



ChartJS & NMBS Leaflet

Datarepresentatie

Contents

1	LEERDOELEN	3
2	OEFENING 1: EEN GRAFIEK MET BEHULP VAN CHART.JS	4
1.1	Opgave	4
1.2	Vooraf.....	4
1.3	Stap 1: Fetch data	5
1.4	Stap 2: Intinialize Chart.js.....	6
1.5	Stap 3: Prepare data.....	6
1.6	Stap 3: Draw Chart	6
1	OEFENING 2: LEAFLET NMBS	8
1.1	Vooraf.....	9
1.2	HTML.....	9
1.3	Javascript.....	10

1 Leerdoelen

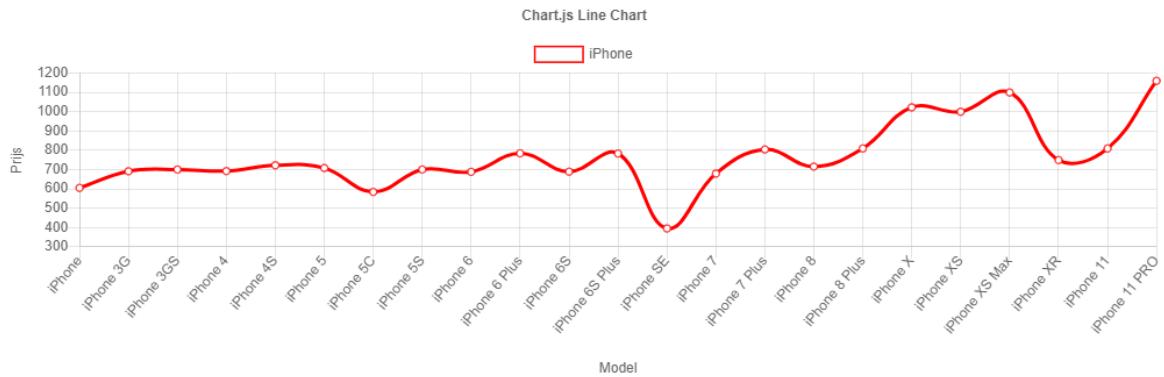
Externe applicaties kunnen gebruiken om data visueel weer te geven.

2 Oefening 1: Een grafiek met behulp van Chart.js

1.1 Opgave

Maak met [Chart.js](#) een grafiek van de gegevens in het bestand `iphone.json`. Het resultaat zou er als volgt moeten uitzien.

Pricing through the years



1.2 Vooraf

Voordat je start neem - op de website van dit project - de documentatie door van dit javascript framework.

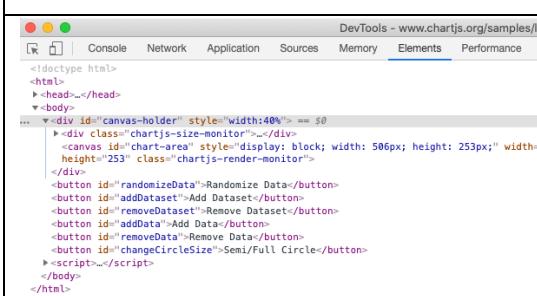
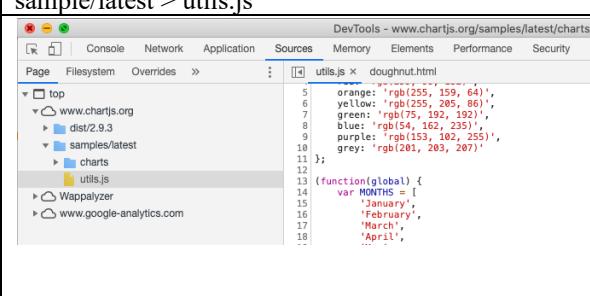
Je hoeft niet alles door te lezen maar bekijk alvast eens deze pagina's om een idee te krijgen wat chart.js is en kan. De kans bestaat dat je niet alles onmiddellijk begrijpt. Dit is niet erg. Als je het labo verder doorloopt zal het een en ander wel duidelijk worden.

- <https://www.chartjs.org/docs/latest/getting-started/>
- <https://www.chartjs.org/docs/latest/getting-started/usage.html>
- <https://www.chartjs.org/docs/latest/configuration/>
- <https://www.chartjs.org/docs/latest/charts/line.html>
- <https://www.chartjs.org/docs/latest/charts/bar.html>

Bekijk ook enkele demo's op:

<https://www.chartjs.org/samples/latest/>

Gebruik Chrome Dev Inspector om de html en javascriptcode van de voorbeelden te verkennen.

Tab Elements	Tab Sources > Page > Top > www.chartjs.org/samples/latest/charts/doughnut.html > utils.js
	 <pre>5 orange: 'rgb(255, 159, 64)', 6 yellow: 'rgb(255, 205, 86)', 7 green: 'rgb(76, 175, 80)', 8 blue: 'rgb(54, 162, 235)', 9 purple: 'rgb(153, 102, 255)', 10 grey: 'rgb(201, 203, 207)' 11 }; 12 13 (function(global) { 14 var MONTHS = [15 'January', 16 'February', 17 'March', 18 'April', 19]; 20 }());</pre>

In de codevoorbeelden zie je dat ze var en getElementById gebruiken ... hoe kan je dit aanpassen naar de werkwijze die je aanleerde in FSWD?

Het zal je ook opvallen dat ze in plaats van DOMContentLoaded gebruik maken van de JQuery wijze.

Chart.js werkt samen met het <canvas> html element. Wil je hier meer over weten? Lees dan zeker eens https://www.w3schools.com/html/html5_canvas.asp door.

Nu je dit framework voor de eerste maal verkend heb, leer je hoe je dit implementeert in de werkwijze van FSWD.

Deze kennismaking met chart.js is het ideale opstapje naar je project 1.

1.3 Stap 1: Fetch data

We zullen de grafiek opvullen met data die we uit de JSON halen die je kan vinden in iphone.json. De JSON bevat een lijst van iPhone-modellen met daarbij het introductiejaar en de kostprijs. Haal de iPhones op via een fetch (de in de handleData functie zit) en geef deze weer in de console.

De app.js zal volgende structuur hebben.

Kopieer deze niet onmiddellijk maar probeer deze eerst zelf op te bouwen!

```
//#region *** Callback-Visualisation - show___ ***
const showData = function(data) {
    console.log(data);
};

//#endregion

//#region *** Data Access - get___ ***
const getIphoneData = function(){
    handleData('./iphone.json', showData);
}
//#endregion

//#region *** INIT / DOMContentLoaded ***
const init = function() {
    console.info('init geladen');
    getIphoneData();
};

document.addEventListener('DOMContentLoaded', function() {
    console.info('DOM geladen');
    init();
});
//#endregion
```

Zoals je kan zien, zetten we geen volledige backend code op. Maar laden we een json bestand in dat zich reeds in de frontend folder bevindt. Dit is dezelfde werkwijze als in de backend, maar met een andere URL.

1.4 Stap 2: Intinialize Chart.js

Wanneer we een grafiek willen maken met het Chart.js framework hebben we een canvas element nodig. Dus voorzie een canvas element in de <html> sectie en geef deze een id. We zetten het formaat goed met de 'height' en 'width' eigenschappen.

```
<body>
| <h1>■ Pricing through the years</h1>
| <div>
| | <canvas id="myChart" class="js-chart" width="300" height="100"></canvas>
| </div>
</body>
```

Voeg ook de Chart.js library toe via een extra scriptfile in de <head> sectie.

```
<script src="https://cdn.jsdelivr.net/npm/chart.js@2.8.0"></script>
```

1.5 Stap 3: Prepare data

Een grafiek bestaat uit een aantal onderdelen. Diegene die we zeker nodig hebben zijn de labels op de horizontale as, nl de iPhone-modellen. Daarnaast hebben we ook de waarden nodig die we zullen tonen op de verticale as, nl de iPhone prijzen.

We gaan de iPhone-modellen en prijzen in een aparte array steken (dit in functie van Chart.js, zie later). Daarna roepen we de functie drawChart op met als parameters beide array's.

```
//#region *** Callback-Visualisation - show____ ***
const showData = function(data) {
    console.log(data);

    let converted_labels = [];
    let converted_data = [];
    for (const iphone of data) {
        converted_labels.push(iphone.unit);
        converted_data.push(iphone.price);
    }
    drawChart(converted_labels, converted_data);
};
//#endregion
```

1.6 Stap 3: Draw Chart

We halen eerst het canvas element op met behulp van `QuerySelector()` en via `getContext` zorgen we ervoor dat we het canvas kunnen vullen. We slaan die referentie op in een variabele met de naam `ctx`.

```
const drawChart = function(labels, data) {
  let ctx = document.querySelector('.js-chart').getContext('2d');
}
```

De configuratie van de grafiek definiëren we in een variable met de naam 'config'. Hierin gaan we zowel de labels als de data binden aan de grafiek, én gaan we ook een aantal properties definiëren die te maken hebben met de layout en het gedrag van de grafiek.

Uiteindelijk maken we een variabele aan die een nieuwe Chart.js grafiek instantieert. We geven ze de naam mychart. De constructor bevat twee argumenten die we al eerder aangemaakt hebben, nl de canvas context en de configuratie van de grafiek.

```

const drawChart = function(labels, data) {
    let ctx = document.querySelector('.js-chart').getContext('2d');

    let config = {
        type: 'line', //specify what type of chart this is
        data: {
            labels: labels, //all of the labels that are going to show along the bottom of the chart
            datasets: [
                {
                    label: 'iPhone', //label that we added at the top
                    backgroundColor: 'white', // styling
                    borderColor: 'red', // styling
                    data: data, //this is basically just the data that we would like to bind to the chart
                    fill: false // styling
                }
            ]
        },
        options: { // options to change style and behaviour of the chart
            responsive: true,
            title: {
                display: true,
                text: 'Chart.js Line Chart'
            },
            tooltips: {
                mode: 'index',
                intersect: true
            },
            hover: {
                mode: 'nearest',
                intersect: true
            },
            scales: {
                xAxes: [
                    {
                        display: true,
                        scaleLabel: {
                            display: true,
                            labelString: 'Model'
                        }
                    }
                ],
                yAxes: [
                    {
                        display: true,
                        scaleLabel: {
                            display: true,
                            labelString: 'Prijs'
                        }
                    }
                ]
            }
        }
    };

    let myChart = new Chart(ctx, config);
}

```

Zo komt onze grafiek tot leven.

1 Oefening 2: Leaflet NMBS

In deze oefening zullen we de NMBS van vorige we(e)k(en) aanvullen met een overzichtspagina waarop we de locatie van de verschillende stations kunnen zien.

De bedoeling is dat we bij het klikken op een van de stations in de lijst de marker van de nmbs willen zien verschijnen, wijzend naar de juiste locatie.

Klik op een bestemming voor meer info.

Beheer

LOCATIES

[Contacteer](#) de beheerde.

Locaties

- Brussel
- Antwerpen
- Gent-Sint-Pieters
- Brugge

De oefening bestaat uit verschillende delen:

1.1 Vooraf

database/update.sql

Door een slordigheid bij het aanleveren van het materiaal van week 6 kregen jullie een database die niet helemaal in orde was. Voordat we met deze oefening starten zullen we dus het bestand database/update.sql even runnen op de treindatabase. Je kan controleren of de database in orde is doordat er in de tabel bestemmingen bij elke bestemming latitude en longitude data staat.

idbestemming	stad	latitude	longitude
100	Brussel	50.8454639	4.3569867
101	Antwerpen	51.217179	4.421361
102	Gent-Sint-Pieters	51.036225	3.710821
103	Brugge	51.223653	3.202935
NULL	NULL	NULL	NULL

1.2 HTML

bestemmingen.html

Voorzie in de HTML de CSS en JS van Leaflet en voorzie een DIV waarin de kaart getoond kan worden.

1.3 Javascript

app.js

Vul in de JS de code aan die zal uitgevoerd worden in het bestand bestemmingen.html.

Eerst en vooral zullen we de bestemmingen moeten ophalen uit de database, dit doen we door gebruik te maken van:

1. getDestinationsForMap: haal alle bestemmingen op uit de backend en voer *showDestinationsForMap* uit.
2. showDestinationsForMap: schrijf voor elke bestemming een stuk html naar de .js-locations:

```
'<li class="c-show-location js-location" data-lat="${waardevoorlat}" data-long="${waardevoorlong}">$
{waardevoorstad}</option>';
```

3. Voer, binnennin *showDestinationsForMap* *listenToClickShowOnMap* uit, zodat er eventhandlers gekoppeld kunnen worden aan de listitems.
4. *listenToClickShowOnMap*: Bij het toekennen van een clickevent op de verschillende li's zorg je ervoor dat het data-attribuut voor de lat en de long uit de li gehaald wordt.
5. Binnenin het clickevent maak je een anonieme functie die een nieuwe marker aanmaakt en deze op de juiste plaats op de map toevoegt.

frontend/assets/marker.svg

frontend/assets/markershadow.svg

Lees : <https://leafletjs.com/reference-1.6.0.html#marker> voor meer info over de marker.

Het is in deze oefening de bedoeling dat er slechts een marker geplaatst kan worden terzelfdertijd. Indien er op Gent en dan Brussel geklikt wordt, moet de marker van Gent weer verdwijnen en enkel deze van Brussel mag getoond worden.



howest
hogeschool