

# software studio

**listener example: star widget**

**Daniel Jackson**

# desiderata

## behavior

- › stars react without reloading page
- › highlight stars when mouse hovers
- › cancel by clicking on selected star

## modularity

- › multiple star selectors on one page without copying
- › clean HTML/Javascript separation
- › degrades smoothly when no JS support? no

## context

- › use within form: setting star widget sets value of element

# a usage

```
<!doctype html>
<html>
<head>
  <link rel="stylesheet" href="stars.css" />
  <script src="star-widget.js"></script>
</head>
<body>
  <form action=rating method=GET>
    <input id=ratingInput type=hidden name=rating
      class=starwidget value=3>
    <input type=submit value="Submit">
  </form>
</body>
</html>
```

- › a usage of the API

# how widget behaves



*initially, all greyed out*

*hover: stars show red*

*click: stars go yellow*

*click on selection: reset*

© source unknown. All rights reserved. This content is excluded from our Creative Commons license. For more information, see <http://ocw.mit.edu/fairuse>.

**`http://...rating?rating=3`**

*click on submit*

# inserting star widgets into DOM

```
window.onload = function () {  
  // find all elements with class starwidget  
  var elements = document.getElementsByClassName("starwidget");  
  each(elements, function (elt) {  
    // if input element, then make editable and set form value updater  
    var editable = (elt.tagName == "INPUT");  
    var set_value = function (v) {elt.value = v;};  
    // get element value to initialize stars; set to zero if missing  
    var initial_value = elt.getAttribute("value");  
    if (initial_value == null)  
      initial_value = 0;  
    // create star widget  
    var stars = makeStarWidget(elt, initial_value, editable, set_value);  
    // insert just before the element  
    elt.parentNode.insertBefore(stars, elt);  
  });  
}
```

# constructing a star widget (1)

```
function makeStarWidget (sibling, initial_value, editable, set_value) {
  var star_element = document.createElement("div");
  star_element.className = "stars";
  var stars = [];
  var value = initial_value;
  fromTo(0, 4,
    (function (i) {
      var star = document.createElement("span");
      star_element.appendChild(star);
      if (i >= initial_value)
        star.className = "star star-basic";
      else
        star.className = "star star-on";
      stars[i] = star;
      var setPrefixClass = function (names) {
        fromTo (0, i,
          function (j) {stars[j].className = names;}
        );
      };
      // if star widget is just for display, don't add listeners
      if (!editable) return;
      // ... attach listeners here
    }));
  return star_element;
}
```

# constructing a star widget (2)

```
...
fromTo(0, 4,
  (function (i) {
    var star = document.createElement("span");
    ...
    star.addEventListener("mouseover", function () {
      if (0 == value)
        setPrefixClass("star star-hover");
    });
    star.addEventListener("mouseout", function () {
      if (0 == value)
        setPrefixClass("star star-basic");
    });
    star.addEventListener("mousedown", function () {
      if (i + 1 == value) {
        value = 0;
        set_value(0);
        setPrefixClass("star star-basic");
      } else if (0 == value) {
        value = i + 1;
        set_value(i + 1);
        setPrefixClass("star star-on");
      }
    });
  }));
return star_element;
}
```

MIT OpenCourseWare  
<http://ocw.mit.edu>

6.170 Software Studio  
Spring 2013

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.