

jQuery Utility Functions

Joe Eames

@josepheames

www.pluralsight.com



Outline

- Introduction
- Prerequisites, Tools & Conventions
- Type Testing Functions
- Collection Manipulation Functions
- `getScript`
- `holdReady`
- `pushStack`
- `each`
- `parseJSON`
- `extend`
- `sub`
- `proxy`

Prerequisites, Tools & Conventions

Prerequisites

- JavaScript
- jQuery
- HTML
- CSS

Tools

- Text Editor
- Browser

Conventions

- \$ or jQuery
- \$("div")

Type Testing Functions

- Determine the type of an object
- Useful for optional parameters & validation

Functions

- `$.isArray(array)`
- `$.isFunction(function)`
- `$.isEmptyObject(object)`
- `$.isPlainObject(object)`
- `$.isXmlDoc(doc)`
- `$.isNumeric(number)`
- `$.isWindow(window)`
- `$.type(object)`

\$.type Function Return Values

- **boolean**
- **number**
- **string**
- **function**
- **array**
- **date**
- **regexp**
- **object**
- **undefined**
- **null**

Collection Manipulation Functions

- **makeArray**
- **inArray**
 - Returns -1 or index
 - Optional parameter to indicate start index
- **unique**
- **merge**
- **map**
 - `$(selector).map`
 - `$.map`
- **grep**
 - Invert parameter
 - Filter Function receives a DOM object
 - Returns a JavaScript array

getScript Function

- Convenience wrapper for \$.ajax
- Executes retrieved javascript immediately

```
$.getScript(url, successCallback);
```

holdReady Function

```
$.holdReady(true);
```

```
$.holdReady(false);
```


pushStack Function

Pushes a new array of DOM objects onto the jQuery stack

each Function


Use Each() on

- array
- \$(selector)
- Javascript object

Flavors

- \$.each(collection, callback(index, element))
- \$(selector).each(callback(index, element))

```
$(“div”).each(function(idx, elem) {  
    ...  
});
```



Using each

```
$(“div”).each(function(idx, elem) {  
    var wrappedElem = $(elem);  
    wrappedElem.addClass(“myclass”);  
});
```

```
$(“div”).each(function(idx, elem) {  
    if( $(elem).html() == “match” ) {  
        // do something  
        return false;  
    }  
});
```

parseJSON function

Turns a JSON string into javascript objects

```
var myObject = $.parseJSON(jsonString);
```

extend Function

- Copy members from a source object to a target object
- Not Inheritance, only runtime member cloning
- Conflicts will cause members to be overwritten

```
var source = { // source members }  
var target = { // target members }  
$.extend(target, source);  
target.targetMember();  
target.sourceMember();
```

Extend with Multiple Base Objects

- Multiple source objects can be used
- Conflicts are resolved from left to right

```
var source1 = { // source1 members }  
var source2 = { // source2 members }  
var target = { // target members }  
$.extend(target, source1, source2);  
target.targetMember();  
target.source1Member();  
target.source2Member();
```

sub Function

- Creates a copy of jQuery for safe modification

```
function myExtendImplementation() { //implementation }  
  
var sub = $.sub();  
sub.extend = myExtendImplementation;
```

proxy Function

```
$.proxy(handler, context);
```

```
$("#myDiv").click( $.proxy(handler, context));
```


Function List

Type Functions

- isArray
- isFunction
- isEmptyObject
- isPlainObject
- isXmlDoc
- isNumeric
- isWindow
- type

Collection Functions

- makeArray
- grep
- map
- inArray
- unique
- merge

Misc Functions

- getScript
- holdReady
- pushStack
- each
- parseJSON
- extend
- sub
- proxy

Summary

- **Benefits**

- Fill in gaps in JavaScript
- Enhance Server Communications
- Provide utilities in plugin authoring
- Ease working in the page
- Enable richer JavaScript coding

- **References**

- <http://api.jquery.com/>