JSXGraph Reference Card

Include JSXGraph in HTML

Three parts are needed: Program files containing the software, an HTML element, and JavaScript code.

Program files:

Three files have to be included: `jsxgraph.css`, `jsxgraphcore.js` and either `prototype.js` or `jquery.js`.

```
<link rel="stylesheet" type="text/css" href="xxx/jsxgraph.css"/>
<script type="text/javascript" src="xxx/prototype.js"></script>
<script type="text/javascript" src="xxx/jsxgraphcore.js"></script>
```

or

```
<link rel="stylesheet" type="text/css" href="xxx/jsxgraph.css"/>
<script type="text/javascript" src="xxx/jquery.min.js"></script>
<script type="text/javascript" src="xxx/jsxgraphcore.js"></script>
```

`xxx` is the location of the files. This can be a local directory or `http://jsxgraph.uni-bayreuth.de/distrib/`

HTML element containing the construction:

```
<div id="box" class="jxgbox"
    style="width:600px; height:600px;"></div>
```

JavaScript code:

```
<script type="text/javascript">
   var brd = JXG.JSXGraph.initBoard('box',{axis:true});
<script>
```

Initialize the board

```
var brd = JXG.JSXGraph.initBoard('box',{attributes});
```

Basic commands

```
var el = brd.createElement('type',[parents],{attributes});
el.setProperty({attributes});
```

Available Elements

'angle', 'arc', 'arrow', 'arrowparallel', 'axis', 'bisector', 'chart', 'circumcircle', 'circumcirclemidpoint', 'curve', 'circle', 'glider', 'group', 'image', 'integral', 'line', 'midpoint', 'mirrorpoint', 'normal', 'parallel', 'parallelpoint', 'perpendicular', 'perpendicularpoint', 'polygon', 'point', 'reflection', 'sector', 'slider', 'spline', 'tangent', 'text', 'ticks', 'transform', 'turtle'

Point

```
brd.createElement('point',[parents],{attributes});
```

Parent elements:

Euclidean coordinates	`[3,-2]`
Homogeneous coords (z in first place)	`[1, 3,-2]`
Functions for x, y, (and z)	`[function(){return p1.X();}, function(){return p2.Y();}]`

Methods

x-coordinate	`p.X()`
y-coordinate	`p.Y()`
(Homogeneous) z-coordinate	`p.Z()`
Distance to other point	`p.Dist(q)`

Glider

```
brd.createElement('glider',[parents],{attributes});
```

Parent elements:

Initial coordinates and object to glide on	`[3, -2, c]`
Object to glide on (initially at origin)	`[c]`

Coordinates may also be defined by functions, see Point.

Line

```
brd.createElement('line',[parents],{attributes});
```

Parent elements:

2 points	`[p1,p2]`
3 coordinates (can also be functions)	`[c,a,b]`

In case of coordinates as parents, the line is the set of solutions of

$$a \cdot x + b \cdot y + c \cdot z = 0.$$

Curve

The supported curve types are:

Function graph

```
brd.createElement('graph',[parents],{attributes});
```

Parameter curve	`'parameter'`
Data plot	`'plot'`
Polar curve	`'polar'`

```
brd.createElement('curve',[parents],{attributes});
```

Parent elements:

– *Function graph:*

function [, start, end] `[function(x){return x*x;},-1,1]`

– *Parameter curve:*

x function, y function [, variable name, start, end]
`[function(x){return x;},function(x){return x*x;}]`

– *Data plot:*

array of x-coordinates,
array of y-coordinates, `[[1,2,3],[4,-2,3]]`
or array of x-coordinates, function
`[[1,2,3],function(x){return x*x;}]`

– *Polar curve:*

Defined by the equation $r = f(\phi)$.

Defining function, [offset, start, end] `[f,[1,2],0,Math.PI]`

Circle

```
brd.createElement('circle',[parents],{attributes});
```

Parent elements:

2 points	`[p1, p2]`
point,radius (constant or function)	`[p, r]`
point,circle	`[p, c]`
circle,point	`[c, p]`
point,line segment	`[p, l]`
line segment, point	`[l, p]`

Turtle

```
brd.createElement('turtle',[],{attributes});
var t = brd.createElement('turtle',[parents],{attributes});
```

The turtle has a position and a direction (in degrees). All angles have to be supplied in degrees.

Parent elements:

Optional start values for x, y, direction `[1,1,70]`

Methods:

Most of the methods have an abbreviated alternative version.

```
t.forward(len); t.fd(len);
t.back(len); or t.bk(len);
t.right(angle); or t.rt(angle);
t.left(angle); or t.lt(angle);
t.penUp(); or t.pu();
t.penDown(); or t.pd();
t.clearScreen(); or t.cs();
t.clean();
t.setPos(x,y);
t.home();
t.hideTurtle(); or t.ht();
t.showTurtle(); or t.st();
t.setPenSize(size);
t.setPenColor(col);
```
(col: colorString, e.g. 'red' or '#ff0000')

```
t.setProperty(key1:value1,key2:value2,...);
t.pushTurtle();
t.popTurtle();
t.lookTo([x,y]);
```
(Turtle looks to a coordinate pair. If t2 is another turtle object: t.lookTo(t2.pos))

```
t.lookTo(dir);
```
(Turtle looks into a given direction)

```
t.moveTo([x,y]);
```

Attributes of geometric elements

Links

Help pages are available at `http://jsxgraph.org`