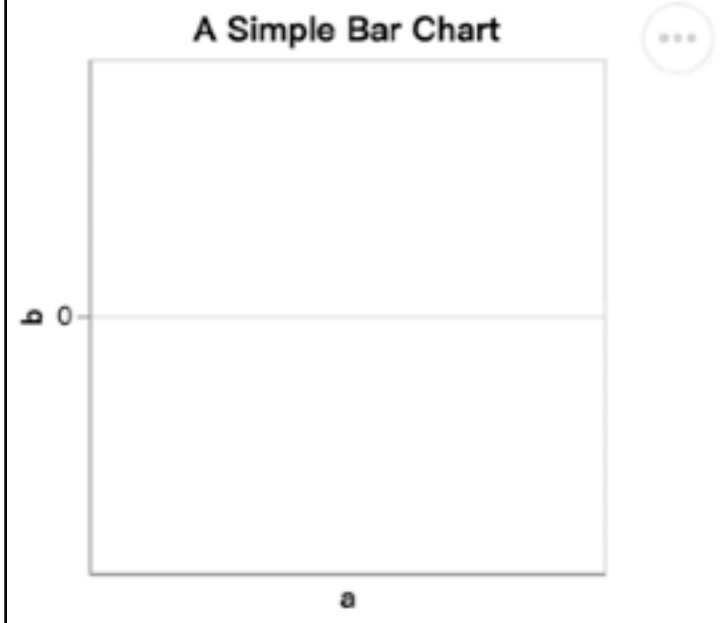
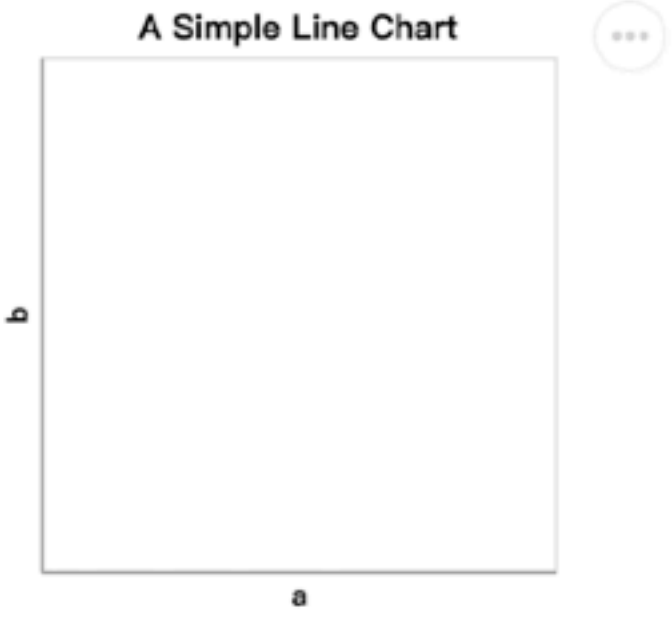
Widgets - Chart

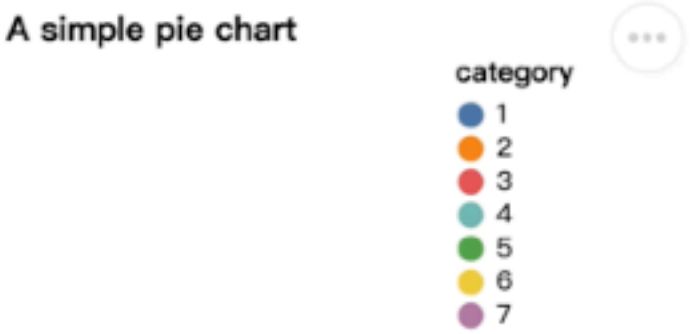
Chart

input: 0

0-100 Play Stop







```
<h1>Chart</h1>
input: <%= text binding: :input %><br/><br/>
0-100<%= slider min:0, max:100, value:10, binding: :input %>
<%= button text:"Play", action:"play()" %>
<%= button text:"Stop", action:"stop()" %>

<br/><br/>

<%= calc_var :chart_val1, "chart_data1(:input.to_i)" %>
<%= calc_var :chart_val2, "chart_data2(:input.to_i)" %>
<%= calc_var :chart_val3, "chart_data3(:input.to_i)" %>

<%= chart binding: :chart_val1 %>
<%= chart binding: :chart_val2 %>
<%= chart binding: :chart_val3 %>
```

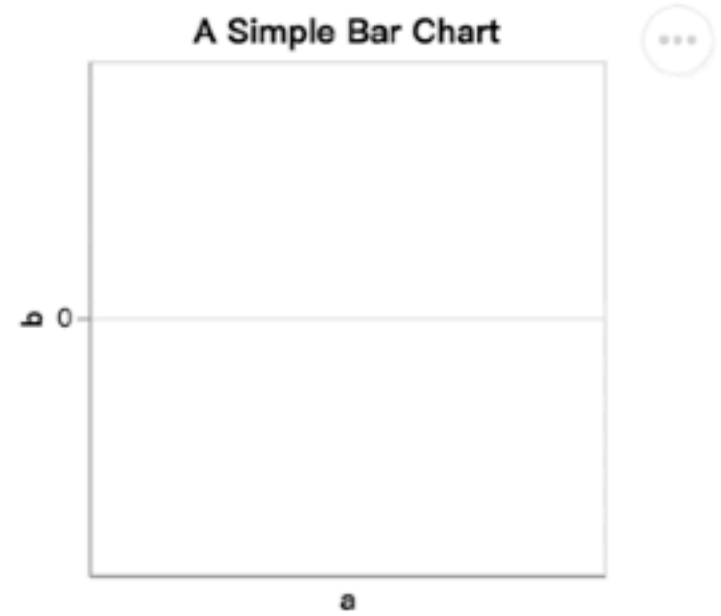
```ruby
def chart_data1(input)
    arr = (1..input).map {|x| x**2}

    spec = {
      "title": "A Simple Bar Chart",
      "width": 200,
      "height": 200,
      "data": {
        "values": []
      },
      "mark": "bar",
      "encoding": {
        "x": {"field": "a", "type": "ordinal"},
        "y": {"field": "b", "type": "quantitative"}

      }
    }
    spec["data"]["values"] = arr.map.with_index do |x,i|  {"a": i,"b": x} end

    return spec
end
```

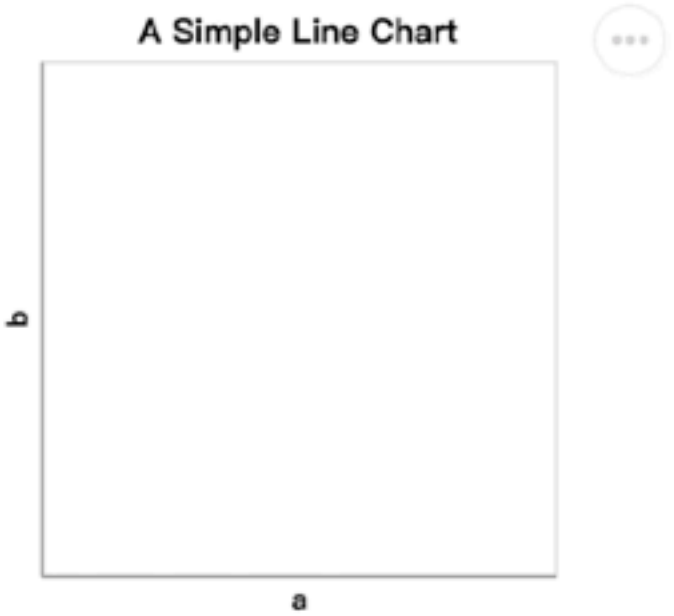
- Chart控件，使用Vega的语法来构建图表

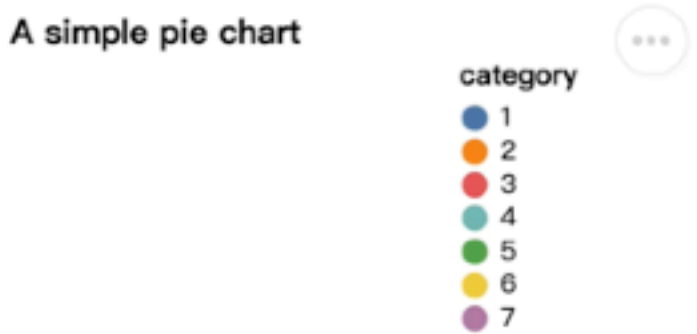
Chart

input: 0

0-100 Play Stop



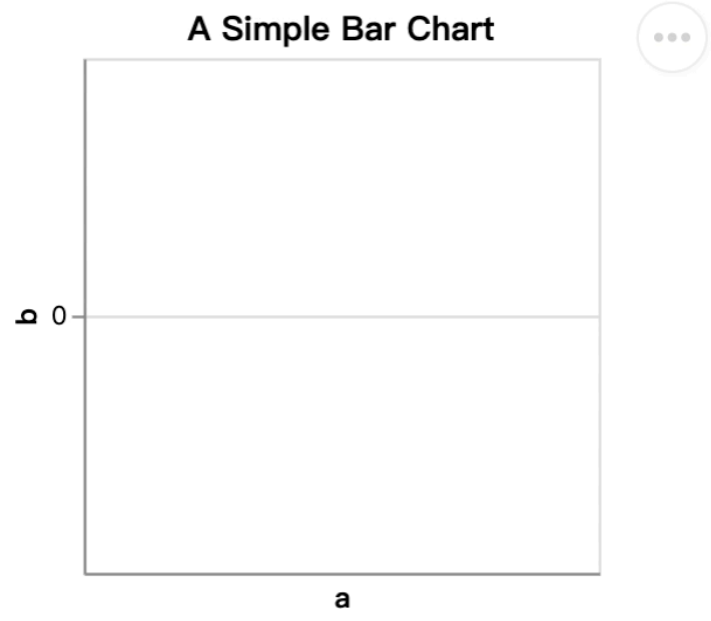


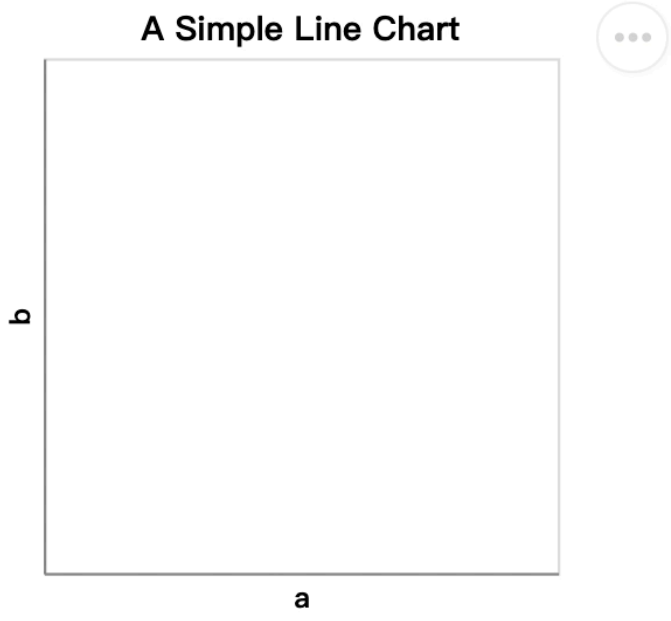


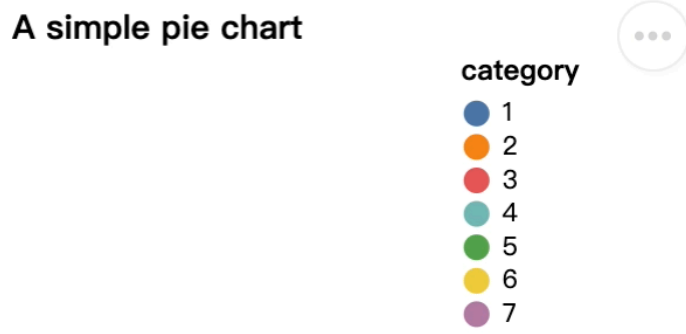
Chart

input: 0

0-100 Play Stop





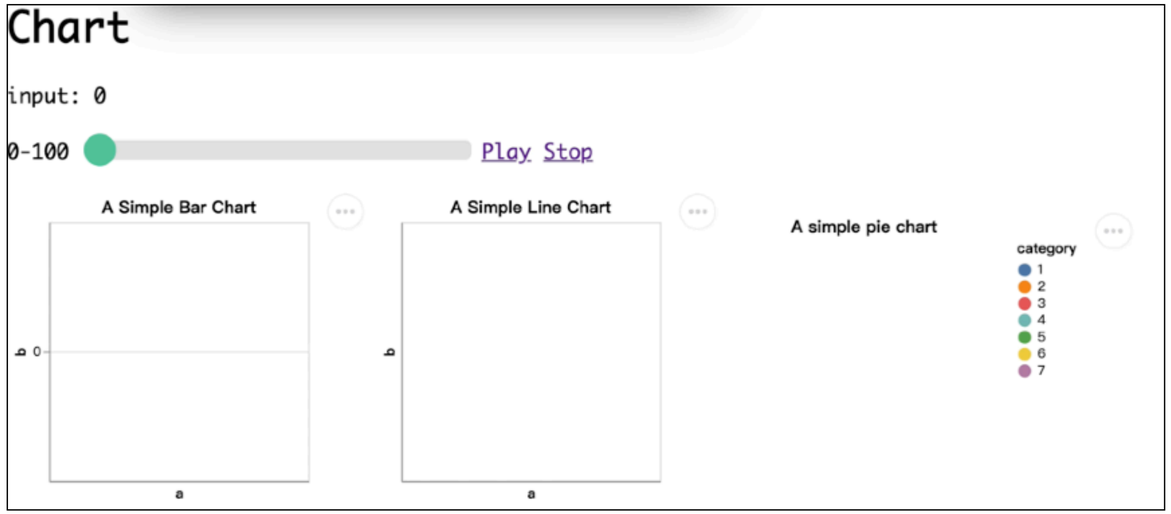


Widgets - Chart

- Chart控件，使用Vega的语法来构建图表

```
<h1>Chart</h1>
input: <%= text binding: :input %><br/><br/>
0-100<%= slider min:0, max:100, value:10, binding: :input %>
<%= button text:"Play", action:"play()" %>
<%= button text:"Stop", action:"stop()" %>

<br/><br/>

<%= calc_var :chart_val1, "chart_data1(:input.to_i)" %>
<%= calc_var :chart_val2, "chart_data2(:input.to_i)" %>
<%= calc_var :chart_val3, "chart_data3(:input.to_i)" %>

<%= chart binding: :chart_val1 %>
<%= chart binding: :chart_val2 %>
<%= chart binding: :chart_val3 %>
```

```
def chart_data1(input)
    arr = (1..input).map {|x| x**2}

    spec = {
      "title": "A Simple Bar Chart",
      "width": 200,
      "height": 200,
      "data": {
        "values": []
      },
      "mark": "bar",
      "encoding": {
        "x": {"field": "a", "type": "ordinal"},
        "y": {"field": "b", "type": "quantitative"}

      }
    }
    spec["data"]["values"] = arr.map.with_index do |x,i|  {"a": i,"b": x} end

    return spec
end
```



Widgets - Table

```
<h4>Table</h4>
<%= slider binding: :size, min:1, max:10, value:5 %>
<%= table binding: ":table=gen_table(:size)" %>

def gen_table(size)
    (1..size.to_i).map {|i| {id:i, name: "Adam", age: rand(i*10)} }
end
```