



a compact javascript framework

## mootools Basics

Full CheatSheet for Javascript-  
Framework mootools rev 1.2  
by mediavrog.net/blog/



### Core

\$chk(m)  
\$clear(timer n)  
\$defined(m)  
\$arguments(index n)  
\$empty  
\$lambda(o)  
\$extend(orig o, ext o)  
\$merge(o, o [, o [, ... ]])  
\$each(o | a, fn [, o])  
\$pick(o, [, o [, ... ]])  
\$random(min n, max n)  
\$splat(o)  
\$time()  
\$try(fn [, fn [, ... ]])  
\$type(o)  
element, textnode, number,  
whitespace, function, date,  
arguments, array, object, string,  
boolean, regexp, class, collection,  
window, document, false

### Native: Class

new Class( props )  
special properties:  
Extends: class  
Implements: class | props  
initialize: fn (=constructor)  
implement(class | props)

### Class.Extras

**Class: Chain**  
new Class(Implements: Chain)  
chain(fn [, fn [, ... ]])  
callChain([args])  
clearChain()

### Class: Events

new Class(Implements: Events)  
addEvent(s, fn [, internal b])  
addEvents(o, fn [, internal b])  
fireEvent(s[,args, delay ms])  
removeEvent(s, fn)  
removeEvents([s])

### Class: Options

new Class(Implements: Options)  
setOptions([opt])

### Hash: Browser

#### Features

xpath  
xhr

#### Engine

IE - trident[4 | 5]  
FF - gecko  
SFI - webkit[419 | 420]  
OP - presto[925 | 950]  
name

#### Plugins

Flash.version  
Flash.build

#### Platform

mac, win, linux, ipod, other  
name

### Native: String

test(regex [,params])  
escapeRegExp()  
contains(s [,sep s])  
trim()  
clean()  
camelCase()  
hyphenate()  
capitalize()  
toInt(), toFloat()  
rgbToHex(retAsArray b)  
hexToRgb(retAsArray b)  
stripScripts(evaluate b)  
substitute(o [, regex])

### Native: Function

create([opt])  
pass([args [, o])  
attempt([args [, o])  
bind([o [, args [, e ] ]])  
bindWithEvent([o [,args [, e ] ]])  
delay([ms [,o [,args ] ]])  
periodical([ms [,o [,args ] ]])  
run([args [, o ]])

### Native: Event

new Event([e [, win ]])  
(shift, control, alt, meta, wheel,  
code, page.x, page.y, client.x,  
client.y, key, target, relatedTarget)  
stop(), stopPropagation()  
preventDefault()

### Hash: Event.Keys

Event.Keys.eName = eKey

### Native: Array

\* each(fn(el,i){ [, o])  
\* every(fn(el,i){ [, o])  
\* filter(fn(el,i){ [, o])  
\* indexOf(el [,from n])  
\* map(fn(el,i){ [, o])  
\* some(fn(el,i){ [, o])  
\* only if not supported natively  
clean()  
associate(a)  
link(o)  
contains(el)  
extend(a)  
getLast()  
getRandom()  
include(el)  
combine(a)  
erase(el)  
empty()  
flatten()  
rgbToHex(retAsArray b)

### Utility Functions

\$(a)

### Native: Hash

new Hash( [props ] )  
each(fn(el,i){ [, o])  
has(key s)  
keyOf(m)  
hasValue(m)  
extend(props)  
combine(props)  
erase(key s)  
get(key s)  
set(key s, val m)  
empty()  
include(key s, val m)  
map(fn(el,i){ [, o])  
filter(fn(el,i){ [, o])  
every(fn(el,i){ [, o])  
some(fn(el,i){ [, o])  
getClean()  
getKeys()  
getValues()  
toQueryString()

### Utility Functions

\$(H( [props ] ) > new Hash

### Native: Number

toInt(), toFloat()  
limit(min n, max n),  
round([n]), times(fn [, o])

### Element

#### Native: Window

\$(el)  
\$\$([el a | id a | el | selector s])  
any combination; commasep

#### Native: Element

new Element(tag s [, opt])  
opt = {  
"styles": setStyles,  
"events": addEvents,  
"anyProp": value  
}

getElement(match)  
getElements(match)  
match(match)  
getElementsById(s)  
set(s, val | o)  
get(s)  
erase(s)  
inject(el [, where s])  
grab(el [, where])  
adopt(el [, el a | el [, ... ]])  
wraps(el [, where])  
appendText(s)  
dispose()  
clone([childs b, keepId b])  
replaces(el)  
hasClass(s)  
addClass(s)  
removeClass(s)  
toggleClass(s)  
getPrevious([match])  
getAllPrevious()  
getNext([match])  
getAllNext()  
getFirst([match])  
getLast([match])  
getParent([match])  
getParents()  
getChildren([match])  
hasChild(el)  
empty()  
destroy()  
toQueryString()  
getSelected()  
getProperty(s)  
getProperties(s [, s [, ... ]])  
setProperty(s, val)  
setProperties( {s: val, ... } )  
removeProperty(s)  
removeProperties(s [, s [, ... ]])  
store(s, val)  
retrieve(s)

#### Hash: Element.Properties

html, [htmlS [,htmlS [, ... ] ]]  
text, textString  
tag (only getter)

#### Native: IFrame

new IFrame([el ] [, opt])

#### Native: Elements

new Elements(el a [,opt])  
filter(sel s)

### Element.Event

#### Native: Element

addEvent(e, fn)  
removeEvent(e, fn)  
addEvents( {e:fn } )  
removeEvents([e])  
fireEvent(e [, args, delay])  
cloneEvents(el [,e])

#### Hash: Element.Events

Element.Events.eName = o  
o = {  
base: e  
condition: fn  
onAdd: fn  
onRemove: fn  
}

### Custom Events

mouseenter  
mouseleave  
mousewheel

### Element.Style

#### Native: Element

setStyle(s, val)  
setStyles( {s : val, ... } )  
getStyle(s)  
getStyles(s [, s [, ... ]])

### Element.Dimensions

#### Native: Element

scrollTo(x,y)  
getSize()  
getScrollSize()  
getScroll()  
getPosition()  
getCoordinates()

o ~ Object  
s ~ String  
a ~ Array  
n ~ Number  
b ~ Boolean  
e ~ Event  
fn ~ Function  
el ~ Element  
el a ~ Array of el  
m ~ mixed

### Selectors

#### Utility Functions

\$(sel s, filter el)  
\$(S(sel s, filter el))

#### Native: Element

getElements(sel s)  
getElement (sel s) > \$E  
match(sel s)

#### Selectors.Pseudo

:enabled, :empty  
:contains(s)  
:even, :odd, :first, :last, :only  
:nth-child(nExpr)  
n - all, 2n - even, 2n+1 - odd  
odd, even, only, first, last

### Class: Swiff

new Swiff(path [, opt])

opt = {  
id: s  
width: n, height : n  
container: el  
params: swfParams  
properties: o  
vars: o  
events: o  
}

swfParams = {  
allowScriptAccess: s  
quality: s  
swLiveConnect: b  
wMode: s  
}

### Object: Cookie

write(key s, value s [, opt])  
opt = {  
domain: s  
path: s  
duration: n  
secure: b  
}

read(cookie s)

dispose(cookie s [, opt])

### Object: JSON

JSON.encode(o)  
JSON.decode(s [, secure b])

### WindowEvent: domready

domready

### Class: Request

new Request( [opt ] )  
opt = {  
url: s  
method: post | get,  
data: s  
async: asyncReq b  
encoding: s (default: utf-8),  
autoCancel: b  
headers: {hdName:hdCont} o  
isSuccess: fn  
onRequest(inst)  
onSuccess(inst)  
onFailure(inst)  
onException(hdName ,val)  
onCancel()  
}  
Properties  
running, response  
setHeader(name s, val s)  
getHeader(name s)  
send( [opt ] )  
cancel()

### Hash: Element.Properties

send [, Request opt]

### Native: Element

send([url s])

### Class: Request.HTML

new Request.HTML([opt])  
opt = { all opt from Request  
update: el,  
evalScripts: b,  
evalResponse: b  
onComplete(rTree, rElems,  
rHTML, rJS)  
}  
get([opt], post([opt])

### Hash: Element.Properties

load [, opt]

### Native: Element

load(url s) > Rq.HTML.get

### Class: Request.JSON

new Request.JSON([opt])  
opt = { all opt from Request  
secure: b  
onComplete(rJSON, text)  
}

mootools Cheat Sheet v1.2  
"Basics" by mediavrog © 08

"FX/Plugins" available soon