



Charts

Plugin for Joomla!

This manual documents **version 12.x** of the Joomla! extension.

<https://www.aimy-extensions.com/joomla/charts.html>

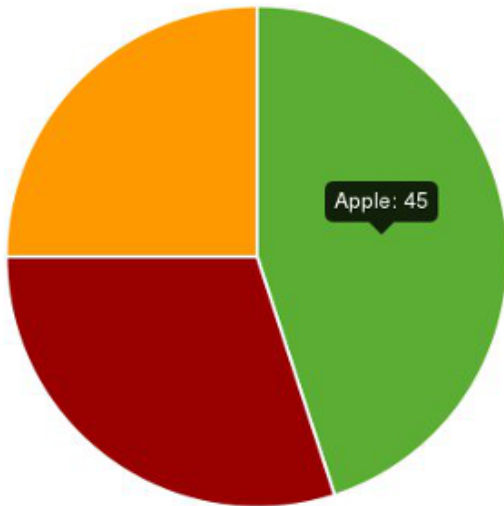
Contents

1	Introduction	3
2	Installation	4
3	Configuration	4
3.1	Enabling the Plugin	4
3.2	Setting Default Options	5
3.2.1	Default Width	5
3.2.2	Default Height	5
3.2.3	Default Animation	5
3.2.4	Responsive (PRO feature)	5
4	Generating Charts	5
4.1	Pie Chart	6
4.1.1	Individual Settings (PRO feature)	6
4.1.2	A Complete Example: This Year's Harvest	6
4.2	Doughnut Chart	8
4.3	Bar Chart	9
4.3.1	Individual Settings (PRO feature)	11
4.3.2	Different Examples	11
4.4	Line Chart	15
4.4.1	A Complete Example: Expecting Next Year's Harvest	15
4.5	Uncontinuous Line Chart (PRO feature)	17
4.5.1	A Complete Example: Showing Crop Failure	17
5	Advanced Configuration	19
5.1	Styling the Container	19
5.2	Using Colors	20
6	Copyright & Trademark Notice	20

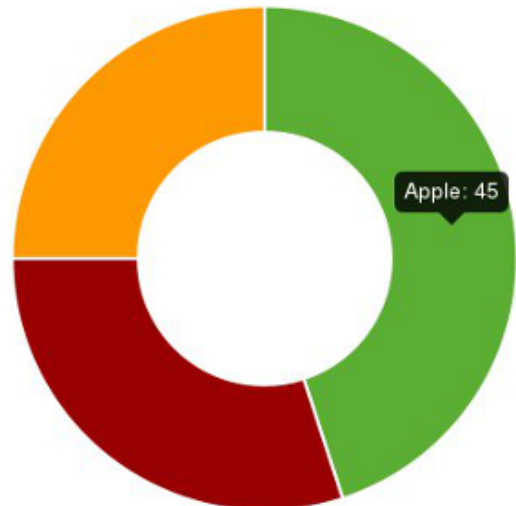
1 Introduction

The Joomla! plugin [Aimy Charts](#) allows you to display various charts on your website to visualize your data.

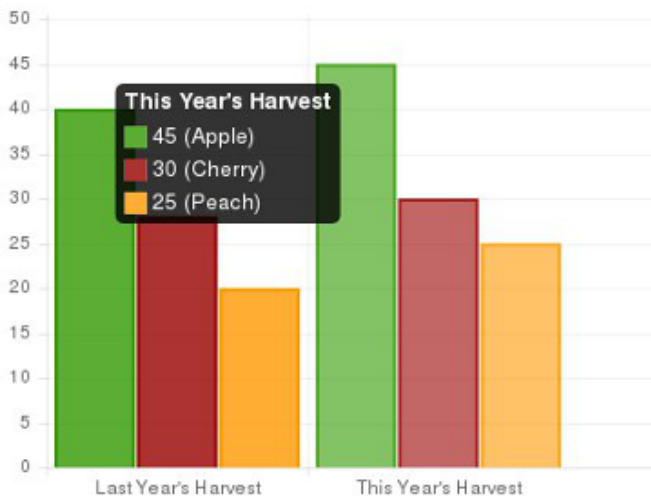
You are free to choose between pie, doughnut, bar and line charts. [Aimy Charts PRO](#) additionally provides support for uncontinuous line charts.



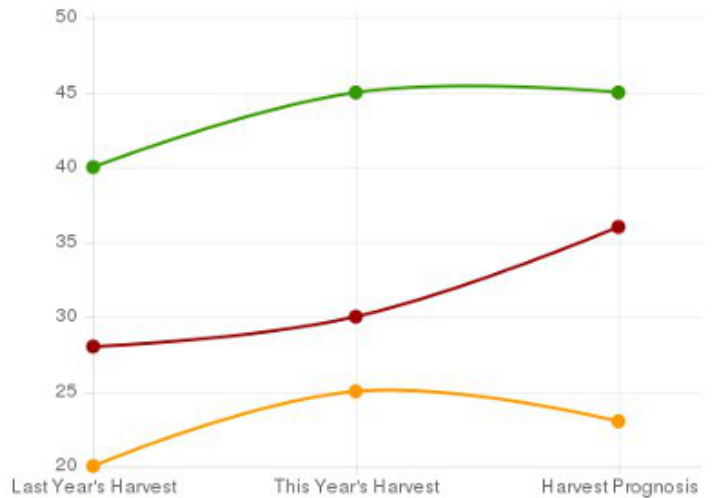
Pie



Doughnut



Bar



Line

[Aimy Charts](#) uses the excellent Chart.js JavaScript library by Nick Downie to visualize your data using plain *HTML 5* and JavaScript. No flash required!

This manual guides you through all steps necessary to install and use the plugin.

2 Installation

The installation of the extension follows the common Joomla! procedures.

In case you are not familiar with these procedures, proceed as follows:

1. Download the extension's ZIP archive
2. Log into your Joomla! backend as "Super User"
3. From the menu, choose "*Extensions*" → "*Manage*" → "*Install*"
4. Click on the "*Or browse for file*" button and select the ZIP archive

The extension's archive will be uploaded and installed afterwards.

For further information, please have a look at the Joomla! documentation [Installing an Extension](#).

NOTE: All plugins are disabled by Joomla! when installed for the first time. To enable [Aimy Charts](#), proceed and configure the plugin.

3 Configuration

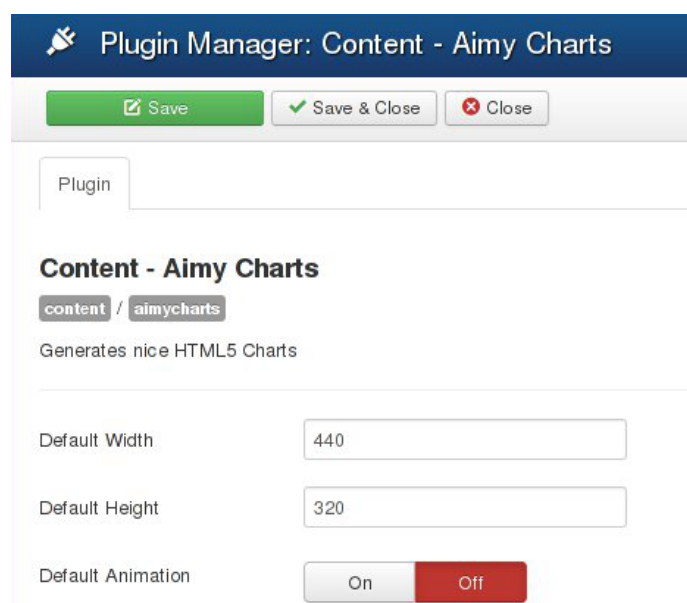
After a fresh installation, click on the "*Configure plugin now*" button on the installation report page.

At any time, you can configure the [Aimy Charts](#) plugin using Joomla!'s Plugin Manager by choosing "*Extensions*" → "*Plugin Manager*" from the menu. Locate the plugin and click on its name in the "*Plugin Name*" column of the plugin listing.

3.1 Enabling the Plugin

In order to use the plugin's functionality, you have to enable it first.

To do so, change the plugin's status from "*Disabled*" to "*Enabled*" and apply your changes by clicking on either the "*Save*" or "*Save & Close*" button in the toolbar.



The screenshot shows the Joomla! Plugin Manager interface for the 'Content - Aimy Charts' plugin. At the top, there is a blue header with the Joomla! logo and the text 'Plugin Manager: Content - Aimy Charts'. Below the header is a toolbar with three buttons: 'Save' (green), 'Save & Close' (green with a checkmark), and 'Close' (red with an X). The main content area has a 'Plugin' tab selected. The plugin name 'Content - Aimy Charts' is displayed, along with a breadcrumb trail 'content / aimycharts' and a description 'Generates nice HTML5 Charts'. Below this, there are three configuration options: 'Default Width' with a text input field containing '440', 'Default Height' with a text input field containing '320', and 'Default Animation' with a radio button selector where 'Off' is selected.

3.2 Setting Default Options

Aimy Charts allows you to customize the way graphics are generated in a couple of ways.

NOTE: This configuration sets the default values. If you use **Aimy Charts** in the PRO version, you may set individual options for each chart (see details in chapter "Generating Charts").

3.2.1 Default Width

Allows you to set the default width of the chart graphic in pixels.

3.2.2 Default Height

Allows you to set the default height of the chart graphic in pixels.

3.2.3 Default Animation

Allows you to set whether the chart graphics should be displayed with an **initial animation** by default.

3.2.4 Responsive (PRO feature)

If enabled, the generated charts will automatically scale on smaller screens. Your entered absolute width and height will be interpreted as a maximum size then.

4 Generating Charts

Enhancing your website's *articles* with nice chart graphics is easy using **Aimy Charts**.

For generating a chart with **Aimy Charts** you simply enclose the data you like to visualize by a *special tag* that is understood and handled by the plugin.

The following chapters introduce you to the different types of charts supported by **Aimy Charts** and empower you to prepare your data in the correct *format*. Each chapter closes with a complete example so you have a template to easily get started.

To create a chart, first open the article it should be part of in Joomla!'s *Article Manager* (*Content* → *Article Manager*).

NOTE ON DECIMAL MARK: **Aimy Charts** expects the integer part of numbers to be separated from their fractional part by a **point**, i.e. *12.34*.

4.1 Pie Chart

The *Pie Chart* allows you to visualize your data in a parts-of-a-whole relation.

Each part consists of an *Entity Name*, its *Numeric Value* and (optionally) an assigned *Color*.

Opening Tag

```
{AimyChartsPie}
```

Closing Tag

```
{/AimyChartsPie}
```

Data Record Fields

1. *Entity Name* (required)
2. *Numeric Value* (required)
3. *Color Code* (optional)

Data Field Separator

Semicolon (";")

The set of records has to be wrapped in the correct opening and closing tags, with each record presented on its own line. Any leading or trailing spaces around fields will be trimmed so you are free to indent your data records.

If no *Color* is manually assigned to an *Entity Name*, one default color for all fields will be used by [Aimy Charts](#) automatically.

The *Entity Name* will be displayed along with its value whenever the mouse hovers over it.

4.1.1 Individual Settings (PRO feature)

You can set individual options per chart when embedding one.

For example:

```
{AimyChartsPie width=300, animate=0} ... {/AimyChartsPie}
```

Each option is separated by a comma. The following options are available:

width and height

Define the chart's size with pixels e.g. width=300

animate

Turn this option on (1) or off (0) e.g. animate=0

responsive

Turn this option on (1) or off (0) e.g. responsive=1

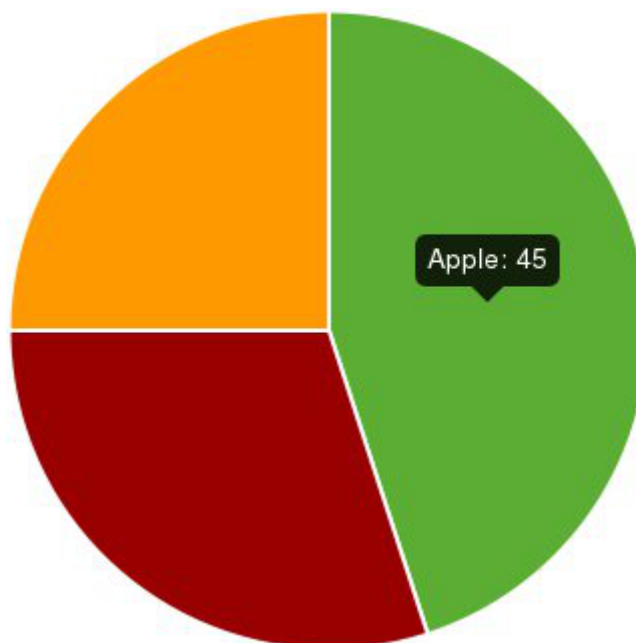
4.1.2 A Complete Example: This Year's Harvest

All examples in this manual are about harvesting some amount of tasty fruits, namely apples, cherries and peaches. This year's harvest has been great and we got 45 kilos of apples, 30 kilos of sweet cherries and 25 kilos of peaches.

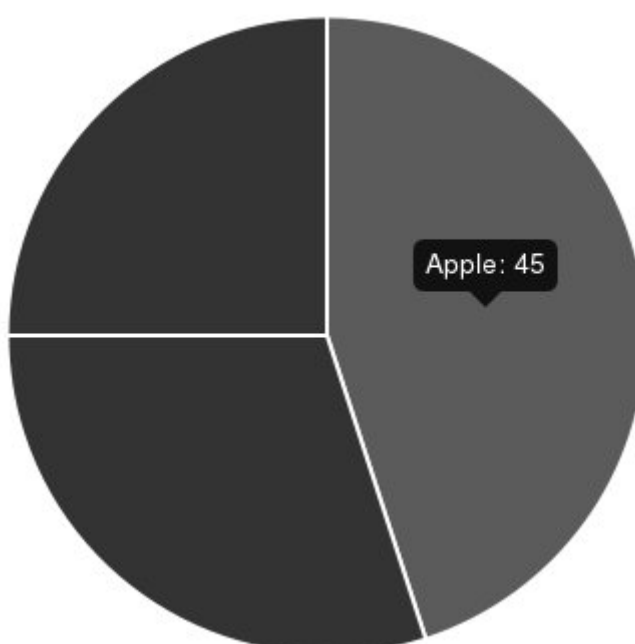
To visualize our harvest, we add the following [Aimy Charts](#) block to our this year's harvesting article.

```
{AimyChartsPie}
Apple; 45; #390
Cherry; 30; #900
Peach; 25; #F90
{/AimyChartsPie}
```

[Aimy Charts](#) picks up the block, extracts its data records and displays the following chart (the "apple" slice is hovered upon):



If no color is chosen, the chart will look like this (the "apple" slice is hovered upon):



4.2 Doughnut Chart

The *Doughnut Chart* is a variant of the *Pie Chart* and thus has exactly the same requirements. Only the opening and closing tags differ:

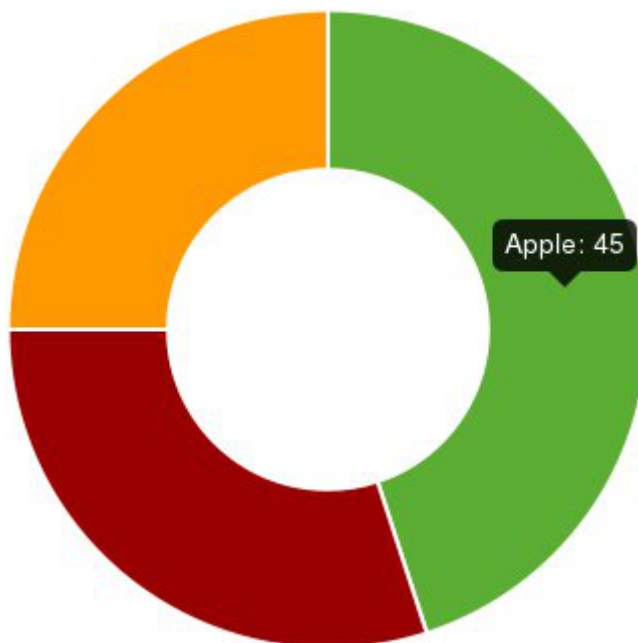
Opening Tag

```
{AimyChartsDoughnut}
```

Closing Tag

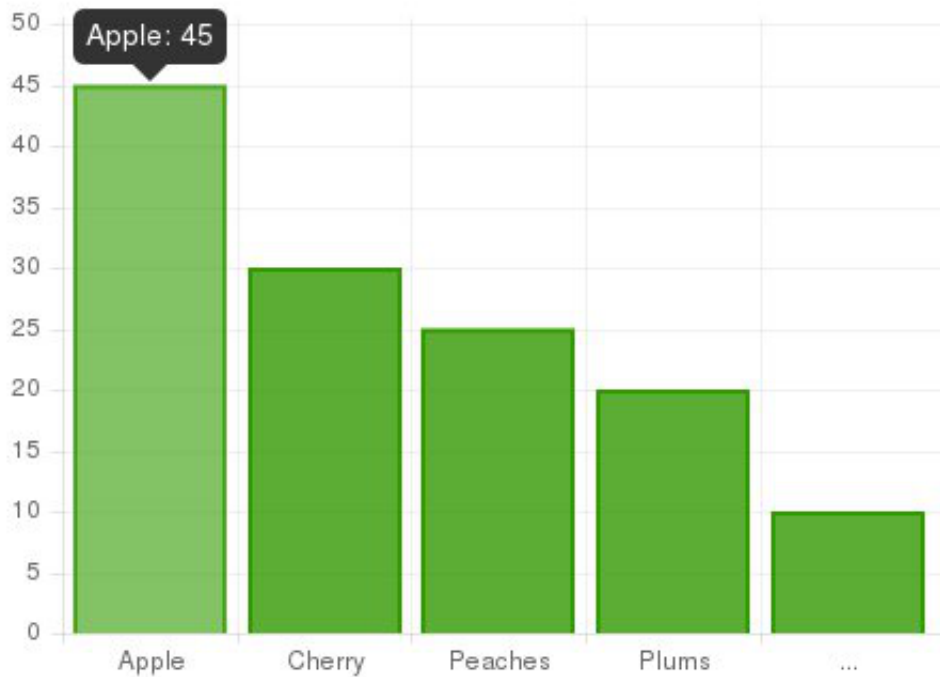
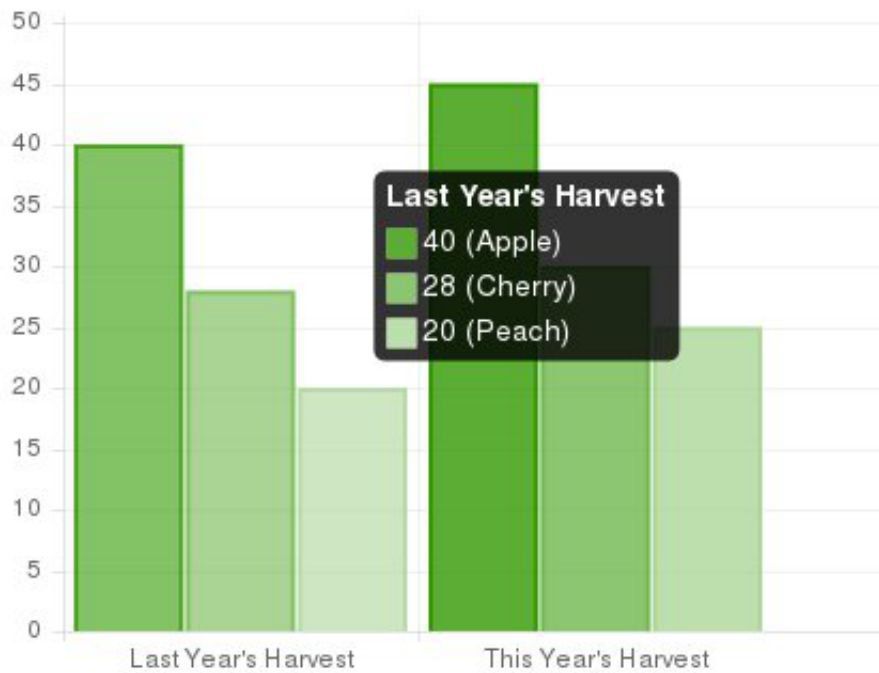
```
{/AimyChartsDoughnut}
```

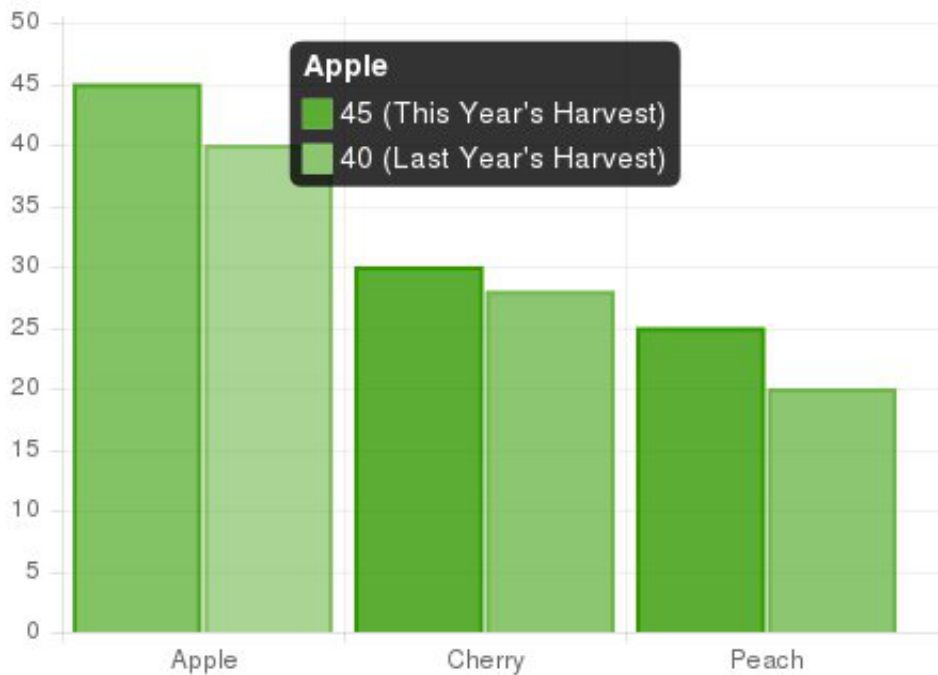
Given the example data used in the pie chart's harvest example, the doughnut chart looks like this (again, the "apple" slice is hovered upon):



4.3 Bar Chart

In contrast to the pie and doughnut charts presented in the previous chapters, the data format of the *Bar Chart* is a bit more complex as it uses x- and y-axis. Additionally there are different possibilities to display the data.





Opening Tag

```
{AimyChartsBar}
```

Closing Tag

```
{/AimyChartsBar}
```

Data Record Fields

1. *Block Name* (→ x-axis, required)
2. *Entity Name* (required) Within a *Block* there may be one or more *Entities*. See examples below.
3. *Numeric Value* (→ y-axis, required)
4. *Color Code* (optional)

Data Field Separator

Semicolon (";")

The set of records has to be wrapped in the correct opening and closing tags, with each record presented on its own line. Any leading or trailing spaces around fields will be trimmed so you are free to indent your data records.

Every *Entity Name* has a distinct color assigned. It is sufficient to set the color on the first definition of the *Entity Name* in a data record. Any subsequent color definitions for an already defined color will overwrite the previous definition. If no *Color* is manually assigned to an *Entity Name*, one default color for all fields will be used by [Aimy Charts](#) automatically.

Whenever the mouse hovers over a *Block Name*, a legend will be shown that contains all *Entity Names* along with their values and the *Block Name* as a heading.

The scale of the axis will be defined automatically depending on the amount of records and their values.

4.3.1 Individual Settings (PRO feature)

You can set individual options per chart when embedding one.

For example:

```
{AimyChartsBar width=200, animate=0} ... {/AimyChartsBar}
```

Each option is separated by a comma. The following options are available:

width and height

Define the chart's size with pixels e.g. width=300

animate

Turn this option on (1) or off (0) e.g. animate=0

responsive

Turn this option on (1) or off (0) e.g. responsive=1

4.3.2 Different Examples

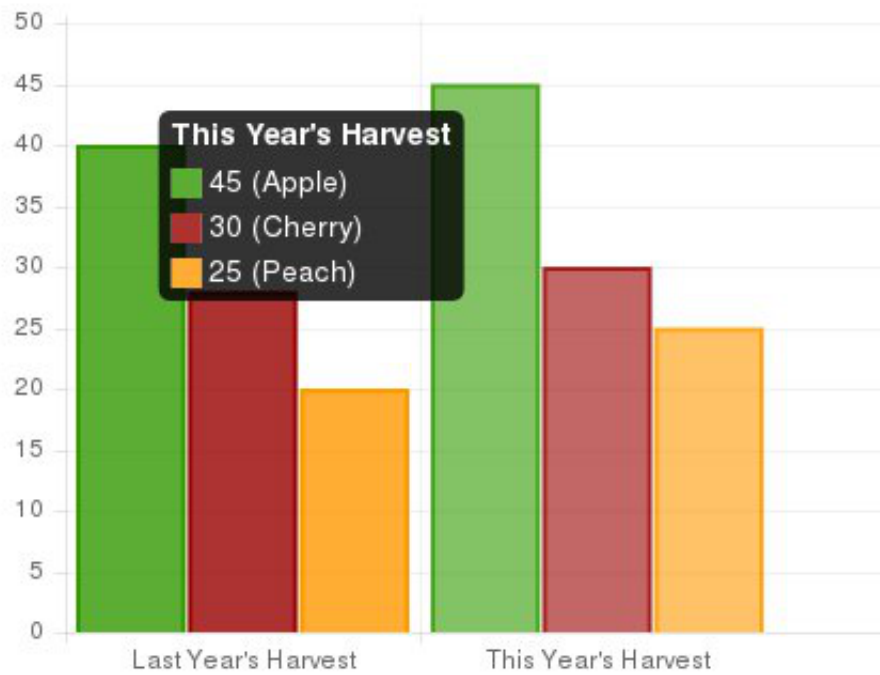
Compare The Last Two Harvests As we have already shown in the pie and doughnut chart, this year's harvest was great - but what about last year's? It hasn't been as good as this year's, yielding only 40 kilos of apples, 28 kilos of cherries and 20 kilos of peaches.

To visualize our harvest not only for the current, but for the previous year as well, we add the following [Aimy Charts](#) block to our harvesting comparison article.

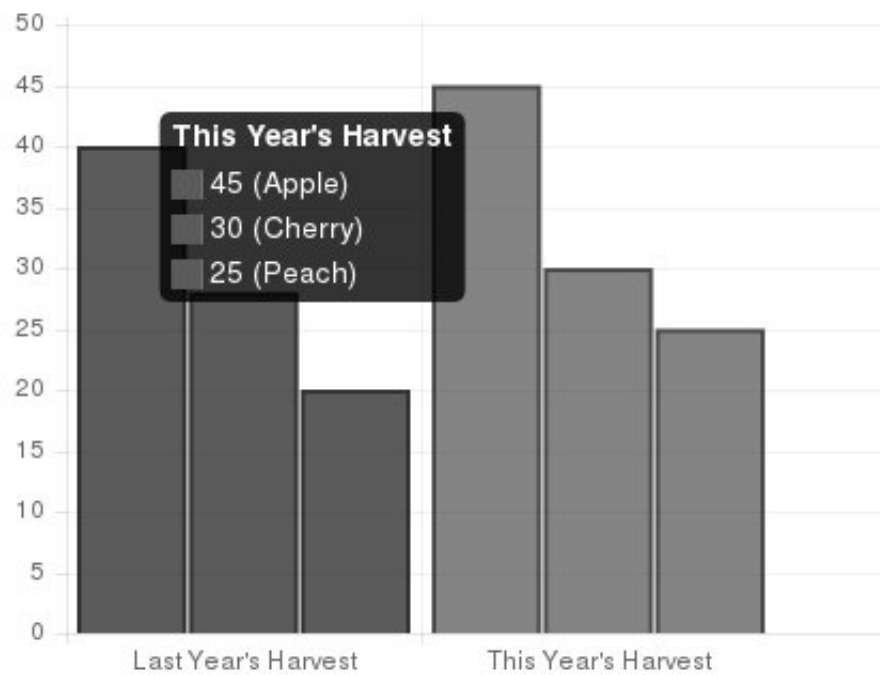
```
{AimyChartsBar}
Last Year's Harvest; Apple; 40; #390
Last Year's Harvest; Cherry; 28; #900
Last Year's Harvest; Peach; 20; #F90
This Year's Harvest; Apple; 45
This Year's Harvest; Cherry; 30
This Year's Harvest; Peach; 25
{/AimyChartsBar}
```

NOTE: The colors are set on each first usage of an *Entity Name* only and are automatically used by [Aimy Charts](#) from this point on.

[Aimy Charts](#) picks up the block, extracts its data records and displays the following chart (the "This Year's Harvest" block is hovered upon):



If no color is chosen, the chart will look like this (the "This Year's Harvest" block is hovered upon):



Show The Current Harvest To visualize just one year's harvest with a bar chart, use the *Block Field* for only one entity - every single fruit in this case.

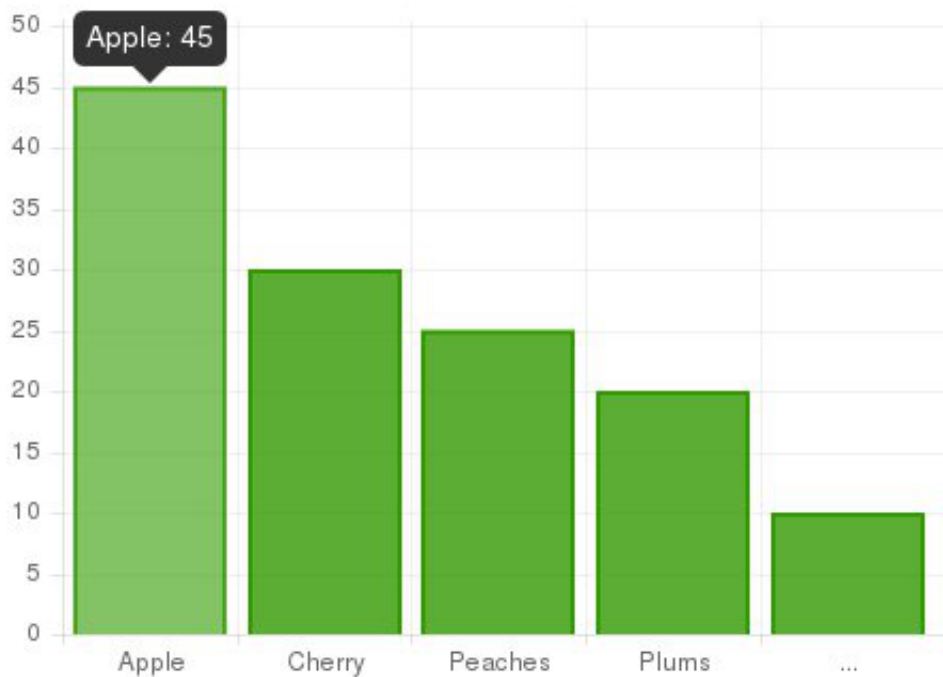
The *Block Field* represents the fruit while the *Entity* stays the same for all entries. The name chosen for *Entity* will not be displayed in the chart.

We add the following **Aimy Charts** block to our harvesting comparison article.

```
{AIMYCHARTSBAR}
Apple; This Year's Harvest; 45; #390
Cherry; This Year's Harvest; 30
Peaches; This Year's Harvest; 25
Plums; This Year's Harvest; 20
... ; This Year's Harvest; 10
{/AIMYCHARTSBAR}
```

NOTE: The colors are set on each first usage of an *Entity Name* only and are automatically used by **Aimy Charts** from this point on. Therefore in this variant **all entries have the same color**.

Aimy Charts picks up the block, extracts its data records and displays the following chart (the "Apple" block is hovered upon):



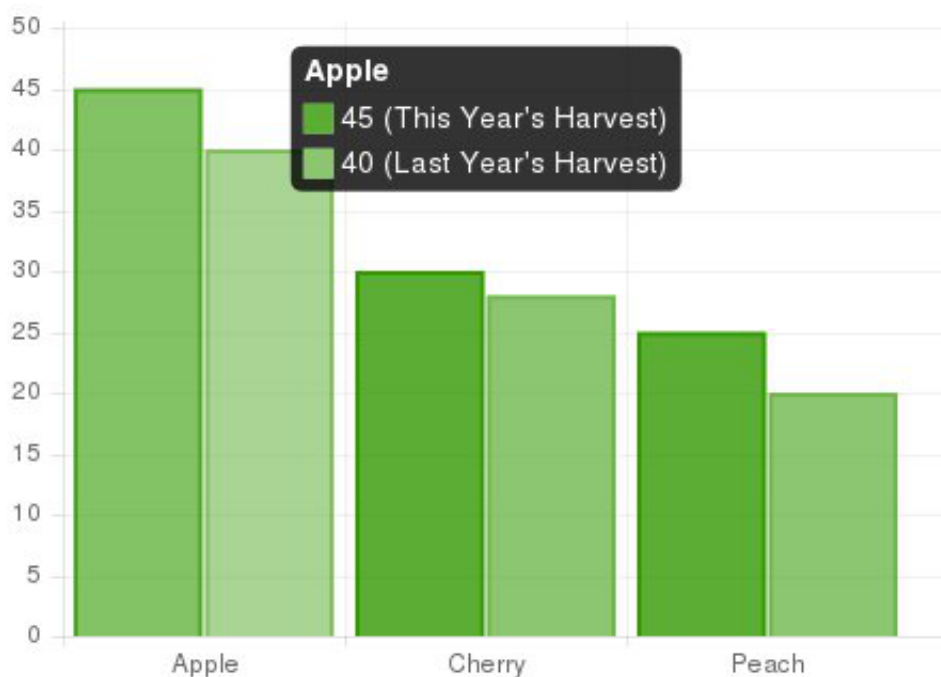
Compare Fruits To visualize the development of each fruit with a bar chart, the *Block Field* represents the fruit while the *Entity* defines the compared harvests.

We add the following **Aimy Charts** block to our harvesting comparison article.

```
{AIMYCHARTSBAR}  
Apple; This Year's Harvest; 45; #339900  
Apple; Last Year's Harvest; 40; #70B84D  
  
Cherry; This Year's Harvest; 30;  
Cherry; Last Year's Harvest; 28;  
  
Peach; This Year's Harvest; 25;  
Peach; Last Year's Harvest; 20;  
{/AIMYCHARTSBAR}
```

NOTE: The colors are set on each first usage of an *Entity Name* only and are automatically used by **Aimy Charts** from this point on.

Aimy Charts picks up the block, extracts its data records and displays the following chart (the "Apple" block is hovered upon):



4.4 Line Chart

The *Line Chart* shares its data format with the *Bar Chart* introduced in the last chapter. Only the opening and closing tags differ:

Opening Tag

```
{AimyChartsLine}
```

Closing Tag

```
{/AimyChartsLine}
```

Setting a *Color* for a line and all other options work the same way as explained for the *Bar Chart*.

4.4.1 A Complete Example: Expecting Next Year's Harvest

The *Line Chart* is ideal to visualize data that is related to events in time. To take our examples one step further we will visualize the prognosis for next year's harvest using a *Line Chart*.

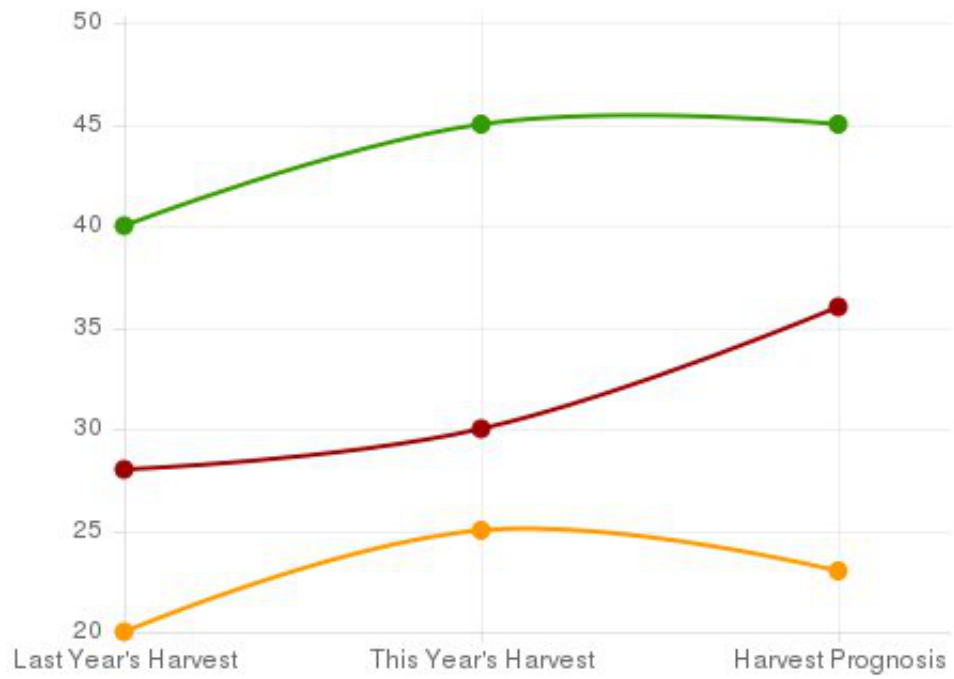
Somehow we are somewhat certain that next year will bring about 45 kilos of apples, 36 kilos of cherries and 23 kilos of peaches. To set this prognosis into relation with the harvest facts we have already collected, we add the following block to our harvesting comparison article.

```
{AimyChartsLine}
Last Year's Harvest; Apple; 40; #390
Last Year's Harvest; Cherry; 28; #900
Last Year's Harvest; Peach; 20; #F90

This Year's Harvest; Apple; 45
This Year's Harvest; Cherry; 30
This Year's Harvest; Peach; 25

Harvest Prognosis; Apple; 45
Harvest Prognosis; Cherry; 36
Harvest Prognosis; Peach; 23
{/AimyChartsLine}
```

Aimy Charts picks up the block, extracts its data records and displays the following chart:



4.5 Uncontinuous Line Chart (PRO feature)

The *Uncontinuous Line Chart* behaves exactly as the *Line Chart* introduced above, except:

1. A data value of zero (0) does not appear as a point on the chart and causes a "break" within its dataset's line.
2. For better readability, the lines between points are not smoothed.

This type of chart uses the following opening and closing tags:

Opening Tag

```
{AimyChartsUncontinuousLine}
```

Closing Tag

```
{/AimyChartsUncontinuousLine}
```

All other options work the same way as explained for the *Line Chart*.

4.5.1 A Complete Example: Showing Crop Failure

As an example, the following chart shows crop failures of different fruits for a couple of recent years. While apples did not suffer from crop failure, cherries and peaches did.

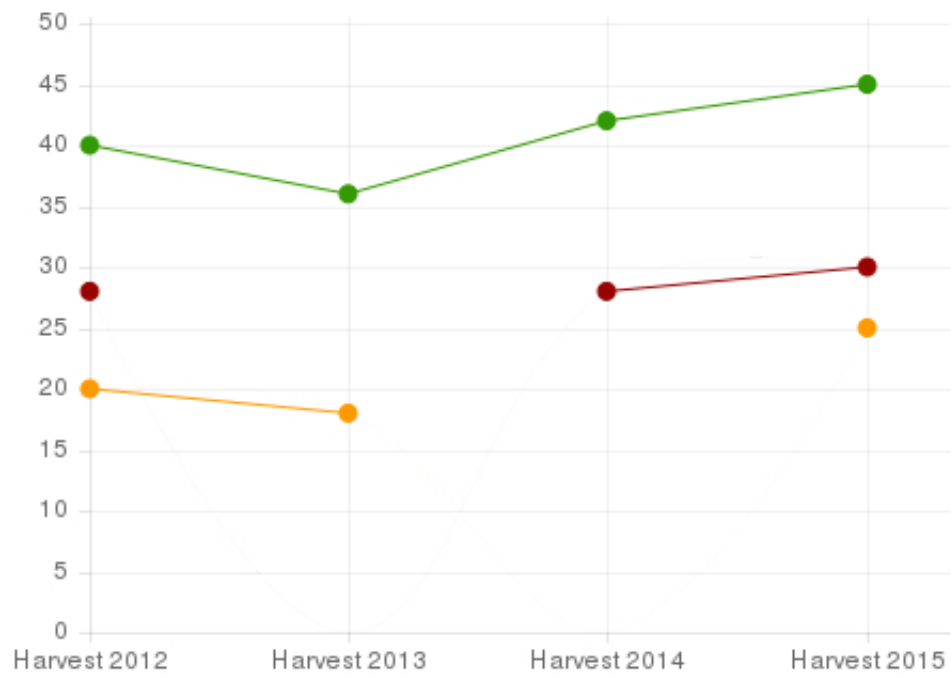
```
{AimyChartsUncontinuousLine}
Harvest 2012; Apple; 40; #390
Harvest 2012; Cherry; 28; #900
Harvest 2012; Peach; 20; #F90

Harvest 2013; Apple; 36
Harvest 2013; Cherry; 0
Harvest 2013; Peach; 18

Harvest 2014; Apple; 42
Harvest 2014; Cherry; 28
Harvest 2014; Peach; 0

Harvest 2015; Apple; 45
Harvest 2015; Cherry; 30
Harvest 2015; Peach; 25
{/AimyChartsUncontinuousLine}
```

Aimy Charts PRO turns this block into the following chart:



5 Advanced Configuration

5.1 Styling the Container

Every chart generated by **Aimy Charts** is wrapped in a `div` HTML element that has both a generic *class* of "AimyCharts" and a specific class that equals the opening tag's name of each chart assigned (i.e. "AimyChartsPie" or "AimyChartsLine").

To reference individual charts, each chart's container provides a unique *id* as well. This id consists of the string "AimyChart", followed by the chart's number (counted ascending from one), like "AimyChart1" or "AimyChart2".

To reference...

...all charts

use the *class* "AimyCharts"

...all charts of one type

use the *class* "AimyCharts**TYPE**", with **TYPE** being either Pie, Doughnut, Bar, Line or UncontinuousLine.

...one specific chart

use the *id* "AimyChart**NUMBER**"

A trimmed HTML example snippet looks like this:

```
<div class="AimyCharts AimyChartsPie" id="AimyChart1">
  <canvas ...></canvas>
</div>
```

You can style the container using standard CSS to meet your requirements. Browse to "[Extensions](#)" → "[Template Manager](#)" → "[Templates](#)" → *Your Template* and extend your template's CSS code, for example:

```
#content div.AimyCharts
{
  text-align: center;
}
#content div#AimyChart1
{
  border: 1px solid #CCC;
}
```

5.2 Using Colors

Aimy Charts allows you to use all colors that are specifiable using CSS.

In a nutshell you are free to specify colors by...

name

CSS specifies a set of predefined color names, like *red*, *green* and *blue*.

Have a look at all defined color names at W3School.

RGB hexadecimal value

CSS allows to specify a color by its hexadecimal red-green-blue value, following the pattern `#RRGGBB` or `#RGB`, like `#F00`, `#008000` or `#00F`.

RGB decimal value

CSS allows to specify a color by its decimal red-green-blue value, following the pattern `rgb(R, G, B)`, like `rgb(255,0,0)`, `rgb(0,128,0)` and `rgb(0,0,255)`.

RGB decimal value plus alpha channel value

Similar to the former, CSS allows to specify an alpha channel for a color, following the pattern `rgba(R, G, B, A)`. The alpha channel parameter *A* is given as a number between 0.0 (→ full transparency) and 1.0 (→ no transparency / fully opaque).

Examples: `rgba(255,0,0,1.0)`, `rgba(0,128,0,0.5)` and `rgba(0,0,255,0.75)`.

W3School also provides a nice color picker that can assist you in selecting your color of choice.

6 Copyright & Trademark Notice

The Joomla!® name and logo are trademarks of Open Source Matters, Inc. in the United States and other countries.

Chart.js is developed and provided under the terms of the MIT license by Nick Downie. Thanks!

Mentioned hard- and software as well as companies may be trademarks of their respective owners. Use of a term in this manual should not be regarded as affecting the validity of any trademark or service mark. A missing annotation of the trademark may not lead to the assumption that no trademark is claimed and may thus be used freely.