

# js\_pygal

July 26, 2022

## 1 pygal

[pygal](#) is one of the most mature and complete libraries using javascript.  
[documentation](#) [source](#) [installation](#) [tutorial](#) [gallery](#)

```
[1]: from jupyterlab import add_notebook_menu
      add_notebook_menu()
```

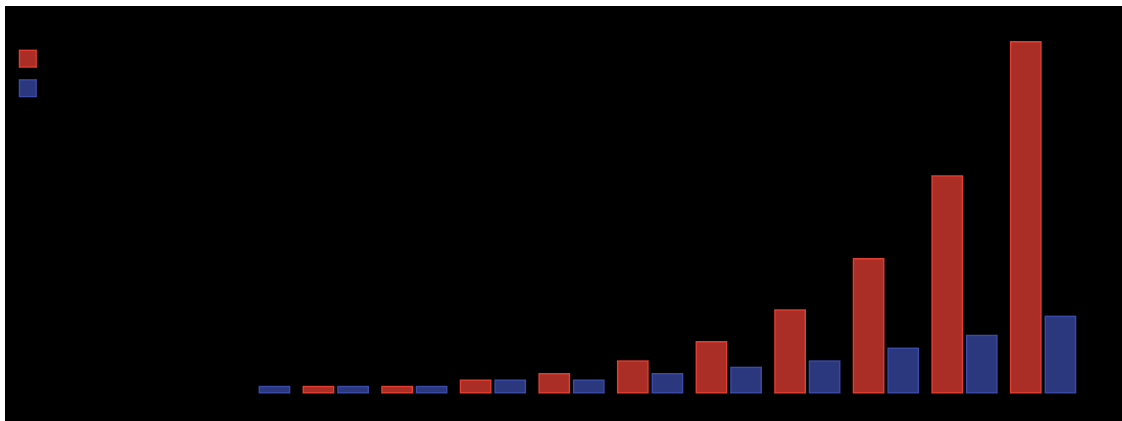
[1]: <IPython.core.display.HTML object>

### 1.1 example

```
[2]: import pygal
      bar_chart = pygal.Bar(height=300)
      bar_chart.add('Fibonacci', [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55])
      bar_chart.add('Padovan', [1, 1, 1, 2, 2, 3, 4, 5, 7, 9, 12])
      svg = bar_chart.render()
```

```
[3]: from IPython.display import SVG
      SVG(svg)
```

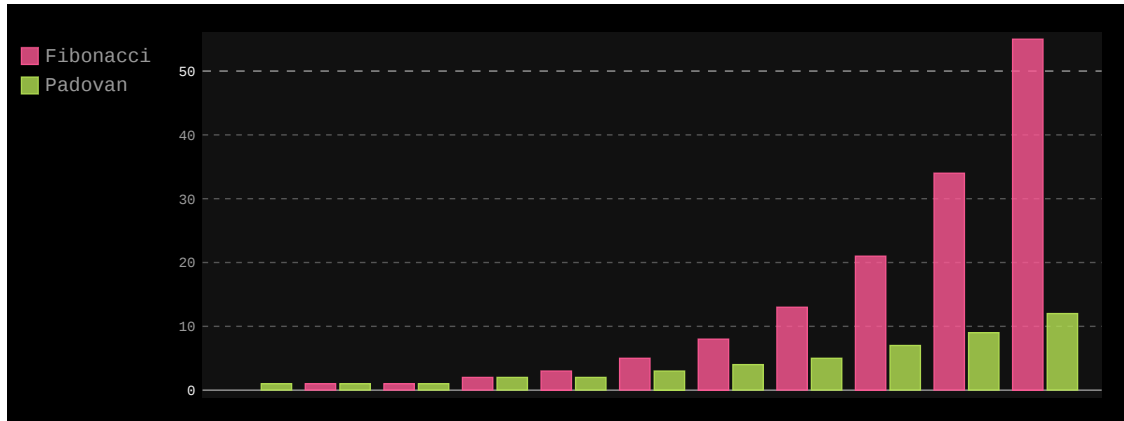
[3]:



### 1.2 Style of the graph

```
[4]: from pygal.style import DarkStyle
bar_chart = pygal.Bar(style=DarkStyle, height=300)
bar_chart.add('Fibonacci', [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55])
bar_chart.add('Padovan', [1, 1, 1, 2, 2, 3, 4, 5, 7, 9, 12])
svg = bar_chart.render()
SVG(svg)
```

[4]:



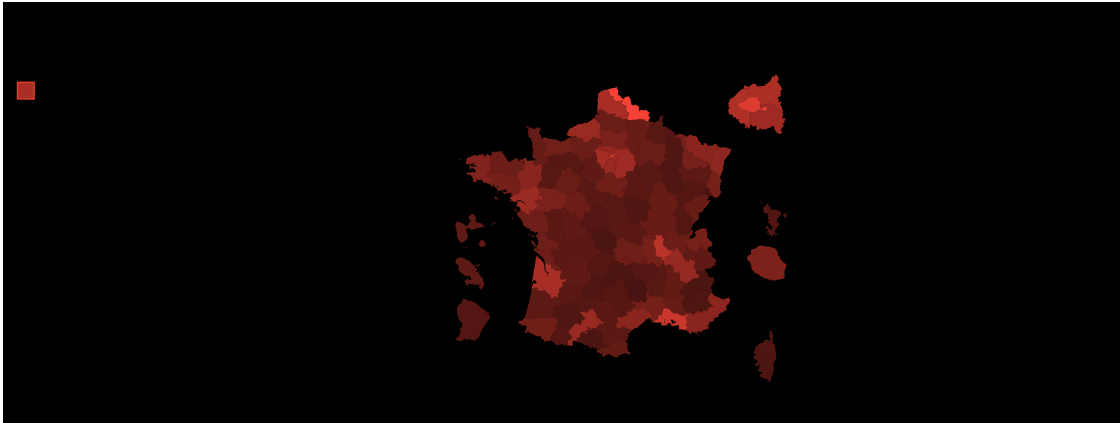
```
[5]: from IPython.display import SVG
import pygal
fr_chart = pygal.maps.fr.Departments(human_readable=True, height=300)
fr_chart.title = 'Population by department'
fr_chart.add('In 2011', {
    '01': 603827, '02': 541302, '03': 342729, '04': 160959, '05': 138605, '06': 1081244,
    '07': 317277, '08': 283110, '09': 152286, '10': 303997, '11': 359967, '12': ↵
    ↵275813,
    '13': 1975896, '14': 685262, '15': 147577, '16': 352705, '17': 625682, '18': ↵
    ↵311694,
    '19': 242454, '2A': 145846, '2B': 168640, '21': 525931, '22': 594375, '23': ↵
    ↵122560, '24': 415168,
    '25': 529103, '26': 487993, '27': 588111, '28': 430416, '29': 899870, '30': ↵
    ↵718357, '31': 1260226,
    '32': 188893, '33': 1463662, '34': 1062036, '35': 996439, '36': 230175, '37': ↵
    ↵593683, '38': 1215212,
    '39': 261294, '40': 387929, '41': 331280, '42': 749053, '43': 224907, '44': ↵
    ↵1296364, '45': 659587,
    '46': 174754, '47': 330866, '48': 77156, '49': 790343, '50': 499531, '51': ↵
    ↵566571, '52': 182375,
    '53': 307031, '54': 733124, '55': 193557, '56': 727083, '57': 1045146, '58': ↵
    ↵218341, '59': 2579208,
    '60': 805642, '61': 290891, '62': 1462807, '63': 635469, '64': 656608, '65': ↵
    ↵229228,
    '66': 452530, '67': 1099269, '68': 753056, '69': 1744236, '70': 239695, '71': ↵
    ↵555999,
    '72': 565718, '73': 418949, '74': 746994, '75': 2249975, '76': 1251282, '77': ↵
    ↵1338427,
```

```

    '78': 1413635, '79': 370939, '80': 571211, '81': 377675, '82': 244545, '83': 1012735,
    '84': 546630, '85': 641657, '86': 428447, '87': 376058, '88': 378830, '89': 342463,
    '90': 143348, '91': 1225191, '92': 1581628, '93': 1529928, '94': 1333702, '95': 1180365,
    '971': 404635, '972': 392291, '973': 237549, '974': 828581, '976': 212645
})
svg = fr_chart.render()
SVG(svg)

```

[5]:



### 1.3 Trick to make pygal reactive

```

[6]: html_pygal = """
<!DOCTYPEg>
<html>
  <head>
    <script type="text/javascript" src="http://kozea.github.com/pygal.js/latest/
    pygal-tooltips.min.js"></script>
    <!-- ... -->
  </head>
  <body>
    <figure>
      {0}
    </figure>
  </body>
</html>
"""

from IPython.display import HTML
HTML(html_pygal.format(svg.decode("utf-8")))

```

[6]: <IPython.core.display.HTML object>

[7]:

```


```