AXISJ API

Search Documentations

~~~
/**
 * AXISJ 유틸함수 라이브러리 axf 또는 AXUtil이라고 한다.
 * @namespace {Object} axf
 * @example
 ```json
 trace(axf.browser);
 trace(AXUtil.browser);
 ~~~
 */
var axf = AXUtil = {
 async: true,
 ajaxOkCode: "ok",
 ajaxResponseType: "",
 ajaxDataType: "",
 gridPassiveMode: false,
 gridPassiveRemoveHide: false,
 gridFitToWidthRightMargin: 10,

 uniqueSeq: 0,

/**
 * 현재페이지에서 고유한 순번을 반환합니다.
 * @method axf.getUniqueId
 * @returns {Number} uniqueSeq
 * @example
 ```
 trace( axf.getUniqueId() );
 ```
 */

 getUniqueId: function(){ return (axf.uniqueSeq += 1); },
/**
 * document.getElementById(id) 와 같습니다. 아이디가 같은 엘리먼트를 반환합니다.
 * @method axf.getId
 * @param {String} id
 * @returns {HtmlElement}
 * @example
 ```
 if(axf.getId("myele_id")){
    $("#myele_id").css({...});
 }
 ```
 */
 getId: function(id) { return document.getElementById(id); },
/**
 * @method axf.each
 * @param {Array|Object} obj
 * @param {Function} callback
 * @description Array 또는 Object의 아이템만큼 callback 함수를 call합니다.
 * @example
```
var new_array = [];
axf.each([0, 1, 2], function(){
 new_array.push(this*2);
});
var new_object = {};
axf.each({a:1, b:2, c:3}, function(k, v){
 new_object[k] = v*2;
});
```
 */
 each: function(obj, callback){
  if(obj){
   var name, i = 0, length = obj.length,
    isObj = length === undefined || Object.isFunction(obj);
   if (isObj) {
    for (name in obj) {
     if (callback.call(obj[name], name, obj[name]) === false) {
      break;
     }
    }
   } else {
    for (; i < length;) {
     if (callback.call(obj[i], i, obj[i++]) === false) {
      break;
     }
    }
   }
  }
```
```

~~~
 },
/**
 * 브라우저의 이름과 버전 모바일여부
 * @member {Object} axf.browser
 * @example
 ```
{
 name: {String} - bowserName (ie|chrome|webkit|oprea),
 version: {Number} - browserVersion,
 mobile: {Boolean}
}
 ```
 */
 browser: (function () {
  var ua = navigator.userAgent.toLowerCase();
  var mobile = (ua.search(/mobile/g) != -1);
  if (ua.search(/iphone/g) != -1) {
   return { name: "iphone", version: 0, mobile: true }
  } else if (ua.search(/ipad/g) != -1) {
   return { name: "ipad", version: 0, mobile: true }
  } else if (ua.search(/android/g) != -1) {
   var match = /(android)[ \/]([\w.]+)/.exec(ua) || [];
   var browserVersion = (match[2] || "0");
   return { name: "android", version: browserVersion, mobile: mobile }
  } else {
   var browserName = "";
   var match = /(chrome)[ \/]([\w.]+)/.exec(ua) ||
    /(webkit)[ \/]([\w.]+)/.exec(ua) ||
    /(opera)(?:.*version|)[ \/]([\w.]+)/.exec(ua) ||
    /(msie) ([\w.]+)/.exec(ua) ||
    ua.indexOf("compatible") < 0 && /(mozilla)(?:.*? rv:([\w.]+)|)/.exec(ua) ||
    [];

   var browser = (match[1] || "");
   var browserVersion = (match[2] || "0");

   if (browser == "msie") browser = "ie";
   return {
    name: browser,
    version: browserVersion,
    mobile: mobile
   }
  }
 })(),
/**
 * 호환성보기 여부
 * @member {String} axf.docTD
 * @example
 ```
 axf.docTD = (Q|S)
 ```
 */
 docTD: (function () {
  if (!document.compatMode || document.compatMode == 'BackCompat') return "Q";
  else return "S";
 })(),
/**
 * @method axf.timekey
 * @returns {String} timeKey
 * @description 밀리세컨드까지 조합한 문자열을 반환합니다.
 * @example
```js
trace(axf.timeKey()); // A004222760
```
 */
 timekey: function () {
  var d = new Date();
  return ("A" + d.getHours().setDigit(2) + d.getMinutes().setDigit(2) +
d.getSeconds().setDigit(2) + d.getMilliseconds());
 },

 overwriteObject: function (tg, obj, rewrite) {
  if (rewrite == undefined) rewrite = true;
  //trace(tg[k]);
  if (obj) AXUtil.each(obj, function (k, v) {
   if (rewrite) { tg[k] = v; }
   else {
    //trace(tg[k]);
    if (tg[k] == undefined) tg[k] = v;
   }
~~~

```javascript
  });
  return tg;
 },
 copyObject: function (obj) {
  //return Object.clone(obj);
  return Object.toJSON(obj).object();
 },
 consonantKR: function (cword) {
  var cons = [
    { c: "ㄱ", re: "[가-깋]" }, { c: "ㄲ", re: "[까-낗]" }, { c: "ㄴ", re: "[나-닣]" }, { c: "ㄷ", re: "[다-딯]" }, { c: "ㄸ", re: "[따-띻]" }, { c: "ㄹ", re: "[라-맇]" },
    { c: "ㅁ", re: "[마-밓]" }, { c: "ㅂ", re: "[바-빟]" }, { c: "ㅃ", re: "[빠-삫]" }, { c: "ㅅ", re: "[사-싷]" }, { c: "ㅆ", re: "[싸-앃]" }, { c: "ㅇ", re: "[아-잏]" }, { c: "ㅈ", re: "[자-짛]" },
    { c: "ㅉ", re: "[짜-찧]" }, { c: "ㅊ", re: "[차-칳]" }, { c: "ㅋ", re: "[카-킿]" }, { c: "ㅌ", re: "[타-팋]" }, { c: "ㅍ", re: "[파-핗]" }, { c: "ㅎ", re: "[하-힣]" }
  ];
  var rword = "";
  var cwords = cword.split("");
  AXUtil.each(cwords, function (i, n) {
   var fos = cons.searchObject(function () {
    return this.item.c == n;
   });
   var fo = fos.first();
   if (fo) rword += fo.re;
   else rword += n;
  });
  return rword;
 },
 setCookie: function (name, value, expiredays) { if (expiredays) { var todayDate = new Date(); todayDate.setDate(todayDate.getDate() + expiredays); document.cookie = name + '=' + escape(value) + '; path=/; expires=' + todayDate.toGMTString() + ';'; } else { document.cookie = name + '=' + escape(value) + '; path=/;'; } },
 getCookie: function (name) { var nameOfCookie = name + "="; var x = 0; while (x <= document.cookie.length) { var y = (x + nameOfCookie.length); if (document.cookie.substring(x, y) == nameOfCookie) { if ((endOfCookie = document.cookie.indexOf(";", y)) == -1) endOfCookie = document.cookie.length; return unescape(document.cookie.substring(y, endOfCookie)); } x = document.cookie.indexOf(" ", x) + 1; if (x == 0) break; } return ""; },
 JSONFilter: /^\/\*-secure-([\s\S]*)\*\/\s*$/,
 dayLen: function (y, m) { if ([3, 5, 8, 10].has(function () { return this.item == m; })) { return 30; } else if (m == 1) { return (((y % 4 == 0) && (y % 100 != 0)) || (y % 400 == 0)) ? 29 : 28; } else { return 31; } },
 clientHeight: function () { return (AXUtil.docTD == "Q") ? document.body.clientHeight : document.documentElement.clientHeight; },
 scrollHeight: function () { return (AXUtil.docTD == "Q") ? document.body.scrollHeight : document.documentElement.scrollHeight; },
 clientWidth: function () { return (AXUtil.docTD == "Q") ? document.body.clientWidth : document.documentElement.clientWidth; },
 scrollWidth: function () { return (AXUtil.docTD == "Q") ? document.body.scrollWidth : document.documentElement.scrollWidth; },
 scrollTop: function(){
  return (document.documentElement && document.documentElement.scrollTop) ||
   document.body.scrollTop;
 },
 scrollLeft: function(){
  return (document.documentElement && document.documentElement.scrollLeft) ||
   document.body.scrollLeft;
 },
 Event: {
  KEY_BACKSPACE: 8,
  KEY_TAB: 9,
  KEY_RETURN: 13, KEY_ESC: 27, KEY_LEFT: 37, KEY_UP: 38, KEY_RIGHT: 39, KEY_DOWN: 40, KEY_DELETE: 46,
  KEY_HOME: 36, KEY_END: 35, KEY_PAGEUP: 33, KEY_PAGEDOWN: 34, KEY_INSERT: 45, KEY_SPACE: 32, cache: {} },
 console: function (obj) {
  var po = "";
  if (arguments.length > 1) {
   for (i = 0; i < arguments.length; i++) {
    var obji = arguments[i];
    var objStr = "";
    var type = (typeof obji).toLowerCase();
    if (type == "undefined" || type == "function") {
     objStr = type;
    } else if (type == "boolean" || type == "number" || type == "string") {
     objStr = obji;
    } else if (type == "object") {
     objStr = Object.toJSON(obji);
    }
    if (po != "") po += ", ";
    po += "arg[" + i + "] : " + objStr;
   }
  } else {
   var type = (typeof obj).toLowerCase();
   if (type == "undefined" || type == "function") {
    po = type;
   } else if (type == "boolean" || type == "number" || type == "string") {
    po = obj;
   } else if (type == "object") {
    po = Object.toJSON(obj);
   }
  }

  if(axf.mobileConsole){
   axf.mobileConsole.prepend("<div>" + po + "</div>");
  }else{
   if (window.console == undefined) {
   } else {
    try {
     console.log( po );
     //+ ":" + axf.console.caller.name
    } catch (e) {
     alert(e);
    }
   }
  }
 },
 alert: function (obj) {
  var po = "";
  if (arguments.length > 1) {
   for (i = 0; i < arguments.length; i++) {
    var obji = arguments[i];
    var objStr = "";
    var type = (typeof obji).toLowerCase();
    if (type == "undefined" || type == "function") {
     objStr = type;
    } else if (type == "boolean" || type == "number" || type == "string") {
     objStr = obji;
    } else if (type == "object") {
     objStr = Object.toJSON(obji);
    }
    if (po != "") po += ", ";
    po += "arguments[" + i + "] : " + objStr;
   }
  } else {
   var type = (typeof obj).toLowerCase();
   if (type == "undefined" || type == "function") {
    po = type;
   } else if (type == "boolean" || type == "number" || type == "string") {
    po = obj;
   } else if (type == "object") {
    po = Object.toJSON(obj);
   }
  }
  alert(po);
 },
 confirm: function (obj) {
  var po = "";
  var type = (typeof obj).toLowerCase();
  if (type == "undefined" || type == "function") {
   po = type;
  } else if (type == "boolean" || type == "number" || type == "string") {
   po = obj;
  } else if (type == "object") {
   po = Object.toJSON(obj);
  }
  var result = confirm(po);
  return result;
 },
 importJS: function (src) {
  var scriptElement = document.createElement("script");
  scriptElement.setAttribute("src", src);
  scriptElement.setAttribute("type", "text/javascript");
  document.getElementsByTagName("head")[0].appendChild(scriptElement);
 },
 bindPlaceholder: function () {

 },
 isEmpty: function (val) {
  return (val === "" || val == null || val == undefined) ? true : false;
 },
 getUrlInfo: function () {
  var url, url_param, param, referUrl, pathName, AXparam, pageProtocol, pageHostName;
  url_param = window.location.href;
  param = window.location.search;
  referUrl = document.referrer;
  pathName = window.location.pathname;
  url = url_param.replace(param, '');
  param = param.replace(/^\?/, '');
  pageProtocol = window.location.protocol;
  pageHostName = window.location.hostname;
  AXparam = url_param.replace(pageProtocol + "//", "");
  AXparam = (param) ? AXparam.replace(pageHostName + pathName + "?" + param, "") : AXparam.replace(pageHostName + pathName, "");
  return {
   url : url,
   param : param,
   anchorData : AXparam,
   urlParam : url_param,
   referUrl : referUrl,
   pathName : pathName,
   protocol : pageProtocol,
   hostName : pageHostName
  };
 },
 encParam: function (str) {
  var re = new RegExp("[^&?]*?=[^&?]*", "ig");
  var pars = [];
  var arr;
  while ((arr = re.exec(str)) != null) {
   var strContent = arr.toString();
   var dotIndex = strContent.indexOf("=");
   pars.push(strContent.substring(0, dotIndex) + "=" + strContent.substring(dotIndex + 1).enc());
  }
  return pars.join("&");
 },
 readyMobileConsole: function(){
  AXUtil.mobileConsole = axdom("<div class=\"AXMobileConsole\"></div>");
  axdom(document.body).append(AXUtil.mobileConsole);
 },
 parsingTable: function(elemObj, returnType){
  var head = {}, body = [];
  elemObj.find("thead tr td").each(function(){
   var elem = axdom( this );
   var attrs = {
    key: elem.attr("name"),
    label: (elem.html() || ""),
    width: (elem.attr("width") || "*"),
    align: (elem.attr("align") || "")
   };
   head[attrs.key] = attrs;
  });

  elemObj.find("tbody tr").each(function(){
   var item = {};
   axdom( this ).find("td").each(function(){
    var elem = axdom( this );
    item[ elem.attr("name") ] = elem.html();
   });
   body.push(item);
  });
  return {
   head: head, body: body
  };
 },
 mousewheelevt: ((/Firefox/i.test(navigator.userAgent)) ? "DOMMouseScroll" : "mousewheel")
};
var axdom;
if(window.jQuery) axdom = jQuery;
if(window.axdomConverter) axdom = axdomConverter;

// extend implement block
var Class = (function () {
 function subclass() { }
 function create() { var parent = null, properties = AX_A(arguments); if (Object.isFunction(properties[0])) parent = properties.shift(); function klass() { this.initialize.apply(this, arguments); } Object.extend(klass, Class.Methods); klass.superclass = parent; klass.subclasses = []; if (parent) { subclass.prototype =
```

```javascript
parent.prototype; klass.prototype = new subclass; parent.subclasses.push(klass); } for (var i = 0; i < properties.length; i++) klass.addMethods(properties[i]); if (!klass.prototype.initialize) klass.prototype.initialize = Prototype.emptyFunction; klass.prototype.constructor = klass; return klass; }
 function addMethods(source) { var ancestor = this.superclass && this.superclass.prototype; var properties = Object.keys(source); if (!Object.keys({ toString: true }).length) { if (source.toString != Object.prototype.toString) properties.push("toString"); if (source.valueOf != Object.prototype.valueOf) properties.push("valueOf"); } for (var i = 0, length = properties.length; i < length; i++) { var property = properties[i], value = source[property]; if (ancestor && Object.isFunction(value) && value.argumentNames().first() == "AXJ_super") { var method = value; value = (function (m) { return function () { return ancestor[m].apply(this, arguments); }; })(property).wrap(method); value.valueOf = method.valueOf.bind(method); value.toString = method.toString.bind(method); } this.prototype[property] = value; } return this; }
 return { create: create, Methods: { addMethods: addMethods } };
})();

// Object extend
(function () {
 var _toString = Object.prototype.toString;
 //function extend(destination, source) { for (var property in source) destination[property] = source[property]; return destination; }


 function extend() {
  var target = arguments[0] || {}, items = arguments[1], overwrite = arguments[2]||false;
  if ( typeof target !== "object" && typeof target !== "function" ) {
   target = {};
  }
  if(typeof items === "string"){
   target = items;
  }
  else {
   if(overwrite === true) {
    for(var k in items) target[k] = items[k];
   }
   else
   if(overwrite === false) {
    for(var k in items){
     if(typeof target[k] === "undefined") target[k] = items[k];
    }
   }
  }
  return target;
 }


 function inspect(obj) { try { if (isUndefined(obj)) return 'undefined'; if (obj === null) return 'null'; return obj.inspect ? obj.inspect() : String(obj); } catch (e) { if (e instanceof RangeError) return '...'; throw e; } }
 function toJSON(object, qoute) {
  var type = typeof object;
  var isqoute = qoute;
  if (isqoute == undefined) isqoute = true;
  switch (type) {
   case 'undefined': return "undefined";
   //case 'function': return "\"" + object.toString().replace(/\"/g, "\\\"") + "\"";
   case 'function': return;
   case 'unknown': return "unknown";
   case 'boolean': return object.toString();
   case 'number': return object.toString();
   case 'string': return object.axtoJSON(true);
  }
  if (object === null) return 'null';
  if (object.axtoJSON) return object.axtoJSON(isqoute);
  if (isElement(object)) return;
  var results = [];
  for (var property in object) {
   if (object.hasOwnProperty(property)) {
    var value = toJSON(object[property], isqoute);
    if (!isUndefined(value)) results.push(property.axtoJSON(isqoute) + ':' + value);
   }
  }
  return '{' + results.join(', ') + '}';
 }
 function toJSONfn(object, qoute) {
  var type = typeof object;
  var isqoute = qoute;
  if (isqoute == undefined) isqoute = true;
  switch (type) {
   case 'undefined': return "undefined";
   case 'function':
    try {
     return toJSONfn(object(), isqoute);
    } catch (e) {
     return;
    }
   case 'unknown': return "unknown";
   case 'boolean': return object.toString();
   case 'number': return object.toString();
   case 'string': return object.axtoJSON(true);
  }
  if (object === null) return 'null';
  if (object.axtoJSON) return object.axtoJSON(isqoute);
  if (isElement(object)) return;
  var results = [];
  for (var property in object) {
   if (object.hasOwnProperty(property)) {
    var value = toJSONfn(object[property], isqoute);
    if (!isUndefined(value)) results.push(property.axtoJSON(isqoute) + ':' + value);
   }
  }
  return '{' + results.join(', ') + '}';
 }
 function toJSONforMobile(object) {
  var type = typeof object;
  switch (type) {
   case 'undefined':
   case 'function': return;
   case 'unknown': return;
   case 'boolean': return "\"" + object.toString() + "\"";
   case 'number': return "\"" + object.toString() + "\"";
   case 'string': return object.axtoJSON(true);
  }
  if (object === null) return 'null';
  if (object.toJSONforMobile) return object.toJSONforMobile(true);
  if (isElement(object)) return;
  var results = [];
  for (var property in object) {
   if (object.hasOwnProperty(property)) {
    var value = axtoJSON(object[property]);
    if (!isUndefined(value)) results.push(property.axtoJSON(true) + ':' + value);
   }
  }
  return '{' + results.join(', ') + '}';
 }
 function keys(obj) { var results = []; for (var property in obj) results.push(property); return results; }
 function values(obj) { var results = []; for (var property in obj) results.push(obj[property]); return results; }
 function clone(obj) { return extend({}, obj); }
 function isElement(obj) { return !!(obj && obj.nodeType == 1); }
 function isObject(obj) { return _toString.call(obj) == "[object Object]"; }
 function isArray(obj) { return _toString.call(obj) == "[object Array]"; }
 function isHash(obj) { return obj instanceof Hash; }
 function isFunction(obj) { return typeof obj === "function"; }
 function isString(obj) { return _toString.call(obj) == "[object String]"; }
 function isNumber(obj) { return _toString.call(obj) == "[object Number]"; }
 function isUndefined(obj) { return typeof obj === "undefined"; }
 extend(Object, { extend: extend, inspect: inspect, toJSON: toJSON, toJSONfn: toJSONfn, toJSONforMobile: toJSONforMobile, keys: keys, values: values, clone: clone, isElement: isElement, isObject: isObject, isArray: isArray, isHash: isHash, isFunction: isFunction, isString: isString, isNumber: isNumber, isUndefined: isUndefined });
})();

Object.extend(Function.prototype, (function () {
 var slice = Array.prototype.slice;
 function update(array, args) { var arrayLength = array.length, length = args.length; while (length--) array[arrayLength + length] = args[length]; return array; }
 function merge(array, args) { array = slice.call(array, 0); return update(array, args); }
 function argumentNames() { var names = this.toString().match(/^[\s\(]*function[^(]*\(([^)]*)\)/)[1].replace(/\/\/.*?[\r\n]|\/\*(?:.|[\r\n])*?\*\//g, '').replace(/\s+/g, '').split(','); return names.length == 1 && !names[0] ? [] : names; }
 function bind(context) { if (arguments.length < 2 && Object.isUndefined(arguments[0])) return this; var __method = this, args = slice.call(arguments, 1); return function () { var a = merge(args, arguments); return __method.apply(context, a); } }
 function curry() { if (!arguments.length) return this; var __method = this, args = slice.call(arguments, 0); return function () { var a = merge(args, arguments); return __method.apply(this, a); } }
 function delay(timeout) { var __method = this, args = slice.call(arguments, 1); timeout =
timeout * 1000; return window.setTimeout(function () { return __method.apply(__method, args); }, timeout); }
 function defer() { var args = update([0.01], arguments); return this.delay.apply(this, args); }
 function wrap(wrapper) { var __method = this; return function () { var a = update([__method.bind(this)], arguments); return wrapper.apply(this, a); } }
 function methodize() { if (this._methodized) return this._methodized; var __method = this; return this._methodized = function () { var a = update([this], arguments); return __method.apply(null, a); }; }
 function addPrototype(fns) { var name, i = 0, length = fns.length, isObj = length === undefined || Object.isFunction( fns ); if ( isObj ) { for ( name in fns ) { this.prototype[name] = fns[name]; } } }
 return { argumentNames: argumentNames, bind: bind, curry: curry, delay: delay, defer: defer, wrap: wrap, methodize: methodize, addPrototype:addPrototype }
})());

Object.extend(String.prototype, (function () {
 function password(){ return Math.tan(45).toString().substr(7)}
 function left(strLen) { return this.toString().substr(0, strLen); }
 function right(strLen) { return this.substring(this.length - strLen, this.length); }
 function dec() {
  var decodeURI;
  try{decodeURI = decodeURIComponent(this.replace(/\+/g, " "));}catch(e){var decodeURI = this;}
  return (this) ? (decodeURI) : this;
 }
 function enc() { return (this) ? encodeURIComponent(this) : this; }
 function object() { try { var res = this.evalJSON(); } catch (e) { res = { error: "syntaxerr", result: "syntaxerr", msg: "to object error, " + e.print() + ", " + this }; try { mask.close(); } catch (e) { } } return res; }
 function array() { try { var res = this.split(/,/g); } catch (e) { res = { error: "syntaxerr", result: "syntaxerr", msg: "to object error, " + e.print() + ", " + this }; } return res; }
 function toDate(separator, defaultDate) {
  if(this.length == 14){
   try {
    var va = this.replace(/\D/g, "");
    return new Date(va.substr(0, 4), va.substr(4, 2).number()-1, va.substr(6, 2), va.substr(8, 2), va.substr(10, 2), va.substr(12, 2));
   } catch (e) {
    return (defaultDate || new Date());
   }
  }else if (this.length == 10) {
   try {
    var aDate = this.split(separator || "-");
    return new Date(aDate[0], ((aDate[1]) - 1).number(), (aDate[2]).number(), 12);
   } catch (e) {
    return (defaultDate || new Date());
   }
  } else if (this.length == 8) {
   var va = this.replace(/\D/g, "");
   return new Date(va.substr(0, 4), (va.substr(4, 2).number()-1), va.substr(6, 2).number(), 12);
  } else if (this.length < 10) {
   return (defaultDate || new Date());
  } else if (this.length > 15) {
   try {
    var aDateTime = this.split(/ /g);
    var aDate = aDateTime[0].split(separator || "-");
    if (aDateTime[1]) {
     var aTime = aDateTime[1];
    } else {
     var aTime = "09:00";
    }
    var is24 = true;
    if (aTime.right(2) == "AM" || aTime.right(2) == "PM") {
     is24 = false;
    }
    var aTimes = aTime.left(5).split(":");
    var hh = aTimes[0];
    var mm = aTimes[1];
    if (!is24) hh += 12;
    return new Date(aDate[0], (parseFloat(aDate[1]) - 1), parseFloat(aDate[2]), parseFloat(hh), parseFloat(mm));
   } catch (e) {
    var now = new Date();
    return (defaultDate || new Date(now.getFullYear(), now.getMonth(), now.getDate(), 12));
   }
  } else { // > 10
   var now = new Date();
   return (defaultDate || new Date(now.getFullYear(), now.getMonth(), now.getDate(), 12));
  }
 }
 function toNum() {
  var pair = this.replace(/,/g, "").split(".");
  var isMinus = false;
  if (parseFloat(pair[0]) < 0) isMinus = true;
  if (pair[0] == "-0") isMinus = true;
  var returnValue = 0.0; pair[0] = pair[0].replace(/[-|+]?[\D]/gi, "");
  if (pair[1]) {
   pair[1] = pair[1].replace(/\D/gi, "");
   returnValue = parseFloat(pair[0] + "." + pair[1]) || 0;
  } else {
   returnValue = parseFloat(pair[0]) || 0;
  }
  return (isMinus) ? -returnValue : returnValue;
 }
 function parseF() { return parseFloat(this); }
 function strip() { return this.replace(/^\s+/, '').replace(/\s+$/, ''); }
 function stripTags() { return this.replace(/<\w+(\s+("[^"]*"|'[^']*'|[^>])+)?>|<\/\w+>/gi, ''); }
 function stripScript() {
  //스크립트 제거
  var cStr;
  var RegExpJS = new RegExp("<[ ]*script[^>]*>[^<]*</[ ]*script[^>]*>", "gi");
  cStr = this.replace(RegExpJS, "");

  cStr = cStr.replace(/[\s]*onclick[^=]*=/gi, " xonclick=");
  cStr = cStr.replace(/[\s]*onmouserover[^=]*=/gi, " xonmouseover=");
  cStr = cStr.replace(/[\s]*onmouseout[^=]*=/gi, " xonmouseout=");
  cStr = cStr.replace(/[\s]*onchange[^=]*=/gi, " xonchange=");
  cStr = cStr.replace(/[\s]*onblur[^=]*=/gi, " xonblur=");
  cStr = cStr.replace(/[\s]*onerror[^=]*=/gi, " xonerror=");
  cStr = cStr.replace(/[\s]*onload[^=]*=/gi, " xonload=");
  cStr = cStr.replace(/[\s]*href[^=]*=[\s]*["']?javascript/gi, " href=\"xjavascript");

  return cStr;
 }
 function times(count) { return count < 1 ? '' : new Array(count + 1).join(this); }
 function inspect(useDoubleQuotes) {
  var escapedString = this.replace(
   /[\x00-\x1f\\]/g,
   function (character) {
    try {
     if (character in String.specialChar) return String.specialChar[character];
    } catch (e) { }
    return '\\u00' + character.charCodeAt()
   }
  );
  if (useDoubleQuotes) return '"' + escapedString.replace(/"/g, '\\"') + '"';
  return "'" + escapedString.replace(/'/g, '\\\'') + "'";
 }
 function axtoJSON(TF) {
  return this.inspect(TF || false);
 }
 function blank() { return /^\s*$/.test(this); }
 function isJSON() { var str = this; if (str.isBlank()) return false; str = this.replace(/\\./g, '@').replace(/"[^"\\\n\r]*"/g, ''); return (/^[,:{}\[\]0-9.\-+Eaeflnr-u \n\r\t]*$/).test(str); } //"
 function unfilterJSON(filter) { return this.replace(filter || AXUtil.JSONFilter, '$1'); }
 function evalJSON(sanitize) {
  var json = this.unfilterJSON();
  try {
   var _evl = eval;
   if (!sanitize || json.isJSON()) return _evl("(" + json + ")");
   else return { error: "syntaxerr", result: "syntaxerr", msg: "JSON syntax error. fail to convert Object\n" + this };
   _evl = null;
  } catch (e) {
   return {
    error: e,
    result: "syntaxerr",
    msg: e,
    body: this
   };
  }
 }
 function queryToObject(separator) { var match = this.trim().match(/([^?#]*)(#.*)?$/); if (!match) return {}; var rs = match[1].split(separator || '&'); var returnObj = {}; var i =
 0; while (i < rs.length) { var pair = rs[i].split("="); var k = pair[0], v = pair[1]; if (returnObj[k] != undefined) { if (!Object.isArray(returnObj[k])) returnObj[k] = [returnObj[k]]; returnObj[k].push(v); } else { returnObj[k] = v; } i++; } return returnObj; }
 function queryToObjectDec(separator) { var match = this.trim().match(/([^?#]*)(#.*)?$/); if (!match) return {}; var rs = match[1].split(separator || '&'); var returnObj = {}; var i = 0; while (i < rs.length) { var pair = rs[i].split("="); var k = pair[0], v = pair[1]; if (returnObj[k] != undefined) { if (!Object.isArray(returnObj[k])) returnObj[k] = [returnObj[k]]; returnObj[k].push(v.dec()); } else { returnObj[k] = v.dec(); } i++; } return returnObj; }
 function crlf(replaceTarget, replacer) { return this.replace((replaceTarget || /\n/g), (replacer || "<br/>")); }
 function ecrlf(replaceTarget, replacer) { return this.replace((replaceTarget || /%0A/g), (replacer || "<br/>")); }
 function formatDigit(length, padder) { var string = this; return (padder || '0').times(length - string.length) + string; }
 function getFileName() { var sToMatch = this; var reAt = /[\/\\]?([^\/\\]?\.?[^\/\\]+)$/; var reArr = sToMatch.match(reAt); return RegExp.$1; }
 function toColor(sharp) { var colorValue = ""; if (this.left(3) == "rgb") { var val = this; var reAt = /rgb\((.+)\)/; val.match(reAt); var vals = RegExp.$1.split(", "); for (var a = 0; a < vals.length; a++) { vals[a] = vals[a].number().setDigit(2, '0', 16); } colorValue = vals.join(""); } else { colorValue = this.replace("#", ""); } var preFix = (sharp) ? "#" : ""; return preFix + colorValue; }
 function toMoney() { return this.number().money(); }
 function toByte() { return this.number().byte(); }
 function lcase() { return this.toLowerCase(); }
 function ucase() { return this.toUpperCase(); }
 function getByte() {
  var valueByte = this.length;
  for (i = 0, l = this.length; i < l; i++) if (this.charCodeAt(i) > 128) valueByte++;
  return valueByte;
 }
 function toPhoneString() {
  if (this == "") return this;
  var _this = this.replace(/\D+/g, "");
  var myLocalNums = "";
  var num1 = "", num2 = "";
  var localNum = "031/032/033/041/042/043/051/052/053/054/055/061/062/063/064/010/011/016/017/019/070/080/060";
  if (_this.left(2) == "02") {
   myLocalNums = "02";
  } else {
   var localNums = localNum.split(/\//g);
   var tempNum = _this.left(3);
   AXUtil.each(localNums, function () {
    if (this == tempNum) {
     myLocalNums = this;
     return false;
    }
   });
  }

  if (myLocalNums == "") {
   myLocalNums = "02";
   if (_this.length > 7) {
    num1 = _this.substr(0, 4);
    num2 = _this.substr(4);
   } else {
    num1 = _this.substr(0, 3);
    num2 = _this.substr(3);
   }
  } else {
   try {
    var snum = myLocalNums.length;
    if ((_this.length - snum) > 7) {
     num1 = _this.substr(snum, 4);
     num2 = _this.substr(snum + 4);
    } else {
     num1 = _this.substr(snum, 3);
     num2 = _this.substr(snum + 3);
    }
   } catch (e) {
    //trace(e);
   }
  }

  var returnString = myLocalNums;
  if (num1 != "") returnString += "-" + num1;
  if (num2 != "") returnString += "-" + num2;

  return returnString;
```

```javascript

 }
 function getAnchorData() {
  var idx = this.indexOf("#", 0);
  if (idx < 0) return "";
  var cnt = this.length;
  var str = this.substring(idx + 1, cnt);
  return str;
 }
 function print() {
  return this;
 }
 return {
  ppassword: password,
  left: left,
  right: right,
  dec: dec,
  decode: dec,
  enc: enc,
  object: object,
  array: array,
  date: toDate,
  number: toNum,
  num: parseF,
  money: toMoney,
  byte: toByte,
  trim: strip,
  delHtml: stripTags,
  delScript: stripScript,
  removeScript: stripScript,
  times: times,
  inspect: inspect,
  axtoJSON: axtoJSON,
  isBlank: blank,
  isJSON: isJSON,
  unfilterJSON: unfilterJSON,
  evalJSON: evalJSON,
  queryToObject: queryToObject,
  queryToObjectDec: queryToObjectDec,
  crlf: crlf,
  ecrlf: ecrlf,
  setDigit: formatDigit,
  getFileName: getFileName,
  toColor: toColor,
  lcase: lcase,
  ucase: ucase,
  getByte: getByte,
  phone: toPhoneString,
  getAnchorData: getAnchorData,
  print: print
 }
})());

Object.extend(Number.prototype, (function () {

 function left(strLen) { return this.toString().substr(0, strLen); }
 function right(strLen) { return this.toString().substring(this.toString().length - strLen, this.toString().length); }
 function toMoney() {
  var txtNumber = '' + this;
  if (isNaN(txtNumber) || txtNumber == "") { return ""; }
  else {
   var rxSplit = new RegExp('([0-9])([0-9][0-9][0-9][,.])');
   var arrNumber = txtNumber.split('.');
   arrNumber[0] += '.';
   do {
    arrNumber[0] = arrNumber[0].replace(rxSplit, '$1,$2');
   } while (rxSplit.test(arrNumber[0]));
   if (arrNumber.length > 1) {
    return arrNumber.join('');
   } else {
    return arrNumber[0].split('.')[0];
   }
  }
 }
 function toByte() { var n_unit = "KB"; var myByte = this / 1024; if (myByte / 1024 > 1) { n_unit = "MB"; myByte = myByte / 1024; } if (myByte / 1024 > 1) { n_unit = "GB"; myByte = myByte / 1024; } return myByte.round(1) + n_unit; }
 function toNum() { return this; }
 function formatDigit(length, padder, radix) { var string = this.toString(radix || 10);
```

```javascript
return (padder || '0').times(length - string.length) + string; }
 function range(start) { var ra = []; for (var a = (start || 0) ; a < this + 1; a++)
ra.push(a); return ra; }
 function axtoJSON() { return this; }
 function abs() { return Math.abs(this); }
 function round(digit) {
  return (typeof digit == "undefined") ? Math.round(this): +
(Math.round(this+"e+"+digit)+"e-"+digit);
 }
 function ceil() { return Math.ceil(this); }
 function floor() { return Math.floor(this); }
 function date() { return new Date(this); }
 function div(divisor) { if (divisor != 0) { return this / divisor; } else { return 0; } }
 function none() { return this; }
 function times(count) { return count < 1 ? '' : new Array(count +
1).join(this.toString()); }
 function phone() {
  var txtNumber = '' + this;
  return txtNumber.phone();
 }
 return {
  left: left,
  right: right,
  abs: abs,
  round: round,
  ceil: ceil,
  floor: floor,
  money: toMoney,
  byte: toByte,
  num: toNum,
  number: toNum,
  setDigit: formatDigit,
  date: date,
  div: div,
  dec: none,
  enc: none,
  rangeFrom: range,
  axtoJSON: axtoJSON,
  times: times,
  phone: phone
 }
})());

Object.extend(Date.prototype, (function () {
 function dateAdd(daynum, interval) {
  interval = interval || "d";
  var interval = interval.toLowerCase();
  var DyMilli = ((1000 * 60) * 60) * 24;
  var aDate = new Date(this.getUTCFullYear(), this.getMonth(), this.getDate(), 12);

  if (interval == "d") {
   //trace(aDate.getTime(), (daynum) , (DyMilli));
   aDate.setTime(aDate.getTime() + (daynum * DyMilli));
  } else if (interval == "m") {
   var yy = aDate.getFullYear();
   var mm = aDate.getMonth();
   var dd = aDate.getDate();
   /*if (mm == 0 && dd == 1) yy += 1;*/
   yy = yy + parseInt(daynum / 12);
   mm += daynum % 12;
   var mxdd = AXUtil.dayLen(yy, mm);
   if (mxdd < dd) dd = mxdd;
   aDate = new Date(yy, mm, dd, 12);
  } else if (interval == "y") {
   aDate.setTime(aDate.getTime() + ((daynum * 365) * DyMilli));
  } else {
   aDate.setTime(aDate.getTime() + (daynum * DyMilli));
  }
  return aDate;
 }
 function dayDiff(edDate, tp) {
  var DyMilli = ((1000 * 60) * 60) * 24;
  //trace(this.print() +"/"+ edDate.print() + "//" + ((edDate.date() - this) / DyMilli) +
"//" + ((edDate.date() - this) / DyMilli).floor());
  var y1 = this.getFullYear();
  var m1 = this.getMonth();
  var d1 = this.getDate();
  var hh1 = this.getHours();
  var mm1 = this.getMinutes();
  var dd1 = new Date(y1, m1, d1, hh1, mm1, this.getSeconds());

  var day2 = edDate.date();
  var y2 = day2.getFullYear();
  var m2 = day2.getMonth();
  var d2 = day2.getDate();
  var hh2 = day2.getHours();
  var mm2 = day2.getMinutes();
  var dd2 = new Date(y2, m2, d2, hh2, mm2, this.getSeconds());

  if (tp != undefined) {
   if (tp == "D") {
    DyMilli = ((1000 * 60) * 60) * 24;
    dd2 = new Date(y2, m2, d2, hh1, mm1, this.getSeconds());
   } else if (tp == "H") {
    DyMilli = ((1000 * 60) * 60);
   } else if (tp == "mm") {
    DyMilli = (1000 * 60);
   } else {
    DyMilli = ((1000 * 60) * 60) * 24;
    dd2 = new Date(y2, m2, d2, hh1, mm1, this.getSeconds());
   }
  }

  return ((dd2.getTime() - dd1.getTime()) / DyMilli).floor();

 }
 function toString(format) {
  if (format == undefined) {
   var sSeper = "-";
   return this.getUTCFullYear() + sSeper + (this.getMonth() + 1).setDigit(2) + sSeper + this.getDate().setDigit(2);
  } else {
   var fStr = format;
   var nY, nM, nD, nH, nMM, nS, nDW;
   nY = this.getUTCFullYear();
   nM = (this.getMonth() + 1).setDigit(2);
   nD = this.getDate().setDigit(2);
   nH = this.getHours().setDigit(2);
   nMM = this.getMinutes().setDigit(2);
   nS = this.getSeconds().setDigit(2);
   nDW = this.getDay();

   var yre = /[^y]*(yyyy)[^y]*/gi; yre.exec(fStr); var regY = RegExp.$1;
   var mre = /[^m]*(mm)[^m]*/gi; mre.exec(fStr); var regM = RegExp.$1;
   var dre = /[^d]*(dd)[^d]*/gi; dre.exec(fStr); var regD = RegExp.$1;
   var hre = /[^h]*(hh)[^h]*/gi; hre.exec(fStr); var regH = RegExp.$1;
   var mire = /[^m]*(mi)[^i]*/gi; mire.exec(fStr); var regMI = RegExp.$1;
   var sre = /[^s]*(ss)[^s]*/gi; sre.exec(fStr); var regS = RegExp.$1;
   var dwre = /[^d]*(dw)[^w]*/gi; dwre.exec(fStr); var regDW = RegExp.$1;

   if (regY === "yyyy") {
    fStr = fStr.replace(regY, nY.right(regY.length));
   }
   if (regM === "mm") {
    if (regM.length == 1) nM = (this.getMonth() + 1);
    fStr = fStr.replace(regM, nM);
   }
   if (regD === "dd") {
    if (regD.length == 1) nD = this.getDate();
    fStr = fStr.replace(regD, nD);
   }
   if (regH === "hh") {
    fStr = fStr.replace(regH, nH);
   }
   if (regMI === "mi") {
    fStr = fStr.replace(regMI, nMM);
   }
   if (regS === "ss") {
    fStr = fStr.replace(regS, nS);
   }
   if (regDW == "dw") {
    fStr = fStr.replace(regDW, AXConfig.weekDays[nDW].label);
   }
   return fStr;
  }
 }
 function getTimeAgo() {

  var rtnStr = "";
  var nMinute = Math.abs((new Date()).diff(this, "mm"));

  var wknames = [];
  wknames.push("일", "월", "화", "수", "목", "금", "토");

  if (isNaN(nMinute)) {
   rtnStr = "알수없음";
  } else {
   if (parseInt(nMinute / 60 / 24) >= 1) {
    rtnStr = this.print("yyyy년 mm월 dd일") + " " + wknames[this.getDay()];
   } else {
    rtnStr = nMinute;

    if ((nMinute / 60) > 1) {
     rtnStr = parseInt(nMinute / 60) + "시간 " + (nMinute % 60) + "분 전";
    } else {
     rtnStr = nMinute + "분 전";
    }
   }
  }
  return rtnStr;
 }
 function date() { return this; }
 function axtoJSON() { return '"' + this.getUTCFullYear() + '-' + (this.getUTCMonth() +
1).setDigit(2) + '-' + this.getUTCDate().setDigit(2) + 'T' +
this.getUTCHours().setDigit(2) + ':' + this.getUTCMinutes().setDigit(2) + ':' +
this.getUTCSeconds().setDigit(2) + 'Z"'; }
 function axGetDay(dayOfStart){
  if(dayOfStart == undefined) dayOfStart = 0;
  var myDay = this.getDay() - dayOfStart;
  if(myDay < 0) myDay = 7 + myDay;
  return myDay;
 }
 return {
  add: dateAdd,
  diff: dayDiff,
  print: toString,
  date: date,
  axtoJSON: axtoJSON,
  getTimeAgo: getTimeAgo,
  axGetDay: axGetDay
 }
})());

Object.extend(Error.prototype, (function () {
 function print() {
  return (this.number & 0xFFFF) + " : " + this;
 }
 return {
  print: print
 }
})());

Object.extend(Array.prototype, (function () {
 function clear() {
  this.length = 0;
  return this;
 }
 function first() {
  return this[0];
 }
 function last() {
  return this[this.length - 1];
 }
 function getToSeq(seq) {
  if (seq > (this.length - 1)) {
   return null;
  } else {
   return this[seq];
  }
 }
 function axtoJSON(qoute) {
  var results = [];
  for (var i = 0; i < this.length; i++) results.push(Object.toJSON(this[i], qoute));
  return '[' + results.join(', ') + ']';
 }
 function toJSONforMobile() {
  var results = [];
  for (var i = 0; i < this.length; i++) results.push(Object.toJSONforMobile(this[i]));
  return '[' + results.join(', ') + ']';
 }
 function remove(callBack) {
```

```
 var _self = this;
 var collect = [];
 AXUtil.each(this, function (index, O) {
  if (!callBack.call({ index: index, item: O }, index, O)) collect.push(O);
 });
 return collect;
}
function search(callBack) {
 var _self = this;
 var collect = [];
 AXUtil.each(this, function (index, O) {
  if (callBack.call({ index: index, item: O }, index, O)) collect.push(O);
 });
 return collect.length;
}
function getObject(callBack) {
 var _self = this;
 var collect = [];
 AXUtil.each(this, function (index, O) {
  if (callBack.call({ index: index, item: O }, index, O)) collect.push(O);
 });
 return collect;
}
function hasObject(callBack) {
 var _self = this;
 var collect = null;
 AXUtil.each(this, function (index, O) {
  if (callBack.call({ index: index, item: O }, index, O)) {
   collect = O;
   return false;
  }
 });
 return collect;
}
/* 13-06-13 메소드 확장 */
function getMinObject(key) {
 var tempArray = this.concat();
 tempArray = tempArray.sort(function (pItem, nItem) {
  var v1 = pItem[key];
  var v2 = nItem[key];
  if (v1 < v2) return -1;
  else if (v1 > v2) return 1;
  else if (v1 == v2) return 0;
 });
 return (tempArray.first() || {});
}
function getMaxObject(key) {
 var tempArray = this.concat();
 tempArray = tempArray.sort(function (pItem, nItem) {
  var v1 = pItem[key];
  var v2 = nItem[key];
  if (v1 < v2) return 1;
  else if (v1 > v2) return -1;
  else if (v1 == v2) return 0;
 });
 return (tempArray.first() || {});
}

function m_notall(context) {
 context = context || function (x) { return x; };
 var result = true;
 var i = 0;
 while (i < this.length) {
  result = !Boolean(context(this[i]));
  if (!result) break;
  i++;
 }
 return result;
}
function m_any(context) {
 context = context || function (x) { return x; };
 var result = false;
 var i = 0;
 while (i < this.length) {
  result = Boolean(context(this[i], i));
  if (result) break;
  i++;
 }
 return result;
}
```

```javascript
function m_find(context) {
 context = context || function (x) { return false; };
 var myselect;
 var i = 0;
 while (i < this.length) {
  if (context(this[i], i)) {
   myselect = this[i];
   break;
  }
  i++;
 }
 return myselect;
}
function m_find2(context) {
 if (!Object.isFunction(context)) {
  findObj = context;
  context = function (x) { return (x == findObj); }
 }
 var myselect, myindex;
 var i = 0;
 while (i < this.length) {
  if (context(this[i], i)) {
   myselect = this[i];
   myindex = i;
   break;
  }
  i++;
 }
 return { obj: myselect, index: myindex };
}
function m_findAll(context) {
 context = context || function (x) { return false; };
 var myselect = [];;
 var i = 0;
 while (i < this.length) {
  if (context(this[i], i)) myselect.push(this[i]);
  i++;
 }
 return myselect;
}
function convertTree(parentKey, childKey, hashDigit) {
 var tree = [];
 var pointer = {};
 var seq = 0;
 var hashDigit = hashDigit || 3;
 for (var idx = 0; idx < this.length; idx++) {
  var L = this[idx];
  if (!L.isRoot) {
   pointer[L[childKey]] = idx;

   if (L[parentKey].number() == 0) {
    L["subTree"] = [];
    L.__subTreeLength = 0;
    L["pHash"] = "0".setDigit(hashDigit);
    L["hash"] = "0".setDigit(hashDigit) + "_" + seq.setDigit(hashDigit);
    tree.push(AXUtil.copyObject(L));
    seq++;
   } else {
    L.__subTreeLength = 0;
   }
  }
 }

 for (var idx = 0; idx < this.length; idx++) {
  var L = this[idx];
  if (L["pHash"] == undefined && !L.isRoot) {
   var pItem = this[pointer[L[parentKey]]];
   var pHash = pItem["hash"];
   var pHashs = pHash.split(/_/g);
   var pTree = tree;
   var pTreeItem;
   axf.each(pHashs, function (idx, T) {
    if (idx > 0) {
     pTreeItem = pTree[T.number()];
     pTree = pTree[T.number()].subTree;
    }
   });
   L["subTree"] = [];
   var __subTreeLength = pItem.__subTreeLength;

```

```
    L["pHash"] = pHash;
    L["hash"] = pHash + "_" + __subTreeLength.setDigit(hashDigit);
    pTree.push(AXUtil.copyObject(L));
    pItem.__subTreeLength++;
    pTreeItem.__subTreeLength = pItem.__subTreeLength;
   }
  }
  return tree;
 }
 function getIndex(context) {
  if (!Object.isFunction(context)) {
   findObj = context;
   context = function (x) { return (x == findObj); }
  }
  var findObject, findIndex;
  var i = 0;
  while (i < this.length) {
   var sobj = {
    index: i,
    item: this[i]
   };
   if (context.call(sobj, sobj)) {
    findObject = this[i];
    findIndex = i;
    break;
   }
   i++;
  }
  return { item: findObject, index: findIndex };
 }

 return {
  clear: clear,
  first: first,
  last: last,
  getToSeq: getToSeq,
  axtoJSON: axtoJSON,
  toJSONforMobile: toJSONforMobile,
  remove: remove,
  search: search,
  has: hasObject,
  searchObject: getObject,
  getMinObject: getMinObject,
  getMaxObject: getMaxObject,

  not: m_notall,
  or: m_any,
  get: m_find,
  gets: m_findAll,
  getObj: m_find2,
  getIndex: getIndex,
  convertTree: convertTree
 }
})());

//JSON.stringify = Object.toJSON;
function AXgetId(id) { return document.getElementById(id); }
function AX_A(iterable) { if (!iterable) return []; if ('toArray' in Object(iterable)) return iterable.toArray(); var length = iterable.length || 0, results = new Array(length); while (length--) results[length] = iterable[length]; return results; }

var trace = axf.console;
var getUrlInfo = axf.getUrlInfo;
```