1. Intro: What is the learning module about? For whom is it of interest? How will you learn?
2. Scenario: Narrative task which is presented in an authentic situation.
3. Development of the concrete tasks, the work plan, (international) division of work, ways of collaboration (Multidisciplinary or multinational) problem solving, implementation of the tasks
4. Assessment of training success
5. Meta-cognitive self-reflexion and evaluation of the learning process

http://www.chartjs.org/

1.Intro: What is the learning module about? For whom is it of interest? How will you learn?

Chart.js is a JavaScript library to create profesional charts to represent different types of statistics. Based on HTML5 canvas and responsive, light-weight, customizable and easy to use.

Types of charts:
- Line chart
- Bar chart
- Radar chart
- Polar area chart
- Pie chart
- Doughnut chart
- Bubble chart

3. Development of the concrete tasks, the work plan, (international) division of work, ways of collaboration (Multidisciplinary or multinational) problem solving, implementation of the tasks

Task 1. Installation and Integration

First download from:
http://www.chartjs.org/docs/latest/getting-started/installation.html

And then install it through NPM

```
npm install chart.js --save
```

Chart.js can be integrated with plain JavaScript or with different module loaders.

ES6 Modules

```
import Chart from 'chart.js';
var myChart = new Chart(ctx, {...});
```

Script Tag

```
<script src="path/to/chartjs/dist/Chart.js"></script>
<script>
    var myChart = new Chart(ctx, {...});
</script>
```

Common JS

```
var Chart = require('chart.js');
var myChart = new Chart(ctx, {...});
```

Require JS

```
require(['path/to/chartjs/dist/Chart.js'], function(Chart){
    var myChart = new Chart(ctx, {...});
});
```

Task 2. Usage of CharJS

Creating a Chart

Inside the .html file place the code where you want to display chart with an ID.

```
<canvas id="myChart"></canvas>
```

And then in an external .js file for example:

```
var ctx = document.getElementById('myChart').getContext('2d');
var myChart = new Chart(ctx, {
```
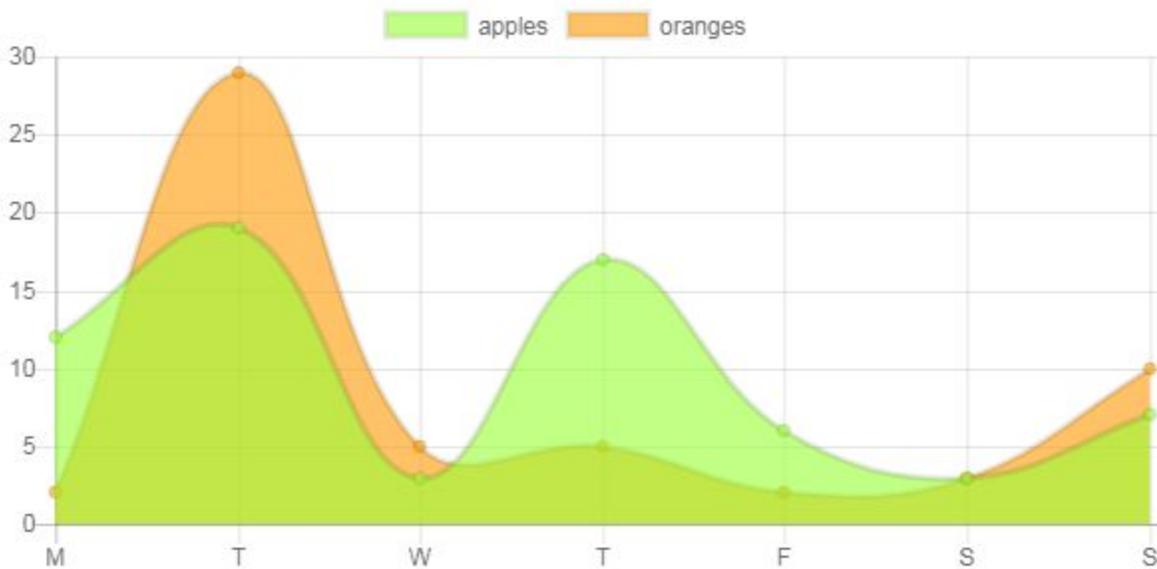
```
  type: 'line',
  data: {
    labels: ['M', 'T', 'W', 'T', 'F', 'S', 'S'],
    datasets: [{
      label: 'apples',
      data: [12, 19, 3, 17, 6, 3, 7],
      backgroundColor: "rgba(153,255,51,0.4)"
    }, {
      label: 'oranges',
      data: [2, 29, 5, 5, 2, 3, 10],
      backgroundColor: "rgba(255,153,0,0.4)"
    }]
  }
});
```

With this result

Or if it changed

```
type: 'line'
```

to:

```
type: 'bar'
```
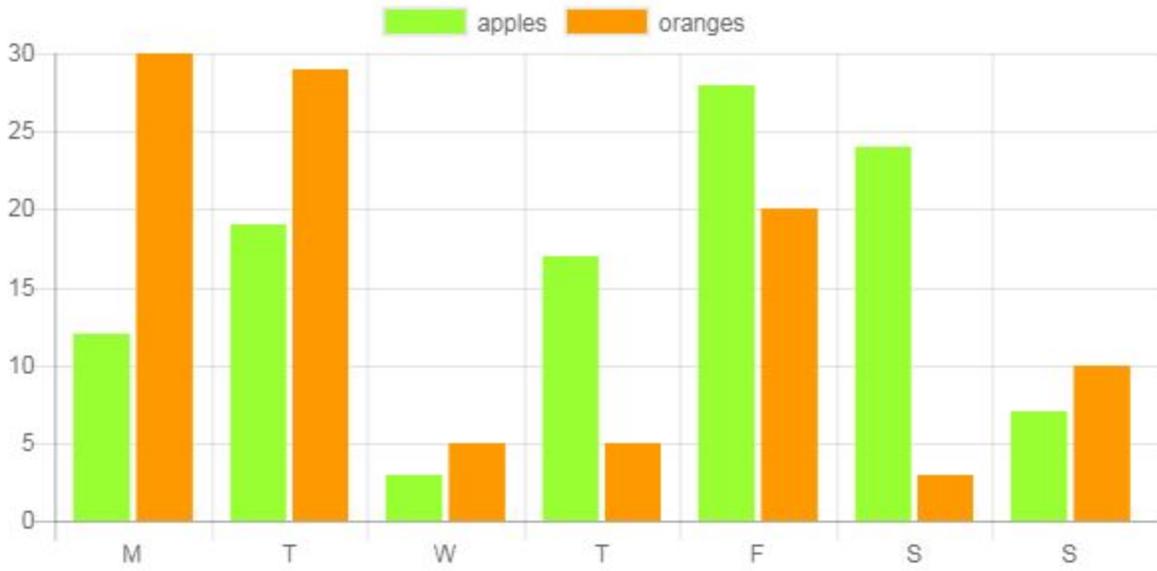
The result will be:

Chart.js is a perfect match for rapid prototyping of simple charts. There are eight main chart types: line, bar, radar, polarArea, pie and doughnut. These diverse charts cover most common ways to visualize data.