



# The Yahoo! User Interface Library

**Simon Willison**

XTech Ajax Developer's Day

16th May, 2006



# The Yahoo! User Interface Library

- A library of reusable JavaScript components
- Over a year of internal development
- Open-sourced (BSD) in February
  - First external bug report within 24 hours!
- Designed for Yahoo! sized problems:
  - Building new applications
  - Integration with legacy code



# Yahoo! UI Library approach

- Full, detailed API documentation
- Abstract away browser differences
- YAHOO namespace to avoid collisions
- Doesn't try to fix JavaScript, or enforce a particular coding style
- Focus on reusable core components
- Verbose comments and documentation; code is minified for production



# Library components

## controls

autocomplete

calendar

container

menu

slider

treeview

---

animation

dragdrop

dom

event

connection

## utilities



- CSS
  - reset
  - font
  - grids
- Design Patterns Library
  - <http://developer.yahoo.com/ypatterns/>





# Getting stuff

---

`document.getElementById ?`



# Getting stuff

```
document.getElementById ?
```

```
YAHOO.util.Dom.get()
```

- Can take a string, an element or an array

```
var $ = YAHOO.util.Dom.get;
```





# getElementsBy

```
function isYahooLink(el) {  
    var re = /www\.yahoo\.com/;  
    return el.href && re.test(el.href);  
}
```

```
YAHOO.util.Dom.getElementsBy(  
    isYahooLink  
);
```

```
YAHOO.util.Dom.getElementsBy(  
    isYahooLink, 'a', 'content'  
);
```



# getElementsByClassName

```
getElementsByClassName: function(  
    className, tag, root) {  
    var re = new RegExp('(?:^|\\s+)' +  
        className + '(?:\\s+|$)');  
    var method = function(el) {  
        return re.test(el['className']);  
    };  
    return this.getElementsBy(  
        method, tag, root  
    );  
}
```



# Class manipulation

```
YAHOO.util.Dom.hasClass(el, className)
```

```
YAHOO.util.Dom.addClass(el, className)
```

```
YAHOO.util.Dom.removeClass(el, className)
```

```
YAHOO.util.Dom.replaceClass(  
    el, oldClassName, newClassName  
)
```



# Style manipulation

---

```
setStyle(el, property);
```

```
getStyle(el, property);
```



# coordinates

```
var pos = YAHOO.util.Dom.getXY(el);  
  
alert(pos[0] + ', ' + pos[1]);  
  
YAHOO.util.Dom.setXY(el, [x, y]);  
  
YAHOO.util.Dom.setXY(el, pos);
```





## The problem

- Browser event handling differs wildly between Internet Explorer and others
- Safari has some weird bugs of its own
- Event handlers in IE are a frequent cause of memory leaks
- ... but event handling is core to writing interactive JavaScript



# addListener

```
function myCallback(e) {  
    alert('Something was clicked');  
}
```

```
YAHOO.util.Event.addListener(  
    el, 'click', myCallback  
);
```

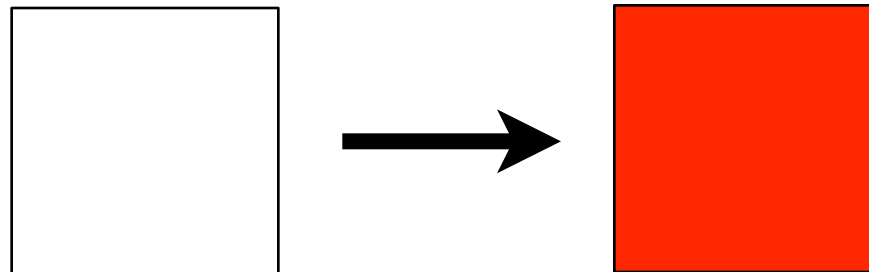
```
YAHOO.util.Event.addListener(  
    el, 'click', function(e) {  
        alert('Something was clicked');  
    }  
);
```





# Callback scope correction

```
YAHOO.util.Event.on(  
    'mydiv', 'click', function(e) {  
        this.style.backgroundColor = 'red';  
    }  
);
```





# Extra callback arguments

```
function msgAlert(e, msg) {  
    alert(msg);  
}
```

```
YAHOO.util.Event.on(  
    'mydiv', 'click', msgAlert,  
    "My div was clicked"  
);
```



# Assign before availability

```
YAHOO.util.Event.on(  
    'mydiv', 'click', myCallback  
);
```

```
YAHOO.util.Event.onAvailable(  
    'mydiv', function() {  
        alert('mydiv has become available');  
    }  
);
```



# Event utility methods

```
YAHOO.util.Event.getCharCode(ev)
YAHOO.util.Event.getPageX(ev)
YAHOO.util.Event.getPageY(ev)
YAHOO.util.Event.getXY(ev)
YAHOO.util.Event.getTarget(ev)
YAHOO.util.Event.getRelatedTarget(ev)
YAHOO.util.Event.stopPropagation(ev)
YAHOO.util.Event.preventDefault(ev)
YAHOO.util.Event.stopEvent(ev)
YAHOO.util.Event.getTime(ev)
```



# Custom events

```
var myEvent = new YAHOO.util.CustomEvent(  
    'myEvent'  
);
```

```
myEvent.subscribe(function() {  
    alert('event fired');  
});
```

```
myEvent.fire();
```





# asyncRequest

```
YAHOO.util.Connect.asyncRequest(  
    'GET', '/ajaxy-goodness', {  
        success: function(o) {  
            alert(o.responseText);  
        },  
        failure: function(o) {  
            alert('Request failed: ' +  
                o.statusText);  
        }  
    }  
);
```



# Changing the scope

```
YAHOO.util.Connect.asyncRequest(  
    'GET', '/ajaxy-goodness', {  
        success: myObject.onSuccess,  
        failure: myObject.onFailure,  
        scope: myObject  
    }  
);
```

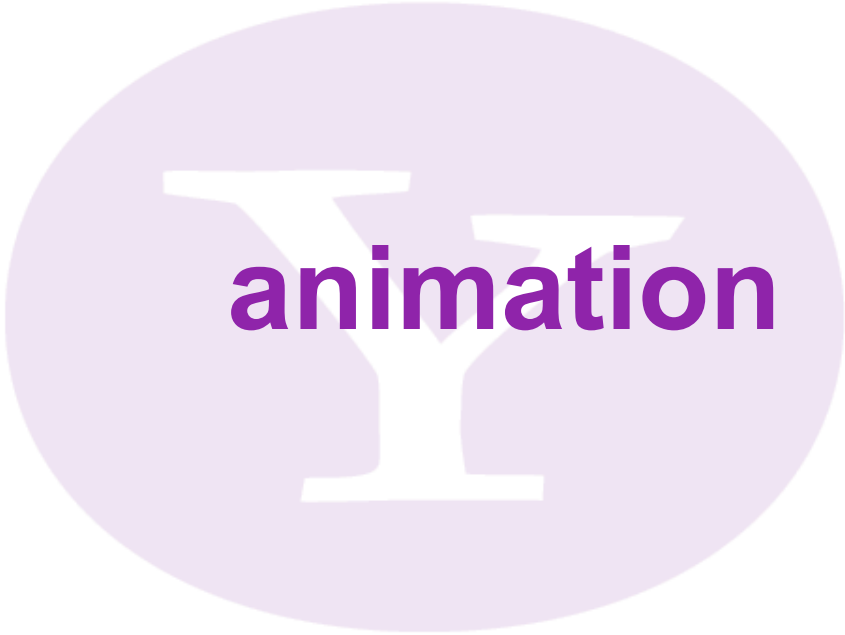




# Callback arguments

```
function onSuccess(o) {  
    alert(o.argument);  
}
```

```
YAHOO.util.Connect.asyncRequest(  
    'GET', '/ajaxy-goodness', {  
        success: onSuccess,  
        argument: 'some extra data'  
    })  
);
```





# Beautiful expressive API

```
var anim = new YAHOO.util.Anim(el, {  
    width: {to: 400},  
    height: {to: 400}  
}, 1);  
anim.animate();
```

```
var anim = new YAHOO.util.Anim(el, {  
    opacity: {from: 0, to: 1}  
    width: {to: 400},  
    height: {to: 400}  
}, 1);
```



# More animation

```
var anim = new YAHOO.util.Anim(el, {  
    width: {by: 100}  
}, 1);
```

```
var anim = new YAHOO.util.Anim(el, {  
    width: {from: 1, to: 10, unit: 'em'}  
}, 1);
```



# Movement

```
var anim = new YAHOO.util.Motion(el, {  
    to: [100, 100]  
}, 1);  
anim.animate();
```

```
var anim = new YAHOO.util.Motion(el, {  
    to: [100, 100],  
    control: [[50, 50], [150, 150]]  
}, 1);
```



# Easing

```
var anim = new YAHOO.util.Motion(el, {  
    to: [100, 100]  
}, 1, YAHOO.util.Easing.easeOut);
```

- easeIn - begin slowly and accelerate
- easeOut - begin quickly and decelerate
- easeBoth - both
- easeNone - the default; uniform speed
- backIn - start below starting value
- backOut - end with bounce beyond ending value
- backBoth - both



# onComplete

```
var anim = new YAHOO.util.Anim(el, {
    opacity: {to: 0}
} 1, YAHOO.util.Easing.easeOut);

anim.onComplete.subscribe(function() {
    var el = this.getEl();
    el.parentNode.removeChild(el);
});

anim.animate();
```







# When does a drag start?

- onmousedown?
- onmousedown + onmousemove?

```
YAHOO.util.DDM.clickTimeThresh = 1000;
```

```
YAHOO.util.DDM.clickPixelThresh = 3;
```

## Drag and Drop Interaction Storyboard

	Page Load	Mouse Hover	Mouse Down	Drag Initiated	Drag Over Valid Target	Drag Over Invalid Target	Drag Over Parent Container	Drop Accepted	Drop Rejected	Drop On Parent Container
Page	drag invitation									
Cursor	normal	draggability grabbable area	selected	dragging	drop will be valid	drop will be invalid	dragging home	drop was accepted	drop was rejected	drop returned home
Tool Tip		draggability grabbable area								
Drag Object	normal	draggability grabbable area	selected	dragging	drop will be valid	drop will be invalid	dragging home	drop was accepted	drop was rejected	drop returned home
Drag Object's Parent Container	normal	draggability grabbable area	selected	dragging			dragging home	drop was accepted	drop was rejected	drop returned home
Drop Target	normal		drop invitation	drop invitation	drop will be valid	drop will be invalid	drop invitation	drop was accepted	drop was rejected	drop returned home
	What does the page contain to indicate drag and drop?	What happens when the mouse hovers over the draggable object?	What happens when the mouse is pressed on the draggable object but dragging has not	What happens when drag starts?	What happens when I drag over a valid drop target?	What happens when I drag over an invalid drop target?	What happens when I drag back to my home area/container/slot?	What happens when the drop is accepted?	What happens when the drop is rejected?	What happens when dropped over the original position/container?

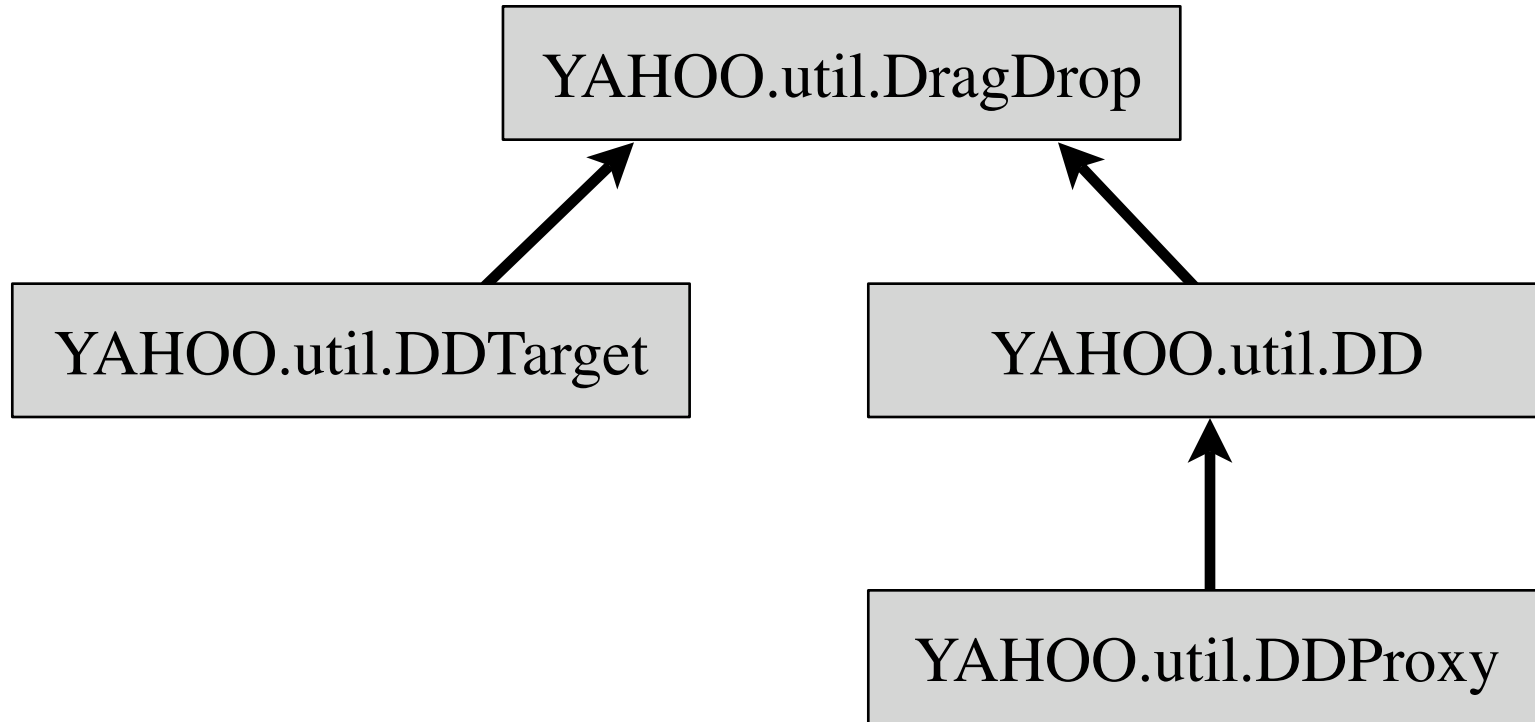


# Make something draggable

```
var dd = new YAHOO.util.DD("element1");  
  
.. new YAHOO.util.DD("element2", "group1");  
  
.. new YAHOO.util.DDProxy("element3");  
  
.. new YAHOO.util.DDTarget("element4");
```



# Class hierarchy





# Interesting moments

---

onMouseDown

startDrag

onDrag

onDragEnter

onDragOver

onDragOut

onDragDrop

endDrag

onMouseUp



# Implementation

```
  
<span id="status">Status message</span>  
<div id="sb">- Shopping basket -</div>
```

```
var basket = new YAHOO.util.DDTarget('sb');  
var item1 = new YAHOO.util.DD('item1');  
item1.onDragDrop = function(e, id) {  
    if (id == 'sb') {  
        YAHOO.util.Dom.get('status').innerHTML =  
            this.getEl().id + ' added to basket';  
    }  
}
```



# POINT vs INTERSECT

- In point mode, dragover / dragdrop calculated based on position of the mouse pointer
- In intersect mode, calculated based on intersection of regions

```
YAHOO.util.DDM.mode = YAHOO.util.DDM.INTERSECT;
```

```
YAHOO.util.DDM.mode = YAHOO.util.DDM.POINT;
```







# autocomplete

```
<input id="i"
  type="text">
<div id="c">
</div>
```

## Yahoo! Search:

The screenshot shows a search dropdown menu for the query 'xtech'. The first result, 'XTech 2006: Building Web 2.0', is highlighted in yellow. Other results include 'Technorati Tag: xtech', 'XTech 2005 Conference', 'Wednesday - 25 May', 'ModBlog - Xtech', 'Flickr: Photos tagged with xtech', 'Extrusion Technology', 'Brendan's Roadmap Updates: XTech', 'XML, SGML, electronic, commerce, EDI, barcode, publishing, printing, digital, Y2K', and 'Xtech - Enterprise IT Solutions for Government'.

```
var datasource = new YAHOO.widget.DS_XHR(
  '/script.php', [ 'Result', 'KeyData' ]
);
var ac = new YAHOO.widget.AutoComplete(
  'i', 'c', datasource
);
```



# autocomplete features

- Data source can be:
  - JavaScript array
  - JavaScript function
  - Remote Web Service
- Optional delimited characters
- Data source caching
- Type ahead option
- and lots more...



# calendar

```
function init() {  
    var cal = new YAHOO.widget.Calendar(  
        "cal", "container"  
    );  
    cal.render();  
}
```

```
<div id="container">  
</div>
```

◀		May 2006					▶	
Su	Mo	Tu	We	Th	Fr	Sa		
30	1	2	3	4	5	6		
7	8	9	10	11	12	13		
14	15	16	17	18	19	20		
21	22	23	24	25	26	27		
28	29	30	31	1	2	3		
4	5	6	7	8	9	10		



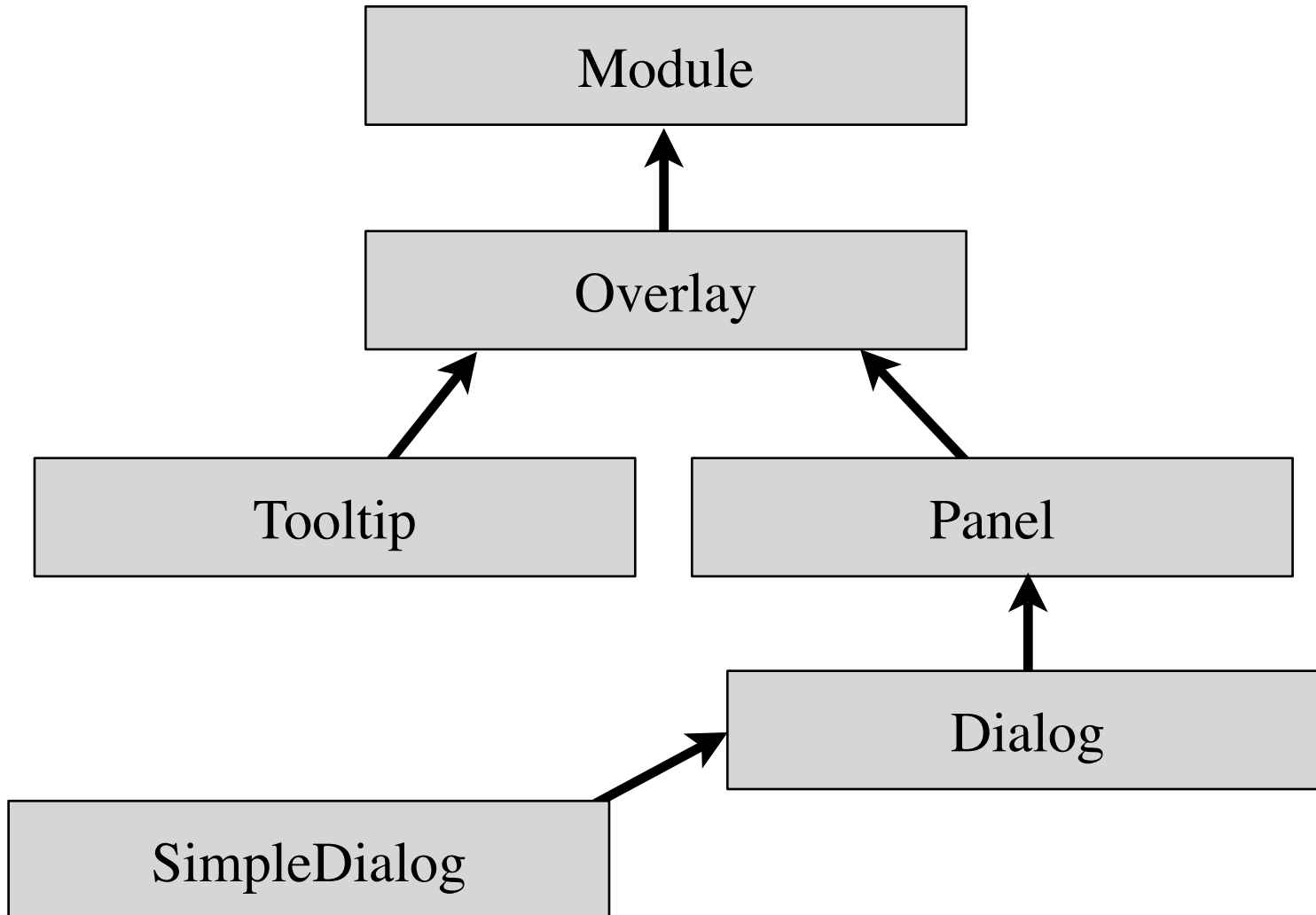
# calendar features

- Highly configurable
  - Skinnable with CSS
  - Restrict selectable date ranges
  - Multi-date-select
- Custom renderers

```
cal.addRenderer("12/25", function(d, cell) {  
    YAHOO.util.Dom.addClass(cell, "xmas");  
})
```



# container classes





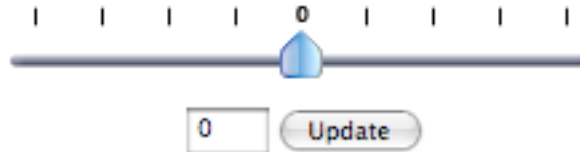
## container

- Simple class hierarchy for building OS-style panels and dialogs
- Modules are skinnable page components:
  - Overlays are absolutely positioned
  - Panels replicate OS windows
  - Dialogs enable text entry



# slider

- A slider widget with accessibility features



```
YAHOO.widget.Slider.getHorizSlider(  
    "sliderbg", "sliderthumb", 0, 200  
);
```



# menu

- Can be created dynamically or from existing markup

```
<div id="menu" class="yuimenu">
```

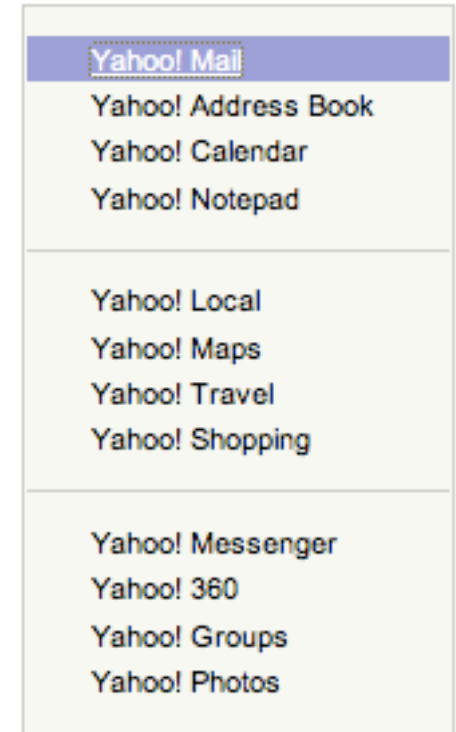
```
<div class="bd">
```

```
<ul><li class="yuimenuitem"><a href="#">Yahoo! Mail</a></li>...
```

```
var m = new YAHOO.widget.Menu(
```

```
    "menuwithgroups" );
```

```
m.render(); m.show();
```



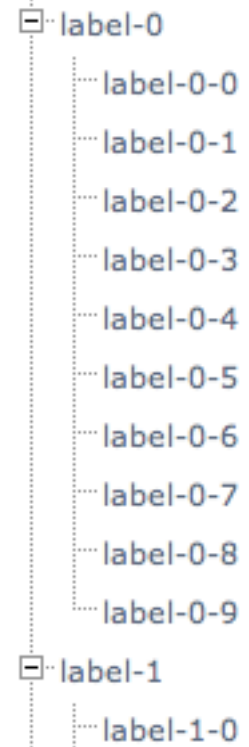




# treeview

```
var t = new YAHOO.widget.TreeView  
    ("treeDiv1");  
var root = t.getRoot();  
var n1 =new YAHOO.widget.TextNode(  
    "mylabel1", root, false);  
var n2 =new YAHOO.widget.TextNode(  
    "mylabel1-1", n1, false);  
tree.draw();
```

[Expand all](#) [Collapse all](#)





## Credits

- **Thomas Sha:** Team lead, connection
- **Adam Moore:** event, dragdrop, treeview, slider
- **Matt Sweeney:** dom, animation
- **Steven Peterson:** calendar, container
- **Jenny Han:** autocomplete
- **Todd Kloots:** menu
- **Matt Sweeny + Nate Koechly:** CSS packages
- **Eric Miraglia:** evangelist and advisor



## More information

- <http://developer.yahoo.com/yui/>
  - official site, documentation and download
- <http://groups.yahoo.com/group/ydn-javascript>
  - official discussion group
  - 20 - 30 messages per day
- <http://www.yuiblog.com/>
  - official blog



**LIFE ENGINE™**