

# ccchart Reference

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、色の透明度は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"	ベジエラインチャート2		
				"bezi"	ベジエラインチャート		
				"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	値	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	チャート上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	"#0000"	rgba(180,180,180,0.3)
		yColor	color	垂直目盛り線の色	for all chartTypes	"#0000"	rgba(180,180,180,0.3)
読み出し専用	ccchart.プロパティ名	プロパティ名	type	説明			
	または、 ccchart.ops[id名].プロパティ名 でアクセス	chartWidth		グラフ領域の幅			ccchart.ops['hogeo0'].chartTop
		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	値	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	
	プロパティ名	type	プロパティ名	type	説明	例	省略時
グラデーション	bgGradient	object		背景色グラデーション			{direction:'vertical',from:'#687478',to:'#222'}
				direction	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	開始色	"#000"	"#687478"
				to	終了色	"#fff"	"#222"
configオブジェクトへ記述		プロパティ名	type	配列のパターン	説明	例	省略時
画像設定	img	array		画像を埋め込む ( <img>要素・<canvas>要素・<video>要素 )			
	*この配列パターンに後日 Alphaも埋め込みたい	[imgsrc]		チャート全体に画像表示	["hogeo.png"]	null	
		[imgsrc, dx, dy]	dx, dy	dxは、描画するイメージを配置する座標	["hogeo.png", 80, 120]	null	
		[imgsrc, dx, dy, dw, dh]	dw, dh	dwは幅と高さ	["hogeo.png", 60, 180, 250, 80]	null	
		[imgsrc, sx, sy, sw, sh, dx, dy, dw, dh]	使用範囲を指定	["hogeo.png", 50, 50, 100, 50, 10, 10, 200, 50]		null	
		複数画像を二次元配列で指定できる (add v1.06.3+)		[[{"img": "hogeo1.png", "x": 450, "y": 135, "w": 100, "h": 30}, {"img": "hogeo2.png", "x": 120, "y": 135, "w": 100, "h": 30}]]		null	
	imgAlpha (add v1.06.3+)	number		画像の透明度	0.5		1
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'"
		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
サブタイトル	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)
		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'"
		subTitle.TextAlign	align	位置	for all chartTypes "start" "end" "left" "right" "center"	"left"	"center"
		subTitleY	number	サブタイトルのtop位置	for all chartTypes	10	55 (if(this.title === "subTitleY = 25;")
水平軸目盛値	xScaleColor	color		水平軸目盛値の色	widthout pie ※水平軸目盛はdata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	widthout pie	"100 10px 'serif'"	"100 12px 'Arial'"
		xScaleAlign	align	位置	widthout pie "start" "end" "left" "right" "center"	"left"	"center"
		xScaleOffset	number	Xオフセット	widthout 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	widthout 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	widthout 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	項目名タイトルオフセットX	widthout pie 項目名タイトルのXは、chartRight + colNamesTitleOffset	15	22
垂直軸目盛値	yScaleColor	color		垂直軸目盛値の色	widthout pie ※yScalePercent=="yes"と"stacked%"は%表示	"#000"	yScaleColor > textColor > textColors.y > #aaa
		yScaleFont	font	フォント	widthout pie	"100 10px 'serif'"	"100 12px 'Arial'"
		yScaleAlign	align	位置	widthout pie "start" "end" "left" "right" "center"	"left"	"right"
		yScaleXOffset	number	Xオフセット	widthout 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	widthout pie ラベルと項目名のYは、chartBottom + xScaleYOffset	10	20
		yScalePercent	string	Y軸目盛のパーセント表示	widthout pie "stacked%"は強制"yes"	"yes"	"no"
凡例	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"
単位	unit	string		単位文字列	for all chartTypes ※language === "ja"なら頭に"単位:"を付けます	"千円"	省略時は単位表示なし
		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
メモ	※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
		textColors	object	各文字列の色を一括指定(データ値valColor以外) for all chartTypes			
				title	タイトルの色	"#000"	"#ccc"
				subTitle	サブタイトルの色	"#000"	"#ddd"
				x	X方向の水平軸ラベル色	"#000"	"#aaa"
				y	Y方向の垂直軸ラベル色	"#000"	"#aaa"
				hanrei	凡例の色	"#000"	"#ccc"
				unit	単位の色	"#000"	"#aaa"
				memo	メモの色	"#000"	"#ccc"
単位一括指定	unit	object		単位を一括指定 for all chartTypes			
				unit	単位文字列	"千円"	省略時は単位表示なし
				left	canvas左辺からの距離	5	unit.left > chartLeft - (unitXOffset    10)
				top	canvas上辺からの距離	10	unit.top > chartTop - (unitYOffset    12)
				align	位置	"left"	unit.align > unitAlign    "right"
				color	色	"#000"	unit.unit > unitColor > extColor > textColors.unit > #aaa
				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	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		

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 data自体は "data"配列 として記述する	<code>useFirstToColName</code>	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
		<code>useFirstToRowName</code>	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
		<code>maxY</code>	number	データの最大値	5000	データから算出
		<code>minY</code>	number	データの最小値	200	データから算出
		<code>roundedUpMaxY</code>	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
		<code>maxX</code>	number	Xデータの最大値 for scatter	5000	データから算出
		<code>minX</code>	number	Xデータの最小値 for scatter	200	データから算出
		<code>roundedUpMaxX</code>	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
		<code>colorSet</code>	array	データ列の順に色付けるための配列	["red", "orange"]	["red", "#FF9114", "#00ABA2", "#0036CD", "#C328FF", "#FF34CD"]

メソッド & プロパティ		名称	説明			例	省略時
初期化		<code>init</code>	function	チャートの初期化	ccchart.init('hoge', chartdataObject )		
			書式 ccchart.init(id, op[, callback])				
			戻値 ccchartオブジェクト				
		引数	<code>id</code> string,oj <code>op</code> object ※上記configオブジェクト参照 <code>callback</code> function	チャートを表示するcanvas要素または、そのid名 チャートオプションオブジェクト[{"config": [...], "data": [...] } ]の形式で指定 各チャートを描画後に発火するコールバック関数	"hoge" 必須 必須 function(){}alert('完了')) 省略可		
		<code>base (add v1.06+)</code>	function	Webページ全体に係るチャートのグローバルデフォルト値ccchart.gcfを設定する	ccchart.base(mySets, 'black', [config:useVal:'yes'])		
			書式 ccchart.base(op0[, op1, op2,...opN])	※優先順位: initのopやmemoのojなどによる個別設定 > このグローバルなデフォルト値 > もともとの各デフォルト値			
			戻値 ccchartオブジェクト				
		引数	<code>op0</code> string,oj <code>opN</code> string,oj	init同様のチャートオプションオブジェクトまたは'white'など文字列で表現されたテ- "black" 前のop引数の設定オブジェクトへ書き追記されるオプション設定	省略時は テーマの 'default' [config:useShadow: 'no']省略可		
メモ		<code>memo</code>	function	ccchart.memo(id, memoOj )	ccchart.init('hoge', chartdataObject ).memo( memoObject )		
			書式 ccchart.memo(memoObject)				
			戻値 ccchartオブジェクト				
		引数	<code>memoOj</code> object <code>val</code> string <code>left</code> number <code>top</code> number <code>font</code> font <code>align</code> align <code>lineTo</code> Array <code>lineToColor</code> string <code>lineToWidth</code> number <code>lineToXOffset</code> number <code>lineToYOffset</code> number	メモの内容 canvas左辺からの距離 canvas上辺からの距離 フォント 位置 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[ "ワーロン", "2011" ] lineTo または left & top のいずれか必須 メモメントのプロパティlineTo時のline色 メモメントのプロパティlineTo時のline幅 メモメントのプロパティlineTo時のメモ側のX位置 メモメントのプロパティlineTo時のメモ側のY位置	"メモです" 必須 150 200 "100 10px serif" "100 14px Arial" "center" "left" "black" "orange" 2 10 (0ならleft値と同じ) 10 (0ならtop値と同じ) 3		
WebSocket		<code>ws</code>	function	WebSocketを接続する	ccchart.init('hoge', chartdataObject ).ws(ws://ccchart.com:8016)		
			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	<code>url</code> string <code>op</code> object <code>autoReConnect</code> boolean <code>maxReConnect</code> number <code>protocol</code> string	WebSocketサーバーのURL 自動再接続フラグ 最大再接続回数 WebSocket のヘッダ Sec-WebSocket-Protocol に使われる文字列	'ws://ccchart.com:8016' 必須 false true 3 5 "mychart" "ws.ccchart.com"		
		<code>wsRecent</code>	object WebSocket	ccchart上で生成されたws接続の最新のインスタンス	ccchart.wsRecent		
			書式 ccchart.wsRecent		ccchart.wsRecent.プロパティ名		
			戻値 WebSocket接続インスタンス				
		プロパティ	<code>op.uid</code> string <code>op.id</code> string <code>op.url</code> string <code>op.useHb</code> boolean <code>op.hbStr</code> string <code>op.hbInterval</code> number <code>op.hbTimer</code> number <code>op.RTT</code> number <code>on</code> event method	"-ccchart-ws-"で始まるWebSocketインスタンス毎の一意なUUID V4 この接続に紐ついているDOM要素の id名 接続先urlすべて小文字で末尾の"/"を除去したスタイル ハートビートを使うかどうか true/ false 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 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 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(wbTimer) で手動停止も可能 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) event method (戻値 WebSocket APIの addEventListener ショートカット)	"-ccchart-ws-d7f04e4d-c813-4337-8ccf-11a46eaa23ae" 修正不可 "chart" 修正不可 "ws://ccchart.com:802" 修正不可 false true ping "Heartbeat" 6000 min 5000 で省略時は 60000 21 undefined エラー時は "-" ccchart.wsuids[uid].on('message', function(e))		
wsuids		<code>wsuids</code>	hash	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"]		
			書式 ccchart.wsuids['-ccchart-ws-' + uidv4]	※keyは"-ccchart-ws-"で始まりUUIDを接続した文字列	ccchart.wsuids[uid].hbStr		
			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.wses["hogeh8-ccchart-ws://ccchart.com:8014"]		
		<code>wses (v1.01で廃止)</code>	hash	ccchart上で生成されたws接続のリスト ws[ id + '-ccchart-' + url ] の書式	ccchart.wses["hogeh8-ccchart-ws://ccchart.com:8014"]		
				※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	<code>id</code> string <code>on</code> event その他 ws API	WebSocketインスタンスを作成したccchartのcanvas id名 WebSocket APIの addEventListener ショートカット 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)
		<code>wsCase</code>	object				
			<code>oneColAtATime</code> function	一度に1列ずつ[ "23:41:47", "58,41" ]といった配列で届く場合用	[ "23:41:47", "58,41" ]		
			<code>someColsAtATime</code> function	一度に複数列ずつ届く場合	[ ["製品A", "435,600"], ["製品B", "332,335"], ["製品C", "524,584"] ]		
wsCloseAll		<code>wsCloseAll</code>	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する	ccchart.wsCloseAll()		
		<code>wsCloseByUid</code>	function	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')		
			書式 ccchart.wsCloseByUid(uid)				
			戻値 ccchartオブジェクト				
		引数	<code>uid</code> string	uuid "-ccchart-ws-"で始まりUUIDを接続した文字列	-ccchart-ws-d7f04e4d- 必須		
		<code>wsCloseById</code>	function	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する ccchart.wsCloseById('hogeo0')			
			書式 ccchart.wsCloseById(id)				
			戻値 ccchartオブジェクト				
wsCloseByUrl		<code>wsCloseByUrl</code>	function	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する ccchart.wsCloseByUrl('ws://ccchart.com:80')			
			書式 ccchart.wsCloseByUrl(url)				
			戻値 ccchartオブジェクト				
		引数	<code>url</code> string	WebSocketの接続先URL	"ws://ccchart.com:80" 必須		
		<code>wsCloseByIdUrl</code>	function	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する ccchart.wsCloseById('hogeo0', 'ws://ccchart.com:80')			
			書式 ccchart.wsCloseById(id)				
			戻値 ccchartオブジェクト				
		引数	<code>id</code> string <code>url</code> string	DOMのid WebSocketの接続先URL	"hogeh2" 必須 "ws://ccchart.com:80" 必須		
wsDelTarget		<code>wsDelTarget</code>	function	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-c813-433****'])		
			書式 ccchart.wsDelTarget(target)				
			戻値 ccchartオブジェクト				
		引数	<code>target</code> object WebSocket	削除する WebSocketインスタンス	必須		
		<code>getWs</code>	function	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-c813-433****')		
			書式 ccchart.getWs(uid)				
			戻値 WebSocket				
		引数	<code>uid</code> string	取得したい WebSocketインスタンスの uid	必須		

### 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, "茶": "紅茶", "生産量": 432}, {"年度": 2008, "茶": "紅茶", "生産量": 308}, {"年度": 2009, "茶": "紅茶", "生産量": 356}, {"年度": 2010, "茶": "紅茶", "生産量": 308}, {"年度": 2011, "茶": "紅茶", "生産量": 432}, {"年度": 2012, "茶": "紅茶", "生産量": 308}, {"年度": 2013, "茶": "紅茶", "生産量": 432}, {"年度": 2007, "茶": "烏龍", "生産量": 164}, {"年度": 2008, "茶": "烏龍", "生産量": 104}, {"年度": 2009, "茶": "烏龍", "生産量": 156}, {"年度": 2010, "茶": "烏龍", "生産量": 104}, {"年度": 2011, "茶": "烏龍", "生産量": 164}, {"年度": 2012, "茶": "烏龍", "生産量": 104}, {"年度": 2013, "茶": "烏龍", "生産量": 164}, {"年度": 2007, "茶": "鐵觀音", "生産量": 308}, {"年度": 2008, "茶": "鐵觀音", "生産量": 164}, {"年度": 2009, "茶": "鐵觀音", "生産量": 308}, {"年度": 2010, "茶": "鐵觀音", "生産量": 164}, {"年度": 2011, "茶": "鐵觀音", "生産量": 308}, {"年度": 2012, "茶": "鐵觀音", "生産量": 164}, {"年度": 2013, "茶": "鐵觀音", "生産量": 308}, {"年度": 2007, "茶": "茉莉花茶", "生
```

