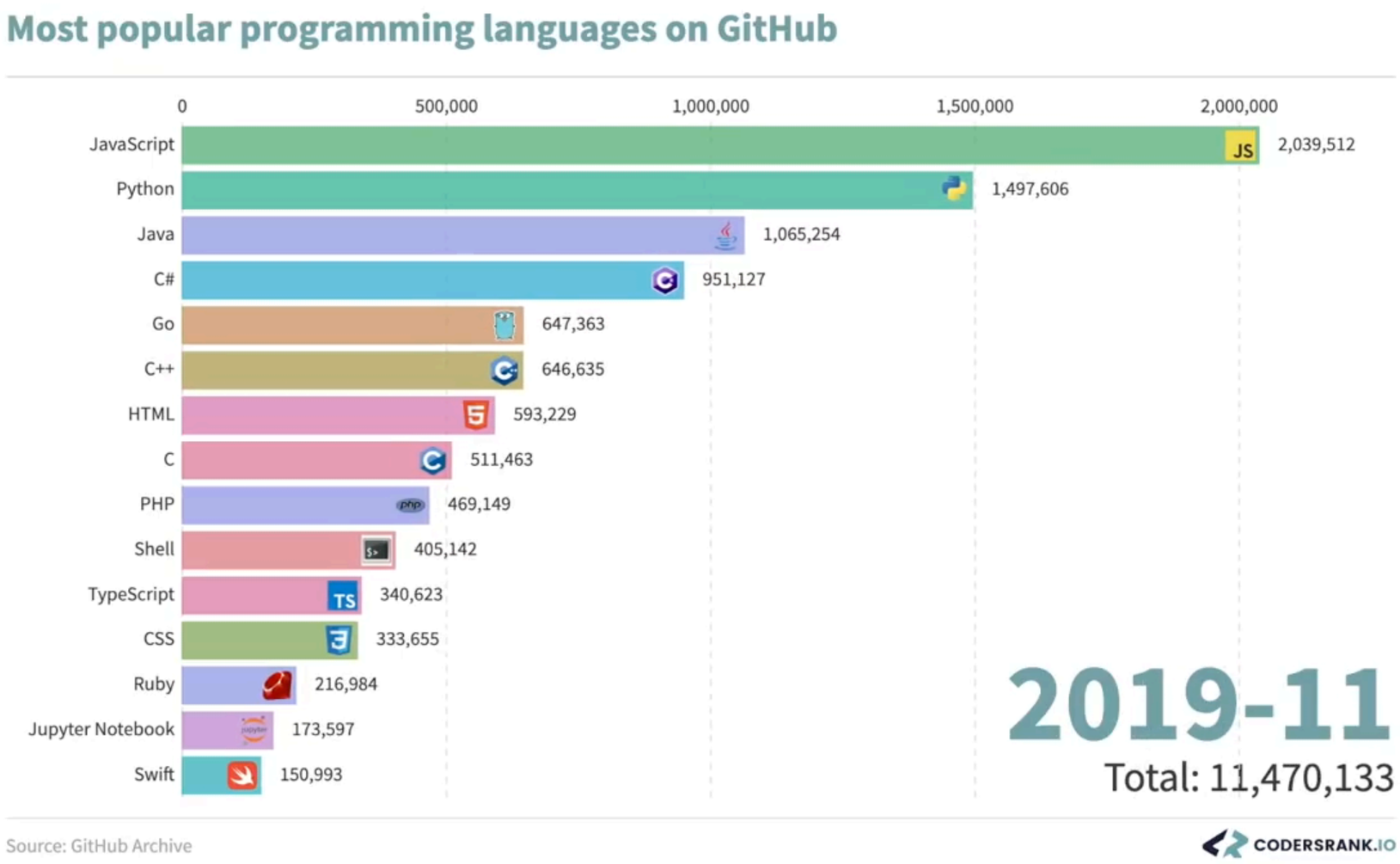
In case of

`java.lang.IllegalArgumentException`



From Java To JavaScript

Computer Science

kea
KØBENHAVNS ERHVERVSAKADEMI

The Purpose

Introduce you to Object-Oriented JavaScript (based on your knowledge of Object-Oriented Java).

Work with JavaScript concepts like classes, objects, arrays, loops & DOM Manipulation.

Last time: Introduction to JavaScript, browser environment, the console and its development tools

Person Objects

http://127.0.0.1:5501/class-person/index.html

Person Objects

Birgitte Kirk Iversen

bki@eaaa.dk
Birthdate: 1966-01-14
Age: 56 years old

Martin Aagaard Nøhr

mnor@eaaa.dk
Birthdate: 1989-05-02
Age: 33 years old

Rasmus Cederdorff

race@eaaa.dk
Birthdate: 1990-09-15
Age: 31 years old

top

Filter

Default levels

1 Issue: 1

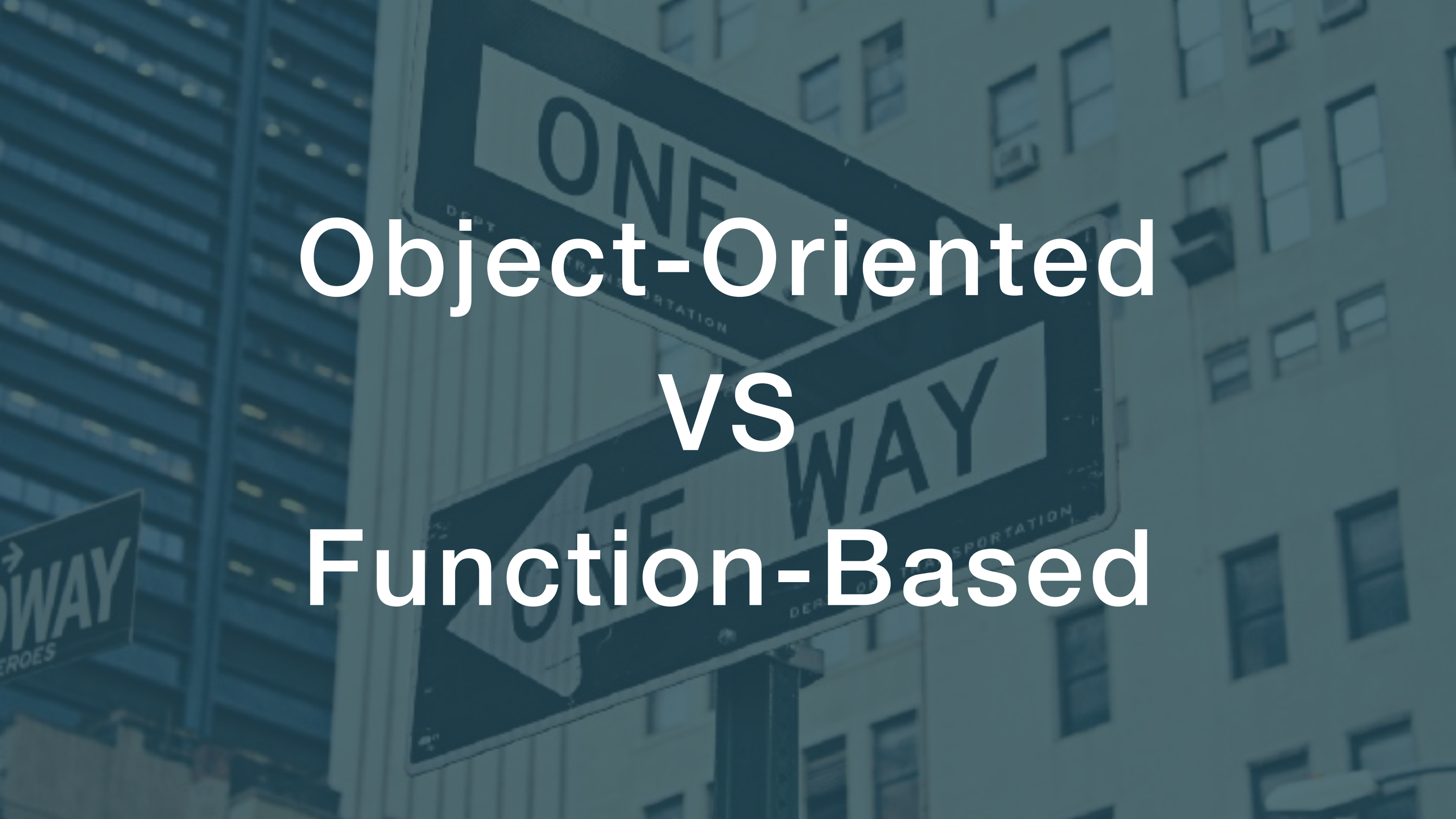
main.js:61

```
(3) [Person, Person, Person]
  0: Person
    birthdate: "1966-01-14"
    img: "https://www.eaaa.dk/mec
    mail: "bki@eaaa.dk"
    name: "Birgitte Kirk Iversen'
    [[Prototype]]: Object
  1: Person
    birthdate: "1989-05-02"
    img: "https://www.eaaa.dk/mec
    mail: "mnor@eaaa.dk"
    name: "Martin Aagaard Nøhr"
    [[Prototype]]: Object
  2: Person
    birthdate: "1990-09-15"
    img: "https://www.eaaa.dk/mec
    mail: "race@eaaa.dk"
    name: "Rasmus Cederdorff"
    [[Prototype]]: Object
  length: 3
  [[Prototype]]: Array(0)
```

class-person

AGENDA

```
const agenda = [
    "It's all Objects & Arrays",
    "Classes in JavaScript",
    "Working with Classes, Objects & Arrays",
    "JS Concepts & DOM Manipulation"
];
```



Object-Oriented VS Function-Based

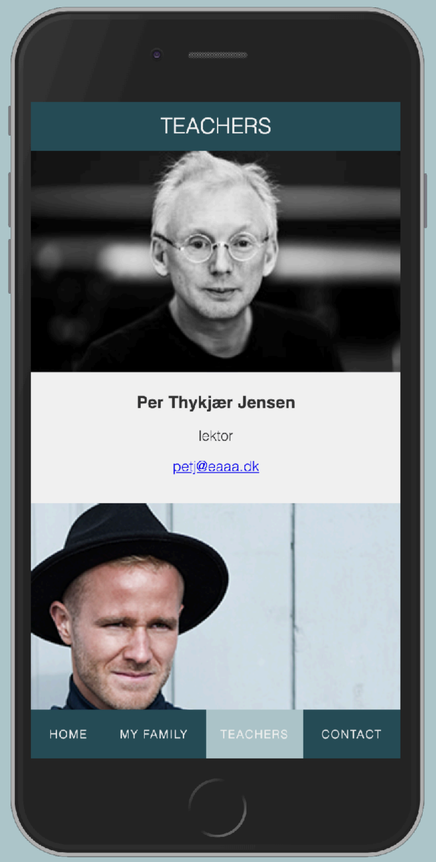

Objects & Arrays
Fundamental Programming Knowledge
MacBook Air

Web Development

WEB APP

API

DATA SOURCE



FETCH(...)

REQUEST

JSON DATA

RESPONSE

JSON

Objects

A set of named values

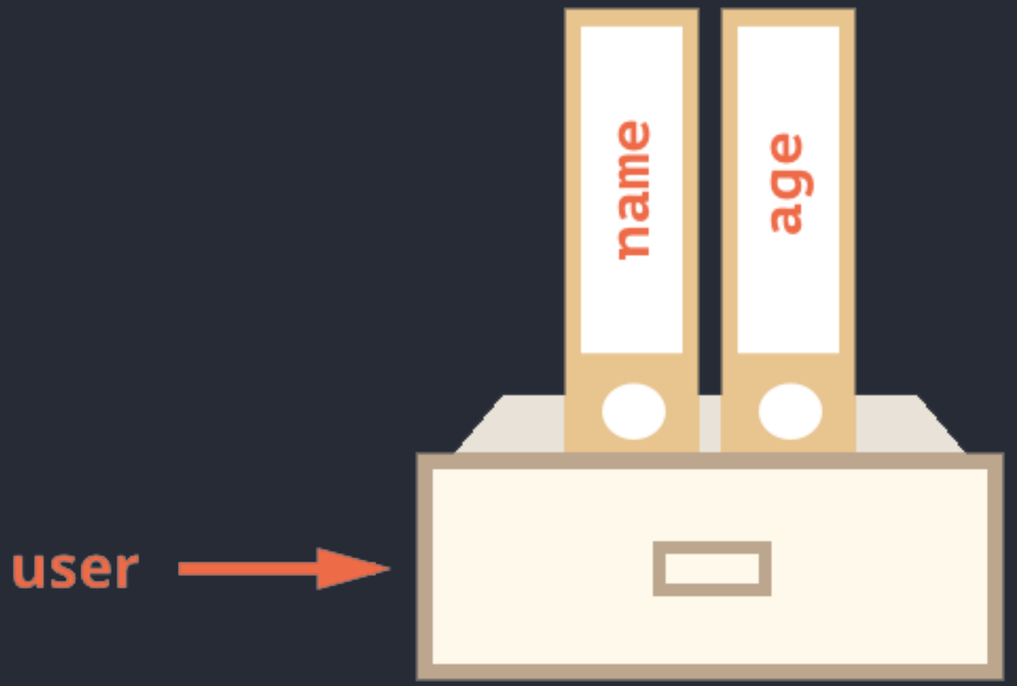
Objects are used to store keyed collections of various data

Containers for named values called properties. A property is a “`key: value`” pair

Objects

A set of named values

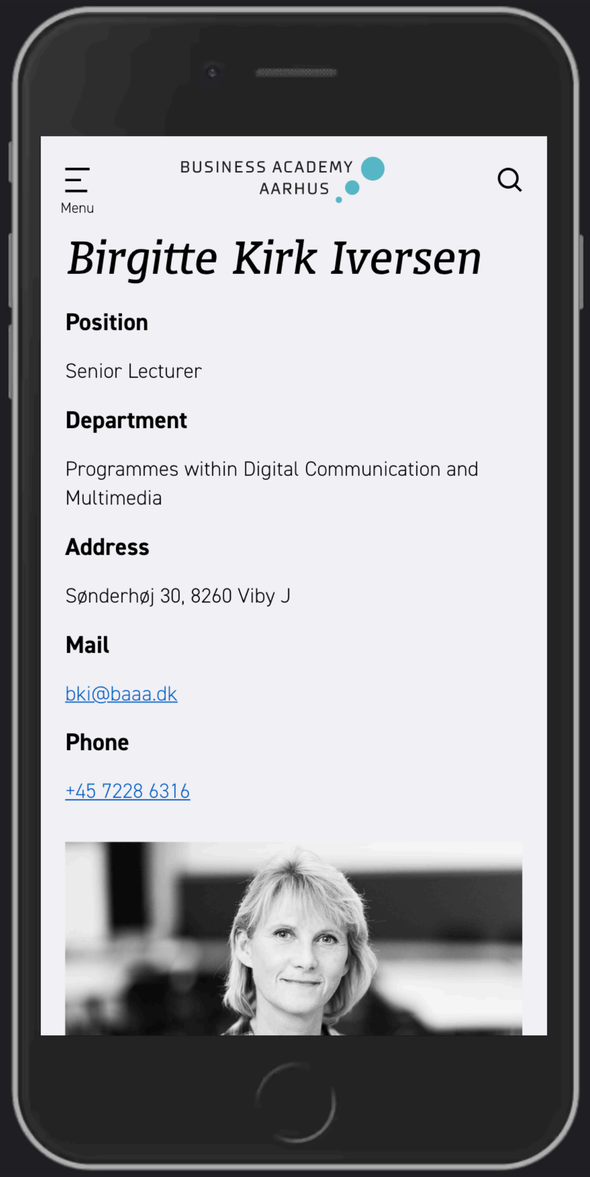
```
let user = {
  name: 'Alicia',
  age: 6
};

console.log(user.name +
  " is " + user.age +
  " years old.");
```




```
Alicia is 6 years old.          main.js:11
>
```

Objects

A set of named values



property

value

```
let person = {
    name: "Birgitte Kirk Iversen",
    position: "Senior Lecturer",
    department: "Programmes within Digital Communication and Multimedia",
    address: "Sønderhøj 30, 8260 Viby J",
    mail: "bki@baaa.dk",
    phone: "+45 7228 6316",
    image: "https://www.baaa.dk/media/u4gorzsd/birgitte-kirk-iversen2.jpg"
};
```

https://www.baaa.dk/contact/find-employee/employee/birgitte-kirk-iversen

netflix.com

NETFLIX

Hvem ser?

Personen der rent faktisk betaler for profilen

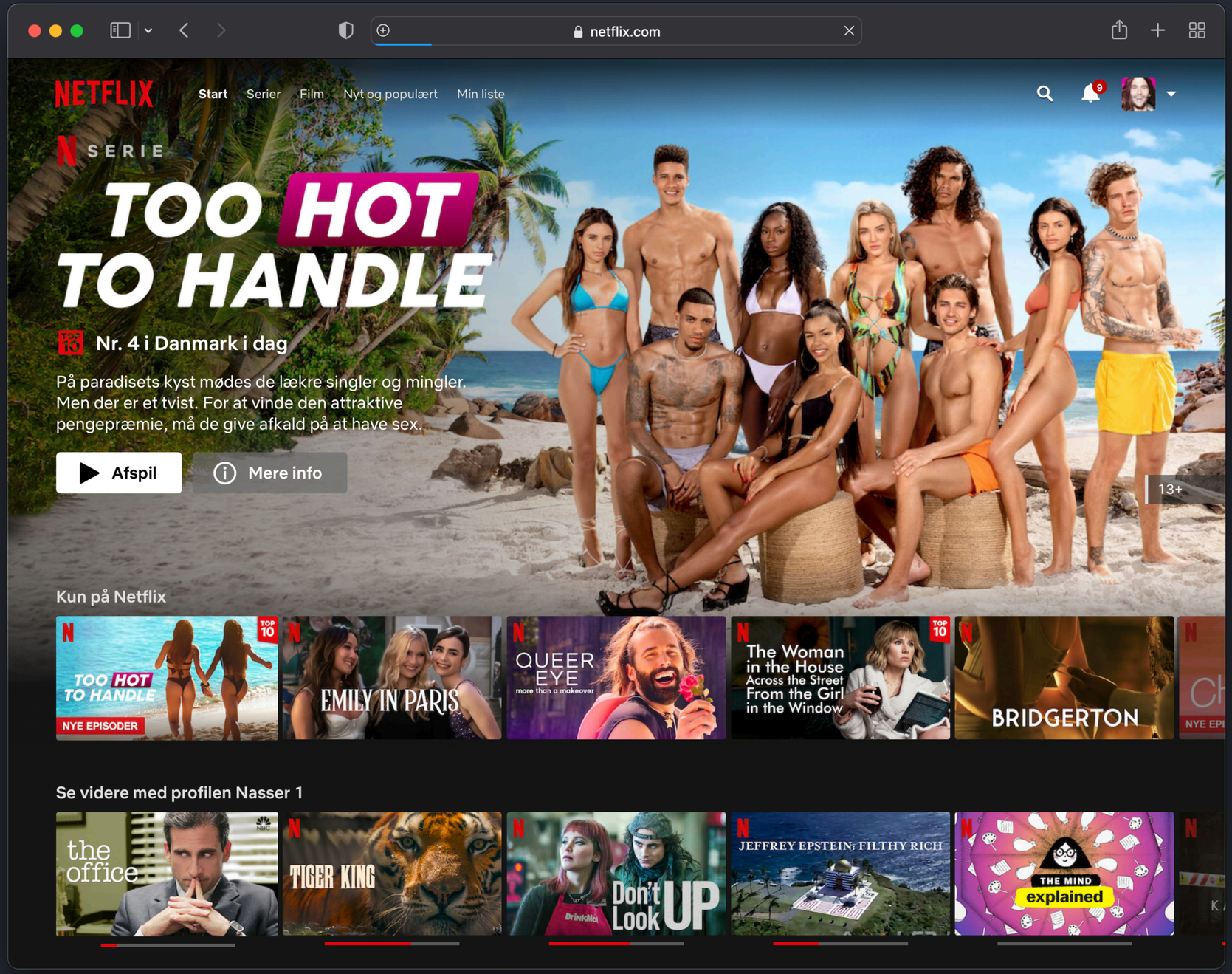
Nasser 1

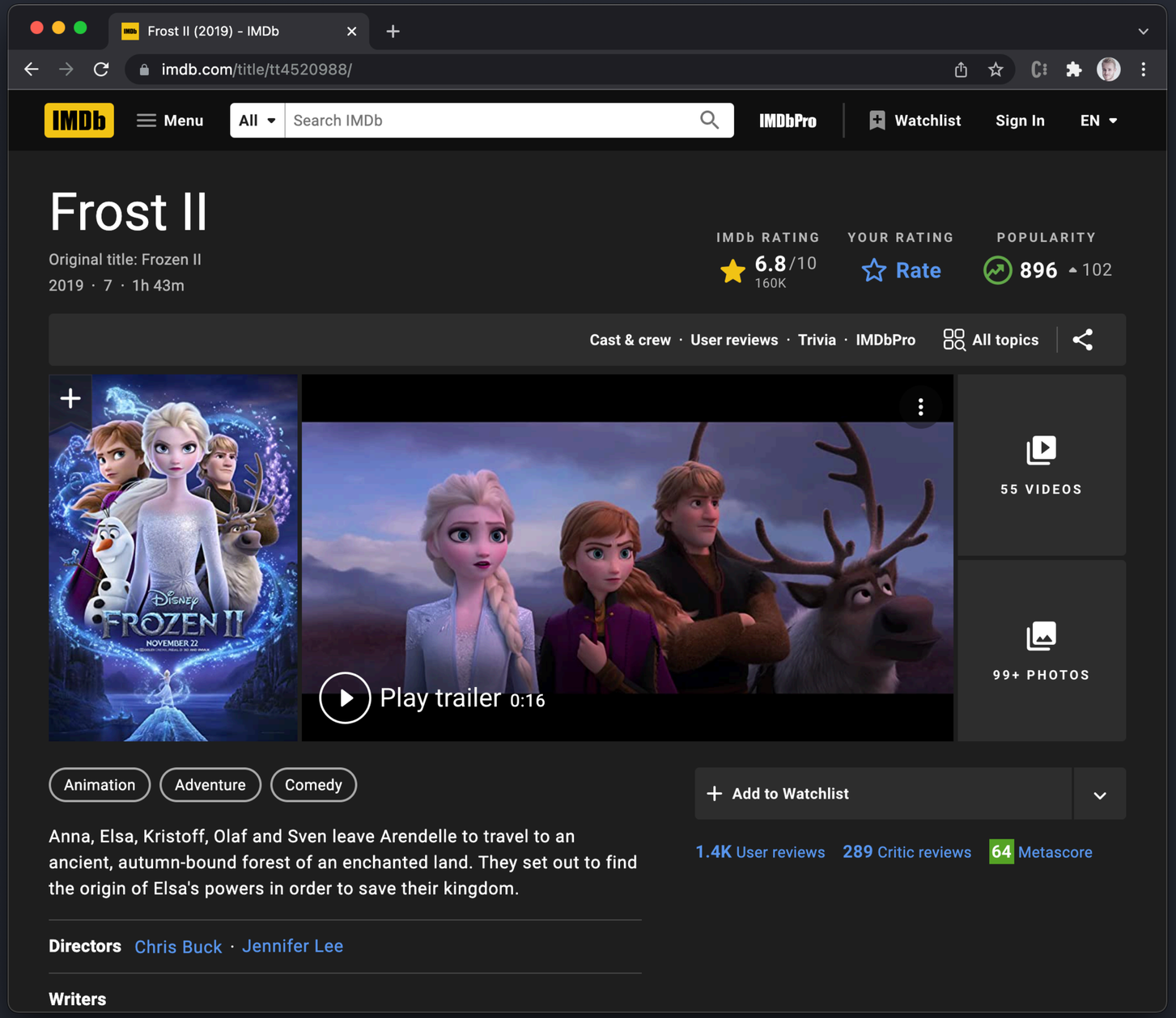
Nasser 2

Nasser 3

Nasser 4 Khader

Administrer profiler


netflix.com
NETFLIX
Start
Serier
Film
Nyt og populært
Min liste
SERIE
TOO HOT TO HANDLE
Nr. 4 i Danmark i dag
På paradisets kyst mødes de lækre singler og mingler. Men der er et tvist. For at vinde den attraktive pengepræmie, må de give afkald på at have sex.
Afspil
Mere info
13+
Kun på Netflix
NYE EPISODER
EMILY IN PARIS
QUEER EYE
more than a makeover
The Woman in the House Across the Street From the Girl in the Window
BRIDGERTON
Se videre med profilen Nasser 1
the office
TIGER KING
Don't Look UP
JEFFREY EPSTEIN: FILTHY RICH
THE MIND
explained




```
let movie = {
  title: "Frozen 2",
  description: "Elsa the Snow Queen has a
  trailer: "https://www.youtube.com/embed
  length: "1h 43m",
  year: "2019"
}
```

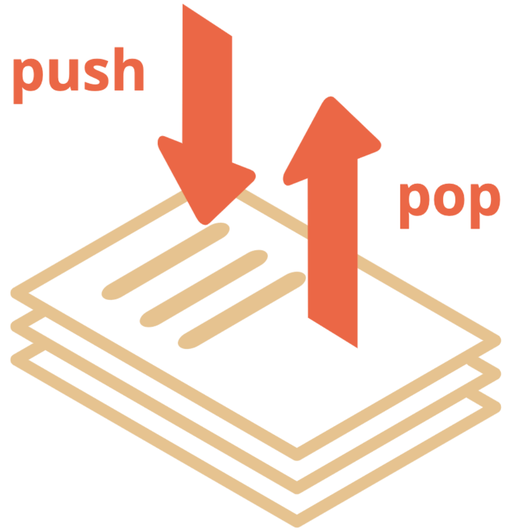
Define yourself as an object

With the following properties

`name, age, mail, phone, city`

Arrays

Ordered collection of values or objects



An array is a way to hold more than one value at a time we have a 1st, a 2nd, a 3rd, a 4th element and so on.

Arrays

Rasmus Cederdorff

Position:
Lecturer

Department/p
Multimedia De
Digital Concep

Address:
Ringvej Syd 1(

Mail:
race@baaa.dk

Phone:
7228 6318

Michael Hvidtfeldt

Position:
Lecturer

Department/p
Multimedia De

Address:
Ringvej Syd 1(

Mail:
mhv@baaa.dk

Phone:
7228 6328

Birgitte Kirk Iversen

Position:
Senior Lecturer

Department/programme:
Multimedia Design

Address:
Sønderhøj 30, 8260 Viby J

Mail:
bki@baaa.dk

Phone:
7228 6316

```
main.js:21
(3) [{…}, {…}, {…}] i
  0: {name: "Birgitte Kirk Iversen", mail: "bki@baaa…
  1: {name: "Michael Hvidtfeldt", mail: "mhv@baaa.dk…
  2: {name: "Rasmus Cederdorff", mail: "race@baaa.dk…
  length: 3
  __proto__: Array(0)
main.js:22
{name: "Michael Hvidtfeldt", mail: "mhv@baaa.dk"}
3
main.js:23
```

```
let teachers = [{
    name: "Birgitte Kirk Iversen",
    mail: "bki@baaa.dk"
  },
  {
    name: "Michael Hvidtfeldt",
    mail: "mhv@baaa.dk"
  },
  {
    name: "Rasmus Cederdorff",
    mail: "race@baaa.dk"
  }
];

console.log(teachers);
console.log(teachers[1]);
console.log(teachers.length);
```

Teachers

http://127.0.0.1:5501/array-teachers/index.html

Teachers

Birgitte Kirk Iversen

Senior Lecturer
bki@baaa.dk

Maria Louise Bendixen

Senior Lecturer
mlbe@baaa.dk

Kim Elkjær Marcher-Jepsen

Lecturer
kije@baaa.dk

Rasmus Cederdorff

Lecturer
race@baaa.dk

Console

top

Filter

Default levels

1 Issue: 1

main.js:46

```
(4) [{…}, {…}, {…}, {…}]
  0:
    address: "Sønderhøj 30, 8260 Viby J"
    department: "Multimedia Design"
    img: "https://www.eaaa.dk/media/u4gorzs
    initials: "bki"
    mail: "bki@baaa.dk"
    name: "Birgitte Kirk Iversen"
    phone: "72286316"
    position: "Senior Lecturer"
    [[Prototype]]: Object
  1: {name: 'Maria Louise Bendixen', initia
  2: {name: 'Kim Elkjær Marcher-Jepsen', in
  3: {name: 'Rasmus Cederdorff', initials:
  length: 4
  [[Prototype]]: Array(0)
```

array-teachers

DR | Nyheder - Breaking - TV

dr.dk

DR

NYHEDER DRTV DR LYD

KONTAKT DR

DR1: Løvens Hule

DR3: Nationens stærkeste

P1: LSD kælderen

DR LYD: Annas Margrethe

DR3: Du fucker med de forkerte

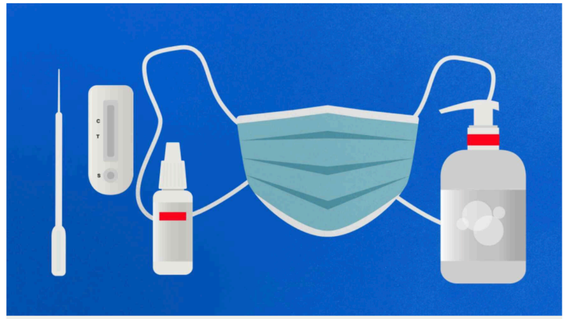
A Very British Scandal

Seneste nyt

6 MIN. SIDEN
EU klager over Kinas hårde kurs over for Litauen

13 MIN. SIDEN
Børn og skoleelever opfordres stadig til to ugentlige coronatest

25 MIN. SIDEN
England skrotter størstedelen af coronarestriktionerne fra i dag



UDLAND

15 lande bakker Danmark op: Danske soldater skal blive i Mali

Liveblog

PENGE

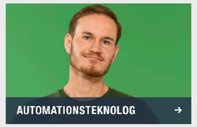
Regeringen har meldt genåbning - men ikke

Erhvervsakademi - København

https://kea.dk/uddannelser/erhvervsakademi

ALLE UDDANNELSER UDDANNELSER UD FRA INTERESSE

ALLE ERHVERVSAKADEMI-UDDANNELSER



AUTOMATIONSTEKNOLOG →

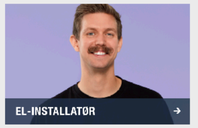
BYGGEKOORDINATOR →

BYGGETEKNIKER →

DATAMATIKER →

DESIGNTEKNOLOG →

ENTREPRENØRSKAB OG DESIGN →



EL-INSTALLATØR →

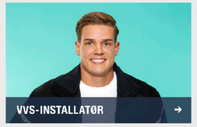
ENERGITEKNOLOG →

IT-TEKNOLOG →

KORT- OG LANDMÅLING →

MULTIMEDIEDESIGNER →

PRODUKTIONSTEKNOLOG →

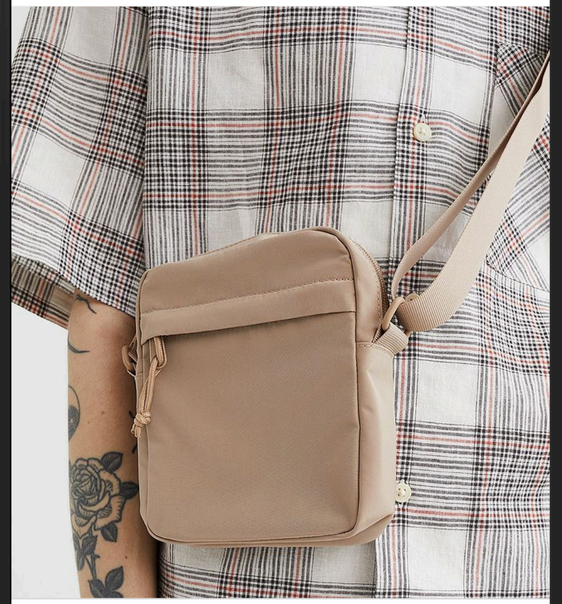


VVS-INSTALLATØR →

Instagram

Your story | fhergram | vladaafa... | kristybra... | gabriel

hm_man



Instagram

Liked by **praaajyot** and **others**

hm_man Shades, style and summer vibes, they all fit inside. #HMMan... more

View all 5 comments

44 MINUTES AGO

programmerhumor_io

I Am Devloper
@iamdevloper

Elon Musk: I'm putting people on Mars!
Developers: Fantastic, more timezones to support

cederdorff

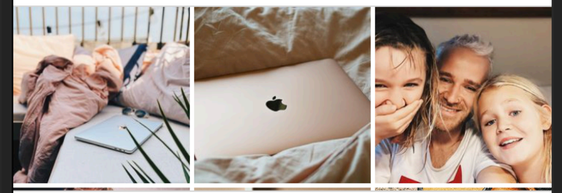
cederdorff

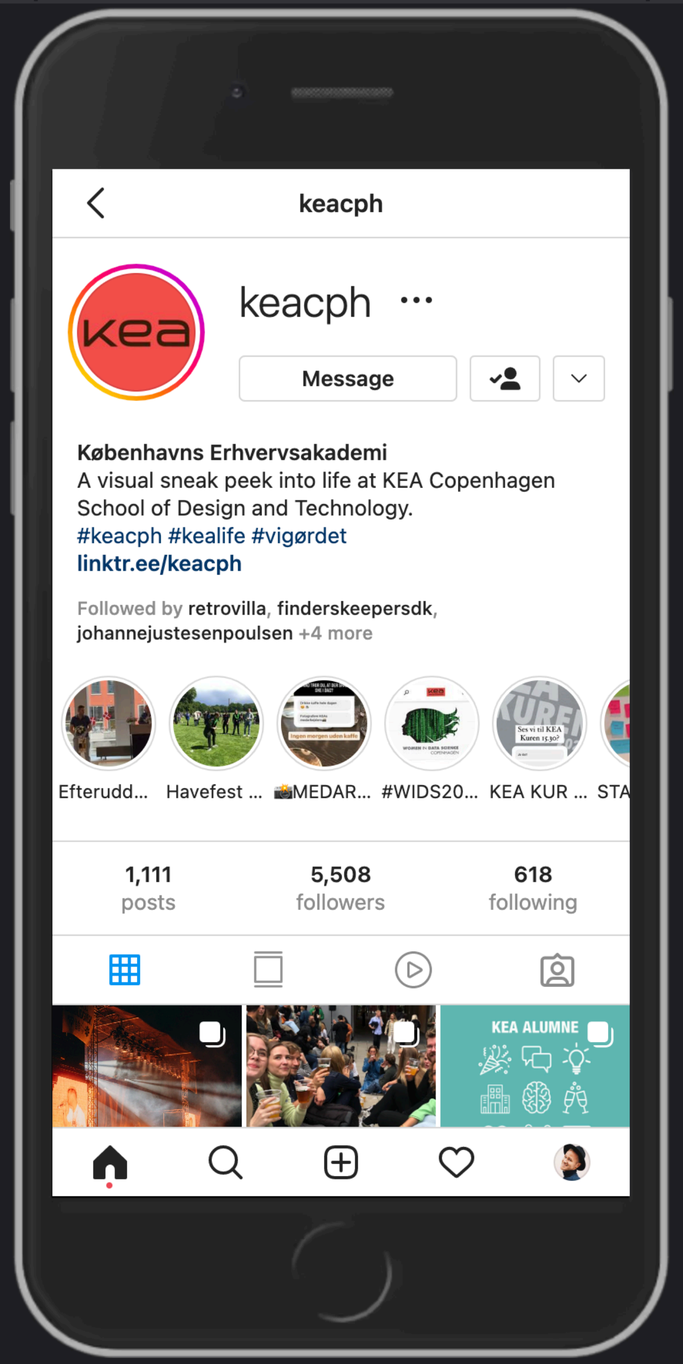
Edit Profile

Rasmus Cederdorff
Senior Lecturer & Web App Developer
Alicia & Ida
Aarhus, Denmark
cederdorff.com

AMSTER... | LONDON ... | Karla & Pi... | @villacph | IDA

181	5,441	635
posts	followers	following





Dimensions: iPhone ... 375 x 667 100%

Elements Components Console Performance insights Recorder Sources Network

Preserve log Disable cache No throttling

Invert Hide data URLs All Fetch/XHR JS CSS Img Media Font Doc WS Wasm Manifest Other

Has blocked cookies Blocked Requests 3rd-party requests

2000 ms 4000 ms 6000 ms 8000 ms 10000 ms 12000 ms 14000 ms

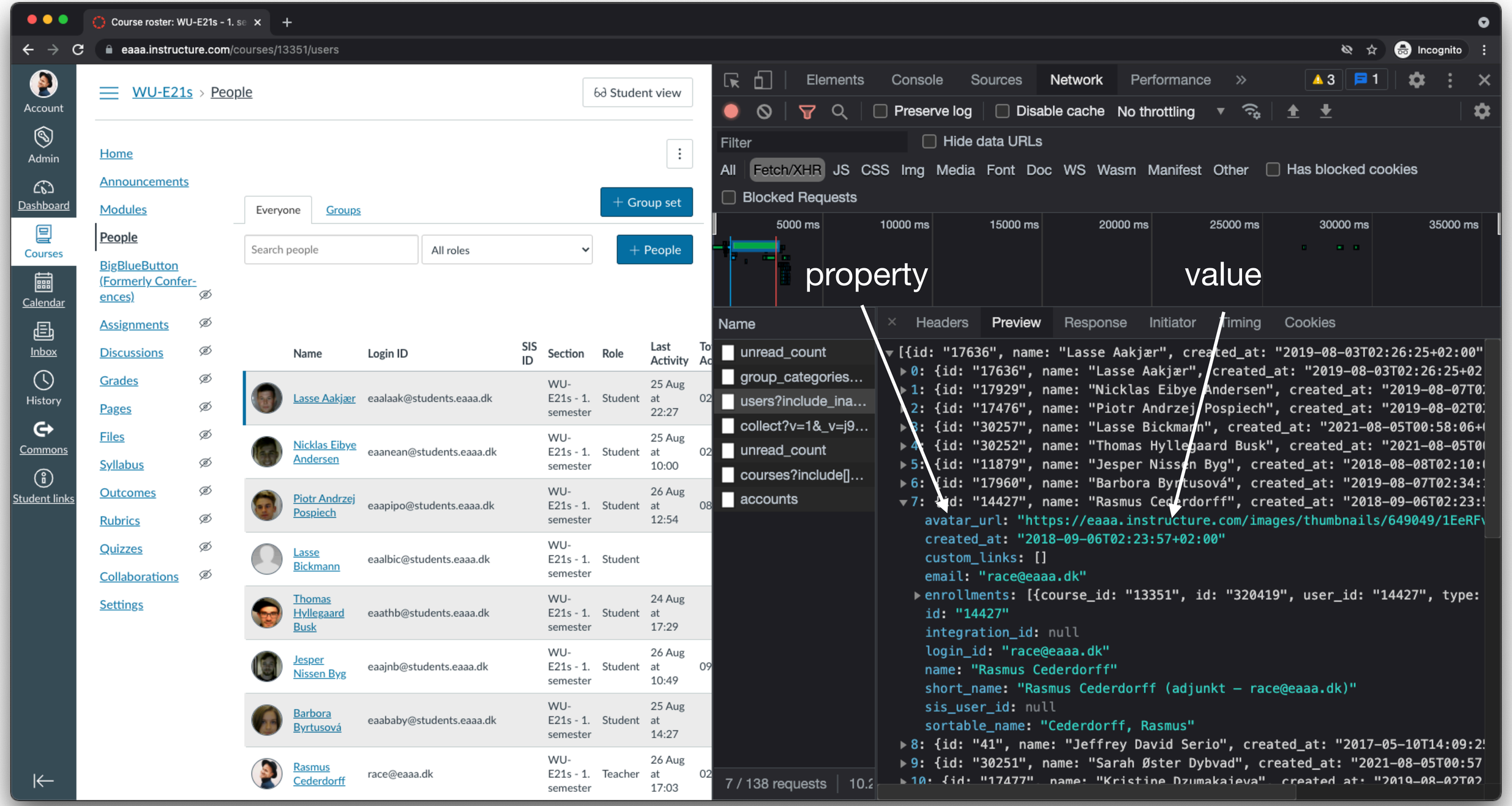
Name

?modules=PolarisBD...
?content_type=PROF...
?username=keacph
reels_tray/
timeline/
ig_sso_users/
logging_client_events
bz?__d=dis
falco
bz?__a=1&__ccg=G...
batch_fetch_web/
get_encrypted_crede...
highlights_tray/
?target_user_id=140...
badge/
story/
bulk-route-definitions/
bulk-route-definitions/
?query_hash=69cba4...
bz?__a=1&__ccg=G...
bz?__a=1&__ccg=G...
bulk-route-definitions/
bz?__a=1&__ccg=G...
logging_client_events
bz?__d=dis

26 / 96 requests 257 kB

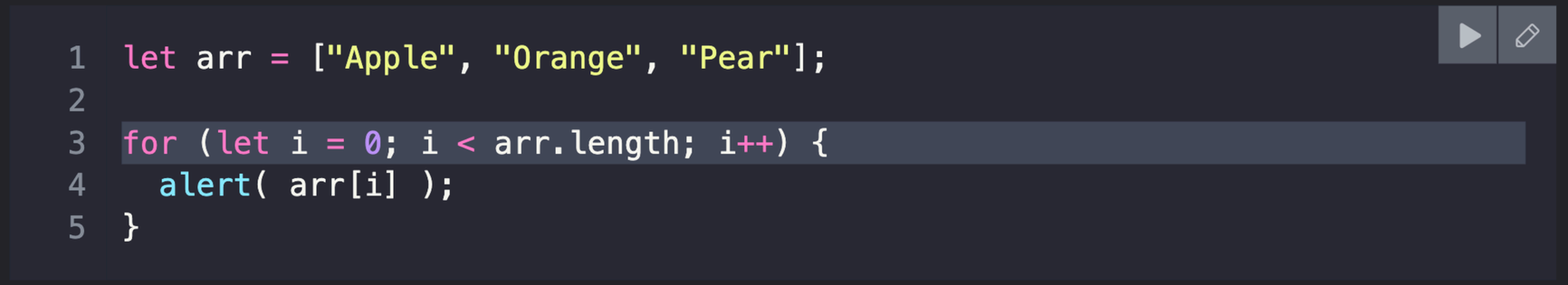
Headers Payload Preview Response Initiator Timing Cookies

{data: {user: {,…}}, status: "ok"}
data: {user: {,…}}
user: {,…}
bio_links: []
biography: "A visual sneak peek into life at KEA Copenhagen School of Design and Technology. \n#keacph
biography_with_entities: {,…}
blocked_by_viewer: false
business_address_json: null
business_category_name: "Home Services"
business_contact_method: "CALL"
business_email: null
business_phone_number: null
category_enum: "LOCAL"
category_name: null
connected_fb_page: null
country_block: false
edge_felix_video_timeline: {count: 116, page_info: {has_next_page: true,…},…}
edge_follow: {count: 618}
edge_followed_by: {count: 5508}
edge_media_collections: {count: 0, page_info: {has_next_page: false, end_cursor: null}, edges: []}
edge_mutual_followed_by: {count: 7, edges: [{node: {username: "retrovilla"}}, {node: {username: "finde
edge_owner_to_timeline_media: {count: 1111, page_info: {has_next_page: true,…},…}
edge_saved_media: {count: 0, page_info: {has_next_page: false, end_cursor: null}, edges: []}
external_url: "https://linktr.ee/keacph"
external_url_linkshimmed: "https://l.instagram.com/?u=https%3A%2F%2Flinktr.ee%2Fkeacph&e=ATNF7APjyvCLD
fbid: "17841400899781985"
followed_by_viewer: true
follows_viewer: true
full_name: "Københavns Erhvervsakademi"
guardian_id: null
has_ar_effects: false
has_blocked_viewer: false
has_channel: false
has_clips: false


Course roster: WU-E21s - 1. se
eaaa.instructure.com/courses/13351/users
Incognito
Account
Admin
Dashboard
Courses
Calendar
Inbox
History
Commons
Student links
WU-E21s › People
Student view
Home
Announcements
Modules
People
BigBlueButton (Formerly Confer-ences)
Assignments
Discussions
Grades
Pages
Files
Syllabus
Outcomes
Rubrics
Quizzes
Collaborations
Settings
Everyone
Groups
+ Group set
Search people
All roles
+ People
Name
Login ID
SIS ID
Section
Role
Last Activity
Lasse Aakjær
eaalaak@students.eaaa.dk
WU-E21s - 1. semester
Student
25 Aug at 22:27
Nicklas Eibye Andersen
eaanean@students.eaaa.dk
WU-E21s - 1. semester
Student
25 Aug at 10:00
Piotr Andrzej Pospiech
eaapipo@students.eaaa.dk
WU-E21s - 1. semester
Student
26 Aug at 12:54
Lasse Bickmann
eaalbic@students.eaaa.dk
WU-E21s - 1. semester
Student
Thomas Hyllegaard Busk
eaathb@students.eaaa.dk
WU-E21s - 1. semester
Student
24 Aug at 17:29
Jesper Nissen Byg
eaajnb@students.eaaa.dk
WU-E21s - 1. semester
Student
26 Aug at 10:49
Barbora Byrtusová
eaababy@students.eaaa.dk
WU-E21s - 1. semester
Student
25 Aug at 14:27
Rasmus Cederdorff
race@eaaa.dk
WU-E21s - 1. semester
Teacher
26 Aug at 17:03
Elements
Console
Sources
Network
Performance
Preserve log
Disable cache
No throttling
Filter
Hide data URLs
All
Fetch/XHR
JS
CSS
Img
Media
Font
Doc
WS
Wasm
Manifest
Other
Has blocked cookies
Blocked Requests
5000 ms
10000 ms
15000 ms
20000 ms
25000 ms
30000 ms
35000 ms
property
value
Name
unread_count
group_categories...
users?include_ina...
collect?v=1&_v=j9...
unread_count
courses?include[]...
accounts
7 / 138 requests
Headers
Preview
Response
Initiator
Timing
Cookies
[{id: "17636", name: "Lasse Aakjær", created_at: "2019-08-03T02:26:25+02:00"
0: {id: "17636", name: "Lasse Aakjær", created_at: "2019-08-03T02:26:25+02
1: {id: "17929", name: "Nicklas Eibye Andersen", created_at: "2019-08-07T0
2: {id: "17476", name: "Piotr Andrzej Pospiech", created_at: "2019-08-02T0
3: {id: "30257", name: "Lasse Bickmann", created_at: "2021-08-05T00:58:06+
4: {id: "30252", name: "Thomas Hyllegaard Busk", created_at: "2021-08-05T0
5: {id: "11879", name: "Jesper Nissen Byg", created_at: "2018-08-08T02:10:
6: {id: "17960", name: "Barbora Byrtusová", created_at: "2019-08-07T02:34:
7: {id: "14427", name: "Rasmus Cederdorff", created_at: "2018-09-06T02:23:
avatar_url: "https://eaaa.instructure.com/images/thumbnails/649049/1EeRF
created_at: "2018-09-06T02:23:57+02:00"
custom_links: []
email: "race@eaaa.dk"
enrollments: [{course_id: "13351", id: "320419", user_id: "14427", type:
id: "14427"
integration_id: null
login_id: "race@eaaa.dk"
name: "Rasmus Cederdorff"
short_name: "Rasmus Cederdorff (adjunkt – race@eaaa.dk)"
sis_user_id: null
sortable_name: "Cederdorff, Rasmus"
8: {id: "41", name: "Jeffrey David Serio", created_at: "2017-05-10T14:09:2
9: {id: "30251", name: "Sarah Øster Dybvad", created_at: "2021-08-05T00:57

JavaScript.info/array#loops

One of the oldest ways to cycle array items is the `for` loop over indexes:



```
let arr = ["Apple", "Orange", "Pear"];

for (let i = 0; i < arr.length; i++) {
  alert( arr[i] );
}
```

But for arrays there is another form of loop, `for..of`:

```
let fruits = ["Apple", "Orange", "Plum"];

// iterates over array elements
for (let fruit of fruits) {
  alert( fruit );
}
```

Array.map(...)

...iterate over an array and modify each element.

Array.map(...) calls a callback function for each element in the array.

```
const persons = [
    { firstname: "Birgitte", lastname: "Iversen" },
    { firstname: "Lykke", lastname: "Dahlen" },
    { firstname: "Rasmus", lastname: "Cederdorff" }
];

const mapped = persons.map(person => {
    return {
        name: `${person.firstname} ${person.lastname}`
    };
});

console.log(mapped);
```

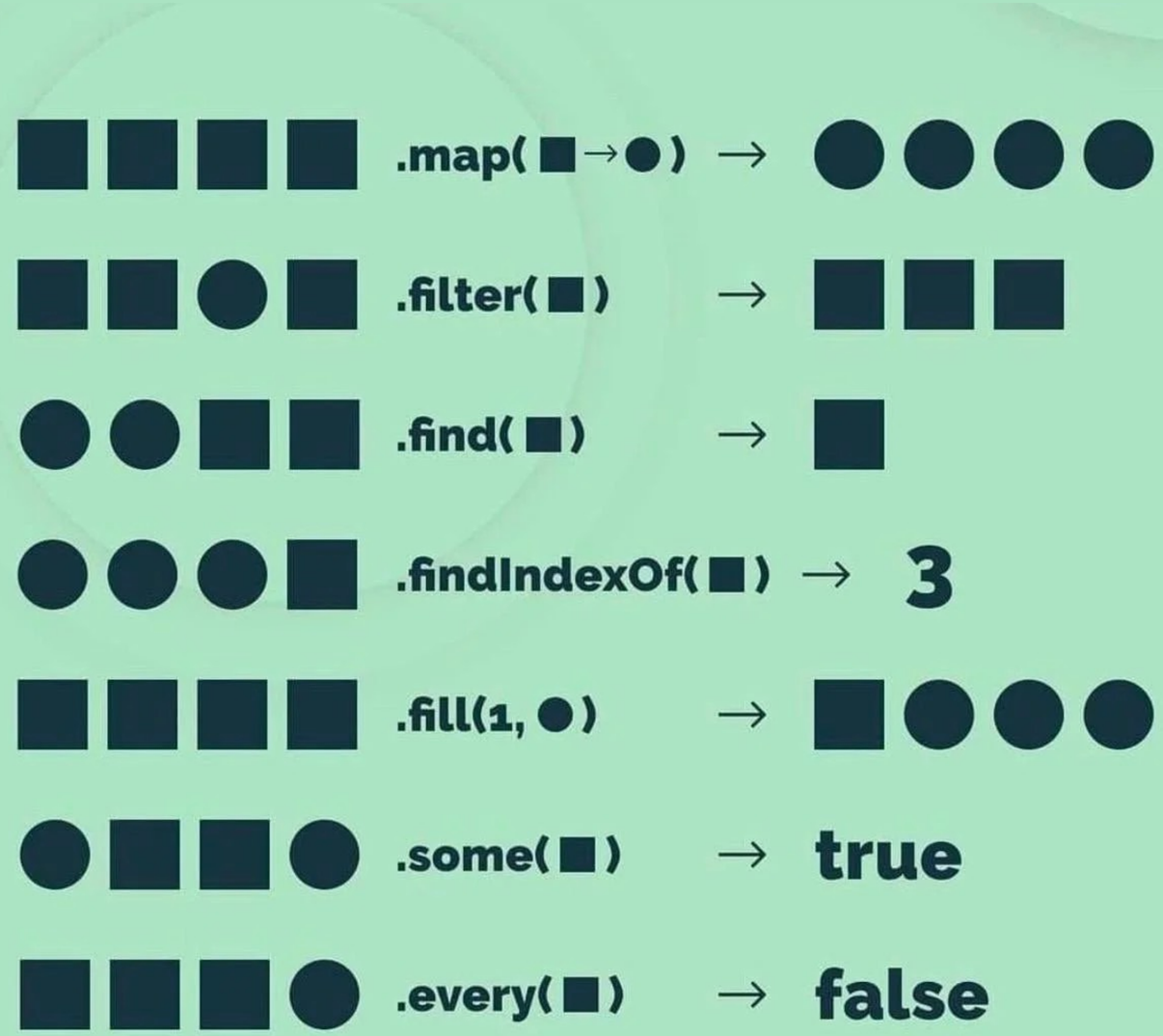
```
▼ (3) [{…}, {…}, {…}] i
  ▶ 0: {name: 'Birgitte Iversen'}
  ▶ 1: {name: 'Lykke Dahlen'}
  ▶ 2: {name: 'Rasmus Cederdorff'}
    length: 3
```

Array

.FILTER(...)

```
let users = [
  { age: 35, name: "John" },
  { age: 40, name: "Pete" },
  { age: 44, name: "Mary" }
];

// returns array of with users older than 39
let someUsers = users.filter(item => item.age > 39);

console.log(someUsers);
```

```
▼ Array(2) i
  ▶ 0: {age: 40, name: "Pete"}
  ▶ 1: {age: 44, name: "Mary"}
    length: 2
```



https://javascript.info/array-methods
https://medium.com/@mandeepkaur1/a-list-of-javascript-array-methods-145d09dd19a0

Array Methods to know

- `.push(…)`
- `.map(…)`
- `.filter(…)`
- `.find(…)`
- `.sort(…)`

Computer science student

Senior developer, 10+ years experience

https://www.instagram.com/p/BxWAgatgSmn/

It’s all objects & arrays!

Object-Oriented JavaScript

“The basic idea of OOP is that we use objects to model real world things that we want to represent inside our programs, and/or provide a simple way to access functionality that would otherwise be hard or impossible to make use of.”

“Objects can contain related data and code, which represent information about the thing you are trying to model, and functionality or behavior that you want it to have.”

https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/Object-oriented_JS

Object-Oriented JavaScript

We use classes to describe a generic structure (data and functionality) of an object.

https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/Object-oriented_JS

Classes

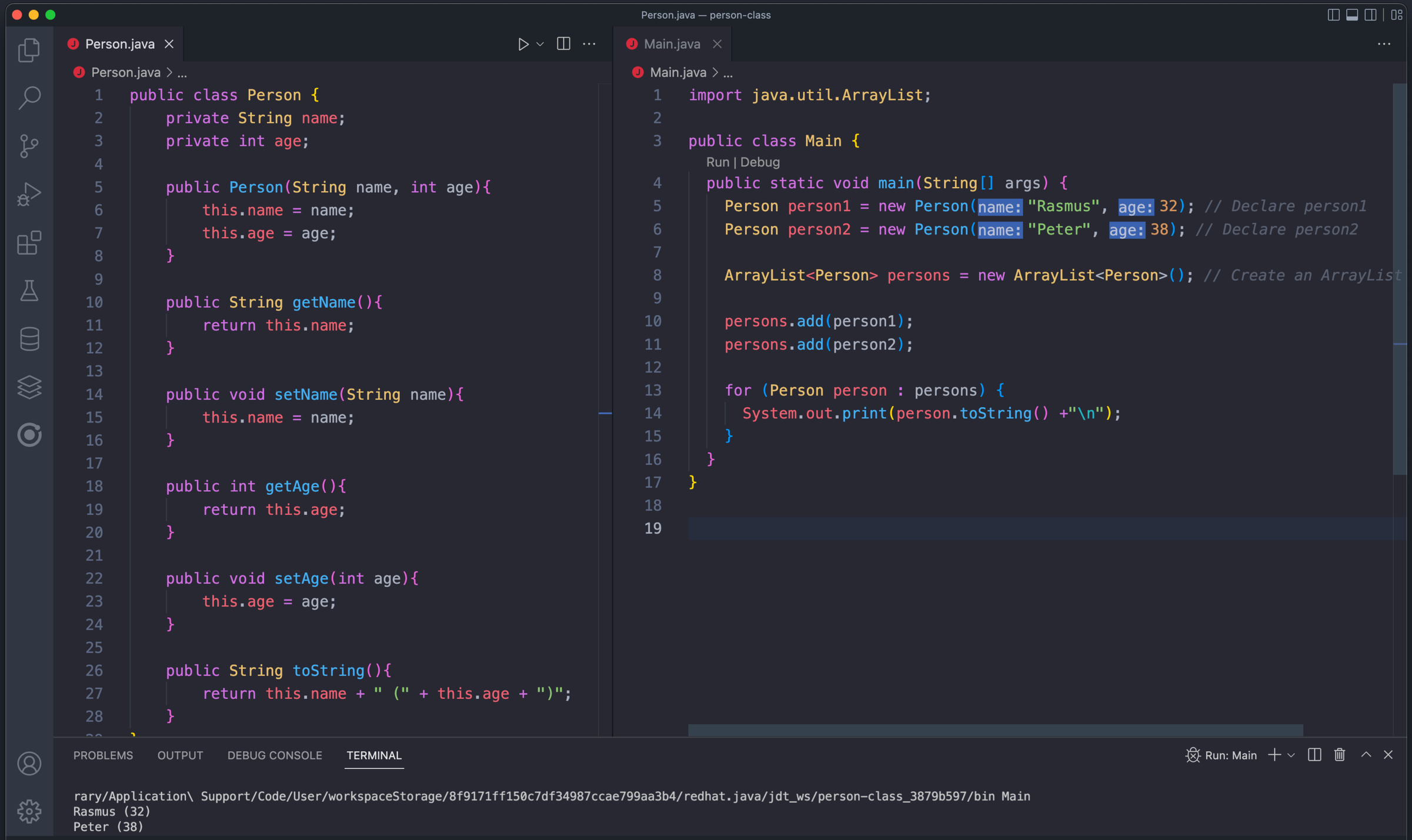
“In object-oriented programming, a class is an extensible program-code-template for creating objects, providing initial values for state (member variables) and implementations of behavior (member functions or methods).”

https://javascript.info/class

Classes

...templates for creating objects

https://javascript.info/class


Person.java — person-class
Person.java
Person.java > ...
1 public class Person {
2 private String name;
3 private int age;
4
5 public Person(String name, int age){
6 this.name = name;
7 this.age = age;
8 }
9
10 public String getName(){
11 return this.name;
12 }
13
14 public void setName(String name){
15 this.name = name;
16 }
17
18 public int getAge(){
19 return this.age;
20 }
21
22 public void setAge(int age){
23 this.age = age;
24 }
25
26 public String toString(){
27 return this.name + " (" + this.age + ")";
28 }
Main.java
Main.java > ...
1 import java.util.ArrayList;
2
3 public class Main {
Run | Debug
4 public static void main(String[] args) {
5 Person person1 = new Person(name: "Rasmus", age: 32); // Declare person1
6 Person person2 = new Person(name: "Peter", age: 38); // Declare person2
7
8 ArrayList<Person> persons = new ArrayList<Person>(); // Create an ArrayList
9
10 persons.add(person1);
11 persons.add(person2);
12
13 for (Person person : persons) {
14 System.out.print(person.toString() +"\n");
15 }
16 }
17 }
18
19
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
Run: Main
rary/Application\ Support/Code/User/workspaceStorage/8f9171ff150c7df34987ccae799aa3b4/redhat.java/jdt_ws/person-class_3879b597/bin Main
Rasmus (32)
Peter (38)

```
class MyClass {
  // class methods
  constructor() { ... }
  method1() { ... }
  method2() { ... }
  method3() { ... }
  ...
}
```

Car class with constructor

```
class Car {
    constructor(name, year) {
        this.name = name;
        this.year = year;
    }
}
```

Create object using a Class

```
class Car {
    constructor(name, year) {
        this.name = name;
        this.year = year;
    }
}

let myCar1 = new Car("Ford", 2014);
let myCar2 = new Car("Audi", 2019);
```

Classes can contain properties & functions

```
class Car {
    constructor(name, year) {
        this.name = name;
        this.year = year;
    }
    age() {
        let date = new Date();
        return date.getFullYear() - this.year;
    }
}

let myCar1 = new Car("Ford", 2014);
let myCar2 = new Car("Audi", 2019);

console.log(myCar1.age());
console.log(myCar2.age());
```

properties

method/function

instantiation

call method on object

Classes

A class is a way to describe a generic structure and functionality of an object. It is a template for creating an object.

A class defines properties and functions of an object.

With a class, we can declare instances (objects) with values based on the given template (class structure).

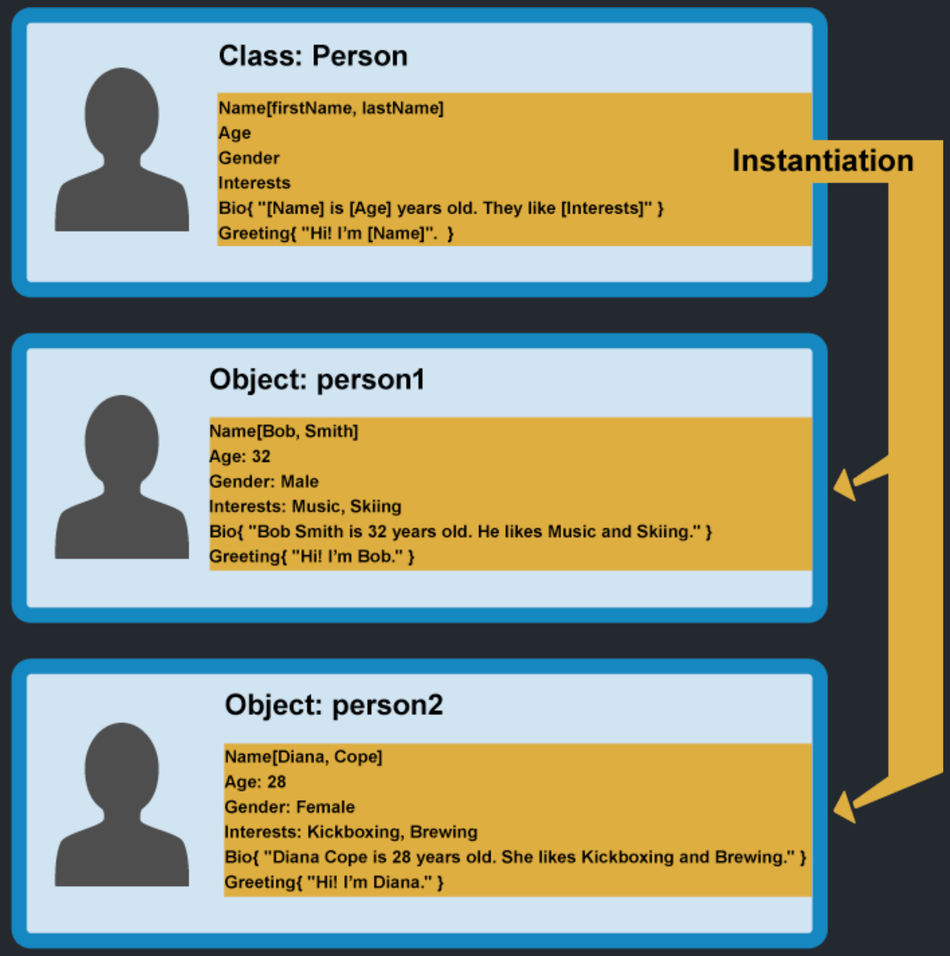
Classes

In practice, we often create many objects of the same kind, like users, products, movies etc.

Instantiation

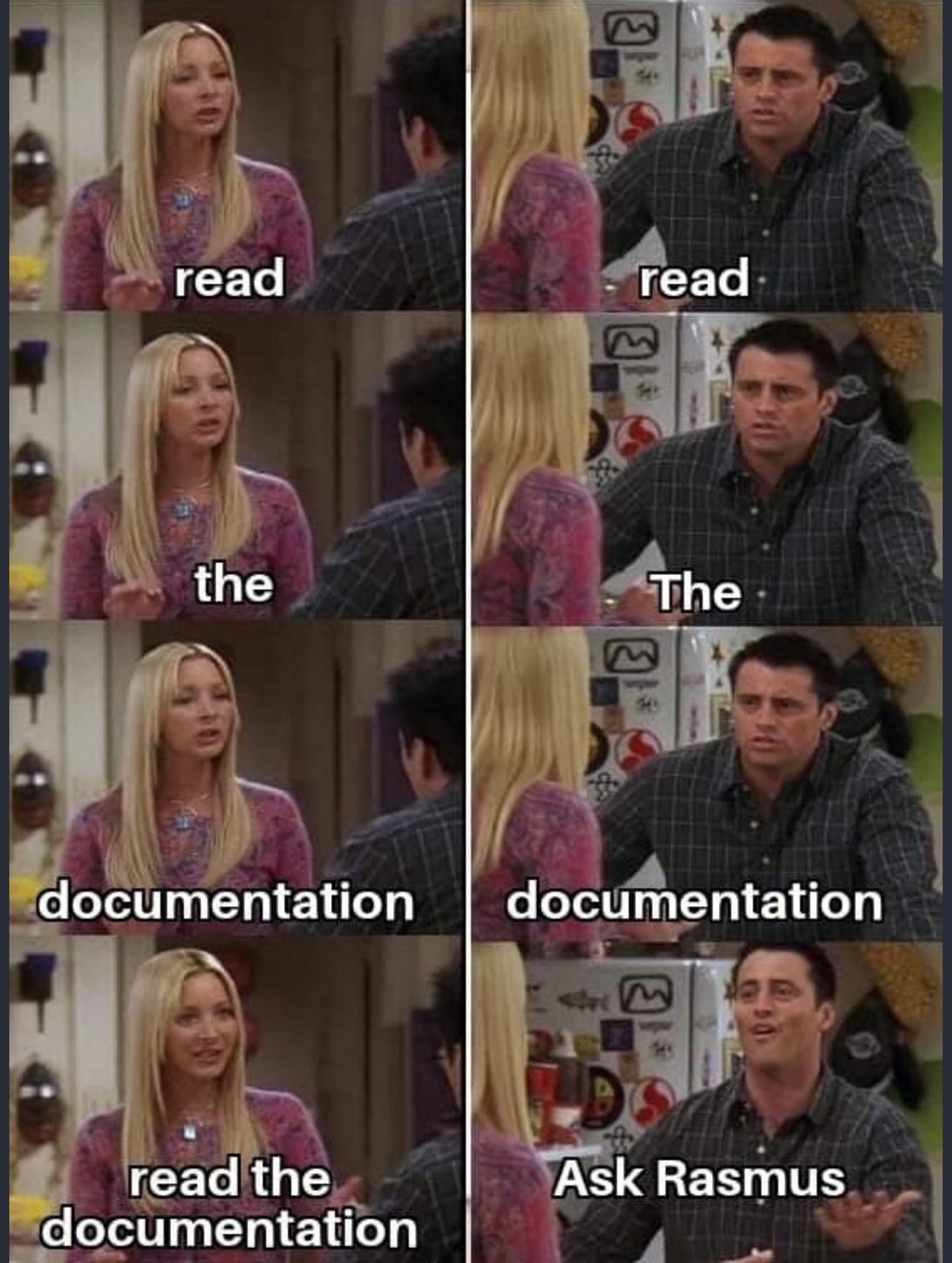
“template”

Instances using class template
(objects with values)



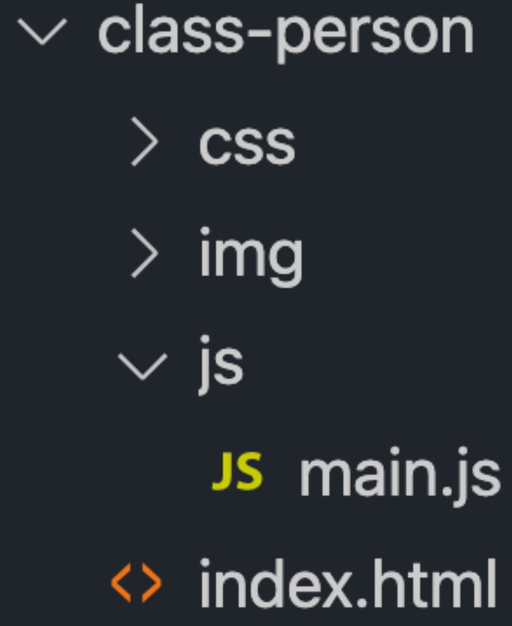
Person Class Exercise #1

1. Review slide 30 to 41.
2. Use `class-js-template` and delete all inside `main.js`
3. Create a class called `Person` inside `main.js`
4. Create a `constructor`
5. Define the properties `name`, `mail`, `birthdate` & `img`
6. Create two `Person` objects based on your newly defined class.
7. Inside of `Person` class, create a function named `log()` logging all properties of the person in one string.
8. Call `log()` on your two Person objects.


read
read
the
The
documentation
documentation
read the documentation
Ask Rasmus

Person Class Exercise #2

1. Inside of `Person` class, create a function named `getAge()` retuning the age of the person. Calculate the age based on the property `birthdate` and `JS Date`.
2. Test `getAge()` on `Person` objects.
3. Inside of `Person` class, create a function called `getHtmlTemplate()`. The function must return a html template string. The string must include `name`, age (`getAge()`), `mail` & `img`
4. Use `getHtmlTemplate()` to add the persons to the DOM.



```
class Person {
  constructor(name, mail, birthDate, img) {
    this.name = name;
    this.mail = mail;
    this.birthDate = birthDate;
    this.img = img;
  }

  log() {
    console.log(`
      Name: ${this.name},
      Mail: ${this.mail},
      Birth date: ${this.birthDate},
      Image Url: ${this.img}
    `);
  }

  getAge() {
    const birthDate = new Date(this.birthDate);
    const today = new Date();
    const diff = new Date(today - birthDate);
    return diff.getFullYear() - 1970;
  }

  getHtmlTemplate() {
    const template = /*html*/`
      <article>
        <img src="${this.img}">
        <h2>${this.name}</h2>
        <a href="mailto:${this.mail}">${this.mail}</a>
        <p>Birth date: ${this.birthDate}</p>
        <p>Age: ${this.getAge()} years old</p>
      </article>
    `;

    return template;
  }
}
```

properties

methods/functions

instantiation inside an Array

```
const persons = [
  new Person("Birgitte", "bki@mail.dk", "1966-01-14", "https://www.eaaa.dk/media/u4gorzsd/birgit
  new Person("Martin", "mnor@mail.dk", "1989-05-02", "https://media-exp1.licdn.com/dms/image/C4D
  new Person("Rasmus", "race@mail.dk", "1990-09-15", "https://www.eaaa.dk/media/devlvvgj/rasmus-
];

console.log(persons);

for (const person of persons) {
  document.querySelector("#content").innerHTML += person.getHtmlTemplate();
}
```

call method on object

Person Class Exercise #3

1. Use your `Person` class solution.
2. Create an array with at least 3 instances (objects) of your `Person` class.
3. Log the array.
4. Use a for-of loop to append all persons to the DOM. Make use of `getHtmlTemplate()`.


Code
Every
Day