

```

; (function ($, window, document) {
    var pluginName = 'checkAll';
    var settings = { container: document, childnode: 'input[type=checkbox]', group: '.abc'
};

    function checkAll(element, options) {
        this.$el = $(element);
        this.oSettings = $.extend({}, settings, this.$el.data(), options);
        this.init();
    }

    checkAll.prototype.init = function () {
        var _self = this;
        _self._checkChildren();
        this.$el.on('ifChanged', function (e) {
            var $children = $(_self.oSettings.childnode,
            _self.oSettings.container).not(_self.$el);
            $children.prop('checked', $(this).prop('checked'));
            $children.iCheck('update');
        });
        $(this.oSettings.container).on('ifChanged', _self.oSettings.childnode, function (e)
    {
        _self._checkChildren();
    });
};

$.fn[pluginName] = function (options) {
    return this.each(function () {
        if (!$.data(this, 'plugin_' + pluginName)) {
            $.data(this, 'plugin_' + pluginName,
                new checkAll(this, options));
        }
    });
};

checkAll.prototype._checkChildren = function () {
    var totalCount = $(this.oSettings.childnode,
    this.oSettings.container).not(this.$el).length;
    var checkedCount = $(this.oSettings.childnode, this.oSettings.container)
        .filter(':checked').not(this.$el).length;

    var indeterminate = this.oSettings.showIndeterminate &&
        checkedCount > 0 && checkedCount < totalCount;

    this.$el.prop('indeterminate', indeterminate);
    this.$el.prop('checked', checkedCount === totalCount && totalCount !== 0);
    this.$el.iCheck("update");
}
})(jQuery, window, document);

function showWaitProcess(oSettings) {
    oSettings = jQuery.extend({ Text: 'Please Wait...', Effect: 'facebook', background:
    'rgba(160, 160, 160, 0.48)', ColorCode: '#000', SizeW: '', SizeH: '', id: 'dvLoading' },
    oSettings);
    var $element = $('#' + oSettings.id);
    if ($element != undefined) {
        $element.removeAttr('class').hide().children().remove();
        $element.css({ 'width': '100%', 'height': '100%', 'position': 'absolute', 'top':
        '0', 'left': '0', 'z-index': '99999', 'display': 'none' });
        $element.waitMe({
            effect: oSettings.Effect,
            text: oSettings.Text,
            bg: oSettings.background,
            color: oSettings.ColorCode,
            sizeW: oSettings.SizeW,

```

```

        sizeH: oSettings.SizeH
    });
    $element.show();
}
};

function hideWaitProcess() {
    $('#dvLoading').removeAttr('class').hide().children().remove();
};

$.extend({
    getUrlVars: function () {
        var vars = [], hash;
        if (window.location.search.substring(1).length >= 1) {
            var hashes = window.location.href.slice(window.location.href.indexOf('?') +
1).split('&');
            for (var i = 0; i < hashes.length; i++) {
                hash = hashes[i].split('=');
                vars.push(hash[0]);
                vars[hash[0]] = hash[1];
            }
        }
        return vars;
    },
    getUrlVar: function (name) {
        return $.getUrlVars()[name];
    },
    clearURLVar: function () {
        history.pushState("", document.title, window.location.pathname);
    }
});

function copyToClipboard(text) {
    var $temp = $("");
    $("body").append($temp);
    $temp.val(text).select();
    document.execCommand("copy");
    $temp.remove();
}

Array.prototype.insert = function (index, item) {
    this.splice(index, 0, item);
};

function Enum() {
    this.self = arguments[0];
}

Enum.prototype = {
    keys: function () {
        return Object.keys(this.self);
    },
    values: function () {
        var me = this;
        return this.keys(this.self).map(function (key) {
            return me.self[key];
        });
    },
    getValueByName: function (key) {
        return this.self[this.keys(this.self).filter(function (k) {
            return key === k;
        }).pop() || ''];
    },
    getNameByValue: function (value) {

```

```
var me = this;  
return this.keys(this.self).filter(function (k) {  
    return me.self[k] === value;  
}).pop() || null;  
}  
};
```