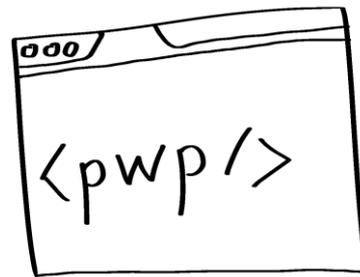


Practical Course: Web Development

Intro

Winter Semester 2016/17

Juliane Franze & Tobias Seitz



Today's Agenda

- Introduction
- Topics
 - Project topics
 - Frameworks
- Groups

Welcome!

Plan for the Semester

	Topic	Homework
28.10.	Organization,	---
04.11.	Requirements Engineering	Web-App Requirements
11.11.	Guest Talk: Julie Wagner on DevOps	UI Concepts / Wireframes
18.11.	NodeJS, APIs	Data Model / APIs
25.11.	Angular Part 1 / Polymer Part 1	Presentation
02.12.	Angular Part 2 / Polymer Part 2	Presentation
09.12.	Angular Part 3 / Polymer Part 3	First Stable Release
16.12.	Angular Part 4 / Polymer Part 4	Release
23.12.	No Session	
13.01.	Guest Talk: SIXT	Release
20.01.	Guest Talks: Maximilian Körner on Meteor + Coding Best Practices	Release
28.01.	Scaling	Release
03.02.	Final Project Presentation	
01.03.	Shipping	

What You'll Need

https://mimuc.slack.com/files/tobi.seitz/F2PCNRW4T/Required_Toolkit



Required Toolkit

Last edited 2 days ago

Over the course of the semester you will need to install a couple of tools. We're trying to keep it as simple as possible. To do the Break-Out exercises during the tutorials and to complete all assignments, make sure to install these tools **beforehand**. I've ordered them according to their importance.

1. Git. <https://git-scm.com/> install it, then go straight ahead to this tutorial, if you don't know git: <https://rogerdudler.github.io/git-guide/>
2. NodeJS + npm. <https://nodejs.org/en/> - If you have npm running, install these (you can do that from the Git-Bash):
 - a. Bower `npm install -g bower`
 - b. Express Generator `npm install -g express`
 - c. Polymer-CLI `npm install -g polymer-cli`
 - d. gulp `npm install -g gulp`
 - e. Browser-Sync `npm install -g browser-sync`
3. Text Editor / Web IDE - **choose one !** -

Syllabus

- About 45 minutes of input in every session
- 45 minutes of hands-on exercises
- Homework each week until the actual project work starts.

Our Expectations During the Semester

- Ask questions. A lot.
- Always attend the sessions on Friday
- Practice saying “Yes, **and...**” to build on others’ ideas
- Collaborate. Distribute tasks equally.
- Weekly status presentations. Every team member does at least one presentation (round robin procedure)

Your Expectations

- Take 5 minutes time to think about your expectations
- Afterwards, we're going to collect them and share them with you.

TOPICS

Topic 1: Platform for Media Technology Course

Elevator pitch:

“Teaching assistants need to manage a diverse range of tasks for the Media Technology Course. Apart from weekly tutorials, there are three practical courses within a very limited timeframe that have individual dates for each group of 4 students. Beforehand, tutors must be recruited and assigned to the dates. The online platform reduces the complexity of the organization because it guides the assistants through the entire process.”

Topic 2: User Study Management Platform

Elevator pitch:

“For researchers who need to manage study participants, the online platform allows them to create, announce, and run user studies. They can screen the participants and manage their compensation with either study credit or monetary incentives. Students can also sign up and see their study progress and participate in follow-up studies.”

Expected Outcome

- Production-level solution. We really want to use these platforms!
- Stability before features
- Localization: German + English
- Documentation:
 - Who did what? → Module breakdown
 - System set up (Dependencies, Installation, Usage)
 - API

Frameworks & Technology

- Front End:
 - Angular
 - Polymer, Web-Components
- Back End:
 - NodeJS + Express + LoopBack
 - SQL and NoSQL databases

Angular

Pro's

- Totally STRUCTURED client-side JS Framework
- 2-WAY data binding → easy & no boilerplate code
- CLEAN modularized code (Templates, Extension, Injection, Filter, Services, Factories)
- AMAZING support from debug tools and community

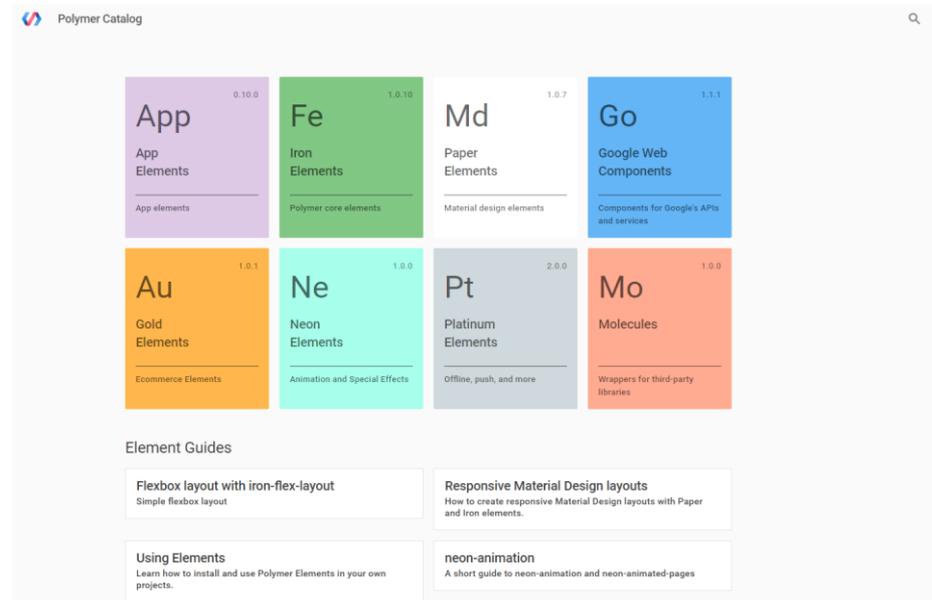
Con's

- don't try to think of DOM-manipulation like in JS
- you have to work structured - no scribble down of all functionality in one file

Polymer



- “Opinionated view on web components”
- Core idea: Extend HTML elements, encapsulate functionality
- <https://www.polymer-project.org/1.0/>
- Many element libraries, e.g. [Polymer Elements](#)



Polymer in Action

- Google Play Music: <https://play.google.com/music/>
- YouTube Gaming: <https://gaming.youtube.com/> (soon all of YouTube)
- USA Today Rio 2016: <https://rio16.usatoday.com/olympics-rio-2016/schedule-results>
- Electronic Arts: <http://www.ea.com/>
- Net-A-Porter: <https://www.net-a-porter.com/>
- GitHub:
<https://twitter.com/ebidel/status/464102546114506752>
- Google I/O 2016: <https://events.google.com/io2016/>

Polymer Evaluation

Pro

- Easy learning curve
- Thousands of modules out of the box
- Material design set up (optional)
- Less JavaScript-y.
- Very close to native browser API #UseThePlatform
- Strong support by Google
- ES6 compliant
- Out of the box responsiveness

Con

- Data flow paradigm
- If you're not careful, performance might suffer
- Widespread browser support yet to come (until then: polyfills)

Why didn't we choose Framework XYZ?

- We know that there is a plethora of front-end frameworks
- Angular and Polymer have gained momentum and will probably stay for a couple of years. If you understand their ideas, it will be easy to adapt them to React / Ember / Vue.
- Besides, we are experts in these Angular/Polymer. It wouldn't make sense to teach things we have to learn along with you.
- Interesting article:
<https://hackernoon.com/how-it-feels-to-learn-javascript-in-2016-d3a717dd577f#.xtrtzz9d9>

Groups

- 16 Participants / 2 Topics / 2 Frameworks
→ 4 groups à 4 people

	Angular	Polymer
Media Technology	Eugenia Nikolai Mareike Felix	Philipp Martin Sarah Bettina
User Study	Alex Mathis Jan Fabian	Marius Thomas Barbara Laura