# A glimpse of

# humanity

by a visualisation amateur called Maike Meuris.

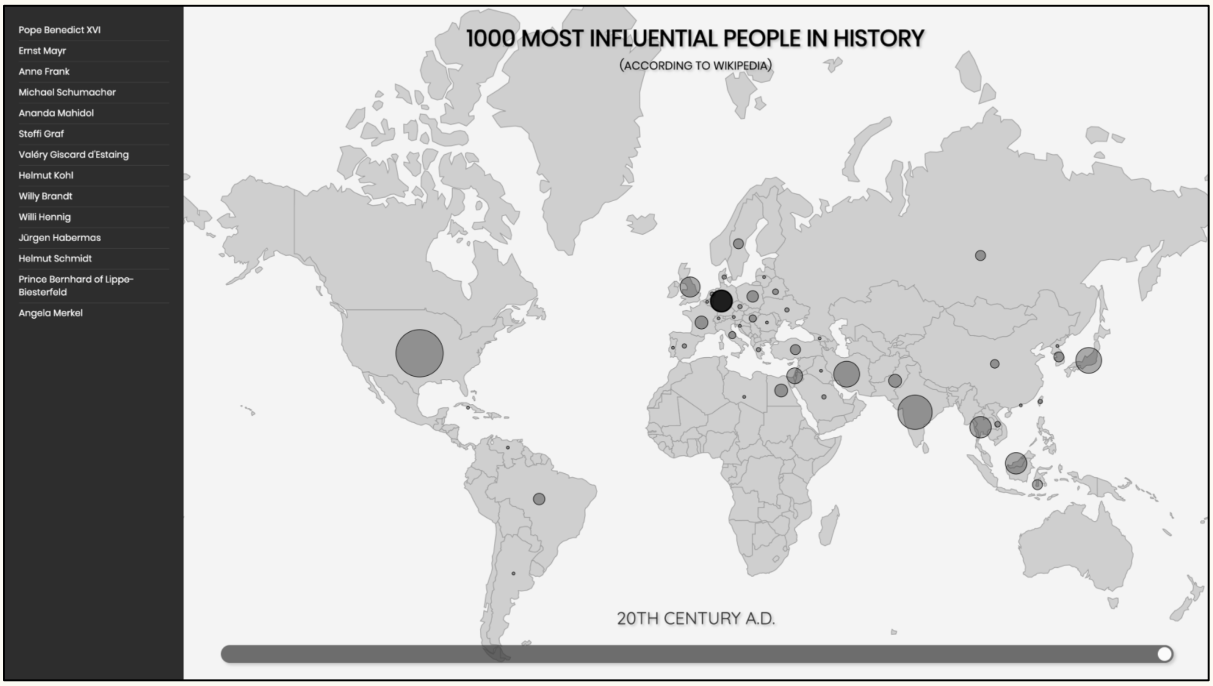
## The Most Influential People In History (According To Wikipedia)

EXPLORATORY – INTERACTIVE – D3

### Visualisation

A web application that shows a world map with a slider underneath. Each point on the slide corresponds with a century. If the slider gets changed, de world map adjusts itself with the information for that century. The map shows people from that century which are found in the top 1000 “most influential people” according to Wikipedia.

It is interesting to evaluate the data on the map. For example, you can see in which century America for discovered.



# Process

The data for this visualisation comes from http://www.quantware.ups-tlse.fr/QWLIB/topwikipeople/. The data was composed by a few people who used a bunch of algorithms on Wikipedia, to determine a list of the "most famous people in history". These are a few lists that can be found on the website. The one I used is http://www.quantware.ups-tlse.fr/QWLIB/topwikipeople/tables/ListOfPageRankGlobalHFs.txt. This list contains more than a 1000 results, but I used the top 1000 for this visualisation because it's a more appealing number.

To create the visualisation in D3, first I followed a course on Lynda.com because D3 was entirely new to me. Then, I had to have a JSON version of the data. To get to this JSON file, I had to do a few thing:

1. I saved the data from the website in a TXT file. I put this file into an online converter (https://www.browserling.com/tools/text-to-csv) to get a CSV file from it.
2. The CSV-version wasn't immediately usable. Some data in the TXT file was left open, so the CSV file didn't add up entirely. I cleaned these mistakes until I got a usable version of the data.
3. Next, I put the CSV file into an online converter (https://shancarter.github.io/mr-data-converter/) to get a JSON file from it.

To determine the data's starting century and ending century, and to know which starting and ending value I had to give to the slider on the web page, I looped all data and put the starting and ending values into variables.

To show the data on the correct spot on the map, each item in the dataset had to have a longitude and latitude that could be used. Unfortunately, this wasn't the case. Each person in the dataset only had a country code. So, I went searching for a dataset with country codes and corresponding longitude and altitude values. I found this this at https://gist.github.com/sindresorhus/1341699. Some country codes were written a little differently, so I had to fix that first too.

In the code, I have a ‚prepareData' function, which transforms both datasets into one array. This array contains all centuries, which contain the country codes that have people in them for that century (+ longituge and latitude data), which contain the corresponding people for that country in that century. That way, the data was easily usable in combination with the map and slider.

# The world in which we live

## Average annual precipitation of the different continents

EXPLORATORY – PHYSICAL – INTERACTIVE – 3D

### Visualisation

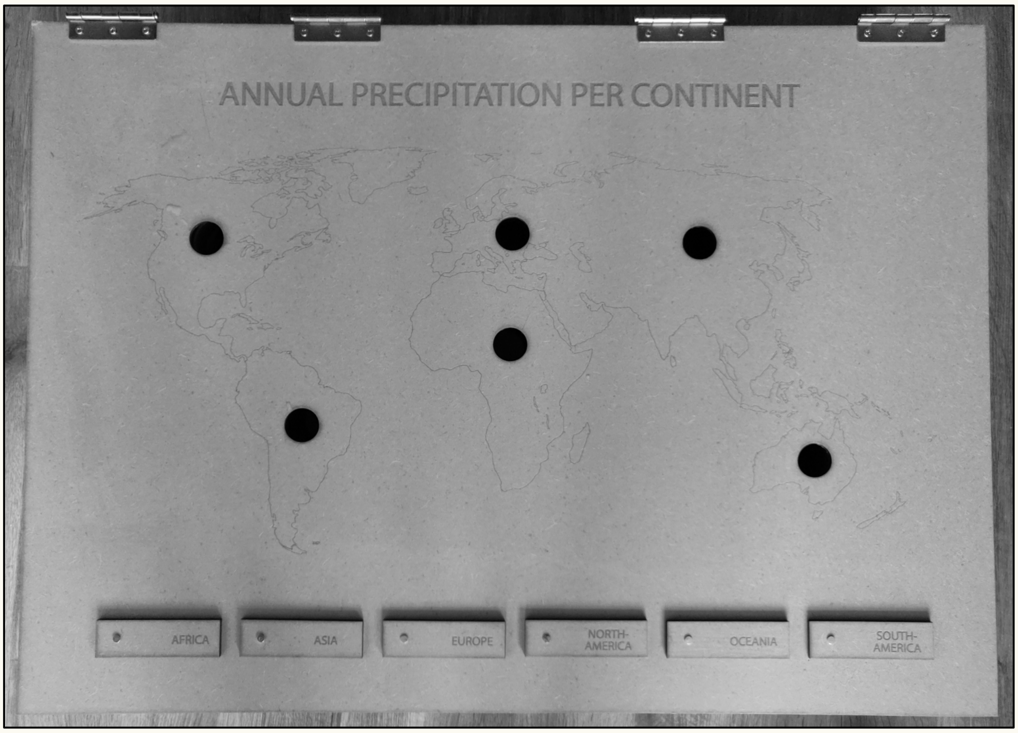
A physical installation showing the average precipitation per continent in 2011. Precipitation means:

> *"Any product of the condensation of atmospheric water vapor that falls under gravity. The main forms of precipitation include drizzle, rain, sleet, snow, graupel and hail."*
>
> \- Wikipedia, Bron: https://en.wikipedia.org/wiki/Precipitation

It's the intention that the "user" of the visualisation gets a world map in front of him. On this map, each continent has a hole in it, where the user can go into with his fingers. Underneath the hole, there's a cup that contains water, in which the amount of water corresponds with the amount of precipitation of that continent.

To show the user the actual data and confirm his possible suspicions, the precipitation data per continent can be read on top of the box, underneath moving cards.

To make the whole even more interactive, the user can also use a measuring stick to read the depths of the cups and make an estimate of the amount of precipitation for that continent.



## Process

### Data processing

I couldn't find the data per continent, so instead I used data from Mecometer.com (http://mecometer.com/topic/average-yearly-precipitation/), which shows the average precipitation in millimetres per country in 2011. I collected this data into an Excel document, with the country's names and the average annual precipitation in millimetres per square meter.

To convert this data per country to data per continent, a few things had to happen:

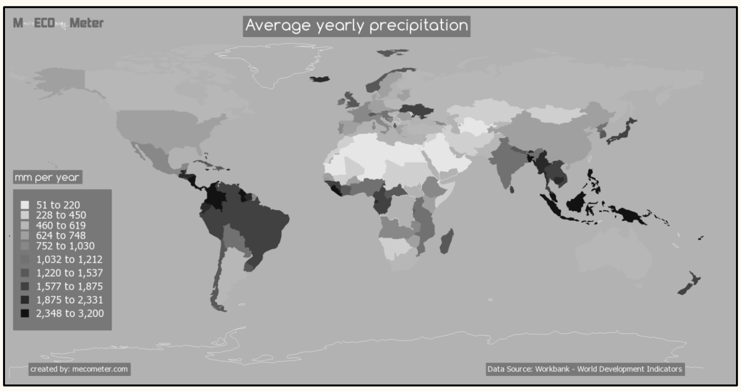
1. Firstly, not all countries could be found on the list. The missing countries had to be added to the list, taking into account the world map had changed a bit since 2011. Many of these missing countries were very small islands, which wouldn't have much influence on a continental level. But some of them were actually very big, like for example the Democratic Republic of the Congo. So I decided to fill in the missing data. The data which I found on the internet was never exactly from 2011, but I went with the assumption that the data is reliable enough for this visualisation.
2. To determine the influence a country and its average annual precipitation has on a continental level, I had to take into account the surface area of each country. I added this information into the Excel file. For some countries this was a harder task than for others. Take Russia for example: it's partly in Europe and partly in Asia. So I had to take into account stuff like that too.
3. This surface area was almost always found in square kilometres, so I had to convert it to square metres by multiplying it by a million, so that it corresponds with the precipitation data, which is also expressed in square metres.
4. Next, I calculated the total amount of precipitation over the surface area of the country, by multiplying the surface area with the precipitation.
5. At last, I added up all country's surface area in square metres for each continent. Then, I added up the tot precipitation over the surface area of each country for each continent. The first got divided by the second, and the result was the average annual precipitation per continent.

This gave us the following results:

| | | | |
|---|---|---|---|
| Asia | 700,869564 | North-America | 681,705229 |
| Africa | 667,668628 | Oceania | 733,250091 |
| Europe | 729,058385 | South-America | 1597,2873 |

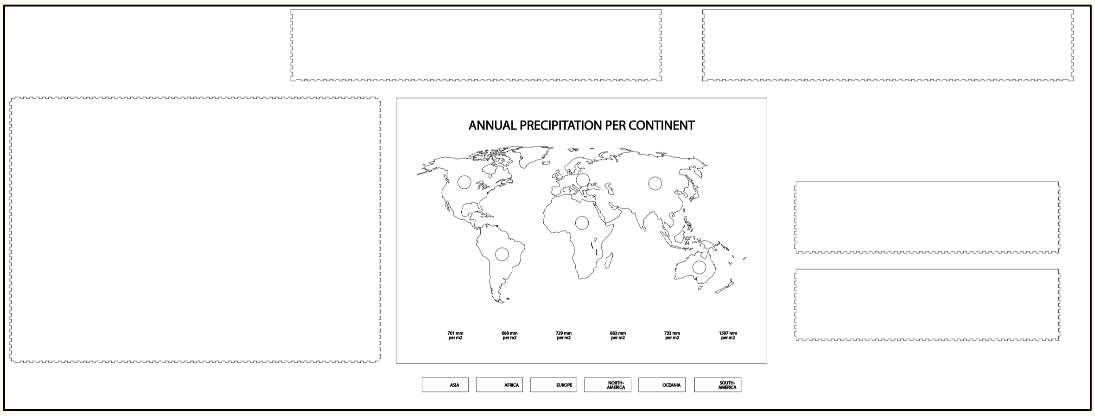
We can see that one of them stands out: South-America. This can be due to the fact South-America has a lot of rain forests. Surprisingly, Africa is very close to the rest of the continent (except for South-Africa, of course). But if we think about it, we know that Africa consist of very dry areas, like Egypt, but also areas with a lot of precipitation, like the Republic Of Congo and Gabon. These probably cancel out each other.

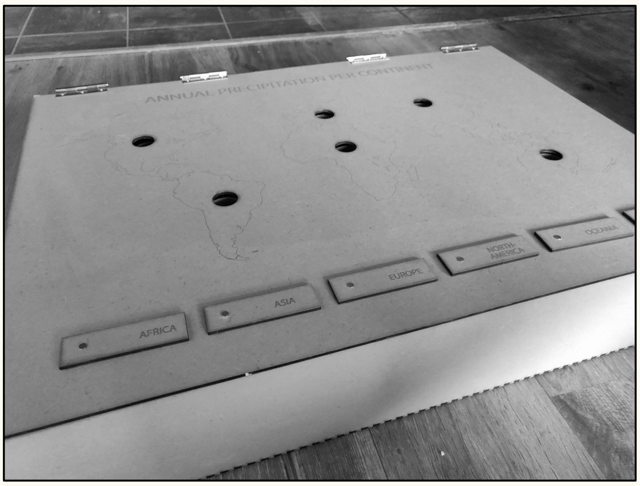
The map that can be found on Mecometer.com can further support this data:



## Building the visualisation

The world map and the box in which the cups will be hold (so that they aren't visible for the user), are made of wood, using the school's lasercutter. This box was made using http://www.makercase.com/ and Illustrator. The box is made waterproof using waterproof wood glue and acrylic varnish. To save some acrylic varnish, the inside of the box is covered with plastic. The world map on top of the box comes from https://simplemaps.com/resources/svg-world. The top of this wooden box can be opened. That way the cups can be taken out to fill them with water or empty them.



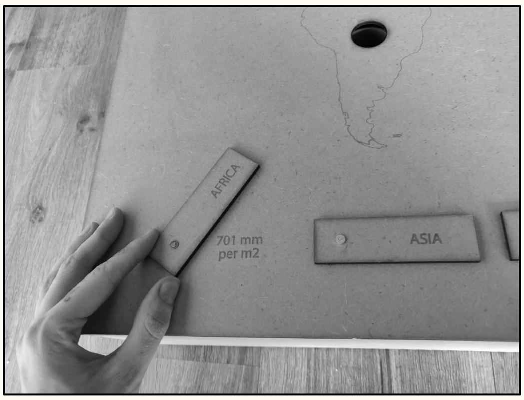



I decided to use the depth of the cup to visualise the data instead of the amount of water in millimetres it contains. The depth in millimetres with which the cup is filled, is calculated by manipulating the precipitation data. I multiplied each value by 0.65, divided it by 100 and rounded the result, so that the millimetres depths, except for South-America, are so that the user can touch the bottom of the cup with his finger. This way, the user can get a better picture of the data. It's sad that most of the data lies so close together the user probably can't feel the differences in depth very well between these continents, but the value of South-America is so high he can't touch the bottom of the cup these. That way, the user knows the value of South-America must be much higher than the rest of the continents. Because of this, the entire visualisation relies on the much higher value of South-America for explaining the data.

For the cups, I used PVC tubes. I sawed these by hand so I got the right length for the cup. I made it so that every cup is a bit longer than the centimetre value with which it will be filled with water. I did this to compensate the user putting his finger into the cup and the water level rising. The PVC-tubes got closed at the bottom with hard plastic and some hot glue. This way, the whole is waterproof.

The cups are filled with water using a stick with measurements on is, which show where the water level had to be per continent. The same stick is used by the user make an estimate of the cup's content.


The actual precipitation values on the bottom of the map aren't shown immediately, but have to be revealed underneath moving cards with the name of the continent on them. This way, the discovery aspect of the visualisation stays.

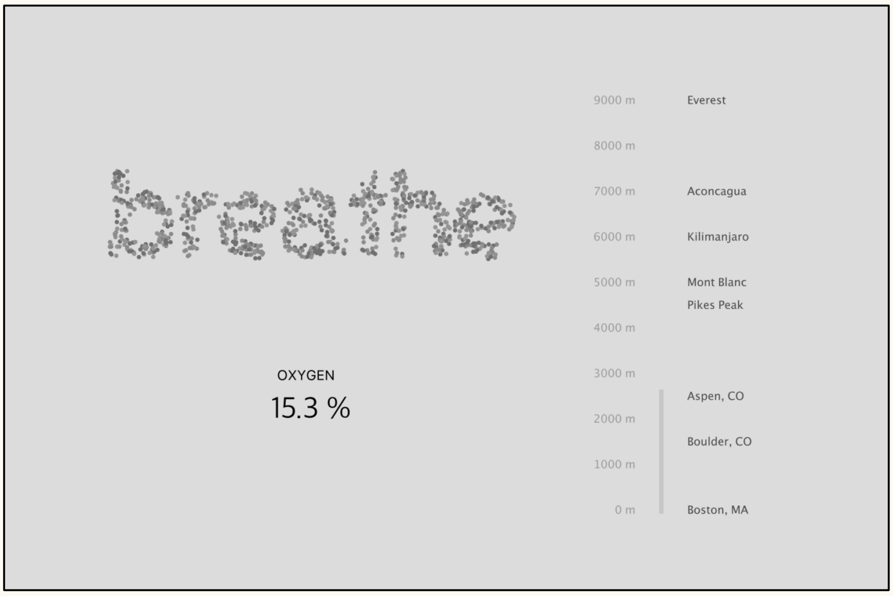



## Oxygen levels in the air

**EXPLANATORY — LINEAR — NODE-BASED**

### Visualisation

A video that shows how the amount of oxygen goes down the higher you go up.

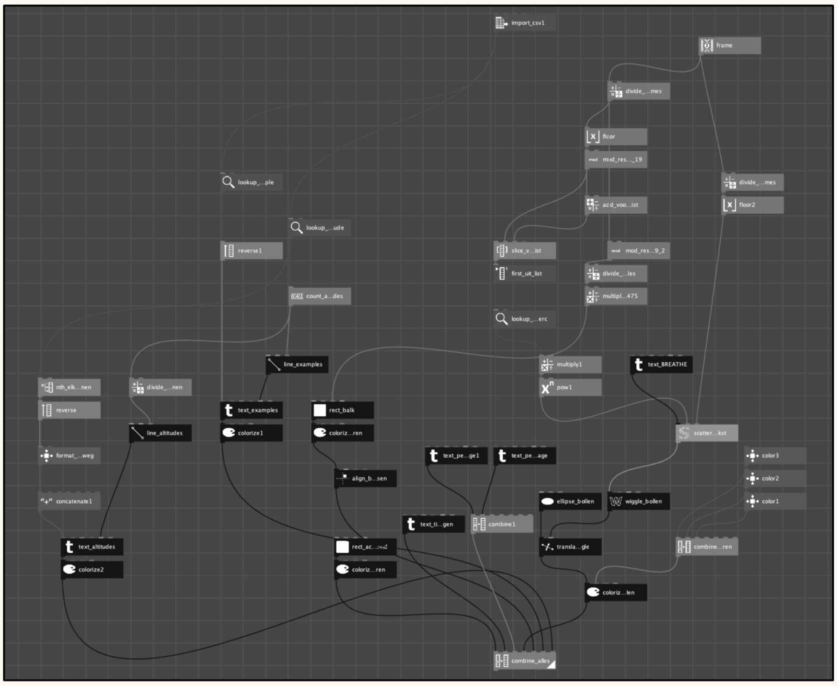


# *Process*

The data was found on the following website: https://www.higherpeak.com/altitudechart.html. It shows the altitude leven, the corresponding percentage of oxygen en some examples of what lies on that height.

I converted the data into an Excel file and saved this as a CSV file.

The visualisation was made using Nodebox and the CSV file. It shows the word “breathe”, which can be easily read in the beginning of the clip. The word is made of different little balls that are in relation with the percentage of oxygen. The starting altitude is 0 metres. The visualisation works its way to 9000 metres, and the word “breathe” gets more and more unreadable. Less and less little balls make up the word, because of course the amount of oxygen lowers.



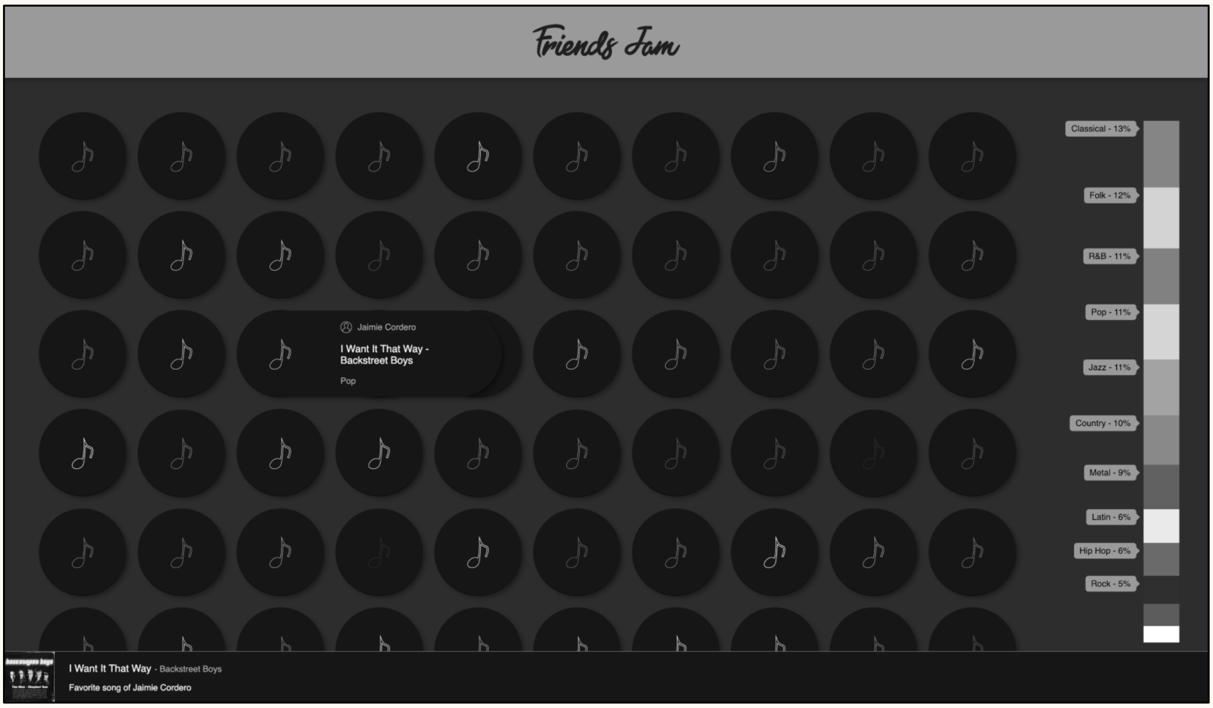
# The interests we share

## ‹Friends jam’: What music do your friends listen to?

**EXPLANATORY & EXPLORATORY – INTERACTIVE**

### Visualisation

A web application that is connected to your Spotify account and shows your friend’s taste in music. On the right, there is a chart showing the percentage of your friends that listen to the music genres that are shown. In the middle, there are a number of music notes. Each music note represents one of your friends. The music note corresponds with one of the music genres. When you hover over a music note, you get to see more info like your friend’s name, their favourite song in their most listened to genre, and the name of the genre. When you click on a note, the corresponding song begins to play.



# Process

It was quite a long process to get this visualisation working. There were a lot of steps involved, which were in complete chaos with each other a lot of the time, because the Spotify API isn't always that clear or up to date for that matter. A lot of function that were given by the API, had to be changed or rewritten because they simply don't work (anymore?).

But ultimately, it came down to this (in a perfect world):

1. Firstly, I had to figure out how to get a list of friends. The teacher advised me to work with a random generated list of friends. So I started coding in Javascript. I made a long list of names, which I randomly generated with http://listofrandomnames.com/index.cfm?textarea, because I simply can't think of that many names.
2. Then, I created a list of music genres. In Spotify, I created a playlist for each music genre (because via the API of Spotify, you can only access playlists or albums). These playlists have a special "uri", which is a sort of ID to access it via a HTTP-request. I saved these uri's with each genre in the array. Each genre also gets its own music note in the right color, and that color's RGB code.
3. I had to get all the songs that were in the playlists, so I wrote a function that does a HTTP-request for each genre and puts the songs in the genre array with the right genre.
4. An array of friends was created in which each name in the names list got a randomly chosen favourite genre and favourite song of that genre. At the same time, I counted how many times each genre was used to make calculations for the chart on the right of the webpage.
5. At the same time with all previous stepts, I figured out the Spotify API. There were a few different HTTP-requests I had to figure out and rewrite. Not everything was equally logical in the API. I put all the spotify-API-stuff is a separate object to divide that logic from the rest of the code and make it manageable. I have to say, now that it's all written and working, the code looks easy and logical. I'm sorry that wasn't actually the case.
6. I created the actual layout and made the app work!

The Spotify API that I used: https://developer.spotify.com/console/

IMPOTANT NOTE:

> The web application doesn't work without refreshing the access token each hour. The access token can be acquired on this URL https://developer.spotify.com/documentation/web-playback-sdk/quick-start/#source-code under "Authenticating with Spotify" and can only be obtained with a paying Spotify account. The token needs to be changed in the file "spotify.js" at the top of the code after "token:".

## Decreasing Arctic Sea Ice Volume

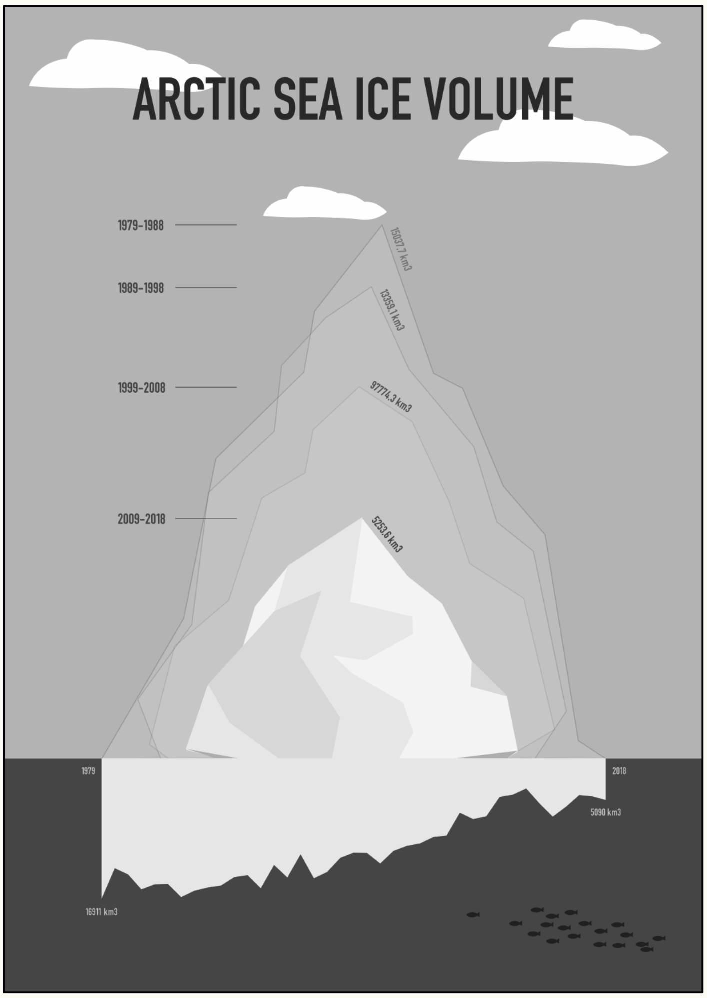
EXPLANATORY – STATIC

### Visualisation

A poster showing in 2 ways the decreasing arctic sea ice volume.

Above the water, multiple icebergs are shown. Each iceberg shows the average volume over ten years. The years with which it corresponds are shown next to the iceberg.

Underneath the water, there is a chart that shows the actual values of each September month over a period of 40 years. Sometimes the volume increases a bit again, but if you look at the whole picture, you can see the ice volume has decreased significantly between 1979 and 2018.



# Process

De source of this data is http://psc.apl.uw.edu/research/projects/arctic-sea-ice-volume-anomaly/data. The actual data textfile can be found on http://psc.apl.uw.edu/wordpress/wp-content/uploads/schweiger/ice_volume/PIOMAS.2sst.monthly.Current.v2.1.txt.

The volume data is available for every month of the year, but I based my visualisation on the data of the September months, because September is the month on which every year the lowest point is reached. I put the data in an Excel file and calculated everything I needed for the visualisation.

These icebergs are made using Illustrator and a script that calculates how much volume a shape area has in Illustrator. First, I drew the biggest iceberg. I took its shape area volume, and put it in relation to the actual ice volume data for those ten years. Next, I drew the other icebergs based on their percentage against the biggest iceberg. I kept running the script and changing the iceberg a bit, until it was almost exactly the same value of shape area volume as it should be. The script I used is https://gist.github.com/nanoSpawn/8e8fca48d0fb85dd961e.

The chart underneath the waterline has been but upside down to make it visually more interesting.