mootools Basics

mootools a compact javascript framework

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

N: Class

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

Class.Extras

C: Chain

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

C: 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])

C: Options

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

H: 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

N: 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])

N: 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])

N: 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()

H: Event.Keys

Event.Keys.eName = eKey

N: 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(a)

N: 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

N: Number

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

Element

N: Window

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

N: 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)
retreive(s)

H: Element.Properties

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

N: IFrame

new IFrame([el] [, opt])

N: Elements

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

Element.Event

N: Element

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

H: Element.Events

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

Custom Events

mouseenter
mouseleave
mousewheel

Element.Style

N: Element

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

Element.Dimensions

N: Element

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

Selectors

Utility Functions

$E(sel s, filter el)
$ES(sel s, filter el)

N: 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

C: 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
}
remote(o, fn)

O: Cookie

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

O: JSON

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

WindowEvent: domready

domready

C: 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()

H: Element.Properties

send [, Request opt]

N: Element

send([url s])

C: 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)

H: Element.Properties

load [, opt]

N: Element

load(url s) > Rq.HTML.get

C: Request.JSON

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

o ~ Object	e ~ Event	[ ]	~ optional	mootools Cheat Sheet v1.2 "Basics" by mediavrog © 08	
s ~ String	fn ~ Function	|	~ choice / or		
a ~ Array	el ~ Element	>	~ see also	"FX/Plugins" available soon	
n ~ Number	el a ~ Array of el	ms	~ Milliseconds		
b ~ Boolean	m ~ mixed	opt	~ Options Obj		