

# ccchartReference

(v.1.01)

chartSetting

configオブジェクトへ記述	プロパティ名	type	値	説明	例	省略時		
種類設定	チャート	type	chartname				-	"line"
			"line"	ラインチャート				
			"bar"	バーチャート				
			"stacked"	積み上げチャート				
			"area"	面チャート。色の透過度はcolorSetへRGBAやHSLA形式で指定				
			"stackedarea"	積み上げ面チャート。縦方向に積み上げる面(Area)チャート				
			"stacked%"	構成比推移チャート。縦方向に%で積み上げる面(Area)チャート				
			"bezi2"	ベジェラインチャート2				
			"bezi"	ベジェラインチャート				
			"pie"	円(ドーナツ)チャート				
			"scatter"	散布図。useMarkerが必須。				
			"ampli"	始点と終点を結ぶライン上の振幅を塗るチャート				

configオブジェクトへ記述	プロパティ名	type	説明	例	省略時		
領域設定	チャート領域	width	number	canvasの幅	for all chartTypes	300	600
		height	number	canvasの高さ	for all chartTypes	200	400
		axisXWidth	number	水平目盛り線の幅	for all chartTypes	2	1
		axisXLen	number	水平目盛り線の本数	for all chartTypes (垂直目盛り線はdata列数から自動計算)	20	10
		axisYWidth	number	垂直目盛り線の幅	for all chartTypes xScaleSkip時は強制でスキップ線が1、節目の線が3	2	1
		axisYLen	number	垂直目盛り線の本数	for all chartTypes (垂直目盛り線はdata列数から自動計算)	20	10
		axisYLen	number	scatter時の垂直目盛り線の本数	for scatter (垂直目盛り線はdata列数から自動計算)	20	10
		onlyChart	string	チャートのみを表示	for all chartTypes (タイトル,サブタイトル,水平垂直目盛無し)	"yes"	"no"
		onlyChartWidthTitle	string	チャートとタイトルのみを表示	for all chartTypes (サブタイトル,水平垂直目盛無し)	"yes"	"no"
		paddingTop	number	チャートの上パディング	for all chartTypes	100	自動計算
		paddingBottom	number	チャートの下パディング	for all chartTypes	100	自動計算
		paddingLeft	number	チャートの左パディング	for all chartTypes	100	自動計算
		paddingRight	number	チャートの右パディング	for all chartTypes	100	自動計算
		xColor	color	水平目盛り線の色	for all chartTypes	"#000"	rgba(180,180,180,0.3)
		yColor	color	垂直目盛り線の色	for all chartTypes	"#000"	rgba(180,180,180,0.3)

読み出し専用	ccchart.プロパティ名	説明	例
	ccchart.ops[id名].プロパティ名		ccchart.ops['hoge0'].chartTop
	chartWidth	グラフ領域の幅	
	chartHeight	グラフ領域の高さ	
	chartTop	グラフ領域の上端	
	chartBottom	グラフ領域の下端	
	chartLeft	グラフ領域の左端	
	chartRight	グラフ領域の右端	
	yGap	水平目盛線用の目盛間隔 Y方向のギャップ	
	xGap	垂直目盛線用の目盛間隔 X方向のギャップ	
	axisYLen	垂直目盛本数	

configオブジェクトへ記述	プロパティ名	type	説明	例	省略時	
背景設定	背景色	bg	string	背景色	bg > bgGradient > vertical, from #687478, to #222	
		bgGradient	object	背景色グラデーション	{direction:"vertical",from:"#687478",to:"#222"}	
	グラデーション	direction	color	グラデーションの方向 vertical(上から下へ)   horizontal(左から右へ)	"horizontal"	"vertical"
		from	color	開始色	"#000"	"#687478"
	to	color	終了色	"#fff"	"#222"	

configオブジェクトへ記述	プロパティ名	type	説明	例	省略時		
文字列設定	タイトル	title	string	タイトル文字列	for all chartTypes	"タイトルです"	"" (省略時のpaddingTopは自動調整)
		titleColor	color	色	for all chartTypes	"red"	titleColor > textColor > textColors.title > #ddd
		titleFont	font	フォント	for all chartTypes http://www.w3.org/TR/2dcontext/#text-styles	"900 10px 'serif"	"100 28px 'Arial"
	サブタイトル	titleTextAlign	align	位置	for all chartTypes "start" "end" "left" "right" "center"	"left"	"center"
		titleY	number	タイトルのtop位置	for all chartTypes	10	38
		subTitle	string	サブタイトル文字列	for all chartTypes	"サブタイトルです"	"" (省略時のpaddingTopは自動調整)
	水平軸目盛値	subTitleColor	color	色	for all chartTypes	"#000"	subTitleColor > textColor > textColors.subTitle > #ccc
		subTitleFont	font	フォント	for all chartTypes http://www.w3.org/TR/2dcontext/#text-styles	"900 10px 'serif"	"100 12px 'Arial"
		subTitleTextAlign	align	位置	for all chartTypes "start" "end" "left" "right" "center"	"left"	"center"
	垂直軸目盛値	subTitleY	number	サブタイトルのtop位置	for all chartTypes	10	55 (if(this.title === "")subTitleY = 25;)
		xScaleColor	color	水平軸目盛値の色	without pie ※水平軸目盛はdadaの1行目を項目名とする場合に表示	"#000"	xScaleColor > textColor > textColors.x > #aaa
		xScaleFont	font	フォント	without pie	"100 10px 'serif"	"100 12px 'Arial"
	凡例	xScaleAlign	align	位置	without pie "start" "end" "left" "right" "center"	"left"	"center"
		xScaleXOffset	number	Xオフセット	without pie 目盛のXは、垂直目盛線X + xGap/2 - xScaleXOffset	5	0
		xScaleYOffset	number	Yオフセット	without pie 目盛と項目名タイトルのYは、chartBottom + xScaleYOffset	10	18
		xScaleSkip	number	X軸目盛のスキップ 数値	without pie 与えた数値分表示をスキップしY軸幅が大きくなります	3	0
		colNamesTitleOffset	number	項目名タイトルのXは、chartRight + colNamesTitleOffset	without pie	15	22
yScaleColor		color	垂直軸目盛値の色	without pie ※yScalePercent=="yes"と "stacked%"は%表示	"#000"	yScaleColor > textColor > textColors.y > #aaa	
yScaleFont		font	フォント	without pie	"100 10px 'serif"	"100 12px 'Arial"	
yScaleAlign		align	位置	without pie "start" "end" "left" "right" "center"	"left"	"right"	
yScaleXOffset		number	Xオフセット	without pie ラベルのXは、垂直目盛線X + xGap/2 - xScaleXOffset	5	0	
yScaleYOffset		number	Yオフセット	without pie ラベルと項目名のYは、chartBottom + xScaleYOffset	10	20	
単位	yScalePercent	string	Y軸目盛のパーセント表示	without pie "stacked"は強制"yes"	"yes"	"no"	
	hanreiColor	color	凡例文字列の色	for all chartTypes ( マーカーの色はthis.colorSet )	"#000"	hanreiColor > textColor > textColors.hanrei > #ccc	
	hanreiFont	font	フォント	for all chartTypes	"100 12px 'Arial"	項目数によって自動計算 "100" + fontSize + "px 'Arial"	
	hanreiAlign	align	位置	for all chartTypes "start" "end" "left" "right" "center"	"right"	"left"	
	hanreiLineHeight	number	1行の高さ	for all chartTypes	20	項目数によって自動計算 (len < 10) ? 20 : (len < 20) ? 14 : 8	
	hanreiXOffset	number	Xオフセット	for all chartTypes ラベルのXは、垂直目盛線X + xGap/2 - xScaleXOffset	5	14	
	hanreiYOffset	number	Yオフセット	for all chartTypes ラベルと項目名のYは、chartBottom + xScaleYOffset	10	40	
	hanreiRadius	number	マーカー半径(または1/2辺)	for all chartTypes hanreiMarkerStyle === "arc"なら半径, "rect"なら1/2辺。	10	40	
	hanreiMarkerStyle	string	マーカーの形	for all chartTypes ラベルと項目名のYは、chartBottom + xScaleYOffset	"rect"	"arc"	
	unit	string	単位文字列	for all chartTypes ※language=="ja"なら頭に'を付けます	"千円"	省略時は単位表示なし	
メモ	unitColor	color	色	for all chartTypes	"#000"	unitColor > textColor > textColors.unit > #aaa	
	unitFont	font	フォント	for all chartTypes	"100 10px 'serif"	"100 12px 'Arial"	
	unitAlign	align	位置	for all chartTypes "start" "end" "left" "right" "center"	"left"	"right"	
	unitXOffset	number	Xオフセット	for all chartTypes 目盛のXは、垂直目盛線X + xGap/2 - xScaleXOffset	5	10	
	unitYOffset	number	Yオフセット	for all chartTypes 目盛と項目名タイトルのYは、chartBottom + xScaleYOffset	10	12	
	※memoを参照						
	プロパティ名	type	oj内の各プロパティ type	説明	例	省略時	
	文字色一括指定	textColors	object	各文字列の色を一括指定(データ値valColor以外) for all chartTypes		{title:"red",subTitle:"blue",x:"green",y:"purple",hanrei:"cyan",unit:"magenta",memo:"black"}	
	単位一括指定	title	color	タイトルの色	"#000"	"#ccc"	
		subTitle	color	サブタイトルの色	"#000"	"#ddd"	
x		color	X方向の水平軸ラベル色	"#000"	"#aaa"		
y		color	Y方向の垂直軸ラベル色	"#000"	"#aaa"		
hanrei		color	凡例の色	"#000"	"#ccc"		
unit		color	単位の色	"#000"	"#aaa"		
memo		color	メモの色	"#000"	"#ccc"		
unit		object	単位を一括指定 for all chartTypes				
unit		string	単位文字列	"千円"	省略時は単位表示なし		
left		number	canvas左辺からの距離	5	unit.left > chartLeft - (unitXOffset    10)		
top	number	canvas上辺からの距離	10	unit.top > chartTop - (unitYOffset    12)			
align	align	位置	"left"	unit.align > unitAlign    "right"			
color	color	色	"#000"	unit.color > unitColor > textColors.unit > #aaa			
font	font	フォント	"100 10px 'serif"	unit.font > unitFont > "100 12px 'Arial"			
影一括指定	shadows	object	影を一括指定				
	all	shadow	指定があれば、下記すべての影のカスタムデフォルト値 [color, x, y, blur]	['#ddd', 3, 3, 3]	undefined		
	hanrei	shadow	凡例の影 [color, x, y, blur]	['#ddd', 5, 10, 5]	['#222', 5, 5, 5]		
	xline	shadow	水平カスタム目盛線の影 [color, x, y, blur]	['#222', 6, 6, 6]	['#444', 7, 7, 5]		
	line	shadow	ラインチャートの影 [color, x, y, blur]	['#eee', 5, 5, 5]	['#222', 5, 5, 5]		
	bar	shadow	バーチャートの影 [color, x, y, blur]	['#aaa', 5, 3, 5]	['#222', 5, 5, 5]		
	stacked	shadow	積み上げチャートの影 [color, x, y, blur]	['#222', 2, 2, 2]	['#222', 5, -5, 5]		
	stackedarea	shadow	積み上げ面チャートの影 [color, x, y, blur]	['#222', 5, 5, 5]	['#222', 5, 5, 5]		
	bezi2	shadow	ベジェラインチャート2の影 [color, x, y, blur]	['#222', 3, 3, 3]	['#222', 5, 5, 5]		
	drawbeziline2	shadow	ベジェラインチャートの影 [color, x, y, blur]	['#222', 5, 5, 5]	['#222', 5, 5, 5]		

configオブジェクトへ記述	プロパティ名	type	説明	例	省略時	
デバッグ	WebSocket用	wsInfo	boolean	WebSocketの接続切断等簡単な情報出力	true	false
		wsDbg	boolean	WebSocketの詳細なデバッグ用情報出力	true	false
configオブジェクトへ記述	プロパティ名	type	説明	例	省略時	
データ設定	データ	useFirstToColName	boolean	trueならデータ配列の1行目を項目名とする for all chartTypes	false	true
		useFirstToRowName	boolean	trueならデータ配列の1列目を項目名とする for all chartTypes	false	true
		maxY	number	データの最大値	5000	データから算出
		minY	number	データの最小値	200	データから算出
		roundedUpMaxY	number	maxYをどの位で切り上げるか for all chartTypes (0 か %表示の時は切り上げなし)	1000	100
		maxX	number	Xデータの最大値	5000	データから算出
		minX	number	Xデータの最小値	200	データから算出
		roundedUpMaxX	number	maxXをどの位で切り上げるか for scatter (0 か %表示の時は切り上げなし)	1000	100
		colorSet	array	データ列の順に色付けするための配列	["red","orange"]	["red","#ff9114","#3CB371","#00A821","#003366","#3399FF","#FF3300"]

メソッド & プロパティ	名称	説明	例	省略時
初期化	<b>init</b>	function 書式 ccchart.init(id, op[, callback]) 戻値 ccchartオブジェクト 引数 <b>id</b> string, oj <b>op</b> object ※上記configオブジェクト参照 <b>callback</b> function	チャートの初期化	ccchart.init('hoge', chartdataObject)
	<b>memo</b>	function 書式 ccchart.memo(memoObject) 戻値 ccchartオブジェクト 引数 <b>memoObj</b> object <b>val</b> string メモの内容 <b>left</b> number canvas左辺からの距離 <b>top</b> number canvas上辺からの距離 <b>font</b> font フォント <b>align</b> align 位置 <b>lineTo</b> Array メモからチャートへ線を引く。線分終点の [ データRowName, データColName ] で指定 <b>lineToColor</b> string memoメソッドのプロパティlineTo時のline色 <b>lineToWidth</b> number memoメソッドのプロパティlineTo時のline幅 <b>lineToXOffset</b> number memoメソッドのプロパティlineTo時のメモ側のX位置 <b>lineToYOffset</b> number memoメソッドのプロパティlineTo時のメモ側のY位置	ccchart.memo(id, memoObj)	ccchart.init('hoge', chartdataObject).memo(memoObject)
WebSocket	<b>ws</b>	function 書式 ccchart.ws(url[, op]) 戻値 WebSocket接続インスタンス ※WebSocket 接続のインスタンスへは ccchart.wsRecent や ccchart.wsuids[uid]、あるいは ccchart.getWs(op)などでアクセスできる。 引数 <b>url</b> string WebSocketサーバーのURL <b>op</b> object <b>autoReConnect</b> boolean 自動再接続フラグ <b>maxReConnect</b> number 最大再接続回数 <b>protocol</b> string WebSocket のヘッダ Sec-WebSocket-Protocol に使われる文字列	WebSocketを接続する	ccchart.init('hoge', chartdataObject).ws('ws://ccchart.com:8016')
	<b>wsRecent</b>	object WebSocket 書式 ccchart.wsRecent 戻値 WebSocket接続インスタンス プロパティ <b>op.uid</b> string “-ccchart-ws-”で始まるWebSocketインスタンス毎の一意な UUID v4 <b>op.id</b> string この接続に紐づいている DOM要素の id名 <b>op.url</b> string 接続先url すべて小文字で末尾の“/”を除去したスタイル <b>op.useHb</b> boolean ハートビートを使うかどうか true/ false <b>op.hbStr</b> string ハートビート用文字列 configオブジェクトのプロパティ hbStr で設定 <b>op.hbInterval</b> number ハートビート文字列送信のインターバルミリ秒 configオブジェクトのプロパティ hbInterval で設定 <b>op.hbTimer</b> number ハートビート文字列送信のインターバルタイマー ※通常は内部用(clearInterval(w.hbTimer) で手動停止も可能) <b>op.RTT</b> number ハートビートのサーバー往復ミリ秒 ラウンドトリップディレイタイム (Round-Trip delay Time) <b>on</b> event method (戻値 WebSocket) WebSocket APIの addEventListener ショートカット その他 new WebSocket で生成されるメソッドやプロパティすべて @see http://www.w3.org/TR/websockets/	ccchart上で生成されたws接続の最新のインスタンス	ccchart.wsRecent
	<b>wsuids</b>	hash 書式 ccchart.wsuids['-ccchart-ws-'+uuidv4] ※keyは“-ccchart-ws-”で始まりUUIDを接続した文字列 戻値 WebSocket接続インスタンスのリスト ※各インスタンスに含まれるプロパティは上記wsRecentのプロパティと同じ	ccchart上で生成されたws接続インスタンスのリスト (UUID v4 を使用)	ccchart.wsuids['-ccchart-ws-d7f04e4d-c813-4337-8ccf-11a46eaa23ae']
	<b>wses (v1.01で廃止)</b>	hash ※id+urlではインスタンスdeleteが間に合わなかった場合などの新規接続生成時に衝突リスクがあるため廃止	ccchart上で生成されたws接続のリスト wses[id+“-ccchart-”+url]の書式	ccchart.wses["hoge8-ccchart-ws://ccchart.com:8014"]
	<b>wscase</b>	object 引数 <b>id</b> string WebSocketインスタンスを作成したccchartのcanvas id名 <b>on</b> event WebSocket APIの addEventListener ショートカット その他 ws API new WebSocket で生成されるメソッドやプロパティすべて @see http://www.w3.org/TR/websockets/ WebSocketの受信パターン関数 ccchart.init('hoge', chartdataObject).ws('ws://ccchart.com:8016').on('message', ccchart.wscase.oneColAtATime)	一度に1列ずつ["23:41:47",58,41]といった配列で届く場合用 一度に複数列ずつ届く場合	[["23:41:47",58,41]] [[["製品A",435,600],[["製品B",332,335],[["製品C",524,584]]]
	<b>wsCloseAll</b>	function	ccchart.wsuids に登録されているすべての ws接続を閉じてインスタンスを削除	ccchart.wsCloseAll()
	<b>wsCloseByUid</b>	function 書式 ccchart.wsCloseByUid(uid) 戻値 ccchartオブジェクト 引数 <b>uid</b> string	引数で与えたUIDのWebSocket接続を閉じてインスタンスを削除する	ccchart.wsCloseByUid("-ccchart-ws-d7f04e4d-c813-4337-8ccf-11a46eaa23ae")
	<b>wsCloseById</b>	function 書式 ccchart.wsCloseById(id) 戻値 ccchartオブジェクト 引数 <b>id</b> string	引数で与えたidに紐付いたWebSocket接続すべてを閉じてインスタンスを削除	ccchart.wsCloseById('hoge0')
	<b>wsCloseByUrl</b>	function 書式 ccchart.wsCloseById(url) 戻値 ccchartオブジェクト 引数 <b>url</b> string	引数で与えたurlに紐付いたWebSocket接続すべてを閉じてインスタンスを削除	ccchart.wsCloseByUrl('ws://ccchart.com:80')
	<b>wsCloseByIdUrl</b>	function 書式 ccchart.wsCloseById(id) 戻値 ccchartオブジェクト 引数 <b>id</b> string <b>url</b> string	引数で与えたidかつURLに紐付いたWebSocket接続すべてを閉じてインスタンスを削除	ccchart.wsCloseById('hoge0', 'ws://ccchart.com:80')
<b>wsDelTarget</b>	function 書式 ccchart.wsDelTarget(target) 戻値 ccchartオブジェクト 引数 <b>target</b> object WebSocket	引数で与えたWebSocketインスタンスを削除する	ccchart.wsDelTarget(ccchart.wsuids["-ccchart-ws-d7f04e4d-c813-4337-8ccf-11a46eaa23ae"])	
<b>getWs</b>	function 書式 ccchart.getWs(uid) 戻値 WebSocket 引数 <b>uid</b> string	引数で与えたUIDのWebSocketインスタンスを取得する	ccchart.getWs("-ccchart-ws-d7f04e4d-c813-4337-8ccf-11a46eaa23ae")	

CSSマーカー サンプル <http://ccchart.com/#67>



```

<script src="http://ccchart.com/js/ccchart.js" charset="utf-8"></script>
<canvas id="hoge"></canvas>
<script>
var chartdata67 = {

"config": {
"title": "Option useMarker: css-ring",
"subTitle": "useMarkerにcss-ringを指定するとCanvasではなくCSSで輪を描きHoverします。",
"type": "line",
"useMarker": "css-ring",
"lineWidth": 8,
"borderWidth": 7,
"markerWidth": 28
},

"data": [
["年度",2007,2008,2009,2010,2011,2012,2013],
["紅茶",435,332,524,688,774,825,999],
["コーヒー",600,335,584,333,457,788,900],
["ジュース",60,435,456,352,567,678,1260],
["ウーロン",200,123,312,200,402,300,512]
]
};
ccchart.init("hoge", chartdata67);
</script>

```

WebSocketの受信パターン関数 <http://ccchart.com/test/ws2.htm>

```

ccchart.wscase.oneColAtATime

一度に1列ずつ、たとえば、下記データの場合なら [ ["2013"], [435], [600] ] といった配列で届く場合

[
["2013", "2014", "2015", "2016", "2017"],
[ 435, 332, 524, 688, 774],
[ 600, 335, 584, 333, 457]
];

ccchart
.init('hoge0', chartCfg0)
.ws('ws://ccchart.com:8021') //これ以降のチェンスコープはWS
.on('open', function(){ console.log('opened!') });
.on('message', ccchart.wscase.oneColAtATime);

//oneColAtATimeは、WebSocketの受信パターン関数
//一度に1列ずつ [ ["2013"], [435], [600] ] といった配列で届く場合用

```

