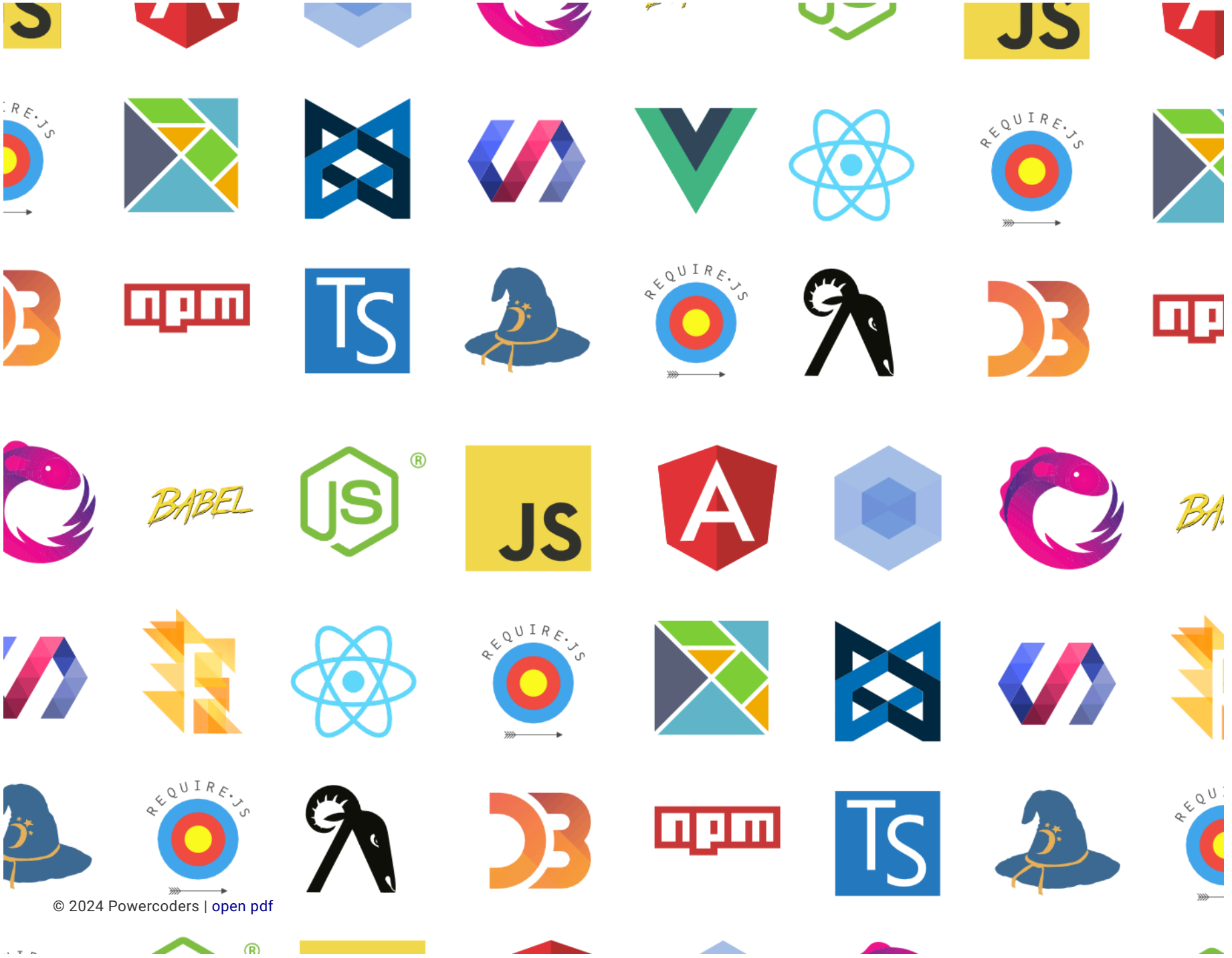
{POWER.CODERS}

Intro to frameworks and libraries

Agenda

Today we will look into

- JS libraries
- JS frameworks



Library and Framework

Both frameworks and libraries are reusable code written by **someone else** that is used to help **solve common problems**.

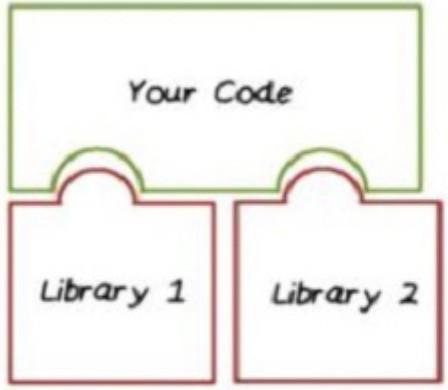
Library vs. Framework

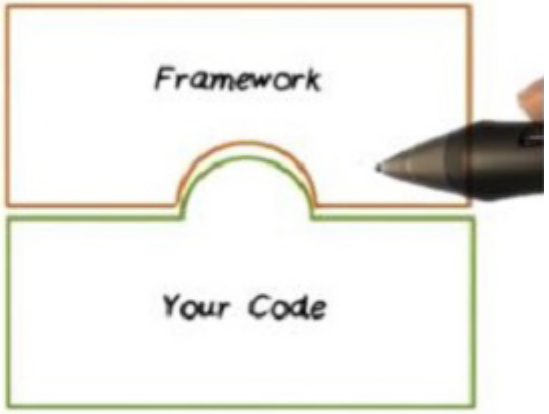
A **library** is like going to IKEA. You already have a home, but you need a bit of help with your funiture. You pick and choose the furniture, you are **in control**.

A **framework** is like building a model home. You have a set of blueprints and a **few limited choices** when it comes to architecture and design.

Inversion of control

What is the technical difference between library and framework





Inversion of control

Library: You are in charge of the flow of the application. You decide when and where to call a library.

Framework: The framework is in charge of the flow. It provides places for you to plugin your code, but it chooses when to call this code.

JS LIBRARIES

What is a library again?

A collection of reusable methods for a particular purpose.

It allows easier and faster development of JavaScript-based applications, especially for AJAX and other web-centric technologies.

A SHORT HISTORY

jQuery as well as other JS libraries were born into a world with cross-browser quirks and standardization issues.

With JavaScript on its own many time-consuming workarounds were needed. And testing was very frustrating.

jQuery, Mootools, Prototype and other libraries helped developers with that by creating an abstraction layer.

Common libraries today

Name	Since	Market share	Popularity	Example
jQuery (UI + Mobile)	2006	95.5% of all websites	34.4% of JS developers	pascualprestige.com
React JS	2013	2.9% of all websites	40.1% of JS developers	criticaltechworks.com

What is jQuery?

jQuery is a feature-rich JavaScript library.

It is not a framework.

With our knowledge of JavaScript so far it will be quite easy to grasp jQuery.

What about today?

Is jQuery still needed?

Today a lot of the features, e.g. CSS selectors syntax and AJAX, which made jQuery so valuable can be done with pure JavaScript.

Some say jQuery is a relic of the past, others still use it day to day.

You might not need any more, but you certainly need to understand it.

jQuery is used by 78% of all websites

Mostly because of Wordpress

Recommendation:
Do not use jQuery for new website projects.

Pro

- jQuery has an extensible plugin system
- jQuery is stable
- jQuery supports older browsers

Con

- Rendering performance
- Increased bundle size
- Not needed if you only support modern browsers

Online Resources jQuery

- Official jQuery Documentation
- w3schools Tutorial
- Freecodecamp Tutorial
- jQuery.each()
- You might not need jQuery
- Using jQuery in 2019
- Should you learn jQuery in 2019?

Some libraries and plugins

- Three.js: creating 3D objects and spaces
- GSAP: all things animated
- D3.js: manipulating documents based on data
- Swiper: best mobile touch slider
- ... and many more

Libraries vs. plugin

Library is a **collection** of classes and functions that helps to develop an application or website.

Plugin is an **extension** that improves the capabilities of your application or website by adding one specific solution.

Our own JS library

It is useful to add common functions you will reuse in your website project to your own library, e.g.

- `addClass`
- `removeClass`
- `hasClass`
- `triggerClass`
- and more...

JS FRAMEWORKS

How are JS frameworks built?

- **Components** or modules are the building blocks. They usually have one specific purpose and need to be imported, before they can be used.
- **Packages** are collection of modules, usually under one namespace and can be imported via package manager like NPM.
- **State management** is to share data across components. Often changes to the state need to be reflected in components and the data has to be synchronizes. Frameworks use libraries for that, e.g. Vues or Redux.

Common frontend frameworks

Name	Since	Usage	Popularity	Example
Vue.js	2014	0.7% of all websites	18.9% of JS developers	umwelt.schweiz.ch
AngularJS	2010	0.4% of all websites	22.9% of JS developers	lamborghiniporrentruy.ch
Electron	2013	~1'600 desktop apps	-	Slack / WhatsApp Desktop

What about node.js?

Node.js is a JavaScript runtime environment. It can be run as a frontend as well as backend framework.

It includes everything you need to execute a program written in Javascript.

It moved Javascript from websites to standalone applications.

e.g. Electron JS uses node.js and Chromium to build and run Desktop apps.

Why do you need to know that?

All companies use a set of technologies to build applications (web, mobile, desktop), a combination of programming languages, frameworks, libraries, servers, tools and so on.

= tech stack.

Tech stacks

Most commonly used stacks are:

- **LAMP**: Linux, Apache, MySQL, and PHP
- **MEAN**: Mongo DB, Express.js, AngularJS, and node.js

... and there are many, many more

Specialization

Even so called **full stack developers** cannot cover all stacks.

Today developers usually need to **specialize**. Switching jobs as a web developer might require for you to learn a new tech stack.

Focus tracks

TIMELINE

The last 4 weeks of our bootcamp are the focus weeks.

Focus tracks planning

We are not sure yet which track we will definitely offer - depends on internships and companies'needs. Usually we have:

Probable tracks

- DevOps
- Data
- Advanced web development
- Object orientated programming
- Software testing

Possible tracks

Additional tracks, only if required by companies

- Cyber Security
- IT Support
- IT Project Management
- Web Design

... and more ...

DevOps

We will expand on our CLI knowledge and dive into Linux. After basics in networking we will learn about Docker, Kubernetes and Cloud computing.

Depending on IT trainers you will work with AWS or Azure.

Data

The data track encompasses training in Data Science / Machine Learning OR Business Analytics / RPA.

Both groups will learn some fundamental knowledge in statistics. Which learning path you will follow depends on your internship or personal preferences (if no internship).

- More info on data-track.powercoders.org
- Profile Data Scientist
- Profile Business Analyst

Advanced web development

We will dive deeper into web development, get to know one JavaScript framework and also learn some backend programming, e.g. PHP or Python. We will also look into helpful tools, like webpack and npm.

We will build a small app using what we've learned.

Object Oriented Programming

We will expand on our knowledge of object oriented programming, getting a sense for using object oriented programming in Java, ASP.Net or C#.

Focus is on general concepts and best practices in Software Architecture and Software Quality.

Software testing

We will learn the essentials for software testing, like its terminology and concepts. You learn how to think like a tester, get to know first tools and strategies to test and report.

There is the option to follow a second course within your testing internship to study for the exam to get certified in ISTQB foundation level.

- Onepager Testing Tracks
- Profile Software Tester

IT SUPPORT

That track is usually based completely on the requirements of the company/companies.

Tell me your favorite track

I will open a slack thread, post your favorite track in there by **Friday 5pm**.

Depending on internship possibilities we might decide to move you to another track, but we will try to honor your choices.

Online resources

- DevOps Engineer
- What is AWS?
- Intro to cloud computig
- Machine learning with Python
- Why learn Java?
- Web development vs Web design
- Why software testing has a better future than development
- UiPath: robotic process automation

Online Resources

- Node JS documentation
- A beginner's guide to NPM
- Choosing a JS framework
- Typescript vs Javascript
- Typescript in 2019 and 2020

Work on your project