# base:

1. find element

method: find(selector)

return: element

1. find all elements

method: findAll(selector)

return nodeList

1. find element from an element

method: element.find(selector)

return element;

1. find all elements from an element

method: element.findAll(selector);

return nodeList;

1. closest
2. get/set attribute

method: element.attr(name [, value])

return value/element

1. get/set html

method: element.html([value])

return html/element

1. set empty html

method: element.empty();

return element

1. append element

method: element.append(dom/html/selector)

return element

1. prepend element
2. appendTo
3. prependTo
4. before
5. after
6. replaceWith
7. remove
8. data
9. hasClass
10. addClass
11. removeClass
12. height
13. width
14. css
15. show
16. hide
17. offset
18. position
19. bind event

method: element.bind(eventName, function, useCapture)

1. unbind event
2. each array/nodeList

method: array.each(function(value, index){})

1. array/nodeList contains

method: array.contains(v)

return bool

1. extend object

method: obj.extend(source[, source1, source2])

obj.extend(true, source[, source1, source2])

return target

1. each object

method: obj.each(function(value, key){})

1. is different object

method: obj.diff(compareObj)

return bool

# form:

1. new form

method: Snow.Form(dom, model)

options: {

model: {},

requestHandler: undefined,

responseHandler: undefined,

response: {},

action: undefined,

method: 'get',

fn: {} //functions

}

return form

1. update model and view

method: form.update(newModel)

1. verify data, submit form, render template

method: form.submit(function(model){})

1. extend view

method: Snow.Form.view.newView = function(model){}

1. extend function

method: form.fn.function = function(model){}

reference: @functionName

1. html tags

data-model: define the mode property

data-value: the value displayed

data-class: the class displayed

data-class-delay: delay to display the class

data-script: script executes

data-script-delay: delay to execute the script

data-view: call the custom view

data-event: execute the function/script when trigger the event (data-click, data-focus...)

data-template: define the template html and the template model

# template:

1. new template

method: Snow.template(html/url, data, function(html){}, nocache)

return: html

description: if define the callback function, then load from url, and use cache or not

# validate:

1. verity data

method: Snow.Validate.int(value)

return: bool

# ajax:

1. ajax call

method: Snow.ajax({

type: 'GET',  
url: '',  
async: true,  
contentType: 'application/x-www-form-urlencoded',  
cache: true,  
data: null,  
success: undefined,  
complete: undefined,  
error: undefined,  
timeout: 0

})

return: data or request