NOKIA Developer

Canceling the default action in Browser 7.1

Overview

On devices using Browser 7.1, using `{{{1}}}` without preventing the default action causes the page to scroll back to the start. This will cause navigational problems especially in widgets using the tabbed navigation mode.

This article demonstrates how the default action can be canceled, depending on how events are handled. Canceling the default action prevents the browser searching for the `#` anchor, so that the page view position is not changed.

Inline event handler

The inline event handler is the oldest way to attach events to elements.

```
<a id="inline" href="#" onclick="inlineEvent();">scroll</a>
```

However, the empty `#` anchor causes the browser to scroll the page back to top. For inline event handlers, the default action can be canceled by simply returning a false `onclick` event.

```
<a id="inline" href="#" onclick="inlineEvent(); return false;">No scroll</a>
```

Traditional event handler

In a traditional event handler, the most of the work is done on the JavaScript side. The default action can be canceled by simply returning `False` from the event handler.

```
var elFalse = document.getElementById("elFalse");
elFalse.onclick = function(){
  alert("page not scrolled. False returned");
  return false;
}
```

```
<a id="elFalse" href="#">click me</a>
```

DOM level 2 event handler

The DOM level 2 event handler allows multiple event handlers to attach to the same event. In addition, it provides a more sophisticated control for canceling the default action.

```
var elDOMII =document.getElementById("elDOMII");
elDOMII.addEventListener( "click", iwasclicked, false );

function iwasclicked(event) {
 event.preventDefault();
 //event.stopPropogation();
```

```
 alert("page not scrolled default action prevented");
}
```

```
<a id="elDOMII"  href="#">Event Link dom 2 </a>
```

JavaScript pseudo-URL

It is also possible to use the JavaScript pseudo-URL as the `href` target. However, the `javascript:` protocol is not a public standard. It functions similarly to writing a JavaScript directly in the browser's address bar: `javascript:alert("Try me!");`

Remember that javascript pseudo-URL should only be used in places where you would normally enter an URL.

```
function pseudourl() {
 alert("called from pseudourl");
}
```

```
<a href="javascript:pseudourl();">click me</a>
<code>[[Category:JavaScript]][[Category:S60 3rd Edition FP2]][[Category:S60 5th
Edition]]
```