



## Core.js

### Functions

\$defined(o)  
 \$type(o)  
 \$merge(o [,o,...])  
 \$extend(o, o)  
 \$native(native o [,native o,...])  
 \$chk(o)  
 \$pick(o, default o)  
 \$random(min n, max n)  
 \$time()

### Abstract (Singleton)

### Window

(ie, ie6, ie7, gecko, webkit, webkit419, webkit420, opera)

## Class.js

### Class

new Class({fname: fn})  
 empty()  
 extend({fname: fn})  
 implement({fname: fn})

## Class.Extras.js

### Chain

chain(fn)  
 callChain()  
 clearChain()

### Events

addEvent(s, fn)  
 fireEvent(s[,arg[]],delay ms)  
 removeEvent(s, fn)

### Options

setOptions(default opt, opt)

## Window.Size.js

### Window

getWidth(), getHeight()  
 getScrollWidth()  
 getScrollHeight()  
 getScrollLeft(),getScrollTop()  
 getSize()

## Window.DomReady.js

### Custom Events

domready

## Array.js

### Array

\* forEach(fn(el,i))  
 \* filter(fn(el,i))  
 \* map(fn(el,i))  
 \* every(fn(el,i))  
 \* some(fn(el,i))  
 \* indexOf(el)  
 each > forEach  
 copy()  
 remove(el)  
 contains(el)  
 associate(a)  
 extend(a)  
 merge(a)  
 include(el)  
 getRandom()  
 getLast()

### Utility Functions

\$A() > copy()  
 \$each(a,fn(el,i))

## String.js

### String

test(regex [,params])  
 toInt(), toFloat()  
 camelCase()  
 hyphenate()  
 capitalize()  
 trim(), clean()  
 rgbToHex(returnArray b)  
 hexToRgb(returnArray b)  
 contains(s [,separator s])  
 escapeRegExp()

### Array

rgbToHex() > rgbToHex  
 hexToRgb() > hexToRgb

## Function.js

### Function

create(opt)  
 pass(arg[], [el])  
 attempt(arg[], [el])  
 bind(fn [,arg[]])  
 bindAsEventListener(o,arg[])  
 delay(ms [,o,arg[]])  
 periodical(ms [,o,arg[]])

## Number.js

### Number

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

## Element.js

### Utility Functions

\$(el | s)  
 \$\$ (el a | id a | el | selector s)  
 (any combination)

### Element

new Element(s, opt)  
 set(opt)  
 opt = {  
   "styles": setStyles,  
   "events": addEvents,  
   "otherKey": setProperty  
 }

injectBefore(el)  
 injectAfter(el)  
 injectInside(el)  
 injectTop(el)

adopt(el)  
 remove(el)  
 clone(withChildnodes b)  
 replaceWith(el)  
 appendText(s)  
 hasClass(s)  
 addClass(s), removeClass(s)  
 toggleClass(s)  
 setStyle(style s, value s | n)  
 setStyles({style:value})  
 setOpacity(n)  
 getStyle(style s)  
 getStyles(style s [,s,...])  
 getPrevious(), getNext()  
 getFirst(), getLast()  
 getParent(), getChildren()  
 hasChild(el)

getProperty(prop s)  
 removeProperty(prop s)  
 getProperties(prop s [,s,...])  
 setProperty(prop, value)  
 setProperties({prop:value})  
 setHTML(html)  
 setText(s)  
 getText()  
 getTag()  
 empty()

## Element.Dimensions.js

### Element

scrollTo(x,y)  
 getSize()  
 getPosition([overflown el a])  
 getTop([overflown el a])  
 getLeft([overflown el a])  
 getCoordinates([overfl el a])

## Element.Selectors.js

### Utility Functions

\$E(selector s, filter el)  
 \$\$E(selector s, filter el)

### Element

getElements(singleSelector s)  
 getElement(selector s) > \$E  
 getElementsBySelector  
 (selector s) > \$\$  
 getElementById(id s) > \$

## Element.Filters.js

### Element

filterByTag(tagname s)  
 filterByClass(classname s)  
 filterById(id s)  
 filterByAttribute(s [,op s,val s])

## Element.Form.js

### Element

getValue(), toQueryString()

## XHR.js

### XHR

new XHR(url, opt)  
 opt = {  
   method: post | get,  
   async: asyncReq b  
   encoding: s (default: utf-8),  
   autoCancel: b  
   headers: {hdName:hdCont} o  
   onRequest: fn,  
   onSuccess: fn,  
   onStateChange: fn,  
   onFailure: fn  
 }

### Properties

running, response  
 setHeader(hdName s,hdVal s)  
 send()  
 cancel()

## Assets.js

### Assets

new Asset.property  
 javascript(src s, opt)  
 css(src s, opt)  
 image(src s, opt)  
 images(srcs a, opt)

## Json.js

### Json

toString(o)  
 evaluate(s, syntaxCheck b)

## Element.Event.js

### Event

(shift,control,alt,meta,wheel,  
 code,page.x,page.y,client.x,  
 client.y,key,target,relatedTarget)  
 stop()  
 stopPropagation()  
 preventDefault()  
 keys.eventName = keycode n

### Element

addEvent(e, fn)  
 removeEvent(e, fn)  
 addEvents({e:fn})  
 removeEvents([type s])  
 fireEvent(type s[,arg[]], delay)  
 cloneEvents([type s])

### Function

bindWithEvent(el [,arg[]])

### Custom Events

mouseenter, mouseleave

## Ajax.js

### Ajax

new Ajax(url, opt)  
 opt = { all opt from XHR,  
   data: s,  
   update: el,  
   evalScripts: b,  
   evalResponse: b  
   onComplete: fn  
 }

### request()

evalScripts()  
 getHeader(hdName s)

### Object

toQueryString()

### Element

send() > for form-Elements

## Cookie.js

### Cookie

opt{domain s, path s,  
   duration days, secure b}  
 set(key s, value s, opt)  
 get(cookieValue s)  
 remove(cookieName s, opt)

## Json.Remote.js

### Json.Remote

new Json.Remote(url, opt)  
 opt = all opt from xhr

## Drag.Base.js

### Drag.Base

new Drag.Base(el, opt)  
 opt = {  
   handle:el,  
   modifiers: {styleX,styleY},  
   limit: [[stX,endX],[stY,endY]]  
   [,grid px, snap px]  
 }

### Element

makeResizable(opt)  
 opt > Drag.Base opt

## FX.Base.js

### Fx.Base

new Fx.Base(opt)  
 opt = {  
   transition: Fx.Transitions,  
   duration: ms,  
   unit: px | em | %  
   wait: waitForCurTransEnd b,  
   fps: framesPerSecond n,  
   onStart: fn,  
   onComplete: fn,  
   onCancel: fn  
 }

start(from n, to n)

stop()

set(to n)

## FX.Style.js

### Fx.Style

new Fx.Style(el, prop, opt)  
 opt = all opt from Fx.Base  
 start(from n, to n)  
 hide()  
 set(to n)

### Element

effect(opt) > Fx.Style

## FX.Styles.js

### Fx.Styles

new Fx.Styles(el, opt)  
 opt = all opt from Fx.Base  
 start({  
   style: [[from n, to n] | to n  
 })

### Element

effects(opt) > Fx.Styles

o ~ Object	e ~ Event	[] ~ optional
s ~ String	fn ~ Function	~ choice / or
a ~ Array	el ~ Element	> ~ see also
n ~ Number	opt ~ Options Object	el a ~ Array of Elements
b ~ Boolean	ms ~ Milliseconds	{key:val} ~ o w/ key/val pairs

## Drag.Move.js

### Drag.Move

new Drag.Move(el, opt)  
 opt = {  
   all opt from Drag.Base,  
   container el,  
   droppables el a,  
   overflown el a  
 }

### Element

makeDraggable(opt)  
 opt > Drag.Move opt

## FX.Elements.js

### Fx.Elements

new Fx.Elements(el, opt)  
 opt = all opt from Fx.Base  
 start({  
   index: {style:[from,to]}  
 })

## FX.Scroll.js

### Fx.Scroll

new Fx.Scroll(el, opt)  
 opt = { all opt from Fx.Base,  
   offset {x,y},  
   overflown el a  
 }

scrollTo(x,y)

toTop(),toBottom()

toRight(),toLeft()

toElement(el)

## FX.Slide.js

### Fx.Slide

new Fx.Slide(el, opt)  
 opt = {all opt from Fx.Base,  
   mode: "vertical" | "horizontal"  
 }  
 slideln(), slideOut()  
 hide(), show()  
 toggle()

## FX.Transitions.js

### Fx.Transitions

linear,Quad,Cubic,Quart,Quint,  
 Pow,Expo,Circ,Sine,Back,  
 Bounce,Elastic  
 each has easln,easeOut,easelnOut