

به نام خدا



دانشگاه صنعتی شریف

دانشکده مهندسی کامپیوتر

برنامه سازی وب - امید جعفرنژاد

نیمسال اول ۹۹-۰۰

تحقیق میان ترم

Chart.js

اعضای تیم:

• امیرحسین رستمی (۹۶۱۰۱۶۳۵)

• امید شرفی (۹۶۱۰۱۸۳۸)

فهرست مطالب

۱	نصب
۲	نحوه‌ی استفاده در کد
۳	اطلاعات کلی
۴	۱.۳ نگاهی دقیق‌تر به canvas
۴	۲.۳ Charts Responsive
۵	۳.۳ ارتباط با نمودار
۷	۴.۳ تنظیمات، رنگ‌ها و فونت
۱۱	۴ تنظیمات نمودار
۱۱	۱.۴ اینیمیشن
۱۲	۲.۴ حاشیه نمودار و عنوان
۱۳	۳.۴ تنظیم عنوان دیتاست ها
۱۴	۴.۴ تنظیم ظاهر المان های نمودار
۱۴	۱.۴.۴ نقاط

١٥	خطوط	٢.٤.٤
١٥	مستطيل ها	٣.٤.٤
١٥	خم ها	٤.٤.٤
١٦	tooltip	٥.٤
١٩	انواع نمودارها	٥
١٩	line	١.٥
١٩	bar	٢.٥
٢٠	radar	٣.٥
٢٠	area polar and pie	٤.٥
٢١	scatter and bubble	٥.٥
٢٣	محورها و تنظيمات ديگر	٦
٢٥	ايجاد تغييرات در نمودار	٧

۱ نصب

برای نصب و استفاده از Chart.js روش های مختلفی وجود دارد. میتوان از npm یا bower برای نصب آن استفاده کنید یا از روش های مانند استفاده از cdn و دانلود پروژه استفاده کنید. فایل های cdn مربوطه اینجا میباشد.

npm

npm v2.9.4 downloads 5.2M/month

npm install chart.js --save

Bower

bower v2.9.4

bower install chart.js --save

شکل ۱: دستورات npm و bower

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.9.4/Chart.min.js" defer></script>
```

شکل ۲: اضافه کردن جاوااسکریپت Chart.js به کد html

۲ نحوه استفاده در کد

برای شروع کار و مشاهده نحوه استفاده از Chart.js درون کدمان، با یک مثال ساده شروع میکنیم.

```

کاسهٔت جو نظر با یک کانواس  

کانواس به مکان می‌بلند نایل  

اعمال کنید نایل این سود  

الاتباد مرسنا - داده‌ای نیازمند است  

نگاه مای مربوط به مرد  

داده‌های کد  

مد نکار دارم هر بار غدار  

سیکل کرد.  

backgroundColor [  

    'rgba(255, 99, 132, 0.2)',  

    'rgba(54, 162, 235, 0.2)',  

    'rgba(255, 206, 86, 0.2)',  

    'rgba(75, 192, 192, 0.2)',  

    'rgba(153, 102, 255, 0.2)',  

    'rgba(255, 159, 64, 0.2)'  

],  

bordercolor [  

    'rgba(255, 99, 132, 1)',  

    'rgba(54, 162, 235, 1)',  

    'rgba(255, 206, 86, 1)',  

    'rgba(75, 192, 192, 1)',  

    'rgba(153, 102, 255, 1)',  

    'rgba(255, 159, 64, 1)'  

],  

borderWidth: 1  

},  

options: {  

    scales: {  

        yAxes: [{  

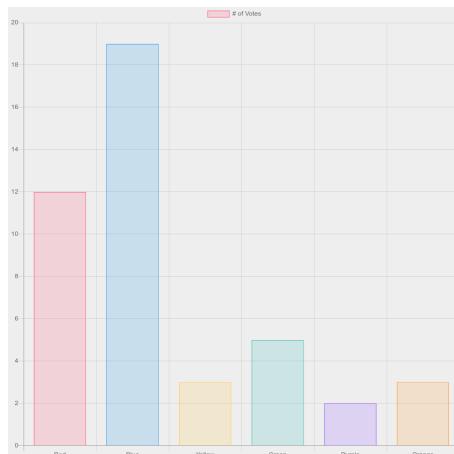
            ticks: {  

                beginAtZero: true
            }
        }]
    }
});

```

شکل ۳: مثال استفاده از Chart.js در کد

همانطور که در این مثال دیده میشود، از تگ canvas برای اضافه کردن کامپوننت مورد نظر به بخش صفحه استفاده شده و نمودار را در فایل جواالاسکریپ برنامه ایجاد میکنیم. بسته به نوع نموداری که مد نظر داریم متغیر type را تنظیم کرده و اطلاعات مربوط به داده ها مانند اندازه هر کدام، رنگ هر کدام، نام هر کدام و هر دسته را مشخص کرده و تنظیمات کلی مربوط به نمودار را در بخش options تنظیم میکنیم. خروجی مثال بالا به شکل زیر است:



شکل ۴: خروجی مثال بالا

۳ اطلاعات کلی

در این بخش به بررسی اطلاعات و تنظیمات کلی نمودارها از جمله ابعاد، فونت، رنگ ها و ... میپردازیم.

۱.۳ نگاهی دقیق تر به canvas

همانطور که در بخش قبل دیدیم برای آن که بتوانیم با استفاده از chart.js نمودار ترسیم کنیم باید از طریق جاوااسکریپت به المان canvas دسترسی داشته باشیم. المانی هست که به صورت پیکس به پیکس بر روی صفحه لود میشوند و دچار تغییر ابعاد نمیشوند، در نتیجه این کامپوننت برای ترسیم نمودارها مفید خواهد بود. همچنین باید توجه شود که المان canvas که بر روی صفحه لود میشود قابل دسترسی نبوده چرا که داخل DOM قرار ندارد. البته این موضوع در مثال های ما تاثیرگذار نخواهد بود ولیکن برای اطلاعات بیشتر در این مورد میتوانید به [این لینک](#) مراجعه کنید.

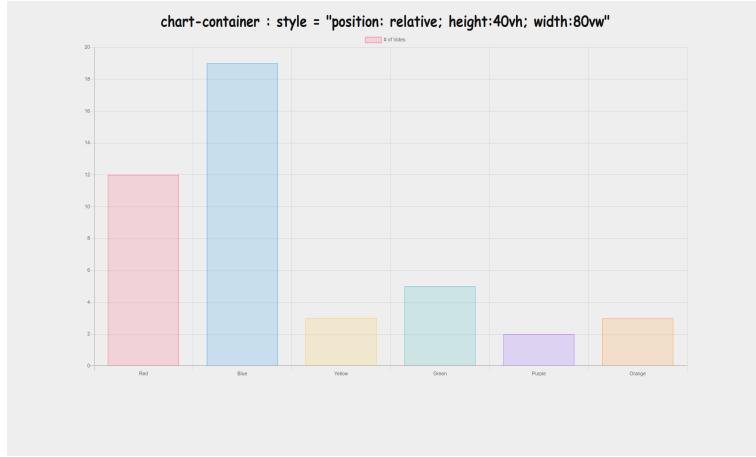
Charts Responsive ۲.۳

بحث ابعاد و اندازه نمودارها طبیعتاً از مباحث قابل اهمیت کاربر خواهد بود. نکته ای که اینجا وجود دارد، این است که المان canvas قابلیت تنظیم ابعاد بر اساس درصدی از ابعاد صفحه را ندارد. در نتیجه تعدادی پرامتر خود Chart.js در اختیارما قرار میدهد که در جدول زیر ارائه شده است.

Name	Type	Default	Description
responsive	boolean	true	Resizes the chart canvas when its container does (important note...).
responsiveAnimationDuration	number	0	Duration in milliseconds it takes to animate to new size after a resize event.
maintainAspectRatio	boolean	true	Maintain the original canvas aspect ratio (width / height) when resizing.
aspectRatio	number	2	Canvas aspect ratio (i.e. width / height , a value of 1 representing a square canvas). Note that this option is ignored if the height is explicitly defined either as attribute or via the style.
onResize	function	null	Called when a resize occurs. Gets passed two arguments: the chart instance and the new size.

شکل ۵: روش های پویا سازی متناسب با ابعاد صفحه

مهم ترین و پرکاربردترین عضو این جدول که در پیاده سازی ها نیز استفاده میشود تنظیم responsive میباشد. در نتیجه صرفاً کافی است یک چارت کانتینر با ابعاد متناسب با صفحه ایجاد کرده و المانمان را داخل این کانتینر قرار دهیم. یک نمونه خروجی در زیر آورده شده است :



شکل ۶: نمودار پویا

همچنین اگر در این شرایط بیایم و ارتفاع و عرض خود المان canvas المان را تعیین کنیم، از یک طرف به خاطر قرار گرفتن در یک کانتینر پویا نمودار نیز پویا میشود اما از طرف دیگر نسبتی که نمودار از ابعاد خود حفظ میکند متناسب با نسبت اعداد خود canvas خواهد بود. برای آشنایی بیشتر میتوانید قطعه کد زیر را اجرا کنید و اعداد آن را در حالت های مختلف تغییر دهید.

```
<div class="chart-container" style="position: relative; height:40vh; width:80vw">
|   <canvas id="myChart" style="height:100px; width:100px"></canvas>
</div>
```

شکل ۷: نمونه کد تغییر ابعاد نمودار

۳.۳ ارتباط با نمودار

برای ارتباط با نمودار میتوانیم پاسخ به فعالیت های مختلف کاربر را در بخش option نمودار تعریف کنیم.

Name	Type	Default	Description
events	string[]	['mousemove', 'mouseout', 'click', 'touchstart', 'touchmove']	The <code>events</code> option defines the browser events that the chart should listen to for tooltips and hovering. more...
onHover	function	null	Called when any of the events fire. Called in the context of the chart and passed the event and an array of active elements (bars, points, etc).
onClick	function	null	Called if the event is of type <code>'mouseup'</code> or <code>'click'</code> . Called in the context of the chart and passed the event and an array of active elements.

شکل ۸: انواع event ها در نمودار Chart.js

در نتیجه میتوانیم برای هر کدام از event های کاربر تابعی تعریف کنیم. این توانایی زمانی کاربرد دارد که به طور مثال بخواهیم ویژگی هایی را به نمودارهای Chart.js اضافه کنیم. مثلاً فرض کنید بخواهیم هنگامی که موس کاربر روی فضای نمودار ما قرار دارد، متناسب با مختصات موس کاربر کاربر تغییری در مورد یا بقیه اجزای صفحه ایجاد کنیم. مثلاً به مثال زیر توجه کنید :

```

<h1 id="myTitle">
  chart-container : style = "position: relative; height:40vh; width:80vw"
</h1>

<div class="chart-container" style="position: relative; height:40vh; width:80vw">
  <canvas id="myChart"></canvas>
</div>

<script>
  var ctx = document.getElementById('myChart');
  var myChart = new Chart(ctx, {
    type: 'line',
    data: {
      labels: ['Red', 'Blue', 'Yellow', 'Green', 'Purple', 'Orange'],
      datasets: [{
        label: '# of Votes',
        data: [12, 19, 3, 5, 2, 3],
      }]
    },
    options: {
      currentDevicePixelRatio : 1,
      onClick: (e) => {
        // Change title color
        document.getElementById("myTitle").style.color = "blue";

        const canvasPosition = Chart.helpers.getRelativePosition(e);
        // Substitute the appropriate scale IDs
        const dataX = chart.scales.x.getValueForPixel(canvasPosition.x);
        const dataY = chart.scales.y.getValueForPixel(canvasPosition.y);
      }
    });
</script>

```

شکل ۹: مثال event در نمودار

در این مثال با کلیک کردن کاربر بر روی نمودار رنگ سرتیتر عوض شده و در عین حال مختصات نقطه‌ی کلیک کاربر را داشته و میتوانیم از آن به طور مثال برای ترسیم د. خط عمودی و افقی که از آن نقطه بگذرد یا هر کاربرد دیگری استفاده کنیم. همچنین توجه کنید که ممکن است در کد بالا نیاز شود که devicePixelRatio نمودار را نیز تنظیم کنید. همچنین چه برای تنظیم tooltips نمودار یا حرکت موس بر روی نمودار میتوانیم انواع حالات انتخاب نقاط را داشته باشیم که بعضی از آن‌ها را به اختصار توضیح میدهیم. در عمل این ویژگی معمولاً کاربرد ندارد و به صورت پیش‌فرض بر روی مد nearest تنظیم شده است. برای اطلاعات بیشتر به این لینک [مراجعه نمایید](#).

- point : تمام آیتم‌هایی که دقیقاً نقطه‌ی ما را قطع میکنند به ما برمی‌گرداند.
- nearest : تمام آیتم یا آیتم‌هایی را که مرکزشان بر روی نمودار کمترین فاصله با نقطه‌ی ما را دارند برمی‌گرداند.
- dataset : برای حالتی که چند دیتاست مختلف داریم، تمام آیتم‌هایی با دیتاست یکسان را به ما برمی‌گرداند.
- x : تمام آیتم‌هایی با مختصات افقی یکسان را به ما برمی‌گرداند.
- y : تمام آیتم‌هایی با مختصات عمودی یکسان را به ما برمی‌گرداند.

دو مورد انتهایی نیز میتوانند مفید باشند. به طور مثال فرض کنید میخواهید بر روی چارت یک سری خط یا کرسر کمکی ترسیم کنید که وقتی کاریر موس خود را روی نمودار حرکت داد این راهنما بر روی داده‌های نمودار ترسیم شود تا کاربر را برای ردیابی نقطه‌ای داده‌های نمودار کمک کند. (مثلاً در شرایطی که تعداد زیادی داده در یک نمودار line داریم) در این شرایط مثلاً میتوانیم از مد x استفاده کرده و در نتیجه ترسیم خطوط کمکی را بر روی نمودار به گونه‌ای انجام دهیم که تقاطع دو خط داده‌ی ما را بر روی نمودار مشخص کنند.

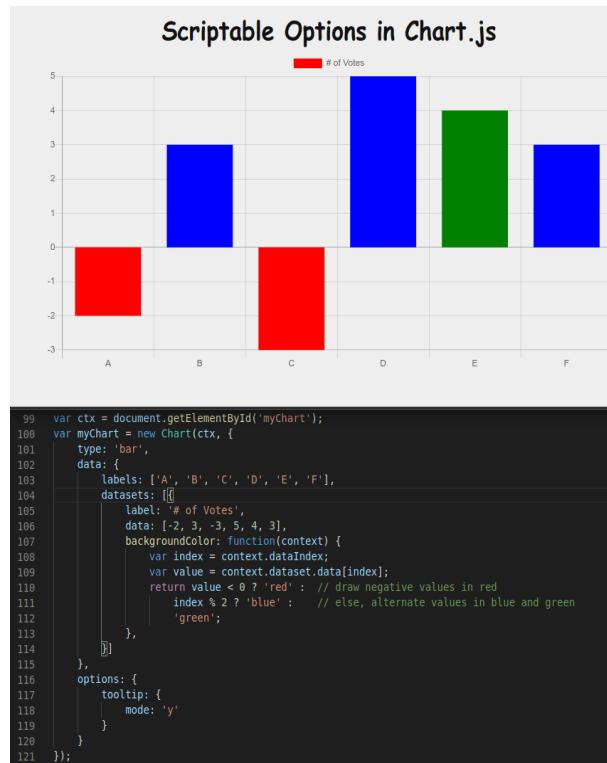
۴.۳ تنظیمات، رنگ‌ها و فونت

در بخش تنظیمات علاوه بر توابع و امکاناتی که تا اینجا توضیح داده شد یا در ادامه توضیح داده خواهد شد تا بتوانیم ظاهر نمودار یا منطق عملکرده نمودار را تنظیم کنیم، میتوانیم خودمان توابعی را متناسب با وضعیت داده‌های نمودار تا تعریف کنیم تا بتوانیم نموداری پویا بر اساس داده‌ها داشته باشیم. برای نوشتن این نوع توابع باید از context استفاده کنیم. این شی امکانات مختلفی از چارت و دیتاست ما در اختیار ما قرار میدهد.

```
color: function(context) {
  var index = context.dataIndex;
  var value = context.dataset.data[index];
  return value < 0 ? 'red' : // draw negative values in red
    index % 2 ? 'blue' : // else, alternate values in blue and green
    'green';
}
```

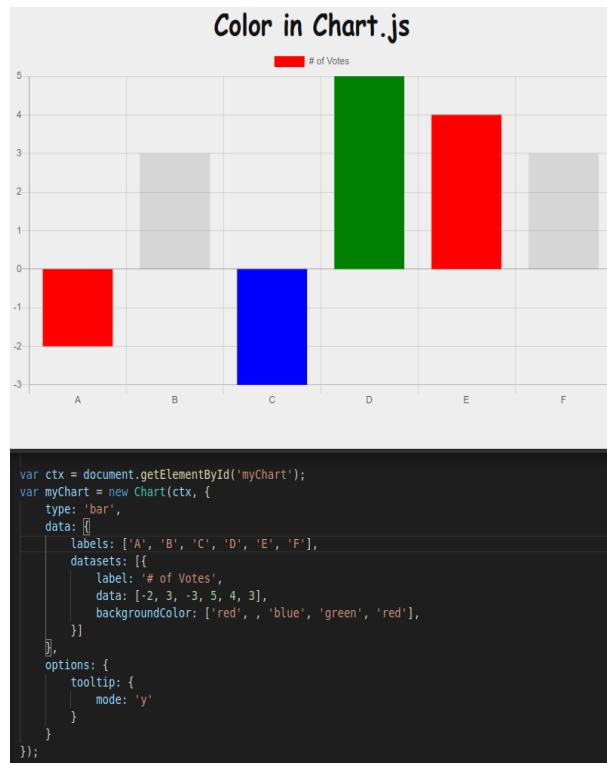
شکل ۱۰: مثال استفاده از شی context در تابع

به طور مثال تابع نوشته شده متناسب با مقدار هز داده وضعیت رنگ آن را تعیین میکند. اگر مقدار منفی باشد، رنگ قرمز، در غیر این صورت برای مقادیر زوج رنگ آبی و در غیر این صورت سبز میشود. در خروجی به صورت زیر میشود.



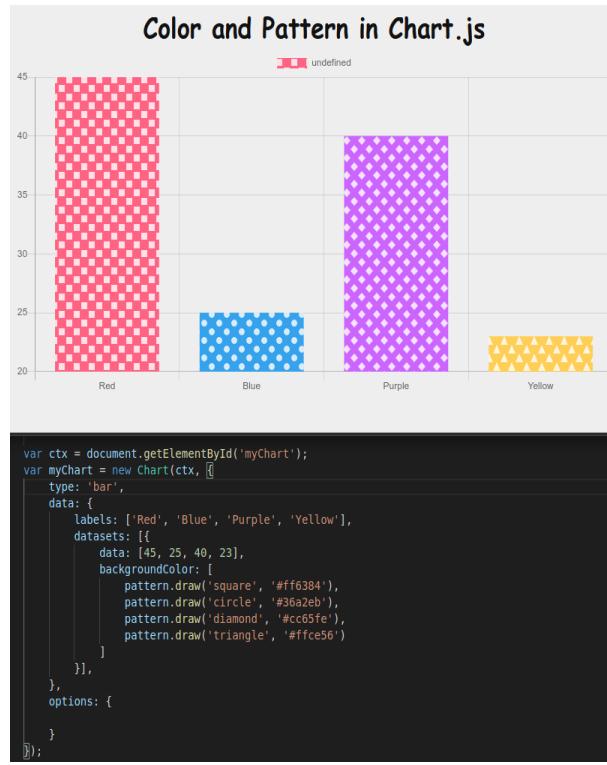
شکل ۱۱: تعریف تابع برای رنگ در نمودار bar

همچنین میتوانیم رنگ های نمودار را یا یک آرایه از رنگ ها هم طول با طول داده های نیز تعیین کنیم که در نتیجه وضعیت رنگ هر داده مشخص خواهد بود. اگر به مثال زیر دقت کنید نکته‌ی جالبی در مورد منطق این بخش وجود دارد. شما یک آرایه به عنوان رنگ داده ها به نمودار میدهید. اگر طول این آرایه کمتر از طول داده ها باشد، از ابتدای داده به همان اندازه ای که در این آرایه رنگ ها مشخص شده اند پیش رفته و رنگ ها را اختصاص میدهد و به بقیه داده های باقی مانده رنگ شیری پیش فرض را اختصاص خواهد داد. در عین حال اگر طول آرایه‌ی رنگ بیشتر باشد نیز از اول آرایه رنگ ها را یکی یکی اختصاص داده و بقیه رنگ های داخل آرایه را در نظر نخواهد گرفت. همچنین هر عنصری از آرایه‌ی رنگ ها را که خالی قرار دهد، معادلاً رنگ شیری پیش فرض برای آن داده قرار میگیرد. این موارد در مثال زیر آورده شده است.



شکل ۱۲: تعریف آرایه رنگ ها

رنگ پیش فرضی که در بالا مشاهده میکنید توسط متغیر `Chart.defaults.global.defaultColor` قابل تنظیم خواهد بود که این مقدار در ابتدا پیش فرض `0/0, 0, 0, 0/1` در معیار `rgba` میباشد. همچنین به جز رنگ های ساده که در مثال بالا دیدیم، میتوانیم انواع تصاویر، گرادیان های رنگی یا پترن ها را ایجاد و اضافه کنیم. به طور مثال با استفاده از این [ریپازیتوری](#) در کد زیر رنگ ها را با استفاده از طرح داخل چارت ایجاد کردیم که به طور مثال برای افراد کوررنگ مفید خواهد بود.



شکل ۱۳: اضافه کردن پترن به نمودارها

همچنین در مورد تنظیمات فونت نیز متناسب با جدول زیر در بخش option نمودار میتوانیم رنگ، فامیلی، سایز و استایل فونت رو به صورت کامل و در کل نمودار تغییر داده و تنظیم کنیم.

Name	Type	Default	Description
defaultFontColor	Color	'#666'	Default font color for all text.
defaultFontFamily	string	"'Helvetica Neue', 'Helvetica', 'Arial', sans-serif"	Default font family for all text.
defaultFontSize	number	12	Default font size (in px) for text. Does not apply to radial/linear scale point labels.
defaultFontStyle	string	'normal'	Default font style. Does not apply to tooltip title or footer. Does not apply to chart title.

شکل ۱۴: تنظیمات فونت

۴ تنظیمات نمودار

در این بخش به طور دقیق تر به امکانات قابل تنظیم نمودارهای Chart.js مانند شکل نمودار، انیمیشن ها، عنوان و موارد دیگر خواهیم پرداخت.

۱.۴ انیمیشن

برای تنظیمات انیمیشن نمودار میتوانیم با استفاده از تنظیم duration سرعت انیمیشن و همچنین با تخصیص یکی از توابع [این لیست](#) برای ویرگی easing نوع انیمیشن نمودار را انتخاب کنیم. در عین حال دو کالبک onAnimationComplete و onAnimationProgress نیز میتوانند بسیار مفید باشند.

Name	Type	Default	Description
duration	number	1000	The number of milliseconds an animation takes.
easing	string	'easeOutQuart'	Easing function to use. more...
onProgress	function	null	Callback called on each step of an animation. more...
onComplete	function	null	Callback called at the end of an animation. more...

شکل ۱۵: تنظیمات انیمیشن

به طور مثال اگر بخواهیم یک انیمیشن به طور همزمان و به صورت سینک با انیمیشن اصلی نمایش داده شود در این صورت کالبک onAnimationProgress میتواند مفید باشد و در حین آن با استفاده currentStep که عدد شماره فریم انیمیشن نمودار و numSteps که تعداد کل فریم های انیمیشن هست بتوانیم انیمیشن جانبی را با انیمیشن اصلی نمودار سینک کنیم. نمونه‌ی استفاده از این کالبک در کد زیر استفاده شده ایت که نسبت پیشرفت حین اجرای انیمیشن را در یک تگ h بالای نمودار نشان میدهد. با اجرای این مثال خواهید دید که این کالبک نه تنها در ابتدا نمودار بلکه در هر جایی تولتیپ روی نمودار نیز که به صورت انیمیشن طور هست اجرا میگردد.

```
options: {
  animation: {
    onProgress: function(animation) {
      document.getElementById('myTitle').innerHTML = animation.animationObject.currentStep / animation.animationObject.numSteps;
    }
  }
}
```

شکل ۱۶: کد استفاده از کالبک انیمیشن

۲.۴ حاشیه نمودار و عنوان

برای نمودار میتوانیم padding تنظیم کنیم که این padding را میتوانیم با یک عدد برای تمام جهت های نمودار یا برای هرکدام از سمت ها به صورت مجزا مانند کد زیر تنظیم کنیم.

```
let chart = new Chart(ctx, {
  type: 'line',
  data: data,
  options: {
    layout: {
      padding: {
        left: 50,
        right: 0,
        top: 0,
        bottom: 0
      }
    }
  }
});
```

شکل ۱۷: تنظیم حاشیه های نمودار

همچنین برای تنظیم عنوان نمودار نیز باید در زیر بخش title زیر بخش options را اضافه و هرکدام از متغیرهایی که از جدول زیر نیاز هست تنظیم کنیم.

Name	Type	Default	Description
display	boolean	false	Is the title shown?
position	string	'top'	Position of title. more...
fontSize	number	12	Font size.
fontFamily	string	"'Helvetica Neue', 'Helvetica', 'Arial', sans-serif"	Font family for the title text.
fontColor	Color	#6666	Font color.
fontWeight	string	'bold'	Font style.
padding	number	10	Number of pixels to add above and below the title text.
lineHeight	number string	1.2	Height of an individual line of text. See MDN .
text	string string[]	''	Title text to display. If specified as an array, text is rendered on multiple lines.

شکل ۱۸: عنوان نمودار

تجربه‌ی شخصی بنده از استفاده از امکان عنوان نمودار Chart.js به گونه‌ای بود که برای تنظیم مثلاً یک متن ثابت خوب است، اما برای دسترسی و ایجاد تغییرات در این عنوان با استفاده از کد جاوااسکریپت با چالش‌ها و محدودیت‌هایی روبرو است که شاید قرار دادن یک المان عنوان بالای المان نمودار گزینه‌ی مناسب‌تری برای همچین شرایطی باشد. مزیت گزینه‌ی عنوان قرار گرفتن عنوان در مکان خیلی بهینه و زیبایی مناسب با نمودار است که این موضوع با قرار دادن یک المان در کنار نمودار مخصوصاً برای عنوان‌های سمت چپ یا راست نمودار به سادگی قابل پیاده‌سازی نیست.

۳.۴ تنظیم عنوان دیتاست ها

هر کدام از دیتاست های نمودار ما یک عنوان دارند که برای نمایش آن تنظیمات مختلفی وجود دارد. در زیر جدولی از این تنظیمات آورده شده است که اکثراً مربوط به تنظیمات style این بخش میباشد.

Name	Type	Default	Description
display	boolean	false	Is the title shown?
position	string	'top'	Position of title. more...
fontSize	number	12	Font size.
fontFamily	string	"'Helvetica Neue', 'Helvetica', 'Arial', sans-serif"	Font family for the title text.
fontColor	Color	'#666'	Font color.
fontStyle	string	'bold'	Font style.
padding	number	10	Number of pixels to add above and below the title text.
lineHeight	number string	1.2	Height of an individual line of text. See MDN .
text	string string[]	''	Title text to display. If specified as an array, text is rendered on multiple lines.

شکل ۱۹: تنظیمات عناوین دیتاست های نمودار

از مواردی که ممکن است ناآشنا باشد یکی گزینه‌ی reverse هست که امکان چاپ به صورت برعکس دیتاست را در اختیار قرار میدهد. در عین حال توابع این عناوین نیز بسیار پرکاربرد هستند. در این توابع ما به موارد مانند text که متن لیل هست، بولین hidden که اگر true باشد یعنی الان دیتاست مربوط به این لیل پنهان هست. همچنین تنظیمات مختلف ظاهر مانند تنظیمات خط، رنگ و غیره در این بخش وجود دارد که برای اطلاعات بیشتر میتوانید به [این لینک](#) مراجعه کنید.
در اینجا ابتدا نگاهی به تابع onClick پیش فرض آیتم های عنوان میاندازیم.

```
function(e, legendItem) {
    var index = legendItem.datasetIndex;
    var ci = this.chart;
    var meta = ci.getDatasetMeta(index);

    // See controller.isDatasetVisible comment
    meta.hidden = meta.hidden === null ? !ci.data.datasets[index].hidden : null;

    // We hid a dataset ... rerender the chart
    ci.update();
}
```

شکل ۲۰: تابع onClick پیش فرض آیتم های عناوین دیتاست ها

در نتیجه با استفاده از اینترفیس legendItem به ایندکس دیتاست مربوط به آیتمی که کلیک شده است دسترسی پیدا میکنیم. سپس با استفاده از این ایندکس به دیتا متناظر با آن در نمودار دسترسی پیدا کرده و نهایتاً بادستور update به نمودار میگیم که دوباره خودش رو آپدیت کرده و در نتیجه تغییری که ما در وضعیت نمایش دیتاست مربوطه دادیم رو بر روی نمودار مشاهده کنیم. و حالا دقیقاً مشابه با همین کد میتوانیم انواع توابع onClick را برای آیتم های بخش عنوانین پیاده سازی کنیم. در انتها توجه کنید که این تابع را باید مانند شکل زیر بر روی نمودار اعمال کنید :

```
options: {
  legend: {
    onClick: legendClickHandler
  }
}
```

شکل ۲۱: اعمال تابع onClick

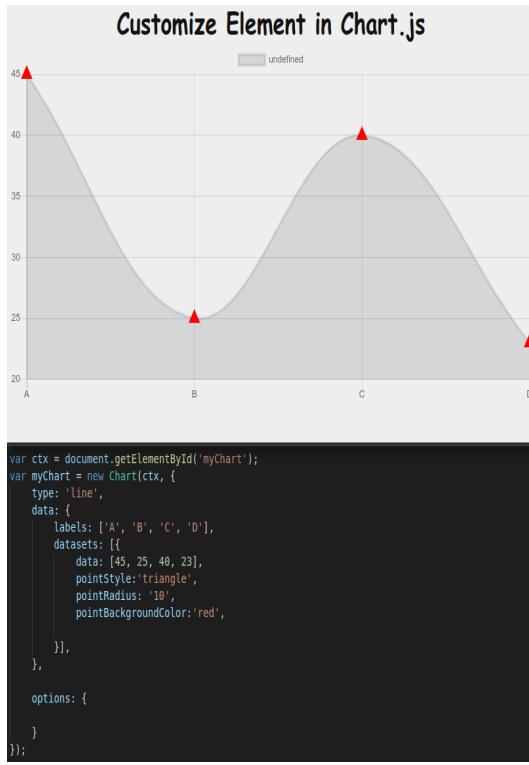
۴.۴ تنظیم ظاهر المان های نمودار

المان های نمودارهای مختلف Chart.js در چهار دسته ی زیر دسته بندی میشوند.

۱.۴.۴ نقاط

در نمودارهای line و bubble کامپونت نقطه وجود دارد که میتوانیم تنظیمات مختلفی مانند شعاع دایره، رنگ پس زمینه، رنگ دور و موارد گرافیکی دیگری را برای آن ها تنظیم کنیم که لیست کامل این موارد آینجا آورده شده است.

همچنین ما برای نقاط صرفاً به شکل دایره محدود نیستیم و اشکالی مانند ضربدر، مربع، ستاره و ... نیز توسط Chart.js برای استفاده فراهم شده اند. یک مثال از این مورد آورده شده است.



شکل ۲۲: مثال تنظیم ظاهر المان نقطه نمودار

۲.۴.۴ خطوط

خطوط صرفا در چارت های دیگری که احیاناً بعداً خارج از کتابخانه رسمی Chart.js دولوپ شده اند مانند نمودارهای ترکیبی استفاده می‌شوند و تنظیماتی مختلفی مانند ضخامت، رنگ و ... دارند که لیست تنظیمات [اینجا](#) آورده شده است.

۳.۴.۴ مستطیل ها

تنظیمات مستطیل در حقیقت شامل تنظیم مستطیل های استفاده شده در نمودار bar و نمودارهای ترکیبی می‌باشد که لیست تنظیماتشان [اینجا](#) آورده شده است.

۴.۴.۴ خم ها

ونها یا خم ها نیز تنظیمات رنگ و زاویه و چند مورد دیگر مربوط به خم ها که در نمودارهای pie و doughnut و نمودارهای قطبی استفاده می‌شوند را دارند که لیست تنظیمات [اینجا](#) آورده شده است.

tooltip ۵.۴

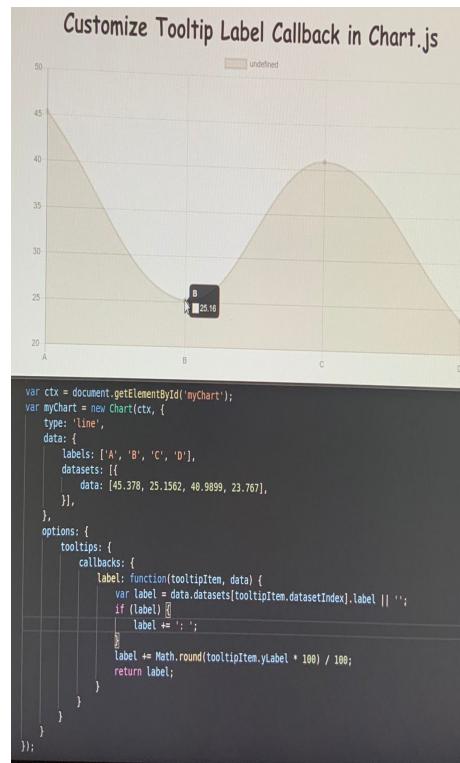
انواع تنظیمات مربوط به tooltip ها از جمله نمایش یا عدم نمایش، انواع مدهای tooltip به این معنا که نزدیکترین داده را نمایش دهم یا انواع دیگر مدهای ارتباطی با نمودار که قبل از آن اشاره شد. میزان خمس کارت دور tooltip در ویژگی cornerRadius قابل تنظیم هست و کلی از تنظیمات ظاهری دیگر که در [اینجا](#) موجود هست.

همچنین مشابه توابعی که قبل از معرفی کردیم، این ویژگی نمودارها نیز امکان پیاده سازی انواع توابع را در کالبک های زیر را دارد.

Name	Arguments	Description
beforeTitle	TooltipItem[], object	Returns the text to render before the title.
title	TooltipItem[], object	Returns text to render as the title of the tooltip.
afterTitle	TooltipItem[], object	Returns text to render after the title.
beforeBody	TooltipItem[], object	Returns text to render before the body section.
beforeLabel	TooltipItem, object	Returns text to render before an individual label. This will be called for each item in the tooltip.
label	TooltipItem, object	Returns text to render for an individual item in the tooltip. more...
labelColor	TooltipItem, Chart	Returns the colors to render for the tooltip item. more...
labelTextColor	TooltipItem, Chart	Returns the colors for the text of the label for the tooltip item.
afterLabel	TooltipItem, object	Returns text to render after an individual label.
afterBody	TooltipItem[], object	Returns text to render after the body section.
beforeFooter	TooltipItem[], object	Returns text to render before the footer section.
footer	TooltipItem[], object	Returns text to render as the footer of the tooltip.
afterFooter	TooltipItem[], object	Text to render after the footer section.

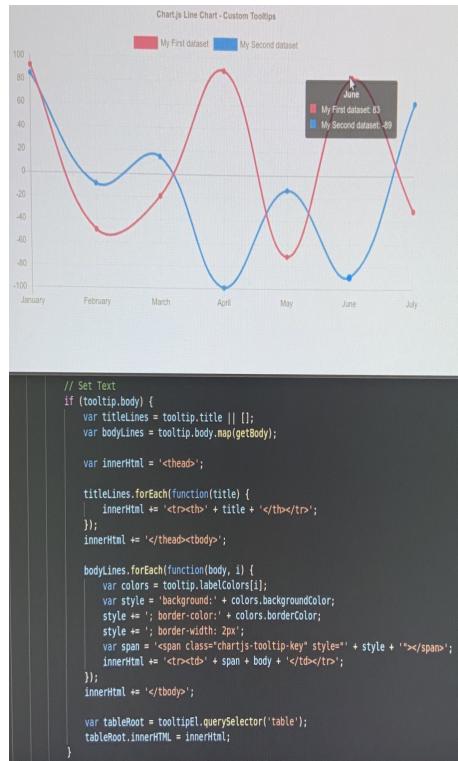
شکل ۲۳: مجموعه کالبک های قابل تنظیم برای پیاده سازی توابع

توجه شود که در توابع این کالبک ها به آبجکت های tooltipItem دسترسی داریم که میتوانیم اطلاعات مورد نیازمان را از آن گرفته و رشته یا آرایه ی از رشته ها را در خروجی تابع return کنیم. به طور مثال در مثال زیر ابتدا از tooltipItem اندیس داده را به دست آورده و لیبل آن را میگیریم و در ادامه مقدار داده ای که tooltip مربوط به آن قرار است نمایش داده شود را با رند کردن مقدار داده تا دو رقم و اضافه کردن آن به رشته ی label رشته حاصل را خروجی میدهیم.



شکل ۲۴: شخصی سازی توابع tooltip

همچنین امکان ایجاد tooltip های پیچیده و شخصی سازی شده نیز موجود است. به این صورت که ابتدا ویژگی enabled را false کنید تا کاملاً tooltip پیش فرض قطع شده و سپس ویژگی custom را تعریف کنید تا tooltip موردنظر خودتان را ایجاد کنید. نمونه زیر خروجی کد به همراه قسمت اصلی کدی است که tooltip را به گونه ای شخصی سازی کرده است که وقتی روی هر داده میرویم داده‌ی هرکدام از نمودارهای دیگری را هم که مقدار ماه یکسانی را دارند برای ما نمایش میدهند. در حقیقت ما در این تابع یک کد html برای نمایش به عنوان tooltip سرو میکنیم. برای پیاده سازی لاجیک این بخش در بدنه‌ی کد html یک حلقه بر روی تمام داده‌های با title یکسان با آبجکت tooltip پاس داده شده زده دقیقاً مدد معادل را شامل یک span با رنگ آن دسته و بدنه‌ی شامل اطلاعات هرکدام از داده‌های با لیبل یکسان را ساخته و نهایتاً نمایش داده میشود. برای مطالعه‌ی کد کامل این مثال به این [لينک](#) مراجعه نمایید.



شکل ۲۵: شخصی سازی توابع tooltip با استفاده ایجاد کد html

۵ انواع نمودارها

در این بخش اشکال مختلف نمودارها را با هم میبینیم. ساختار تمام نمودارها و نحوه داده دادن به هر کدام دقیقاً مانند شکل ۳ این داک میباشد. همچنین هر کدام متناسب با کامپونت هایی در ترسیم آنها استفاده شده است، تنظیمات و امکانات خودشان را دارند. برای بررسی بیشتر تنظیمات هر کدام از نمودارها به [این](#) رجوع کنید.

line 1.0

این نمودارها داده های هر دیتاست را به صورت خطی به هم متصل میکند. انواع تنظیمات از جمله امکان تغییر رنگ و ابعاد نقطه ها به صورت عادی یا هنگام عبور موس و غیره وجود دارد. خطوط نیز قابلیت تغییر استیل دارند که در بخش خطوط در بالا بررسی شد. نمونه این چارت را مفصلان در مثال های بالا دیدید. صرفاً توجه کنید که داده ها را علاوه بر فرمتی که بالا ذکر شد میتوانید برای هر داده هم مقدار x و مقدار y را مانند نمودارهای scatter برای نمودارهای خطی هم تعریف کنید.

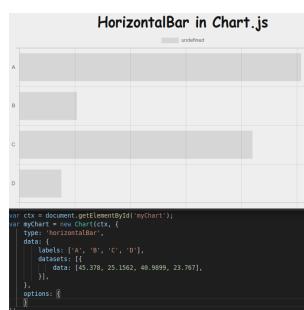
Point[]

```
data: [{  
    x: 10,  
    y: 20  
}, {  
    x: 15,  
    y: 10  
}]
```

شکل ۲۶: روش دیگر تعریف داده ها

bar ۲.۵

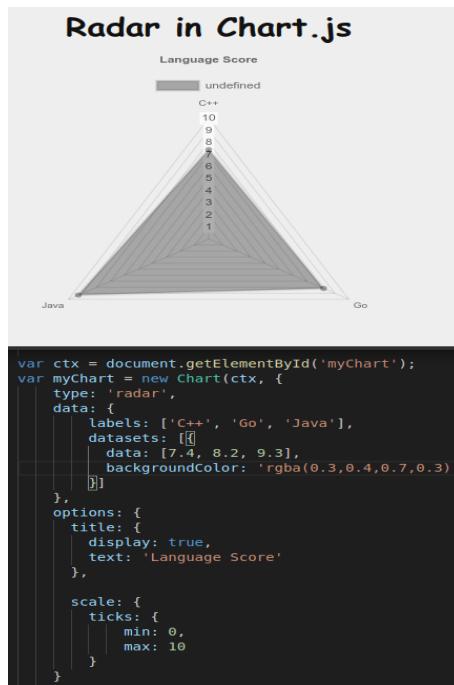
نمودار بار را هم مفصل‌آمدیم و تنظیمات استایل مستطیل های زیر آن هم گفته شد. صرفاً بدانیم که نمودار میتواند مانند شکل زیر افقی نیز باشد.



شکا ۲۷: نمودار بار افقی

radar ۴.۵

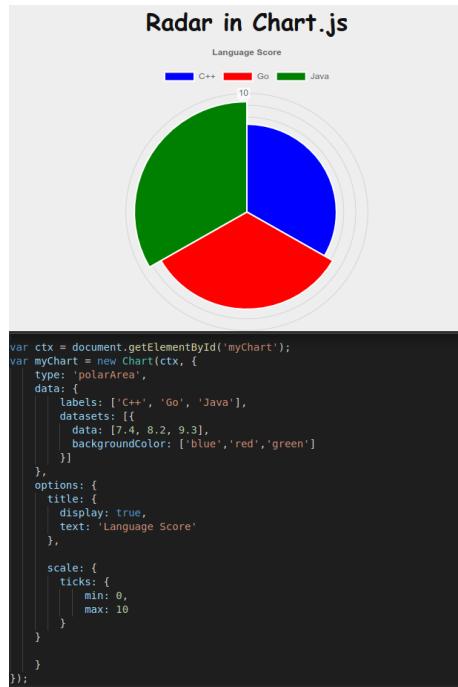
نمودار رادار به شکل زیر در می‌آید. این نمودار معمولاً برای زمانی که تعداد اعضای دیتاست ما کم هست و میخواهیم میزان تمرکز یا توانایی یا علاقه امان در هر کدام از موضوعات نمایش دهیم کاربرد دارد. همچنین مانند نمودارهای دیگر این نمودار هم میتواند چندین دیتاست داشته باشد که دیتاست‌ها روی هم ترسیم خواهند شد.



شکل ۲۸: مثال نمودار رادار

area polar and pie ۴.۶

نمودارهای area polar یا pie هر دو بسیار شبیه به هم بوده و معمولاً برای نمایش سهم موضعیات مختلف در یک عامل استفاده می‌شوند. تهر دو این نمودارها به صورت زاویه‌ای حول دایره هستند، صرفاً نمودار pie برای نشان دادن سهم هر داده متناسب با مقدار آن زاویه تعیین می‌کند ولیکن زاویه‌های داده‌های نمودار area polar ثابت و تفاوت داده‌ها با میزان جلو رفتگی آن‌ها مشابه نمودار رادار مشخص می‌شود. در نتیجه اگر تنظیمات area polar را بررسی کنید ترکیبی از تنظیمات موجود در pie و radar را خواهید دید. یک نمونه از area برای آشنایی بیشتر آورده شده است.

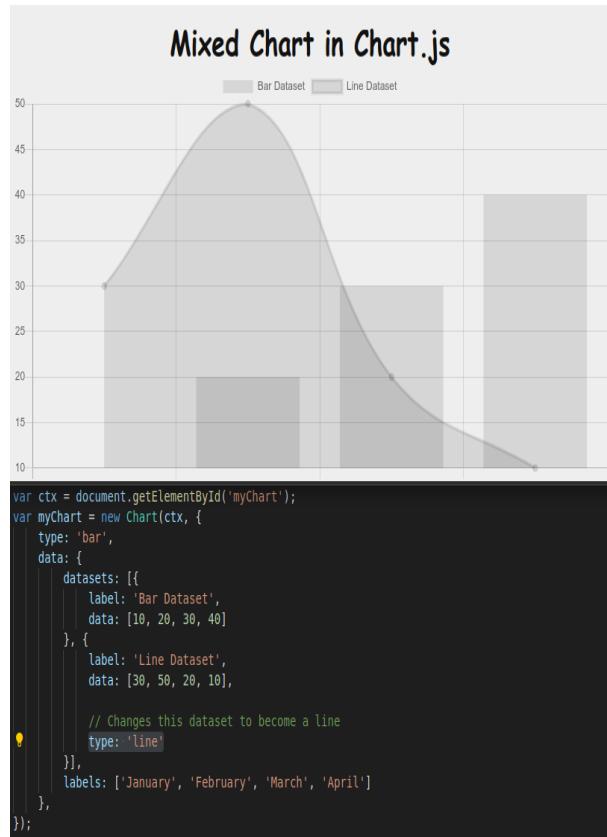


شکل ۲۹: مثال نمودار area polar

scatter and bubble ۵.۵

نمودار scatter برای ترسیم نقاط به صورت نقطه ای بر روی صفحه‌ی دو بعدی هست، در نتیجه طبیعتاً تنها روش دادن روش مختصاتی هست که در شکل ۲۶ نمایش داده شد. همچنین نمودار bubble نیز به همین صورت هست صرفاً داده‌ها به صورت سه بعدی بوده و بعد سوم با استفاده از شعاع دایره‌ها مشخص می‌شود و خب طبیعتاً تمام استایل‌های دایره‌ها قابل تنظیم هست. برای آشنایی بیشتر با جدول موارد قابل تنظیمات به این [لينك](#) مراجعه نمایید.

در انتهای این بخش دو نکته مهم است. اولاً که امکان ترکیب دو نمودار در یک نمودار وجود داره و صرفاً کافیه که داخل دیتاست‌ها دوباره type را تعریف کنیم.



شکل ۳۰: ترکیب دو نمودار

همچنین در نمودارهای radar و line امکان ایجاد فضای بین دیتاست های مختلف با یکدیگر و با مرزهای نمودار وجود دارد که برای فهم دقیق نحوه ای کارکرد این ویژگی به این [مثال](#) رجوع کنید.

۶ محورها و تنظیمات دیگر

در مورد محورها دو دسته تنظیمات داریم. برای حالتی که محورها به صورت شعایی هستن مثل نمودار radar و نمودارهای شعایی دیگر، یک مجموعه از تنظیمات به صورت خطی داریم که نمونه نحوه‌ی استفاده ویژگی max و min در شکل ۲۸ نشان داده شد. تمام این تنظیمات از طریق تنظیم زیر دسته‌ی ticks انجام می‌شوند. طبیعتاً تنظیمات لیل‌ها و استایل‌ها نیز در هر دو مدل کارتزین یا شعایی ممکن است که مجموعه‌ی تنظیمات و کالبک‌های موجود برای محورها در [اینجا](#) موجود هست.

Name	Type	Default	Description
backdropColor	Color	'rgba(255, 255, 255, 0.75)'	Color of label backdrops.
backdropPaddingX	number	2	Horizontal padding of label backdrop.
backdropPaddingY	number	2	Vertical padding of label backdrop.
beginAtZero	boolean	false	if true, scale will include 0 if it is not already included.
min	number		User defined minimum number for the scale, overrides minimum value from data. more...
max	number		User defined maximum number for the scale, overrides maximum value from data. more...
maxTicksLimit	number	11	Maximum number of ticks and gridlines to show.
precision	number		if defined and stepSize is not specified, the step size will be rounded to this many decimal places.
stepSize	number		User defined fixed step size for the scale. more...
suggestedMax	number		Adjustment used when calculating the maximum data value. more...
suggestedMin	number		Adjustment used when calculating the minimum data value. more...
showLabelBackdrop	boolean	true	If true, draw a background behind the tick labels.

شکل ۳۱: تنظیمات ticks

همچنین برای حالت کارتزین نیز امکان تنظیم هر کدام از محورها به صورت خطی یا لگاریتمی وجود داشته و جدول جدول تنظیمات ticks در این حالت به صورت زیر است.

Name	Type	Default	Description
beginAtZero	boolean		if true, scale will include 0 if it is not already included.
maxTicksLimit	number	11	Maximum number of ticks and gridlines to show.
precision	number		if defined and <code>stepSize</code> is not specified, the step size will be rounded to this many decimal places.
stepSize	number		User defined fixed step size for the scale. more...
suggestedMax	number		Adjustment used when calculating the maximum data value. more...
suggestedMin	number		Adjustment used when calculating the minimum data value. more...

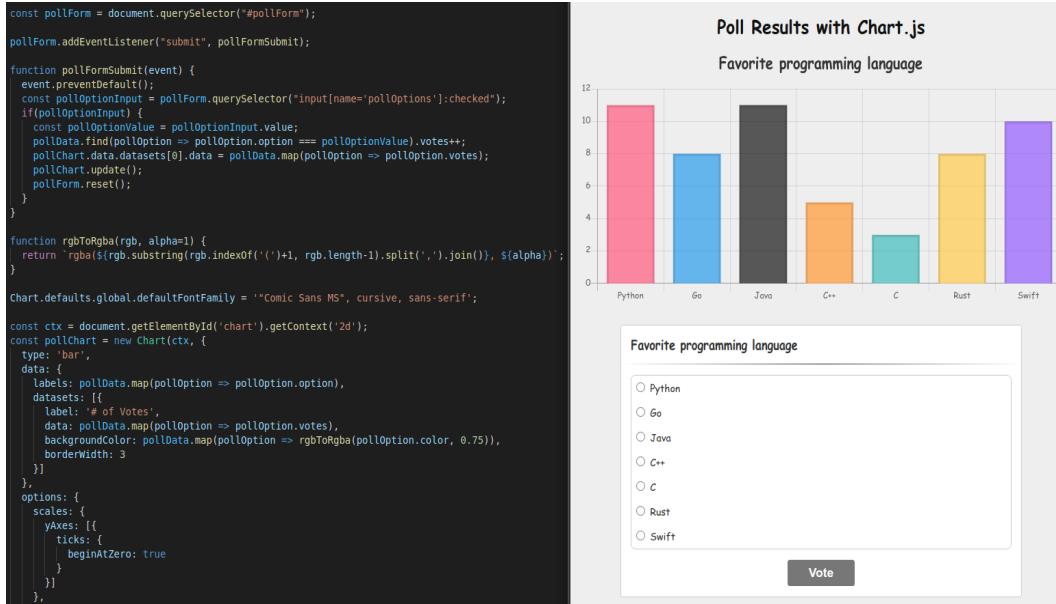
شکل ۳۲: تنظیمات ticks

همچنین در صورتی که بخواهیم لیبل های محور ما به صورت زمانی باشد، Chart.js ابزارهای مختلفی را برای ما فراهم کرده است که در [اینجا](#) لیست این ابزارها موجود است. نهایتاً مانند هر ابزار ترسیم نمودار دیگر امکان فعل سازی و تنظیم grid line نمودار نیز وجود دارد که اطلاعات بیشتر در مورد استایل خط ها [اینجا](#) موجود است.

۷ ایجاد تغییرات در نمودار

در مورد ایجاد تغییر در نمودار و آپدیت آن همانطور که قبلا هم اشاره شد میتوانیم مثلا از چارت لیبل برداریم یا به آن اضافه کنیم یا کلا تنظیماتی را تغییر دهیم و نهایتا با دستور `chart.update()` تغییرمان را بر روی چارت اعمال کنیم. طبیعتاً اگر نخواهیم که این تغییر به صورت اینیمیشن وار انجام شود صرفا کافی است مدت زمانیمیشن را صفر کنیم تا نمودار درجا نندر محدد شود. همچنین یکی دیگر از روش های آپدیت نمودار استفاده از پلاگین های آن میباشد.

برای آن که بیشتر با بحث آپدیت نمودار آشنا شوید در اینجا یک مثال میاوریم. در این مثال یک لیست رای گیری داریم که پس از کلیک بر روی دکمه ی سابمیت اطلاعات نمودار آپدیت میشود. طبیعتاً همین مثال میتواند بسط پیدا کند و زدن بر روی دکمه ی سابمیت معادل رسیدن داده ی نمودار از سمت سرور باشد. در نتیجه ما آرایه ی اطلاعاتمان را که اطلاعات نمودار به آن وابسته هست رو متناسب با نتیجه ی نظرسنجی آپدیت میکنیم و سپس دستور آپدیت نمودار را میدهیم. در نتیجه نمودار ریندر شده و چون ما آرایه ی داده های نمودار را آپدیت کرده بودیم، نمودار هم با استفاده از اینیمیشن (پیش فرض) آپدیت میشود.



شکل ۳۳: یک مثال از آپدیت نمودار

نهایتاً تا اینجا ما امکانات Chart.js را معرفی کردیم، اما همانطور که مشاهده میشود همچنان کلی از امکانات متعدد مانند امکان زوم بر روی نمودار، نمودارهای کنسل و تنظیمات متعددی وجود ندارد. این موارد توسعه افروزنه هایی که دیگران توسعه داده اند برای ما تا حدی فراهم شده اند که لیست کاملی از افزونه های معروف Chart.js موجود هست.