

Yahoo! Ajax UI - Design Pattern Library

Color Picker





R H

G S

B V

#

RGB Slider





RGB

Hex: #

```
var hue; var picker; var gLogger; var dd1, dd2; var r, g, b; function
init() { if (typeof(ygLogger) != "undefined") {
ygLogger.init(document.getElementById("logDiv")); gLogger = new
ygLogger("slider.php"); } rgblnit(); pickerlnit(); } // Picker
----- function
pickerlnit() { hue = YAHOO.widget.Slider.getVertSlider("hueBg",
"hueThumb", 0, 180); hue.onChange = function(newVal) {
```

```
hueUpdate(newVal); }; picker =
YAHOO.widget.Slider.getSliderRegion("pickerDiv", "selector", 0,
180, 0, 180); picker.onChange = function(newX, newY) {
pickerUpdate(newX, newY); }; hueUpdate(); dd1 = new
YAHOO.util.DD("pickerPanel"); dd1.setHandleElId("pickerHandle");
dd1.endDrag = function(e) { // picker.thumb.resetConstraints(); //
hue.thumb.resetConstraints(); }; } window.onload = init; function
pickerUpdate(newX, newY) { pickerSwatchUpdate(); } function
hueUpdate(newVal) { var h = (180 - hue.getValue()) / 180; if (h ==
1) { h = 0; } gLogger.debug("hue " + hue.getValue()); var a =
YAHOO.util.Color.hsv2rgb(h, 1, 1);
document.getElementById("pickerDiv").style.backgroundColor =
"rgb(" + a[0] + ", " + a[1] + ", " + a[2] + ")"; pickerSwatchUpdate(); }
function pickerSwatchUpdate() { var h = (180 - hue.getValue()); if
(h == 180) { h = 0; } document.getElementById("pickerhval").value
= (h*2); h = h / 180; gLogger.debug("h " + hue.getValue()); var s =
picker.getXValue() / 180;
document.getElementById("pickersval").value = Math.round(s *
100); gLogger.debug("s " + s); var v = (180 - picker.getYValue()) /
180; document.getElementById("pickervval").value = Math.round(v
* 100); gLogger.debug("v " + v); var a = YAHOO.util.Color.hsv2rgb(
h, s, v);
document.getElementById("pickerSwatch").style.backgroundColor
= "rgb(" + a[0] + ", " + a[1] + ", " + a[2] + ")";
document.getElementById("pickerrval").value = a[0];
document.getElementById("pickergval").value = a[1];
```

```

document.getElementById("pickerbval").value = a[2];
document.getElementById("pickerhexval").value =
YAHOO.util.Color.rgb2hex(a[0], a[1], a[2]); } // RGB slider
----- function
rgbInit() { r = YAHOO.widget.Slider.getHorizSlider("rBG", "rthumb",
0, 128); r.onChange = function(newVal) { listenerUpdate("r",
newVal*2); }; g = YAHOO.widget.Slider.getHorizSlider("gBG",
"gthumb", 0, 128); g.onChange = function(newVal) {
listenerUpdate("g", newVal*2); }; b =
YAHOO.widget.Slider.getHorizSlider("bBG", "bthumb", 0, 128);
b.onChange = function(newVal) { listenerUpdate("b", newVal*2); };
initColor(); dd2 = new YAHOO.util.DD("rgbPanel");
dd2.setHandleElId("rgbHandle"); dd2.endDrag = function(e) { //
r.thumb.resetConstraints(); // g.thumb.resetConstraints(); //
b.thumb.resetConstraints(); } } function initColor() { var ch = " "; d
= document.createElement("P"); d.className = "rb";
r.getEl().appendChild(d); d = document.createElement("P");
d.className = "rb"; g.getEl().appendChild(d); d =
document.createElement("P"); d.className = "rb";
b.getEl().appendChild(d); for (var i=0; i

```



야후에서 제공하는 Ajax 기반의 [디자인 패턴 라이브러리](#)(Yahoo! Ajax UI Design Pattern Library)입니다. 드래그 앤 드롭 예제(Drag and Drop - Slider Widgets) 중 '컬러피커'를 자바스크립트 구현한 것입니다. 로드 하는 스크립트는 슬라이더, 드래그 앤 드롭, 컬러, 휴, 돔, 이벤트, 로그의 JS

페이지를 사용합니다. 로딩이 약간 무겁게 느껴지지만, 작동은 매우 원활한편 입니다.

역시 대기업에서 만든 라이브러리는 뭔가 다르긴 다르네요. 깔끔하고 간결하며, 어느 환경에서나 잘 작동하도록 세분화가 잘 되어있습니다. 다만 지나치게 세분화 되어있어 파악하는데에 적지 않은 시간을 투자해야 합니다. 또한 약간의 속성을 변경하려면 스크립트를 다시 짜야하는 수준입니다. script.aculo.us처럼 효과별 속성을 지원해 주었으면 하는 바램입니다. 완성되기만 하면 최적화가 너무 잘되어 정말 좋지만, 만들기가 너무 힘들군요. 저에겐 그림의 떡입니다.

※ 이 페이지는 RSS리더기에서 비정상적으로 보일 수 있으며 정상 작동하지 않을 수 있습니다.