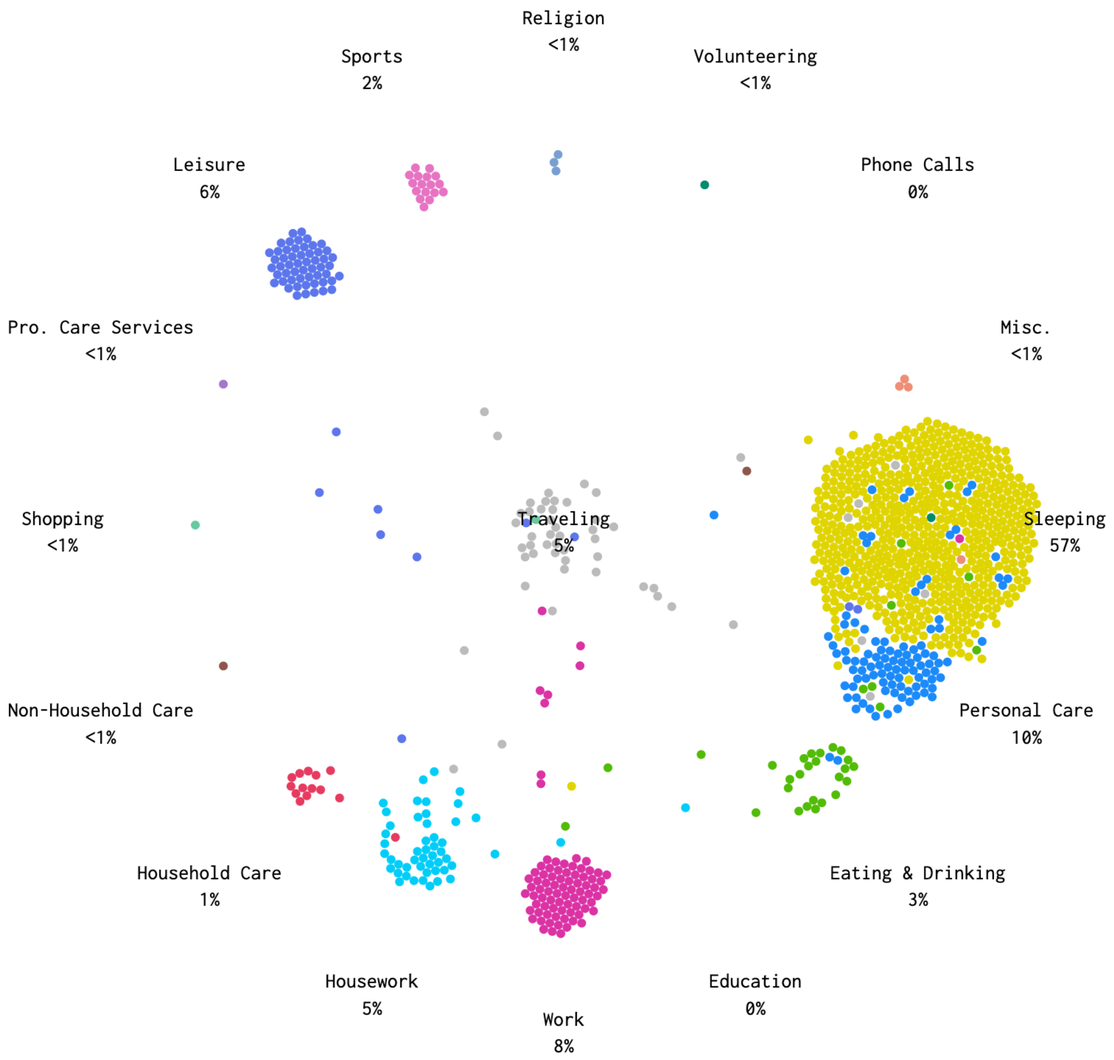
6:21am

SLOW | MEDIUM | FAST

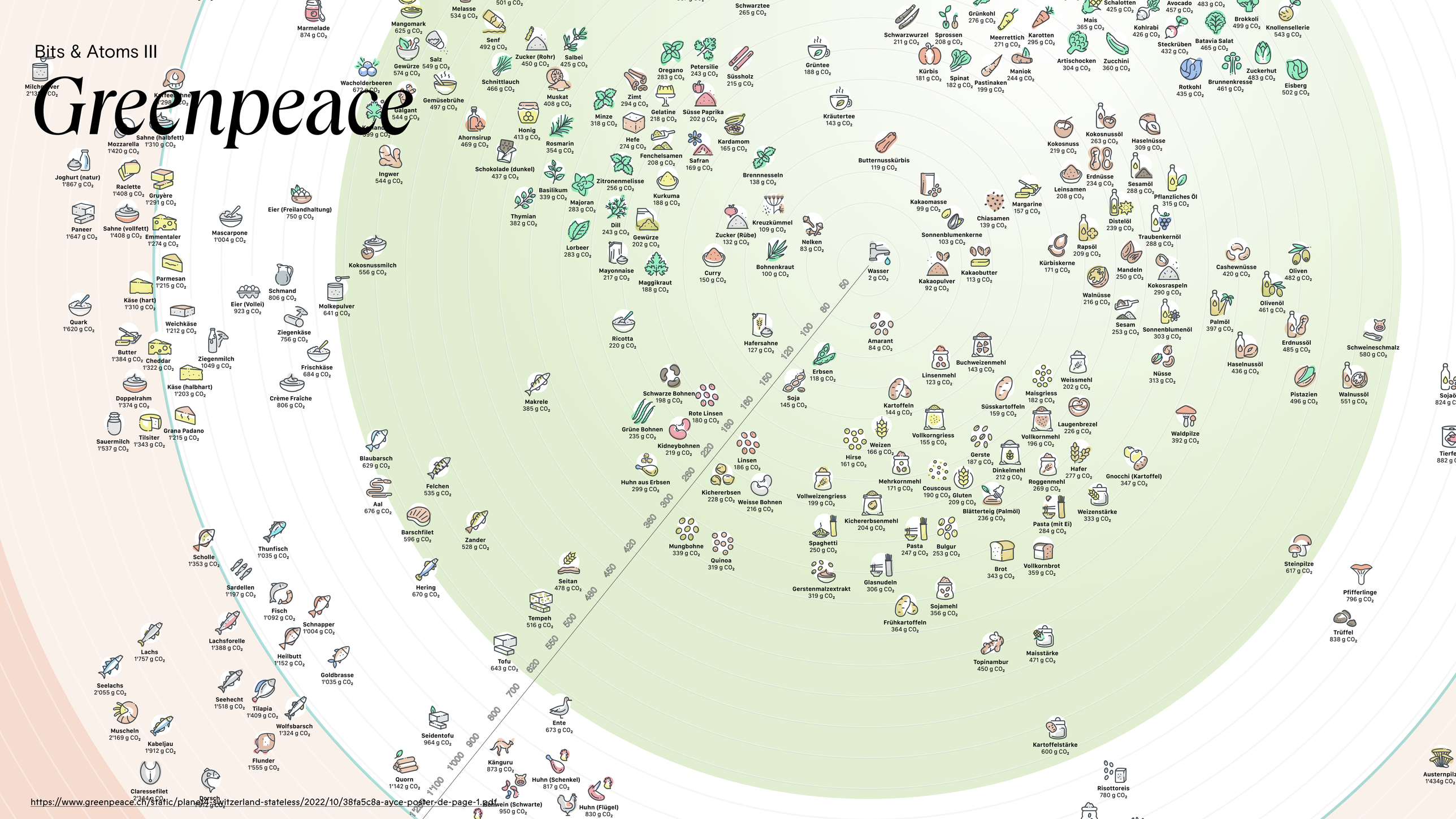


This is a simulation of 1,000 people's average day. It's based on 2014 data from the American Time Use Survey, made way more accessible by the ATUS Extract Builder.

https://flowingdata.com/2015/12/15/a-day-in-the-life-of-americans/

Bits & Atoms III

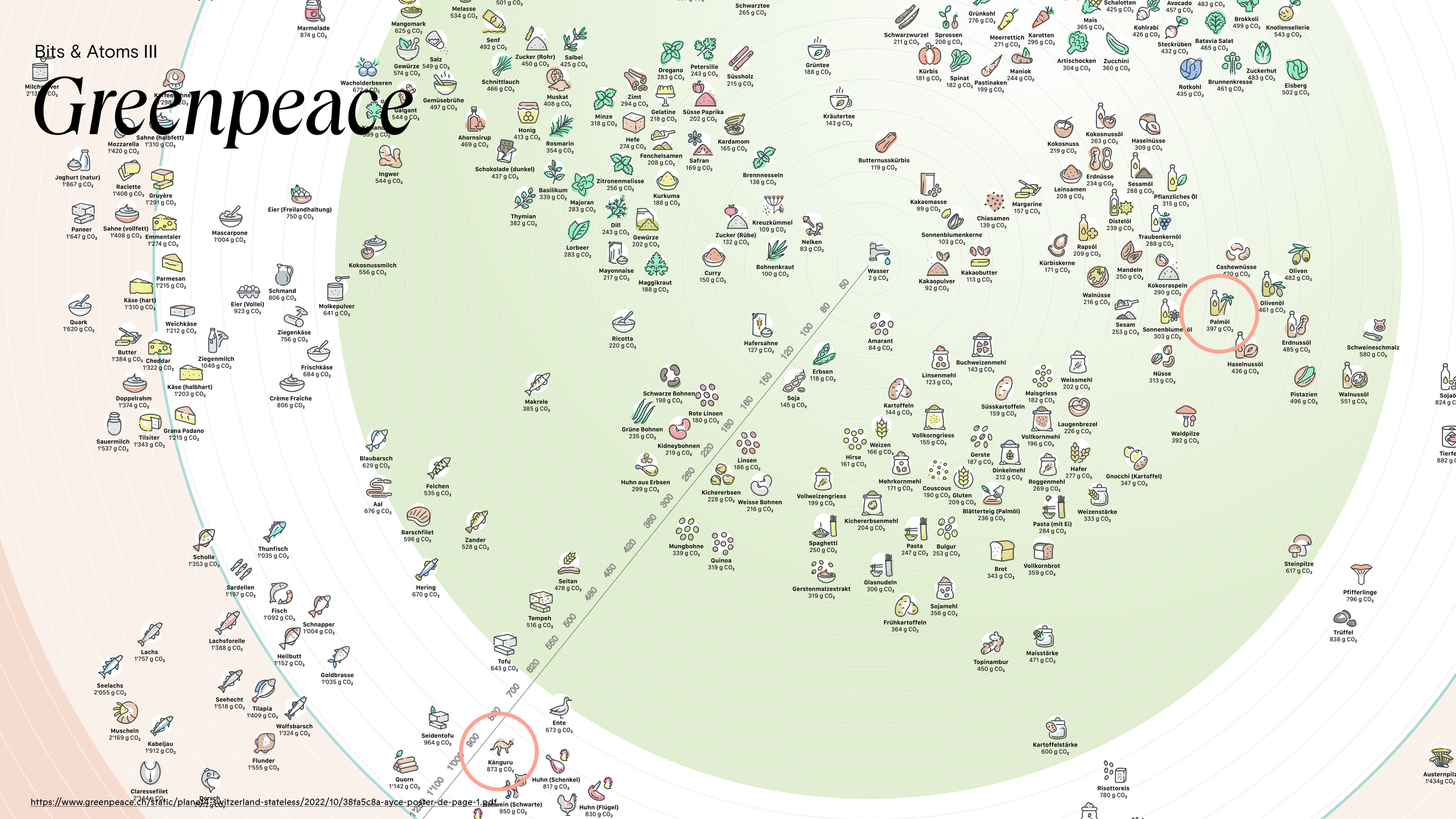
Greenpeace



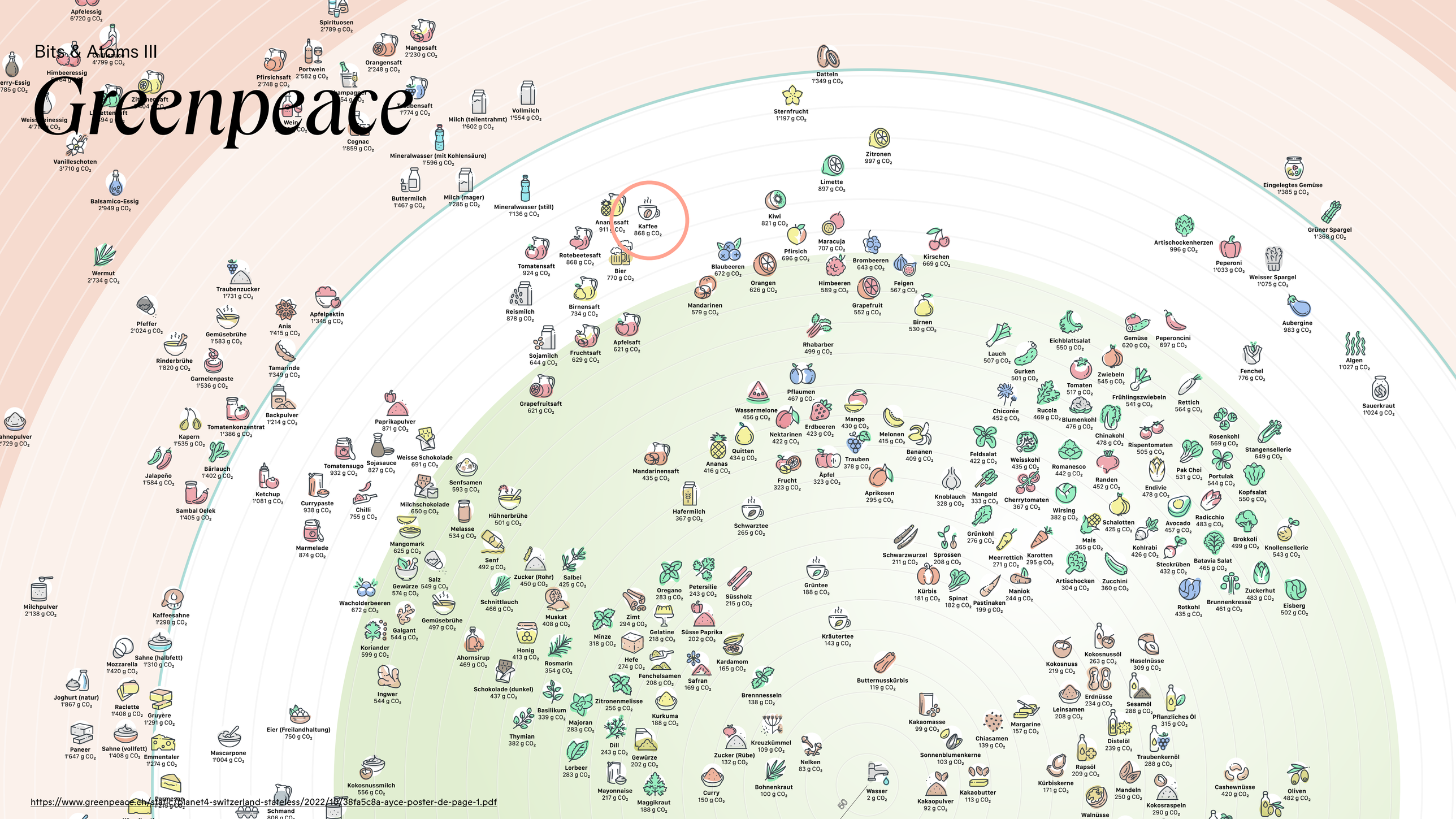
https://www.greenpeace.ch/static/planet4-switzerland-stateless/2022/10/38fa5c8a-ayce-poster-de-page-1.pdf

Bits & Atoms III

Greenpeace



https://www.greenpeace.ch/static/planet/a-switzerland-stateless/2022/10/38fa5c8a-ayce-poster-de-page-1.pdf

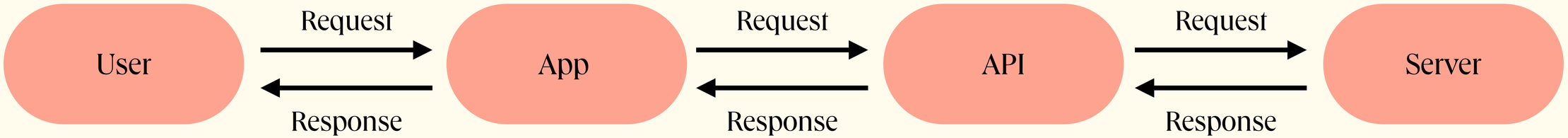

Bits & Atoms III
Greenpeace
Apfelessig 6'720 g CO2
Spirituosen 2'789 g CO2
Himbeeressig 4'799 g CO2
Mangosaft 2'230 g CO2
Orangensaft 2'248 g CO2
Pfirsichsaft 2'748 g CO2
Portwein 2'582 g CO2
Traubensaft 1'774 g CO2
Milch (teilentrahmt) 1'602 g CO2
Vollmilch 1'554 g CO2
Cognac 1'859 g CO2
Mineralwasser (mit Kohlensäure) 1'596 g CO2
Vanilleschoten 3'710 g CO2
Balsamico-Essig 2'949 g CO2
Buttermilch 1'467 g CO2
Milch (mager) 1'285 g CO2
Mineralwasser (still) 1'136 g CO2
Datteln 1'349 g CO2
Sternfrucht 1'197 g CO2
Zitronen 997 g CO2
Limette 897 g CO2
Kiwi 821 g CO2
Pfirsich 696 g CO2
Maracuja 707 g CO2
Kaffee 868 g CO2
Ananassaft 911 g CO2
Rotebeetesaft 868 g CO2
Tomatensaft 924 g CO2
Bier 770 g CO2
Blaubeeren 672 g CO2
Orangen 626 g CO2
Brombeeren 643 g CO2
Himbeeren 589 g CO2
Kirschen 669 g CO2
Feigen 567 g CO2
Grapefruit 552 g CO2
Birnen 530 g CO2
Mandarinen 579 g CO2
Reismilch 878 g CO2
Birnensaft 734 g CO2
Sojamilch 644 g CO2
Fruchtsaft 629 g CO2
Apfelsaft 621 g CO2
Grapefruitsaft 621 g CO2
Rhabarber 499 g CO2
Wermut 2'734 g CO2
Traubenzucker 1'731 g CO2
Apfelpektin 1'345 g CO2
Pfeffer 2'024 g CO2
Anis 1'415 g CO2
Gemüsebrühe 1'583 g CO2
Rinderbrühe 1'820 g CO2
Tamarinde 1'349 g CO2
Garnelenpaste 1'536 g CO2
Backpulver 1'214 g CO2
Tomatenkonzentrat 1'386 g CO2
Kapern 1'535 g CO2
Jalapeño 1'584 g CO2
Bärlauch 1'402 g CO2
Sambal Oelek 1'405 g CO2
Ketchup 1'081 g CO2
Currypaste 938 g CO2
Chilli 755 g CO2
Marmelade 874 g CO2
Paprikapulver 871 g CO2
Tomatensugo 932 g CO2
Sojasauce 827 g CO2
Weisse Schokolade 691 g CO2
Senfsamen 593 g CO2
Milchschokolade 650 g CO2
Hühnerbrühe 501 g CO2
Melasse 534 g CO2
Mangomark 625 g CO2
Senf 492 g CO2
Zucker (Rohr) 450 g CO2
Salbei 425 g CO2
Salz 549 g CO2
Gewürze 574 g CO2
Wacholderbeeren 672 g CO2
Schnittlauch 466 g CO2
Muskat 408 g CO2
Gemüsebrühe 497 g CO2
Galgant 544 g CO2
Koriander 599 g CO2
Honig 413 g CO2
Ahornsirup 469 g CO2
Schokolade (dunkel) 437 g CO2
Rosmarin 354 g CO2
Ingwer 544 g CO2
Basilikum 339 g CO2
Thymian 382 g CO2
Majoran 283 g CO2
Zitronenmelisse 256 g CO2
Lorbeer 283 g CO2
Dill 243 g CO2
Gewürze 202 g CO2
Mayonnaise 217 g CO2
Maggikraut 188 g CO2
Kokosnussmilch 556 g CO2
Eier (Freilandhaltung) 750 g CO2
Mascarpone 1'004 g CO2
Milchpulver 2'138 g CO2
Kaffeesahne 1'298 g CO2
Mozzarella 1'420 g CO2
Sahne (halbfett) 1'310 g CO2
Joghurt (natur) 1'867 g CO2
Raclette 1'408 g CO2
Gruyère 1'291 g CO2
Paneer 1'647 g CO2
Sahne (vollfett) 1'408 g CO2
Emmentaler 1'274 g CO2
Parmesan 1'218 g CO2
Schmand 806 g CO2
Mandarinensaft 435 g CO2
Hafermilch 367 g CO2
Schwarztee 265 g CO2
Ananas 416 g CO2
Quitten 434 g CO2
Wassermelone 456 g CO2
Nektarinen 422 g CO2
Pflaumen 467 g CO2
Erdbeeren 423 g CO2
Mango 430 g CO2
Melonen 415 g CO2
Bananen 409 g CO2
Frucht 323 g CO2
Äpfel 323 g CO2
Trauben 378 g CO2
Aprikosen 295 g CO2
Oregano 283 g CO2
Petersilie 243 g CO2
Süssholz 215 g CO2
Zimt 294 g CO2
Gelatine 218 g CO2
Süsse Paprika 202 g CO2
Minze 318 g CO2
Hefe 274 g CO2
Fenchelsamen 208 g CO2
Safran 169 g CO2
Kardamom 165 g CO2
Brennnesseln 138 g CO2
Kurkuma 188 g CO2
Kreuzkümmel 109 g CO2
Zucker (Rübe) 132 g CO2
Nelken 83 g CO2
Curry 150 g CO2
Bohnenkraut 100 g CO2
Grüntee 188 g CO2
Kräutertee 143 g CO2
Butternusskürbis 119 g CO2
Wasser 2 g CO2
Kakaomasse 99 g CO2
Sonnenblumenkerne 103 g CO2
Kakaopulver 92 g CO2
Kakaobutter 113 g CO2
Chiasamen 139 g CO2
Margarine 157 g CO2
Eingelegtes Gemüse 1'385 g CO2
Grüner Spargel 1'368 g CO2
Artischockenherzen 996 g CO2
Peperoni 1'033 g CO2
Weisser Spargel 1'075 g CO2
Aubergine 983 g CO2
Algen 1'027 g CO2
Sauerkraut 1'024 g CO2
Fenchel 776 g CO2
Lauch 507 g CO2
Eichblattsalat 550 g CO2
Gemüse 620 g CO2
Peperoncini 697 g CO2
Gurken 501 g CO2
Zwiebeln 545 g CO2
Tomaten 517 g CO2
Frühlingszwiebeln 541 g CO2
Chicorée 452 g CO2
Rucola 469 g CO2
Blumenkohl 476 g CO2
Rettich 564 g CO2
Chinakohl 478 g CO2
Rispentomaten 505 g CO2
Rosenkohl 569 g CO2
Stangensellerie 649 g CO2
Feldsalat 422 g CO2
Weisskohl 435 g CO2
Romanesco 442 g CO2
Pak Choi 531 g CO2
Portulak 544 g CO2
Randen 452 g CO2
Endivie 478 g CO2
Kopfsalat 550 g CO2
Knoblauch 328 g CO2
Mangold 333 g CO2
Cherrytomaten 367 g CO2
Wirsing 382 g CO2
Radicchio 483 g CO2
Schalotten 425 g CO2
Avocado 457 g CO2
Brokkoli 499 g CO2
Grünkohl 276 g CO2
Meerrettich 271 g CO2
Karotten 295 g CO2
Mais 365 g CO2
Kohlrabi 426 g CO2
Knollensellerie 543 g CO2
Schwarzwurzel 211 g CO2
Sprossen 208 g CO2
Steckrüben 432 g CO2
Batavia Salat 465 g CO2
Kürbis 181 g CO2
Spinat 182 g CO2
Pastinaken 199 g CO2
Maniok 244 g CO2
Artischocken 304 g CO2
Zucchini 360 g CO2
Zuckerhut 483 g CO2
Rotkohl 435 g CO2
Brunnenkresse 461 g CO2
Eisberg 502 g CO2
Kokosnuss 219 g CO2
Kokosnussöl 263 g CO2
Haselnüsse 309 g CO2
Erdnüsse 234 g CO2
Leinsamen 208 g CO2
Sesamöl 288 g CO2
Pflanzliches Öl 315 g CO2
Distelöl 239 g CO2
Traubenkernöl 288 g CO2
Rapsöl 209 g CO2
Kürbiskerne 171 g CO2
Mandeln 250 g CO2
Kokosraspeln 290 g CO2
Walnüsse 182 g CO2
Cashewnüsse 420 g CO2
Oliven 482 g CO2
https://www.greenpeace.ch/static/planet4-switzerland-stateless/2022/10/38fa5c8a-ayce-poster-de-page-1.pdf

API

- **API** (Application Programming Interface) is an interface through which one application can access the services of another.The API acts as a middleman between any two machines that want to connect with each other for a specified task.

API

- Advantage: Abstraction of functionality between systems.



REST

- Representational State Transfer (REST)
- REST defines certain specific operations that applications should be able to do in order to satisfy all of the CRUD (create, read, update, delete) requirements.

«REST API»

- RESTful API is an Application-Programmer Interface for communicating with an application across a network using HTTP methods.
- Not all HTTP APIs are REST APIs but all REST APIs are HTTP APIs.


```
{
  "response_code": 0,
  "results": [
    {
      "category": "Science: Computers",
      "type": "multiple",
      "difficulty": "medium",
      "question": "In the programming language &quot;Python&quot;, which of these statements would display the string &quot;Hello World&quot; correctly?",
      "correct_answer": "print(&quot;Hello World&quot;)",
      "incorrect_answers": [
        "console.log(&quot;Hello World&quot;)",
        "echo &quot;Hello World&quot;",
        "printf(&quot;Hello World&quot;)"
      ]
    }
  ]
}
```


https://opentdb.com/api.php?amount=1&category=18

«REST API»

- There are four parts of a REST API request
- **URI** (Uniform Resource Identifier): the URL address, also known as an «endpoint»
- **HTTP**: PUT, POST, GET, DELETE methods
- **Headers**: include authentication tokens, define the data format of the response, impose rate limits
- **Body**: (payload) the actual part of the request

https://opentdb.com/api.php?amount=1&category=18

Fetching data from an API

```
const requestOptions = {
  method: 'GET',
  body: formdata,
  redirect: 'follow',
  headers: { authentication: 'Basic' }
};


fetch(url, requestOptions)
  .then(function(response) {
    return response.text();
  })
  .then(function(text) {
    console.log('Request successful', text);
  })
  .catch(function(error) {
    log('Request failed', error)
  });

```

More: https://developer.mozilla.org/en-US/docs/Web/API/fetch

Exercise:

- Get future_cities_data.csv from email
- Transform .csv into .json

Stack

```
1   function multiply(a, b) {
2       return a * b
3
4   }
5
6   function square(n) {
7       return multiply(n, n)
8
9   }
10
11  function printSquare(n) {
12      console.log(square(n))
13
    }

    printSquare(4)
```

Single threaded JavaScript

- JavaScript is normally «single threaded», meaning we have only one stack running
- … but …

AJAX

- **AJAX** (Asynchronous JavaScript and XML) is a technique for communicating with a server and dynamically altering a page without leaving the page. It is made possible with the XMLHttpRequest object, a built-in feature of a web-browser.

Callback

- A callback(-function) can be passed as a parameter.
- Useful when working with asynchronous functions, because we can tell the programm what is to be executed whenever the first task has completed.

p5.js and AJAX

- loadStrings - loads .txt files
- loadJSON - loads .json files
- loadXML - loads .xml files
- loadTable - loads .csv files

loadTable()

```
const table = loadTable(
  filename,
  [extension],
  [header],
  [callback],
  [errorCallback]
);
```

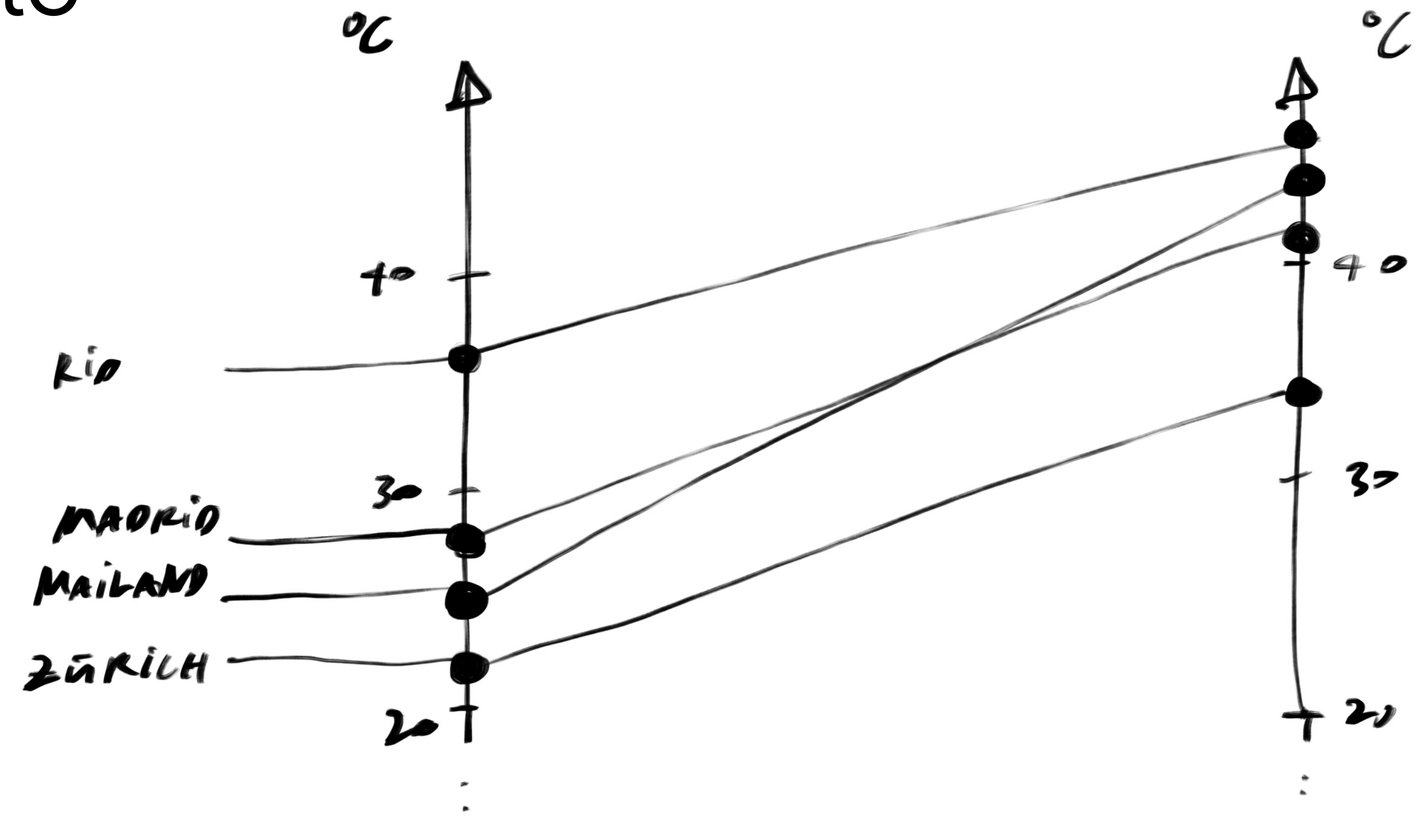
Exercise:

1. Try to find p5 methods that
   - Count number of rows
   - Get the value of a specific cell
   - Get all values from a specific column
2. Load city table (.csv) into your sketch.

current_city	selfdissim_current_future	future_city_1_source	Annual_Mean_Temperature	future_Annual_Mean_Temperatu
Budapest	1.894243591	Skopje	11.23333359	14.26666667
Milan	1.697220278	Dallas	13.38333321	15.86666667
Tbilisi	1.598058111	Bishkek	12.86250019	15.73333333
Rostov	1.425731006	Skopje	9.729166985	12.66666667
Konya	1.429877977	Tashkent	11.13749981	13.93333333
Skopje	1.479205099	Austin	12.60833359	14.96666667
Berlin	1.473413669	Canberra	9.858333588	11.63333333
Paris	1.714245199	Canberra	12.1875	13.63333333
Seattle	1.283094458	San Francisco	11.03750038	13.6
Minsk	1.270259119	Sofia	6.329166889	9.166666667
Athens	1.033231598	Fez	17.86666679	20.36666667

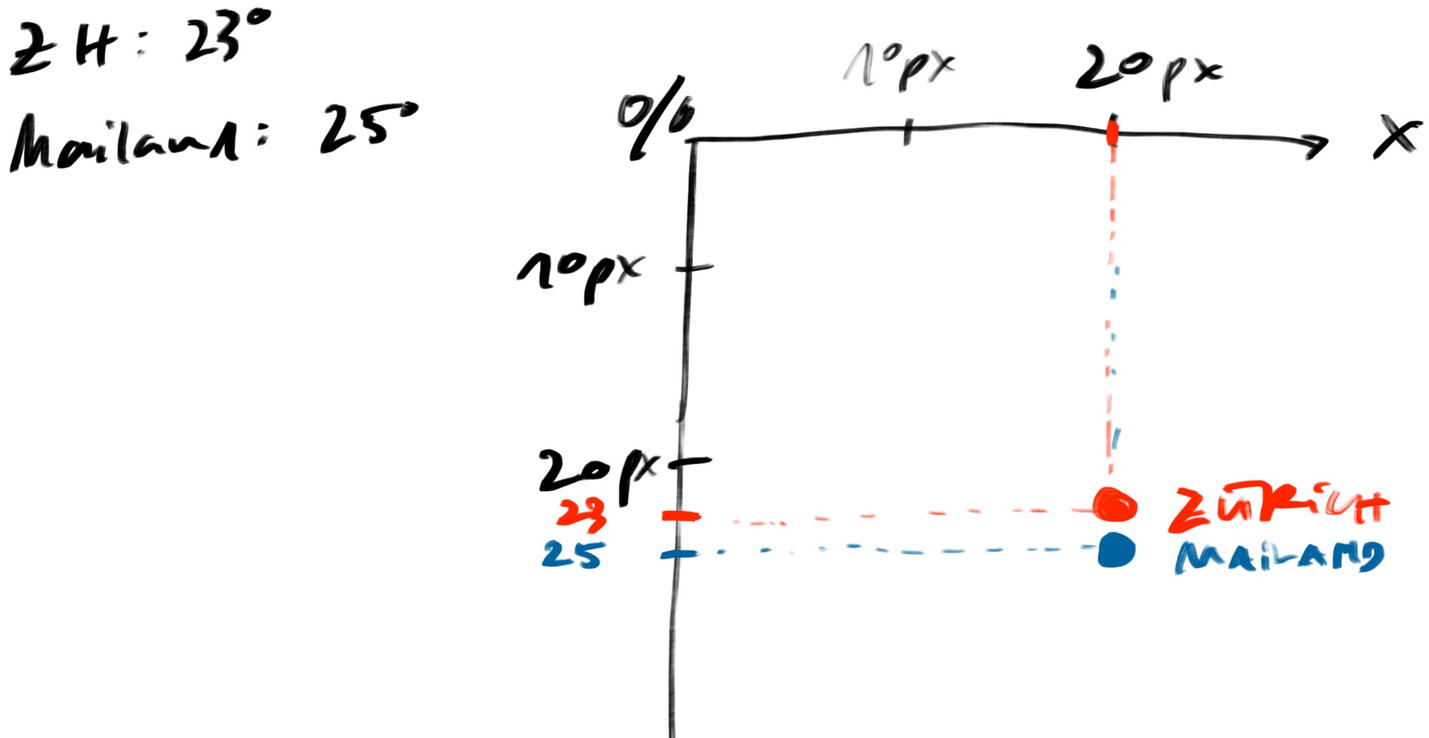
Exercise: Display today/future

- Break down task into single steps



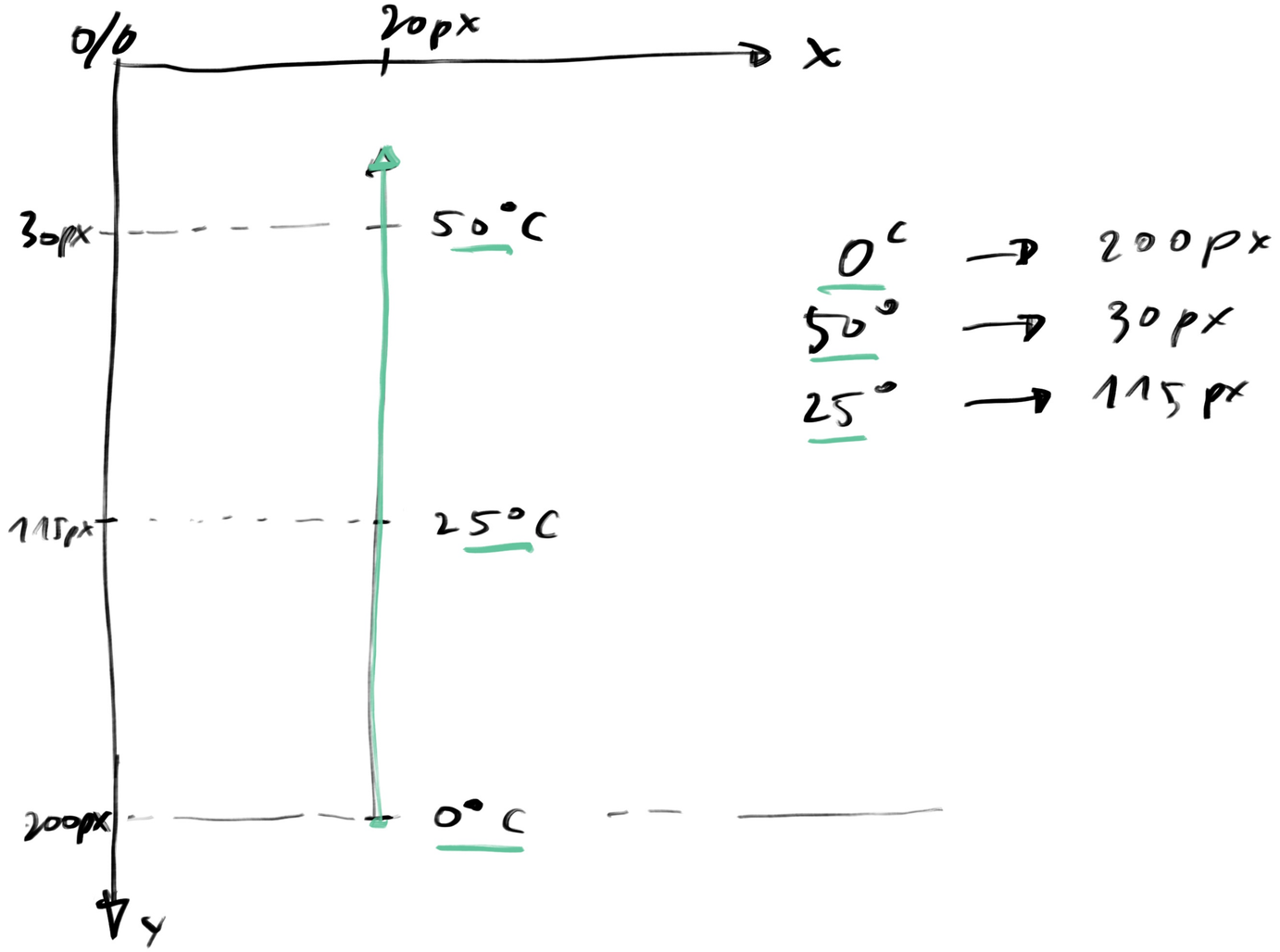
Problem: Get the axis right

- Try to find a solution how to solve this problem: Y-position in relation to temperature?



Map()

- Wee need to “map” the Celsius range to a position range (y) that fits our needs.



Map()

Description

Re-maps a number from one range to another.

In the first example above, the number 25 is converted from a value in the range of 0 to 100 into a value that ranges from the left edge of the window (0) to the right edge (width).

Syntax

```
map(value, start1, stop1, start2, stop2,
[withinBounds])
```

Parameters

value	Number: the incoming value to be converted
start1	Number: lower bound of the value's current range
stop1	Number: upper bound of the value's current range
start2	Number: lower bound of the value's target range
stop2	Number: upper bound of the value's target range

https://p5js.org/reference/#/p5/map

Map()

```
function convertDegreesToPosition(temp) {
  // we need to map the temperatures to a new scale
  // 0° = 200px, 25° = 115px, 50° = 30px
  // https://p5js.org/reference/#/p5/map
  const position = map(temp, 0, 50, 200, 30);
  return position;
}
```

https://p5js.org/reference/#/p5/map

Exercise

- Draw mapped data points on to y-axis
- Add labels for each data point:
  - City name
  - Temperature

https://p5js.org/reference/#/p5/map