이다.

```
<form method="post" name="vars">
  <input type="hidden" name="articleId" value="23" />
  <input type="hidden" name="page" value="3" />
  <input type="hidden" name="keyord" value="" />
  <input type="hidden" name="searchType" value="" />
  <!-- 등등 -->
</form>

...

<a href="javascript:ArticleRead()">글읽기</a>
<a href="javascript:GoList()">리스트 보기</a>
```

url이 간단해 지고 다루기 쉽다는 이유로 이러한 방식으로 개발을 하는 경우가 많은 것 같은데 절대로 사용해서는 안되는 방식이다.

우선 위와 같이 모든 기능을 자바스크립트를 이용해서 구현을 하게 되면 자바스크립트오류가 있거나 핸드폰과 같이 자바스크립트가 정상적으로 작동 하지 않는 상황에서는 접근을 할 수 없게 된다. 또한 모든 변수를 post를 통해서 전달하기 때문에 url에 표시가 되지 않고 해당 페이지를 따로 북마크 한다든지 저장을 할 수가 없게된다. 게시판의 경우 해당 게시물로의 접근을 쉽게 해주어야 하는데 사용자가 url을 알 수 없게 함으로써 접근을 원천적으로 막게 된다.

자바스크립트를 이용하지 않고도 `<a>`와 url만으로도 작동 가능한 페이지를 만들고 QueryString을 효율적으로 다루는 방법을 모색하여 개발을 진행하는 것이 가장 좋은 방법이다.

결론

본인이 웹사이트 개발을 하면서 접할 수 있었던 몇몇 잘못된 자바스크립트사용 예를 적어 보았다. 그런데 놀라운 것은 이러한 개발 방식을 초보부터 어느정도 경력이 된 개발자들 까지 아무런 고민 없이 사용하고 있다는 것이다. 책이 잘못된 것인지 교육이 잘못 된 것인지 정확히 근원을 알 수는 없지만 많은 개발자들이 웹이 가지고 있는 기본적인 특성을 무시한채 잘못된 방법을 이용하여 개발을 하고 있는 것이 사실이다. 그리고 또 초보 개발자들은 HTML을 잘 모르기 때문에 이러한 개발 방식을 아무 고민 없이 그냥 받아들이고 있는 것이 사실이다.

브라우저 접근성을 가장 크게 낮추고 있는 것이 바로 이 잘못된 자바스크립트의 사용이다. 부디 조금이라도 많은 개발자들이 링크에 기초한 웹의 특성을 이해하고 접근성 높은 웹사이트, 웹 어플리케이션을 구축하기를 바라는 바이다.

[이 글에 대한 의견](#)