

# Y! YUI Library: The YAHOO Global Object

2007-12-4

v2.4

## Simple Use Case: YAHOO Object

In its simplest usage, the YAHOO global object requires no implementer action; it serves as a container and provider of utility methods to all other components of the YUI Library.

## Usage: YAHOO.namespace()

```
YAHOO.namespace(str namespace)
```

Arguments:

- (1) **namespace**: A string containing a single namespace (e.g. "myproduct") or a deeper namespace (e.g. "myproduct.weatherModule"). Namespace objects are created within the YAHOO object.

**Note:** Be careful when naming packages. JavaScript reserved words may work as property names in some browsers and not others.

## Usage: YAHOO.lang.augmentObject()

```
YAHOO.lang.augmentObject(fn receiver, fn  
    supplier[, str property1, str property2, ... ,  
    str propertyn])
```

Arguments:

- (1) **receiver**: The object to be augmented.
- (2) **supplier**: The object serving as the source of the augmentation.
- (3-n) **properties**: By default, YAHOO.lang.augmentObject will apply all members of the supplier object to the receiver if the receiver doesn't already have them; arguments 3 through *n* can be used to supply string member names that designate the specific members to be augmented from the supplier to the receiver.

**Note:** The default operation, in which all of the supplier's members are applied to the receiver, `YAHOO.lang.augment` will avoid overwriting existing members on the receiver. If you specify supplier members to use for augmentation (via arguments 3 through *n*), the augmentation will overwrite those members if they already exist on the receiver.

## Usage: YAHOO.lang.augmentProto()

```
YAHOO.namespace(fn receiver, fn supplier[, str  
    property1, str property2, ... , str propertyn])
```

This function is symmetrical with `YAHOO.lang.augmentObject` (see above); however, it augments only from the supplier's prototype to the receiver's prototype. Instance members are not copied from the supplier to the receiver.

## Dependencies

The YAHOO Global Object is a dependency for all YUI components; it has no dependencies of its own.

## YAHOO Object: Default Members

See online docs for complete documentation on each default member of the YAHOO object.

Member	Type	Description
env	object	Environment object. Contains information about what YUI modules are loaded and provides a method for obtaining version information.
example	object	An empty object used as a namespace for example implementations.
lang	object	Contains utility methods. Full list at right.
util	object	Namespace for YUI utilities. Do not add your own members to this object
tool	object	Namespace for developer tools like YUITest. Do not add your own members to this object.
widget	object	Namespace for YUI controls (widgets). Do not add your own members to this object.
log	method	Calls <code>YAHOO.widget.Logger.log</code> ; prevents log messages from throwing errors when the Logger Control is not present.
register	method	Registers a module with the YAHOO object.

## Usage: YAHOO.lang.extend()

```
YAHOO.extend(obj subclass, obj superclass[, obj  
    overrides])
```

Arguments:

- (1) **subclass**: The object you're using to extend the base object.
- (2) **superclass**: The base object being extended by the "subclass".
- (3) **overrides**: An object whose members will be added to the subclass prototype, overriding members of the same name if they exist on the superclass prototype.

## Solutions

`YAHOO_config` is not included as part of the YUI library. Instead it is an object that can be defined by the implementer immediately before including the YUI library. Use `YAHOO_config` to set up a listener that fires when YUI components are loaded:

```
var YAHOO_config = {  
    listener: function(moduleInfo) {  
        //executes when any YUI module loads, including YAHOO  
        object  
    }  
}
```

**Note:** See Module Info table at right for the format of object passed to your listener function.

**Get version information** for a YUI component that has been loaded on the page:

```
var YAHOO.env.getVersion("animation"); //returns module info  
object
```

## YAHOO.lang Methods

`dump(obj or arr)` returns string representation  
`isArray(any)` returns boolean  
`isBoolean(any)` returns boolean  
`isFunction(any)` returns boolean  
`isNull(any)` returns boolean  
`isNumber(any)` returns boolean  
`isObject(any)` returns boolean  
`isString(any)` returns boolean  
`isUndefined(any)` returns boolean  
`hasOwnProperty(obj, property)` returns boolean  
`augmentObject()` general mixin function  
`augmentProto()` see usage section  
`extend()` see usage section  
`isValue(any)` returns false for null/undefined/NaN, else true — note that `false` has value and returns true  
`merge(obj1, obj2, ...)` returns object with all properties of all args  
`substitute()` see docs  
`trim(string)` removes leading/trailing space

## YAHOO.env Method:

`getVersion(str yuimodulename)` returns module info; see below

## Module Info

`YAHOO.env.modules` is an object indexed by *module name*; each member contains information about a single YUI module. Module info objects contain the following information:

name	str module name
version	str last loaded version
build	n last loaded build
versions	arr all loaded versions
builds	arr all loaded builds
mainClass	fn reference to main class for this module

**YUI module names:** animation, autocomplete, base, button, calendar, colorpicker, connection, container, containercore, datasource, datatable, dom, dragdrop, editor, element, event, fonts, grids, history, imageloader, logger, menu, reset, slider, tabview, treeview, yahoo, yuiloader, yuitest.