

Simple Use Case (AutoComplete)

Markup:

```
<input id="myInput" type="text">
<div id="myContainer"></div>
```

Script:

```
var myAutoComp = new YAHOO.widget.AutoComplete
("myInput", "myContainer", myDataSource);
```

Instantiates a new AutoComplete object, `myAutoComp`, which queries an existing DataSource `myDataSource`.

Constructor (AutoComplete)

```
YAHOO.widget.AutoComplete(str | el ref input field,
    str | el ref suggestion container, obj DataSource
    instance[, obj configuration object]);
```

Arguments:

- (1) **HTML element (string or object)**: Text input or textarea element.
- (2) **HTML element (string or object)**: Suggestion container.
- (3) **DataSource instance (obj)**: An instantiated DataSource object; see below for DataSource types and constructor syntax.
- (4) **Configuration object (object)**: An optional object literal defines property values of an AutoComplete instance.

Constructors (DataSource Classes)

```
YAHOO.widget.DS_JsArray(array js array[, obj
    configuration object]);
```

Arguments:

- (1) **JS Array (array)**: A JavaScript array of strings.
- (2) **Configuration object (object)**: An optional object literal defines property values of a DataSource instance.

```
YAHOO.widget.DS_JsFunction(function js function[,
    obj configuration object]);
```

Arguments:

- (1) **JS Function (function)**: A JavaScript function which returns an array of strings.
- (2) **Configuration object (object)**: See above.

```
YAHOO.widget.DS_XHR(string script uri, array
    schema[, obj configuration object]);
```

Arguments:

- (1) **Script URI (string)**: Server URI (local domains only – use a proxy for remote domains).
- (2) **Schema (array)**: Schema description of server response data.
- (3) **Configuration object (object)**: See above.

Interesting Moments (AutoComplete)

Event	Arguments (passed via <i>args</i> array)
textBoxFocusEvent/ textBoxBlurEvent	[0] AC instance
textBoxKeyEvent	[0] AC instance; [1] keycode int
dataRequestEvent/ dataErrorEvent	[0] AC instance; [1] query string
dataReturnEvent	[0] AC instance; [1] query string; [2] results array
containerExpandEvent/ containerCollapseEvent	[0] AC instance
itemArrowToEvent/ itemArrowFromEvent	[0] AC instance; [1] element
itemMouseOverEvent/ itemMouseOutEvent	[0] AC instance; [1] element
itemSelectEvent	[0] AC instance; [1] element; [2] item data object or array
selectionEnforceEvent	[0] AC instance
unmatchedItemSelectEvent	[0] AC instance; [1] user selected string
typeAheadEvent	[0] AC instance; [1] query string; [2] prefill string
Subscribe to AutoComplete Custom Events on your AutoComplete instance: <code>myAC.containerExpandEvent.subscribe(myFn[, myObj, bScope]);</code>	

Simple Use Case (DataSource)

```
var myDataSource = new YAHOO.widget.DS_JsArray(
    ["a", "b", "c"]);
```

Instantiates a new DataSource object, `myDataSource`, which is an array of strings that queries can be matched against.

Custom Formatting (AutoComplete)

The `formatResult` function gets passed (1) an array that holds result data and (2) the original query string. Override this function to return custom markup to populate each element in the container.

```
myAC.formatResult = function(aResultItem, sQuery) {
    var sKey = aResultItem[0]; //the query match key
    // Additional data mapped by schema
    var attribute1 = aResultItem[1];
    var attribute2 = aResultItem[2];
    return (sKey + ": " + attribute1); }
```

Dependencies

AutoComplete requires the YAHOO object, Dom, and Event; Connection Manager and Animation are optional.

YAHOO.widget.
AutoComplete
Properties:

animVert (b)
animHoriz (b)
animSpeed (int)
delimChar (char || array)
maxResultsDisplayed (int)
minQueryLength (int)
queryDelay (int)
autoHighlight (b)
highlightClassName (string)
prehighlightClass (string)
useShadow (b)
useFrame (b)
forceSelection (b)
typeAhead (b)
allowBrowserAutocomplete (b)
alwaysShowContainer (b)

YAHOO.widget.
DataSource Properties:

maxCacheEntries (int)
queryMatchCase (b)
queryMatchContains (b)
queryMatchSubset (b)

YAHOO.widget.DS_XHR
Properties:

responseType (constant) TYPE_FLAT, TYPE_JSON, or TYPE_XML
scriptQueryParam (string)
scriptQueryAppend (string)
responseStripAfter (string)
connTimeout (int)