

```

1  package org.wicket.sapporo.handson.ajax;
2
3  import org.apache.wicket.ajax.AjaxRequestTarget;
4  import org.apache.wicket.ajax.markup.html.AjaxLink;
5  import org.apache.wicket.markup.html.WebMarkupContainer;
6  import org.apache.wicket.markup.html.WebPage;
7
8  /**
9   * Ajaxでコンポーネントの表示・非表示を設定するページの例.
10  */
11  public class VisibleChangePage extends WebPage {
12      private static final long serialVersionUID = 1L;
13
14      /**
15       * コンストラクタ.
16       */
17      public VisibleChangePage() {
18          // 匿名クラスの内部から参照される変数はfinal宣言が必要
19          final WebMarkupContainer green = new WebMarkupContainer("visible") {
20              private static final long serialVersionUID = 1L;
21
22              @Override
23              protected void onInitialize() {
24                  super.onInitialize();
25                  // Ajaxの更新対象となるコンポーネントには setOutputMarkupId(true) を実行する.
26                  // これにより、JavaScriptがフックするための id がタグに加えられる.
27                  setOutputMarkupId(true);
28                  // 非表示 (visible = false) 状態から表示 (visible = true) になる可能性があるタグには、
29                  // setOutputMarkupPlaceholderTag(true) を実行する。これにより、表示上は消えても、JavaScriptが
30                  // フックできるタグが残る.
31                  setOutputMarkupPlaceholderTag(true);
32                  setVisible(false);
33              }
34
35              @Override
36              protected void onConfigure() {
37                  super.onConfigure();
38                  // コンポーネントがAjaxなどで更新される度に実行されるメソッド.
39                  // ここでは、コンポーネントの表示/非表示状態を切り替えている.
40                  setVisible(!isVisibleInHierarchy());
41              }
42          };
43          add(green);
44
45          add(new AjaxLink<Void>("link") {
46              private static final long serialVersionUID = 1L;
47
48              @Override
49              public void onClick(AjaxRequestTarget target) {
50                  // AjaxLinkがクリックされた時の処理
51                  target.add(green);
52              }
53          });
54      };
55  }
56
57  }

```