var graph;

$('select').change(function() {

loadPreset($('select option:selected'));

});

loadPreset($('select option').get(0));

function loadPreset(opt) {

var P = $('#params').html('');

$.each($(opt).data('p').split(','), function(i, p) {

p = p.split(':');

var inp = $('<input />').val(p[1]).appendTo(

$('<label />').html(p[0]).appendTo(P)

);

inp.change(update);

});

function update() {

var args = [];

$('input').each(function(i,el) {

args.push(+$(el).val());

});

$('svg').remove();

renderForceGraph( eval('randomgraph.'+$(opt).html()+'('+args.join(',')+')'));

}

update();

}

function renderForceGraph(graph) {

var width = innerWidth-40,

height = innerHeight-40;

var force = d3.layout.force()

.charge(+$('#charge').val()\*-1)

.linkDistance(+$('#dist').val())

.size([width, height]);

var svg = d3.select("body").append("svg")

.attr("width", width)

.attr("height", height);

force

.nodes(graph.nodes)

.links(graph.edges)

.start();

var link = svg.selectAll(".link")

.data(graph.edges)

.enter().append("line")

.attr("class", "link");

var node = svg.selectAll(".node")

.data(graph.nodes)

.enter().append("circle")

.attr("class", "node")

.attr("r", 5)

.call(force.drag);

node.append("title")

.text(function(d) { return d.name; });

force.on("tick", function() {

link.attr("x1", function(d) { return d.source.x; })

.attr("y1", function(d) { return d.source.y; })

.attr("x2", function(d) { return d.target.x; })

.attr("y2", function(d) { return d.target.y; });

node.attr("cx", function(d) { return d.x; })

.attr("cy", function(d) { return d.y; });

});

$('input[type=range]').off('change').on('change', function() {

force

.charge(+$('#charge').val()\*-1)

.linkDistance(+$('#dist').val())

.start();

});

}