Course information

build passing

The **Web technology materials and resources** of the 2019/20 Web and Database Technology course at TU Delft are listed here.

They were initially created for the 2018/19 edition of the course and have been updated for this year's edition.

The database materials (lectures/assignments) are not included here! You can find them on Brightspace! Due to the many code examples in the web technology lectures we opted to maintain the materials in a GitHub repository.

Table of Contents

Course instructors

The responsible instructors are Associate Professors Claudia Hauff (web) and Professor Alessandro Bozzon (databases). The responsible instructors can be reached at `cse1500-ewi@tudelft.nl`.

The web technology werkcolleges are led by Assistant Professor Julián Urbano. Azqa Nadeem covers some of the werkcolleges as well, will teach the web security lecture and is responsible for the web technology teaching assistants.

Web technology course book

The course book for the web technology part is **Learning Web App Development** by Semmy Purewal. This book covers HTML, CSS, JavaScript and Node.js/Express - exactly the technologies we cover in this course. The book is written for learners without any prior knowledge in these technologies. This book was published in 2014. This could be considered a feature or a bug ... we consider it a feature, as it means that a lot of the recently incorporated changes in the JavaScript language are **not** part of the book. Given that we only have one lecture to introduce the JavaScript language itself, the coverage of the JavaScript language in the book is sufficient and complete enough to allow everyone to start coding!

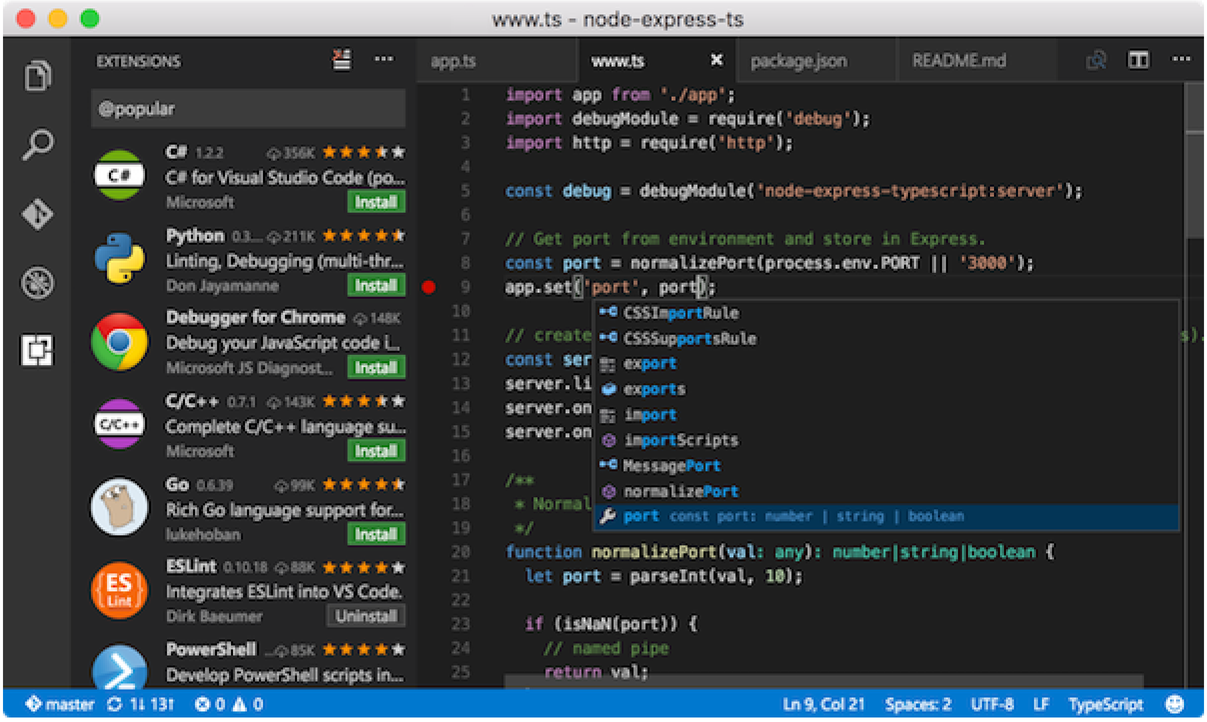
The book also covers version control basics (`git` to be specific) which can be ignored for the purposes of our lectures. You will learn all about `git` in the academic quarters that follow!

O'REILLY
Learning Web App Development
BUILD QUICKLY WITH PROVEN JAVASCRIPT TECHNIQUES
Semmy Purewal

Tooling

Recommended

Visual Studio Code (VSC) is the strongly recommended development environment for the web technology assignments. It is a free and open-source IDE available for all major operating systems. It was originally designed to support Node.js programmers (a server-side JavaScript framework we use in this class), but now has extensions for many programming languages. Best of all, **Visual Studio Code itself is written in JavaScript**!



```
import app from './app';
import debugModule = require('debug');
import http = require('http');

const debug = debugModule('node-express-typescript:server');

// Get port from environment and store in Express.
const port = normalizePort(process.env.PORT || '3000');
app.set('port', port);

// create
const ser
server.li
server.on
server.on

/**
 * Normal
 */
function normalizePort(val: any): number|string|boolean {
  let port = parseInt(val, 10);

  if (isNaN(port)) {
    // named pipe
    return val;
```

VSC offers **many extensions**. You can find a guide on how to browse and install them here. A list with all kinds of "delightful" extensions can be found here. Since this may be a bit overwhelming, we made a short list of the extensions we recommend to install for this course. For the programming assignments we strongly recommend the **VS Live Share** extension for those students that do not want to dive into `git` yet!

- ESLint, the most popular linting utility for JavaScript (a linter is a tool that analyzes source code to flag potentially poor code such as unused variables) - use it to improve your code.
- Bracket Pair Colorizer does what the name suggests.
- Quokka.js - a great tool to try out JavaScript snippets without hassle.
- VS Live Share - a collaborative real-time coding extension (especially useful if a student team works mostly remotely).
- Material Icon Theme to make finding the file you are after in your app directory a bit easier.
-
- If you like to add TODOs to your code, try Todo Tree, it makes sure you don't overlook any of your TODOs!
- Finally, a good theme for your IDE to make coding more enjoyable is also not a bad idea. Dracula is a popular one. Cobalt2 is also nice. If you want to pick based on visuals, head over to https://vscodethemes.com/.

If you need to be convinced about why Visual Studio Code is a great IDE to use, check out this podcast by Wes Bos and Scott Tolinski on this very topic.

Browser developer tools are built into all modern browsers; familiarize yourself with the developer tools of your favorite browser. In our lectures, we will showcase Firefox's developer tools - an elaborate description of them can be found at MDN.

If you are new to team programming, and want to use VSC's features efficiently, read our guide.

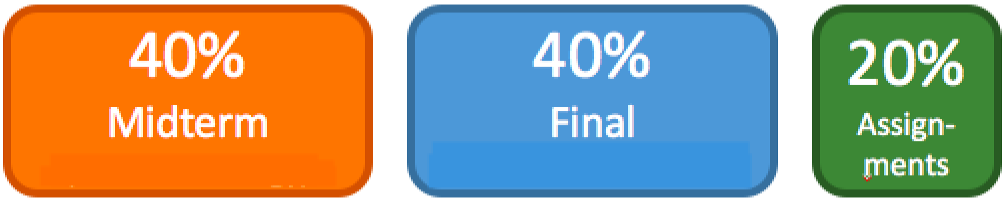
Required

The latest stable version of Node.js.

Two modern web browsers such as Chrome, Firefox, Edge, etc. to test your project code across browser implementations.

Telnet (for Assignment 4).

Course grading



Passing requirements: the overall grade must be **5.8** or higher AND the midterm grade must be at least a **5** AND the final exam grade must be at least a **5** in accordance with TU Delft's exam regulations.

Assignments are done in **pairs of two students** and are graded by student assistants in a binary manner (pass or fail). There are six assignments in total, three cover database topics and three cover web technology topics. While the assignments together make up only 20% of the final grade and are not required, we strongly recommend to tackle the assignments - they cover a lot of exam materials in a practical manner!

We have two assessment moments: one for the database assignments and one for the web technology assignments. The assessment is interview-based. Each assessment moment can yield anything between 3x pass and 3x no-pass. The grading is conducted per student.

The table below shows how the number of passes (maximum 6: 3x database and 3x web) are converted to the assignment grade:

Number of passes	Grade
6	10.0
5	8.3
4	6.7
3	5.0
2	3.3
1	1.7
0	0.0

Course topics

Both databases and web are covered in eight lectures each. The midterm covers only database topics. The final covers only web technology topics.

Week	Monday	Wednesday	Thursday
2.1	Introduction to databases	Relational database systems	---
2.2	Design of relational database systems	SQL 1	---
2.3	SQL 2	NoSQL: graph stores	---
2.4	NoSQL: document stores	SQL (advanced)	---
2.5	---	---	**Midterm exam**
2.6	HTTP	HTTP/HTML5	---
2.7	JavaScript	Node.js	---
2.8	CSS	Node.js (advanced)	---
2.9	Cookies and sessions	Web security	---
2.10	**Final exam**	---	---

Web technology tutorials

We offer three tutorials (interactive sessions to solve small problems) for interested students - these tutorials are optional and do not count towards the 140 hours of this course:

- Week 2.7: JavaScript
- Week 2.8: Node.js
- Week 2.9: old exam questions

Check the MyTimeTable app to determine which tutorial session to attend!

Sample exams

Until (and including) the 2017/18 edition, the web and database topics were interwoven (usually 1 lecture of web and 1 lecture of database topics per week) and thus for most years, the midterm and final exams also contain questions on both topics. For the 2019/20 edition, the final exam will contain about 40 multiple choice questions on the web technology topics.

Exam topics are all topics covered in the lectures and the required readings. Content of the recommended activitivies/readings that goes beyond what is covered in the lectures/course book is not exam material.

- Midterm 2015/16 (with answers)
- Final 2015/16 (with answers)
- Midterm 2016/17
- Final 2016/117
- Midterm 2018/19 (with answers)
- Midterm Resit 2018/19 (with answers)
- Final 2019/20 with answers

Assignments & assessment

There are six assignments in this course in total.

Assignments are worked on in **teams of 2 students**. *We do not allow larger teams; we do not allow teams of 1.*

Assignments 1-3 cover database topics and can be found on Brightspace. Assignments 4-6 cover web technology topics:

- Assignment 4 (web)
- Assignment 5 (web)
- Assignment 6 (web)

Assessment: the three web assignments are assessed in a **single** interview session with a TA; the interview lasts 20-25 minutes. Each of the three assignments can be passed or failed (binary judgment, there are no grade levels) on an individual basis. The assessment rubric for the web assignments is available here. The assessment takes place during week 2.9.

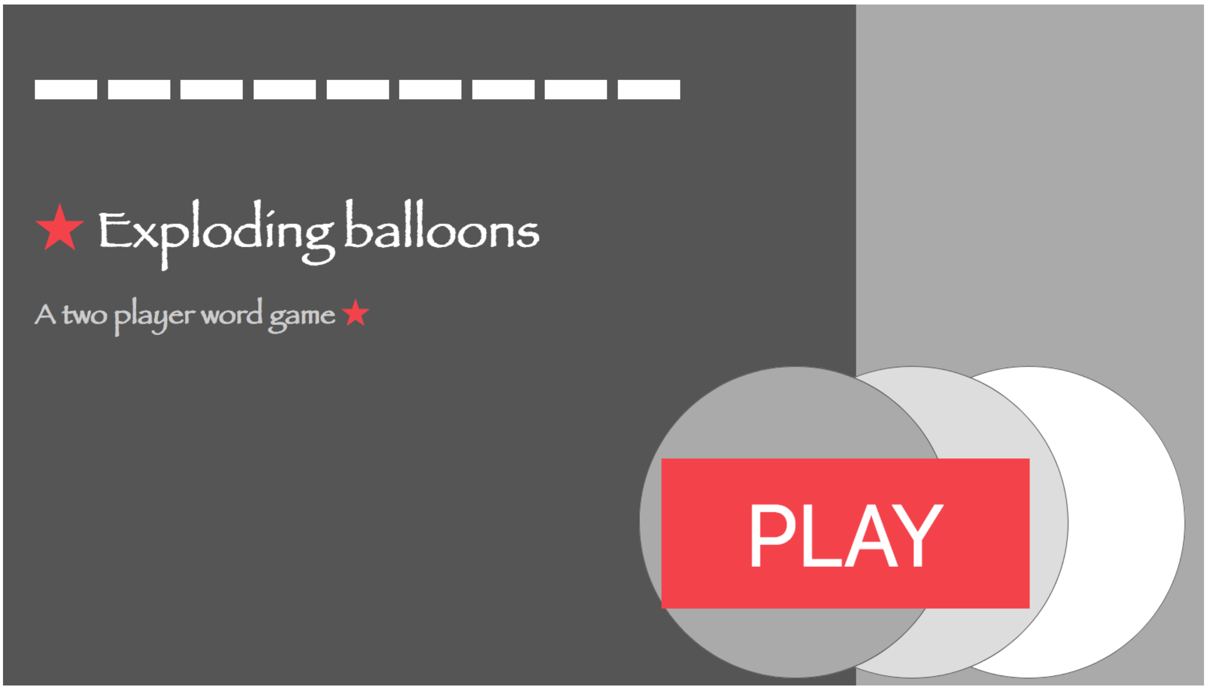
All assignments must be submitted in Brightspace before your assessment timeslot, and in any case no later than January 22, 2020 at 23:30 (hard deadline). All submissions are timestamped. Late submissions (or re-submissions) will not be assessed. We are going to check if your submitted work is consistent with the one used for assessment.

FYI, in previous years, roughly 90% of groups attempted and passed the first assignment, about 60% of groups passed two assignments and about 30% of groups all three assignments.

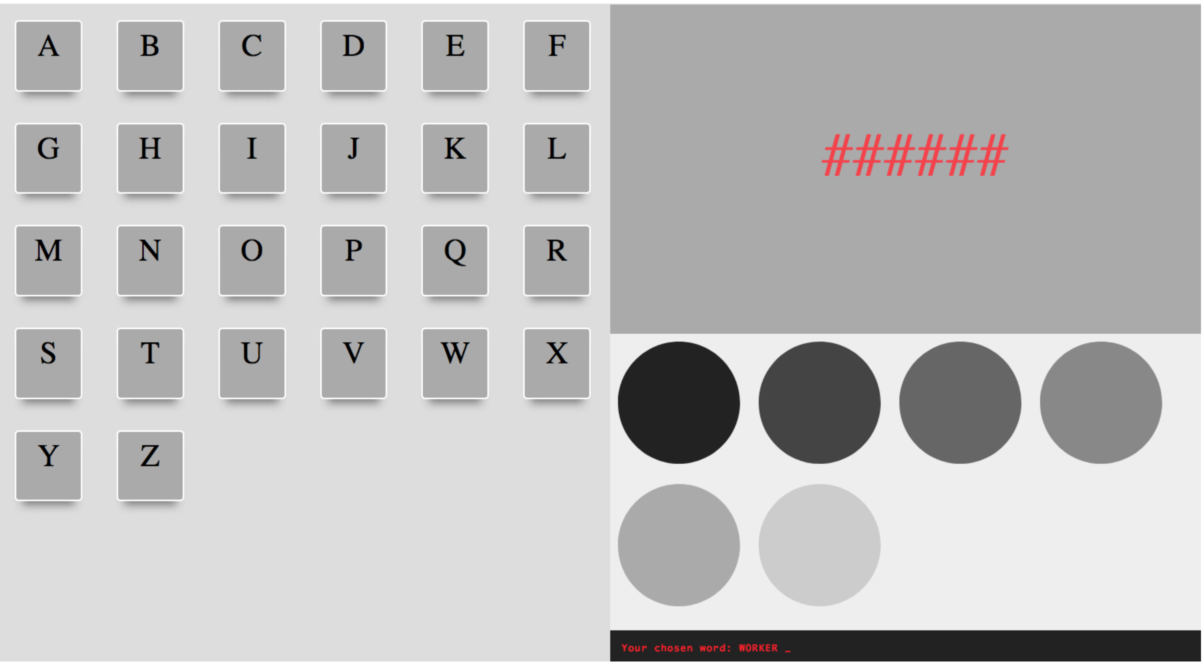
Demo application

A demo board game application (a word guesser) is available as well. It has has been implemented along the lines of the assignments.

The demo app, just as listed in this year's board game project, consists of two screens, a splash screen:



and a game screen:



```
Your chosen word: WORKER _
```

Interactive web technology exercises

Students who do not have a lot of prior knowledge often struggle to get started. Here is a list of useful interactive exercises and how they match up with each lecture. Each of these exercises is small, taking just a few minutes (sometimes less than a minute) to solve.

Twitter accounts to follow

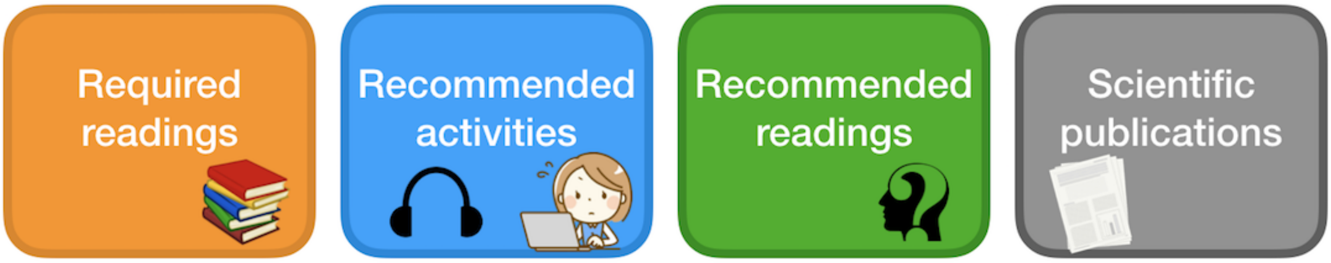
Twitter is a great resource to hear the latest and greatest about the web stack. Here is a list of accounts we recommend to follow

- Franziska Hinkelmann, working on Node.js at Google
- Lin Clark makes technical challenges accessible to the wider public for Mozilla
- V8, Google's JavaScript engine
- JavaScript Daily, does exactly as the name suggests
- AmsterdamJS, local JavaScript community
- Node.js (of course)
- Axel Rauschmeyer, an author of popular JavaScript books
- Addy Osmani, working on Google Chrome and well known for his JavaScript design patterns book
- Wes Bos, a popular teacher of the web stack
- Mozilla Hacks, official Mozilla account for web developers
- Visual Studio Code (of course)
- JavaScript Conference Series - look out for their conference talks, usually posted on YouTube
- Chrome DevTools, tips and tricks of the dev tool trade
- Syntax, podcasts for web developers

Web development resources

The practical assignments of this work often require looking up Web development specifics. Two good resources, in general, for Web engineering are Mozilla's MDN portal and Google's Web Fundamentals.

Lecture material types



For each lecture we provide a range of materials:

- **Required readings**: most of the required readings are book chapters of the web course book. The book introduces each technology from scratch. The lectures go beyond what is covered in the book and assume that you have read the corresponding book chapter(s) **before** the lecture.

- **Recommended activities**: activities that we think will help you get a better grasp of the different web technologies. Activities are either programming exercises or podcasts (there are some great ones out there!). Don't see those as a complete list to work through, that is impossible given the time!
- **Recommended readings**: these may be tweets, blog posts or book chapters that help you understand some of the introduced concepts better; again, do not attempt to read all of the materials.
- **Relevant scientific publications**: we are often asked why this kind of course is part of the computer science curriculum. Isn't this "just" programming? To showcase how web technologies are researched, we here list a number of scientific publications that contribute to our understanding of (the use of) web technologies.

If you are interested in knowing what other universities teach in terms of web technologies, take a look at this list of courses with a web focus.

List of university-level courses with a Web focus

- Deep Dive Into Modern Web Development /U Helsinki
- Design and Implementation of Software for the Web /GMU
- Research on Web Technologies /NTNU
- Web Applications /Stanford
- Web Programming Fundamentals /Stanford
- Creating Modern Web Applications /Brown
- Web Security /NUS
- Programming Languages for Web Applications /Virginia
- Web Development /MIT
- Web Technologies /USC
- Web Development /NYU
- Cutting-edge Web Technologies /Berkeley
- Web Technologies /U Texas
- Web Programming /U Washington
- Web and Mobile Systems /U Virginia
- Javascript /UPenn

Web lectures

All lecture transcripts and assigments (which were developed in Markdown) were **automatically** converted to PDF format, for those students preferring PDFs. Warning: due to the automatic conversion not all pages will look great. You can find them in the generatedPDFs folder here on GitHub.

Lectures 1-8 cover database topics and can be found on Brightspace.

Lecture 9

The lecture covers HTTP (intro slides, lecture slides, transcript).

- Required readings: *none*
- Recommended activity:

  - 🎧 Listen to this podcast on how to learn new things quickly in the Web technology world.
  - 📺 A short video (in Dutch) by CWI (Centrum Wiskunde & Informatica) to celebrate 30 years since the Netherlands was connected to the public Internet!
- Recommended readings:
  - 📕 Chapters 1, 2 and 3 of HTTP: The Definite Guide (O'REILLY 2002).
  - MDN overview of HTTP.
  - Developer tools overview of your favorite web browser (e.g. Firefox or Chrome).
  - A brief history of HTTP.
  - Browser fingerprinting showcases how seemingly innocuous data can identify users.
  - A crash course on HTTP and DNS by Mozilla (among other things).
  - A detailed overview of the history of web standards and how to make use of them.
- Relevant scientific publications:
  - Ihm, S. and Pai, V.S., 2011. Towards understanding modern web traffic. In Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference (pp. 295-312). ACM.
  - Naylor, D., Finamore, A., Leontiadis, I., Grunenberger, Y., Mellia, M., Munafò, M., Papagiannaki, K. and Steenkiste, P., 2014. The cost of the S in HTTPS. In Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies (pp. 133-140).
  - Falaki, H., Lymberopoulos, D., Mahajan, R., Kandula, S. and Estrin, D., 2010. A first look at traffic on smartphones. In Proceedings of the 10th ACM SIGCOMM conference on Internet measurement (pp. 281-287).

Lecture 10

This lecture covers web design and HTML5 (lecture slides, transcript).

- Required readings:
  - ⚠️ Chapter 2 of the web course book.
  - ⚠️ Introduction to HTML forms (ignore the section on *Basic form styling*, we will cover CSS in a later lecture) and Sending form data (up to and including the section on *Viewing HTTP requests*).
- Recommended activity:
  - 🎧 Listen to this podcast with Tom Dale discussing modern web applications and the move from web sites to web apps and an almost *native* experience.
- Recommended readings:
  - 📕 Chapter 3 of the Definite Guide to HTML5.
  - Web Fundamentals by Google.
  - Examples of what modern web technologies can achieve in Chrome (if you use another browser, not all examples may work as expected).
  - Atomic Design by Brad Frost
- Relevant scientific publications:
  - Fernandez, A., Insfran, E. and Abrahão, S., 2011. Usability evaluation methods for the web: A systematic mapping study. Information and Software Technology, 53(8), pp. 789-817.
  - Sonderegger, A. and Sauer, J., 2010. The influence of design aesthetics in usability testing: Effects on user performance and perceived usability. Applied Ergonomics, 41(3), pp. 403-410.

  - Mathur, A., Acar, G., Friedman, M., Lucherini, E., Mayer, J., Ghetty, M. and Narayanan, A., 2019. Dark Patterns at Scale: Findings from a Crawl of 11KShopping Websites, ACM Human-Computer Interaction, Vol. 3, CSCW, Article 81.

Lecture 11

This lecture covers client-side JavaScript (lecture slides, transcript).

- Required readings:
  - ⚠️ Chapter 4 of the web course book.
- Recommended activities:
  - Interactive JavaScript exercises.
  - 🎧 Listen to this podcast by Wes Bos and Scott Tolinski on debugging.
  - JavaScript30: 30 day vanilla JS coding challenges
- Recommended readings:
  - 📕 Learning JavaScript design patterns, in particular the sections on the constructor pattern and the module pattern.
  - JavaScript for impatient programmers: online quizzes.
  - MDN's introduction to JavaScript objects.
  - The dynamic nature of JavaScript makes optimization tricky (blog post).
  - Why do we need all those fancy tools for JavaScript development nowadays? (blog post).
  - Learn how to debug with Firefox devtools.
  - Array.prototype.sort is now stable in V8 (tweet): shows that even basic mechanisms like sorting are still being changed in mature implementations.
  - Tooling and conventions are vital in the fast-paced world of JavaScript.
  - The State of JavaScript 2017.
  - Clean Code concepts adapted for JavaScript: a highly popular GitHub repository with many examples of good and bad code patterns.
- Relevant scientific publications:
  - Charland, A. and Leroux, B., 2011. Mobile application development: web vs. native. Queue, 9(4), p. 20.
  - Mowery, K., Bogenreif, D., Yilek, S. and Shacham, H., 2011. Fingerprinting information in JavaScript implementations. In Proceedings of W2SP (Vol. 2, No. 11).
  - Ratanaworabhan, P., Livshits, B. and Zorn, B.G., 2010. JSMeter: Comparing the Behavior of JavaScript Benchmarks with Real Web Applications. In Proceedings of the 2010 USENIX conference on Web application development.
  - Lin, J., 2018. Computing without Servers, V8, Rocket Ships, and Other Batshi*t Crazy Ideas in Data Systems. In Proceedings of DESIRES. A quote ... "*So, the future is. . . JavaScript? Once we get beyond the fact that JavaScript is an undeniably shitty language on which to build an interlingual execution platform, there is at least some so-crazy-it-might-actually-work appeal to this idea.*"
  - Patra, J., Dixit, P.N. and Pradel, M., 2018. ConflictJS: finding and understanding conflicts between JavaScript libraries. In Proceedings of the 40th International Conference on Software Engineering, pp. 741-751.
  - Jangda, A., Powers, B., Berger, E. D., & Guha, A., 2019. Not so fast: analyzing the performance of webassembly vs. native code. In Proceedings of the 2019 USENIX Annual Technical Conference, pp. 107-120.

Lecture 12

This lecture covers Node.js (lecture slides, transcript).

- Required readings:
  - ⚠️ Chapters 5 & 6 of the web course book. We will cover Node.js in the lecture from sratch, use the reading to get a first idea of what Node.js is about!
- Recommended activities:
  - Interactive Node.js exercises.
- Recommended readings:
  - The MDN documentation on Asynchronous JavaScript is an excellent source to learn about the topic in more detail.
  - Node.js State of the Union 2017.
  - The world runs on Node.js (Twitter thread).
  - If you are interested in learning how best to get involved in the Node community, read this blog post on how to go about it.
- Relevant scientific publications:
  - Chaniotis, I.K., Kyriakou, K.I.D. and Tselikas, N.D., 2015. Is Node.js a viable option for building modern web applications? A performance evaluation study. Computing, 97(10), pp.1023-1044.

Lecture 13

This lecture covers CSS (lecture slides, transcript).

- Required readings:
  - ⚠️ Chapter 3 of the web course book.
- Recommended activities:
  - 📺 Watch this talk by Rachel Andrew to learn more about the long (and painful) history of CSS.
  - 📺 Watch this talk by Mandy Michael on text effects in CSS. Data attributes and pseudo-elements are a powerful combination!
  - 🎧 Listen to this podcast by Wes Bos and Scott Tolinski on the present and future of CSS.
  - If you are looking for styling challenges, take a look at the weekly CodePen Challenges.
  - At https://flukeout.github.io/ you find a small game that allows you to practice CSS selectors.
- Recommended readings:
  - 📕 Chapters 1-4 and chapter 13 of The Book of CSS3: A developer's guide to the future of web design by Peter Gasston (2nd edition, 2014).
  - To make sense of CSS positioning, take a look here.
  - If you want to look behind the scenes, read Mozilla's blog post on their CSS engine Quantum CSS.
  - After studying this lecture you have the basic knowledge to consider a career as a Front-End Developer
- Relevant scientific publications:
  - Meyerovich, L.A. and Bodik, R., 2010, April. Fast and parallel webpage layout In Proceedings of the 19th international conference on World wide web (pp. 711-720). ACM.

Lecture 14

This lecture covers advanced Node.js concepts (lecture slides, transcript).

- Required readings: *none*
- Recommended activities:
  - An interactive ejs playground can be found here.
- Recommended readings:
  - To learn more about ejs, take a look at its GitHub repository.
  - To learn more about middleware and Express, take a look at the Express documentation.
  - An overview of best practices in Node.js.
  - Slide deck on High Performance JS in V8.
- Relevant scientific publications:
  - Fard, A.M. and Mesbah, A., 2013, September. JSNose: Detecting javascript code smells. In Source Code Analysis and Manipulation (SCAM), 2013 IEEE 13th International Working Conference on (pp. 116-125). IEEE.
  - Nasehi, S.M., Sillito, J., Maurer, F. and Burns, C., 2012, September. What makes a good code example?: A study of programming Q&A in StackOverflow. In 2012 28th IEEE International Conference on Software Maintenance (ICSM) (pp. 25-34). IEEE.

Lecture 15

This lecture covers cookies and sessions (lecture slides, transcript).

- Required readings: *none*
- Recommended activities: *none*
- Recommended readings:
  - 📕 Chapter 9 of Web Development with Node & Express by Ethan Brown.
  - Instead of cookies to recognize a user, we can also resort to the more stealthy version of browser fingerprinting.
- Relevant scientific publications:
  - Roesner, F., Kohno, T. and Wetherall, D., 2012. Detecting and defending against third-party tracking on the web. In Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation (pp. 12-12). USENIX Association.
  - Park, J.S. and Sandhu, R., 2000. Secure cookies on the Web. IEEE internet computing, 4(4), pp.36-44.
  - Franken, G., Van Goethem, T. and Joosen, W., 2018. Who left open the cookie jar? a comprehensive evaluation of third-party cookie policies. In Proceedings of the 27th USENIX Security Symposium, pp. 151-168.

Lecture 16

This lecture covers web security topics (lecture slides), (transcript).

- Required readings: *none*
- Recommended activities:
  - 🎧 Listen to this podcast on how to navigate the fast-paced Web technology industry.

  - 📺 The Power of the Web Platform - a talk by Feross Aboukhadijeh about what annoying things are possible on the web. Entertaining! Be sure to not open the accompanying website while in a quiet space.
- Recommended readings:
  - Stanford's 2019 Web Security course has a very relevant set of slides.
  - 📕 If you want to know everything there is about security, read Ross Anderson's Security Engineering book. Chapter 21 is most pertinent to the web security lecture (warning: this is an extensive read).
  - Stanford's Computer and Network Security course has a number of lectures on web security (PDFs: here, here, here and here).
  - CERN's web security lecture.
  - The Open Web Application Security Project provides an extensive list of practical tips, best practices and further readings on the topic.
  - Node.js security best practices.
- Relevant scientific publications:
  - Aggarwal, G., Bursztein, E., Jackson, C. and Boneh, D., 2010. An Analysis of Private Browsing Modes in Modern Browsers. In Proceedings of the 19th USENIX conference on Security. USENIX Association.
  - Kieyzun, A., Guo, P.J., Jayaraman, K. and Ernst, M.D., 2009. Automatic creation of SQL injection and cross-site scripting attacks. In Proceedings of the 31st International Conference on Software Engineering (pp. 199-209). IEEE Computer Society.
  - Von Ahn, L., Maurer, B., McMillen, C., Abraham, D. and Blum, M., 2008. recaptcha: Human-based character recognition via web security measures. Science, 321(5895), pp.1465-1468.
  - Acar, G., Eubank, C., Englehardt, S., Juarez, M., Narayanan, A. and Diaz, C., 2014. The web never forgets: Persistent tracking mechanisms in the wild. In Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (pp. 674-689). ACM.