//map 方法封装 \_map

Array.prototype.\_map = function ( callback, context ) {

var temp = [];

for ( var k = 0; k < this.length; k ++ ) {

temp.push ( callback.call ( context, this[k], k, this ) );

}

return temp;

}

//filter 方法封装 \_filter

Array.prototype.\_filter = function ( callback, context ) {

var temp = [];

for ( var k = 0; k < this.length; k ++ ) {

temp.call ( context, this[k] ,k, this ) ? temp.push ( this[k] ) : “”;

}

return temp;

}

//some 方法封装 \_some

Array.prototype.\_some = function ( callback, context ) {

for ( var k = 0; k < this.length; k ++ ) {

if ( callback. call ( context, this[k] ,k, this ) ) {

return true;

}

}

return false;

}

//every 方法封装 \_every

Array.prototype.\_every = function ( callback, context ) {

for ( var k = 0; k < this.length; k ++ ) {

if ( ! callback. call ( context, this[k] ,k, this ) ) {

return false;

}

}

return true;

}

//forEach 方法封装 \_forEach

function \_forEach ( arr, func ) {

var i;

if ( arr instanceof Array || arr.length >= 0 ) {

for ( i = 0; i < arr.length; i ++ ) {

func.call ( arr [ i ], arr [ i ], i );

}

} else {

for ( i in arr ) {

func.call ( arr [ i ], arr [ i ], i );

}

}

return arr;

}