{POWER.CODERS}

Intro to frameworks and libraries

Agenda

Today we will look into

- JS libraries
- JS frameworks

REQUIRE.JS
npm
TS
BABEL
JS

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**.

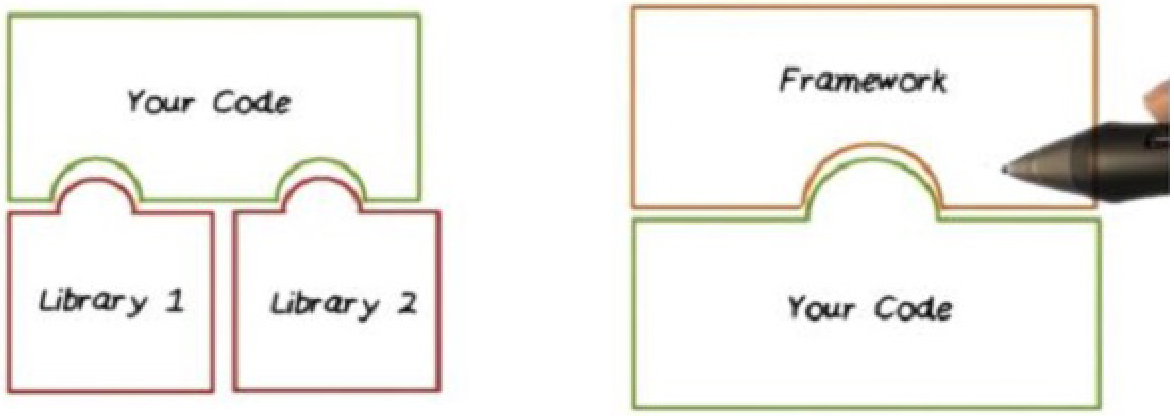
Library vs. Framework

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

Inversion of control

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

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.

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

A SHORT HISTORY

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

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.

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.

What is jQuery?

jQuery is a feature-rich JavaScript library.

It is not a framework.

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?

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.

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.

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 Ressources 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

Libraries vs. plugin

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

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.

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?

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.

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.

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.

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.

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.

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.

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:

Tech stacks

Most commonly used stacks are:

- **LAMP**: Linux, Apache, MySQL, and PHP

TECH STACKS

Most commonly used stacks are:

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

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.

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

After our 2 career weeks there are 4 weeks left. The focus weeks.

Focus tracks planning

Let us know your interests (top 3) by FRI this week. Via Slack.

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

Possible tracks

- DevOps
- Python / Machine learning
- Advanced web development
- Object orientated programming
- RPA business analyst
- Software testing
- Web design maybe

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.

All exercises will be done on AWS Cloud.

MACHINE LEARNING

We will investigate different methods for predicting the future using on data. We will learn how to write classifiers and how to use regression to improve our predictions.

We will learn how to do all of this in Python, and depending on your needs we will also look into SQL.

ADVANCED WEB DEVELOPMENT

We will dive deeper into web development, learn different JavaScript frameworks and also 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#.

We will make a small game using what we’ve learned.

RPA BUSINESS ANALYST

We will investigate what robotic process automation is, learn business analysis fundamentals and get a first introduction to develop for RPA.

This track is based on UIpath Academy and uses UIpath as RPA software.

SOFTWARE TESTING

We will learn the needed foundation for software testing with all its terminology and concepts. Afterwards we will be introduced to different fields in the testing world: Test analysis, technical test analysis, test automation and test management.

After the first 2 weeks of the course, the exam for the certificate on ISTQB foundation level is possible.

WEB DESIGN

We will plan and build a website project from the beginning following the UX process. There are several steps before the coding actually starts: from doing research, to testing, prototyping and design.

We will also look into Content Management Systems and get to know Search engine optimization and more.

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 Ressources

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

Let's code together

Last week's exercises

- Move Box
- About me
- The Book List
- To-do list
- Tip calculator
- Facebook clone

Exercises

How to

Do exercises first on your own or with a peer, then show and tell in the afternoon.

30 minutes per exercise

1. Calculate the Hamming Distance

Write a function to compare two strands of DNA and count the differences between them. The Hamming distance is only defined for sequences of equal length, so an attempt to calculate it between sequences of different lengths should not work.

Example: `GAGCCTACTAACGGGAT` vs `CATCGTAATGACGGCCT`

2. Transpose a matrix

Write a function that transposes a given matrix: rows become columns and columns become rows.

Example:

```
ABC
DE
```

results in

```
AD
BE
C
```

3. Convert arabic numbers to Roman

Write a function to convert from arabic (normal) numbers to Roman Numerals. The Romans wrote numbers using letters: I, V, X, L, C, D, M. There is no need to be able to convert numbers larger than about 3000.

Example: `7` returns `VII`