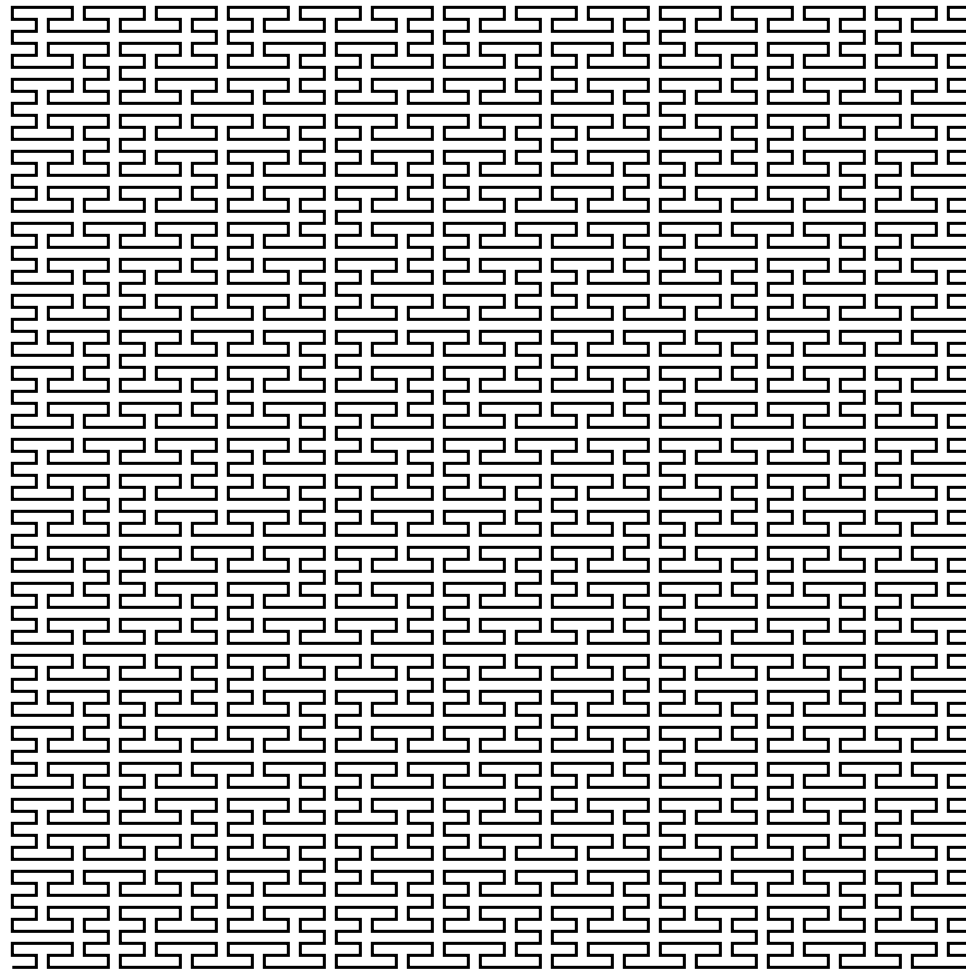


Peano curve (L-system)



A Peano [space-filling curve](#) drawn with an implementation of an [L-system](#) renderer. This is different from [the other curve also called Peano](#) by some.

[Open](#) [↗](#)

This curve, named Peano at least [on Wikipedia](#) and in [this article](#) about L-systems, is also named Hilbert II (in [this page](#)).

The drawing of the line is animated using Mike Bostock's [stroke dash interpolation](#).

index.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Peano curve</title>
    <link type="text/css" href="index.css" rel="stylesheet"/>
    <script src="http://d3js.org/d3.v3.min.js"></script>
  </head>
  <body></body>
  <script src="index.js"></script>
</html>
```

index.coffee

```
### compute a Lindenmayer system given an axiom, a number of steps and rules ###
fractalize = (config) ->
  input = config.axiom

  for i in [0...config.steps]
    output = ''

    for char in input
      if char of config.rules
        output += config.rules[char]
      else
```

```

        output += char

    input = output

    return output

### convert a Lindenmayer string into an SVG path string ###
svg_path = (config) ->
    angle = 0.0
    path = 'M0 0'

    for char in config.fractal
        if char == '+'
            angle += config.angle
        else if char == '-'
            angle -= config.angle
        else if char == 'F'
            path += "l#{config.side * Math.cos(angle)} #{config.side * Math.sin(angle)}"

    return path

peano = fractalize
    axiom: 'L'
    steps: 4
    rules:
        L: 'LFRFL-F-RFLFR+F+LFRFL'
        R: 'RFLFR+F+LFRFL-F-RFLFR'

d = svg_path
    fractal: peano
    side: 6
    angle: Math.PI/2

width = 960
height = 500

svg = d3.select('body').append('svg')
    .attr('width', width)

```

```

        .attr('height', height)

svg.append('path')
    .attr('class', 'curve shadow')
    .attr('d', d)
    .attr('transform', 'translate(240,490)')

### animate the path ###
### from Mike Bostock's stroke dash interpolation example http://bl.ocks.org/mbostock/5649592 ###
tweenDash = () ->
    l = this.getTotalLength()
    i = d3.interpolateString('0,' + l, l + ',' + l)
    return (t) -> i(t)

transition = (path) ->
    path.transition()
        .duration(20000)
        .attrTween('stroke-dasharray', tweenDash)

svg.append('path')
    .attr('class', 'curve')
    .attr('d', d)
    .attr('transform', 'translate(240,490)')
    .call(transition)

```

index.css

```

.curve {
    fill: none;
    stroke: black;
    stroke-width: 1.5px;
}

.shadow {

```

```
    opacity: 0.1;  
}
```

index.sass

```
.curve  
  fill: none  
  stroke: black  
  stroke-width: 1.5px  
  
.shadow  
  opacity: 0.1
```

LICENSE

This block appears to have [no license](#). Please contact [the author](#) to request a license.