

Array.js

Array
 forEach(fn, bind o)
 each = forEach
 copy()
 remove(s)
 test(s)
 extend(a)
 associate([])

Utility Functions
 \$A(a) = Array.copy

Function.js

Function
 pass([args,...], bind o)
 bind(o)
 bindAsEventListener(o)
 delay(ms, bind o)
 periodical(ms, bind o)

Utility Functions
 \$clear(timer)
 \$type(o)

Chain

chain(fn)
 callChain()
 clearChain()

String.js

String
 test(pattern s, params s)
 toInt()
 camelCase()
 capitalize()
 trim()
 clean()
 rgbToHex(returnArray b)
 hexToRgb(returnArray b)

Number

toInt()

Element.js

Element
 initialize(tagname s)
 injectBefore(el)
 injectAfter(el)
 injectInside(el)
 adopt(el)
 remove()
 clone()
 replaceWith(el)
 appendText(s)
 hasClass(s)
 addClass(s)
 removeClass(s)
 toggleClass(s)
 setStyle(property, value)
 setStyles({prop:value,...})
 setOpacity(n)
 getStyle(property)
 addEvent(event, f)
 removeEvent(event, f)
 getPrevious()
 getFirst()
 getNext()
 getLast()
 setProperty(prop, value)
 setProperties({prop:val,...})
 setHTML(html)
 getProperty(property)
 getTag()
 getTop()
 getLeft()
 getValue()

Utility Functions
 \$Element(el, method s, args)
 \$(el)s

Moo.js

Class
 empty()
 create(o)
 extend(o)
 implement(o)

Object (singleton)
 extend(base o, other o)
 Native(o)

Dom.js

Element
 getElements(selector)
 getElement(selector)
 getElementsBySelector(s)

Elements
 action({event:fn, ...})

Utility Functions
 \$\$(selector s, s, ...)
 \$\$() = \$\$S
 \$E(selector s, el)
 #ES(selector s, el)
 document
 .getElementsByClassName(s)

Drag.js

Drag.Base(el, xM, yM, options)
 options = {
 handle: el,
 onStart: fn,
 onComplete: fn,
 onDrag: fn,
 xMax:n, xMin:n,
 yMax:n, yMin:n
 }

Drag.Move(el, options)

options = {
 snap: (true)||false,
 snapDistance: (8) n,
 onSnap: fn,
 xMod: (left) prop,
 yMod: (top) prop,
 container: true||(false)
 }

Element
 makeDraggable(options)
 makeResizable(options)
 getPosition()

Fx.js

Fx.Base
 options = {
 onStart: fn,
 onComplete: fn,
 transition: Fx.Transition,
 duration: (500) ms,
 unit: ('px')||'em'||'%',
 wait: (true)||false,
 fps: (30) n
 }

set(to n)
 custom(from n, to n)
 clearTimer()

Fx.Style(el, property, options)
 hide()
 goTo(n)

Fx.Styles(el, options)
 custom({prop:[start, end],...})

Element
 effect() = Fx.Style
 effects() = Fx.Styles

Fx.Transitions
 linear, sineInOut

Ajax.js

Ajax(url, options)
 options = {
 method: ('post')||'get',
 postBody: s || o || Form el
 async:(true)||false,
 onComplete:fn,
 onStateChange:fn,
 update: el,
 evalScripts: true||(false)
 }

request()
 evalScripts()

Object
 toQueryString()

Element
 send()
 toQueryString()

Fxpack.js

Fx.Scroll(el, options)
 down()
 up()

Fx.Slide(el, options)
 options = {
 mode: ('vertical')||'horizontal'
 }
 hide()
 show()
 toggle()

Fx.Color(el, options)
 custom(fromHex s, toHex s)
 fromColor(hex s)
 toColor(hex s)

Fxutils.js

Fx.Height, Fx.Width, Fx.Opacity(el, options)
 toggle()
 show()

Fxtransitions.js

Fx.Transitions
 linear, quadIn, quadOut,
 quadInOut, cubicIn, cubicOut,
 cubicInOut, quartIn, quartOut,
 quartInOut, quintIn, quinOut,
 quintInOut, sineIn, sineOut,
 sineInOut, expoIn, expoOut,
 expolnOut, circln, circOut,
 circlnOut, elasticIn, elasticOut,
 elasticInOut, backIn, backOut,
 backInOut, bounceIn,
 bounceOut, bounceInOut

Sortables.js

Sortables(els, options)
 options = {
 handles: els,
 fxDuration: (250) ms,
 fxTransition: Fx.Transitions,
 maxOpacity: n,
 onStart: fn,
 onComplete: fn,
 contain: (true)||false
 }

Cookie.js

Cookie
 set(key s, value, days n)
 get(key s)
 remove(key s)

Json.js

Json
 toString(json o)
 evaluate(s)

Tips.js

Tips(els, options)
 options = {
 transitionStart: Fx.Transitions,
 transitionEnd: Fx.Transitions,
 maxTitleChars: (30) n,
 fxDuration: (150) ms,
 maxOpacity: (1) n,
 timeOut: (100) ms,
 className: s
 }

Window.js

Window
 disableImageCache()
 getWidth()
 getHeight()
 getScrollHeight()
 getScrollWidth()
 getScrollTop()
 getScrollLeft()
 onDomReady(fn)

Accordion.js

Fx.Elements(el, options)
 custom({
 '0':{prop:[start, end],...},
 '1':{prop:[start, end],...},
 ... })

Fx.Accordion(els, toggler els, options)

options = {
 start: ('open-first')||'first-open',
 fixedHeight: (false) n,
 fixedWidth: (false) n,
 alwaysHide: true||(false),
 wait: true||(false),
 onActive: fn,
 onBackground: fn,
 height: (true)||false,
 opacity: (true)||false,
 width: true||(false)
 }

showThisHideOpen(index n)