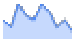
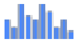
Peity

Star 1,684 Fork 114

Fork me on GitHub

Peity (sounds like deity) is a simple jQuery plugin that converts an element's content into a simple `<canvas>` mini pie line  or bar chart  and is compatible with any browser that supports `<canvas>`: Chrome, Firefox, IE9+, Opera, Safari.

Read about pie charts, line charts, bar charts, setting colours dynamically, updating charts, custom charts, data-* attributes and events, default settings.

Download version 1.2.0

Uncompressed 6.4Kb

jquery.peity.js

Minified 3.0Kb (+gzipped 1.4Kb)

jquery.peity.min.js

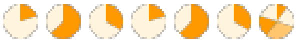
Source

github.com/benpickles/peity

Pie Charts

Simply call `peity("pie")` on a jQuery selection. There are two subtly different pie chart semantics, a "/" delimiter is assumed to mean "three out of five" and only the first two values will be drawn, otherwise all of the values are included in the chart and the total is the sum of all values.

You can also pass `colours`, `delimiter`, `diameter`, `height` and `width` options. Setting a diameter is a shortcut for setting the width and height to the same value.



HTML

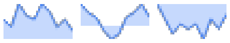
```
<span class="pie">1/5</span>
<span class="pie">226/360</span>
<span class="pie">0.52/1.561</span>
<span class="pie">1,4</span>
<span class="pie">226,134</span>
<span class="pie">0.52,1.041</span>
<span class="pie">1,2,3,2,2</span>
```

JavaScript

```
$("span.pie").peity("pie")
```

Line Charts

Line charts work on a comma-separated list of digits. Line charts can take the following options: `colour`, `strokeColour`, `strokeWidth`, `delimiter`, `max`, `min`, `width` and `height`.



HTML

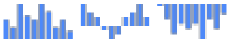
```
<span class="line">5,3,9,6,5,9,7,3,5,2</span>
<span class="line">5,3,2,-1,-3,-2,2,3,5,2</span>
<span class="line">0,-3,-6,-4,-5,-4,-7,-3,-5,-2</span>
```

JavaScript

```
$(".line").peity("line")
```

Bar Charts

Bar charts work in the same way as line charts and take the following options: `colours`, `delimiter`, `max`, `min`, `spacing`, `width` and `height`.



HTML

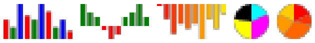
```
<span class="bar">5,3,9,6,5,9,7,3,5,2</span>
<span class="bar">5,3,2,-1,-3,-2,2,3,5,2</span>
<span class="bar">0,-3,-6,-4,-5,-4,-7,-3,-5,-2</span>
```

JavaScript

```
$(".bar").peity("bar")
```

Setting Colours Dynamically

Bar and pie chart colours can be defined dynamically based on the values of the chart. When passing an array its values are cycled, when passing a function it is called once for each value allowing you to define each bar's colour. The function is called with a similar signature to `Array#forEach` with the bar's value, its index, and the full array of values.



HTML

```
<span class="bar-colours-1">5,3,9,6,5,9,7,3,5,2</span>
<span class="bar-colours-2">5,3,2,-1,-3,-2,2,3,5,2</span>
<span class="bar-colours-3">0,-3,-6,-4,-5,-4,-7,-3,-5,-2</span>
<span class="pie-colours-1">4,7,6,5</span>
```

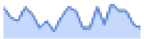
```
<span class="pie-colours-2">5,3,9,6,5</span>
```

JavaScript

```
$(".bar-colours-1").peity("bar", {
  colours: ["red", "green", "blue"]
})

$(".bar-colours-2").peity("bar", {
  colours: function(value) {
    return value > 0 ? "green" : "red"
  }
})

$(".bar-colours-3").peity("bar", {
  colours: function(_, i, all) {
    var g = parseInt((i / all.length) * 255)
    return "rgb(255, " + g + ", 0)"
  }
})

$(".pie-colours-1").peity("pie", {
  colours: ["cyan", "magenta", "yellow", "black"]
})

$(".pie-colours-2").peity("pie", {
  colours: function(_, i, all) {
    var g = parseInt((i / all.length) * 255)
    return "rgb(255, " + g + ", 0)"
  }
})
```

Updating Charts

Charts can be updated by changing the the jQuery selection's text content and calling `change()` on it. The chart will be redrawn with the same options that were originally passed to it.



HTML

```
<span class="updating-chart">5,3,9,6,5,9,7,3,5,2,5,3,9,6,5,9,7,3,5,2</span>
```

JavaScript

```
var updatingChart = $(".updating-chart").peity("line", { width: 64 })

setInterval(function() {
  var random = Math.round(Math.random() * 10)
  var values = updatingChart.text().split(",")
  values.shift()
  values.push(random)

  updatingChart
    .text(values.join(","))
    .change()
}, 1000)
```

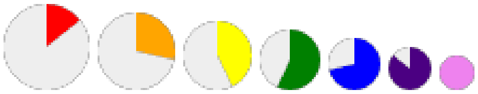
Custom Charts

You can easily add your own custom chart by registering it with Peity with name, defaults and the draw function which receives an options object.

```
$.fn.peity.add("custom", {
    colour: "#FFCC00"
  }, function(opts) {
    ...
  }
)
```

data-* attributes

If you use data attributes, Peity will apply their values to the defaults with the same names automatically. Options passed to the `peity()` function take precedence over data-* attributes.



HTML

```
<span data-colours='["red", "#eeeeee"]'    data-diameter="40">1/7</span>
<span data-colours='["orange", "#eeeeee"]' data-diameter="36">2/7</span>
<span data-colours='["yellow", "#eeeeee"]' data-diameter="32">3/7</span>
<span data-colours='["green", "#eeeeee"]'  data-diameter="28">4/7</span>
<span data-colours='["blue", "#eeeeee"]'   data-diameter="24">5/7</span>
<span data-colours='["indigo", "#eeeeee"]' data-diameter="20">6/7</span>
<span data-colours='["violet", "#eeeeee"]' data-diameter="16">7/7</span>
```

JavaScript

```
$(".pie span").peity("pie")
```

Events

Peity adds a "change" event trigger to your graph elements, so if you update their data your can regenerate one or more charts by triggering `change()` on them.

- 4 ▾

- 1 ▼
- 3 ▼

Chart updated: 3/5

HTML

```
<ul>
  <li>
    <span class="graph"></span>
    <select>
      <option value="0">0</option>
      <option value="1">1</option>
      <option value="2">2</option>
      <option value="3">3</option>
      <option value="4" selected>4</option>
      <option value="5">5</option>
    </select>
  </li>
  <li>
    <span class="graph"></span>
    <select>
      <option value="0">0</option>
      <option value="1" selected>1</option>
      <option value="2">2</option>
      <option value="3">3</option>
      <option value="4">4</option>
      <option value="5">5</option>
    </select>
  </li>
  <li>
    <span class="graph"></span>
    <select>
      <option value="0">0</option>
      <option value="1">1</option>
      <option value="2">2</option>
      <option value="3" selected>3</option>
      <option value="4">4</option>
      <option value="5">5</option>
    </select>
  </li>
</ul>

<p id="notice">Nothing's happened yet.</p>
```

JavaScript

```
$("select").change(function() {
  $(this)
    .siblings("span.graph")
    .text($(this).val() + "/" + 5).change()
}).change()

var chartUpdate = function(event, value, max) {
  $("#notice").text(
    "Chart updated: " + value + "/" + max
  )
}

$("span.graph")
  .peity("pie")
  .bind("change.peity", chartUpdate)
```

Default Settings

Defaults can be overridden globally like so:

```
$.fn.peity.defaults.pie = {
  colours: ["#ff9900", "#fff4dd", "#ffd592"],
  delimiter: null,
  diameter: 16,
  height: null,
  width: null
}

$.fn.peity.defaults.line = {
  colour: "#c6d9fd",
  strokeColour: "#4d89f9",
  strokeWidth: 1,
  delimiter: ",",
  height: 16,
  max: null,
  min: 0,
  width: 32
}

$.fn.peity.defaults.bar = {
  colours: ["#4d89f9"],
  delimiter: ",",
  height: 16,
  max: null,
  min: 0,
  spacing: devicePixelRatio || 1,
  width: 32
}
```