T-213-VEFF: Web Programming I

L1: Introduction

Grischa Liebel

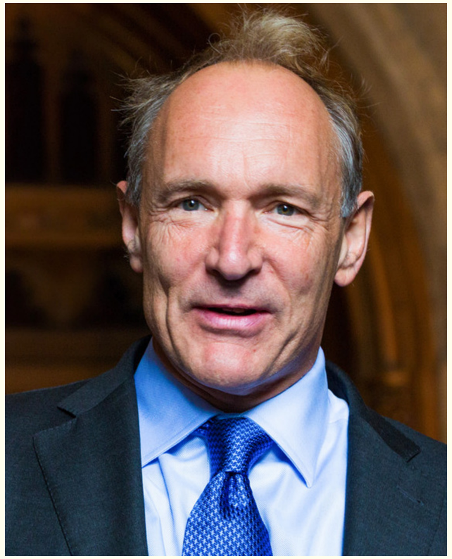
This Lecture

1. A short history of the web
2. The Team
3. Why do the slides look weird?
4. What are we covering here?
5. Organisation

A short history of the web

WWW

- Invented by Tim Berners-Lee (1989)
- Public from 1991
- First website in history: Link
- Contains **hypertext** or **hypermedia**
- Formatting in HyperText Markup Language **(HTML)**
- Documents are viewed in a **browser**



Websites in the old days…

- OJ Simpson trial (CNN)
- Dole-Kemp Presidential Campaign 96
- Welcome to Netscape
- Lego (96)

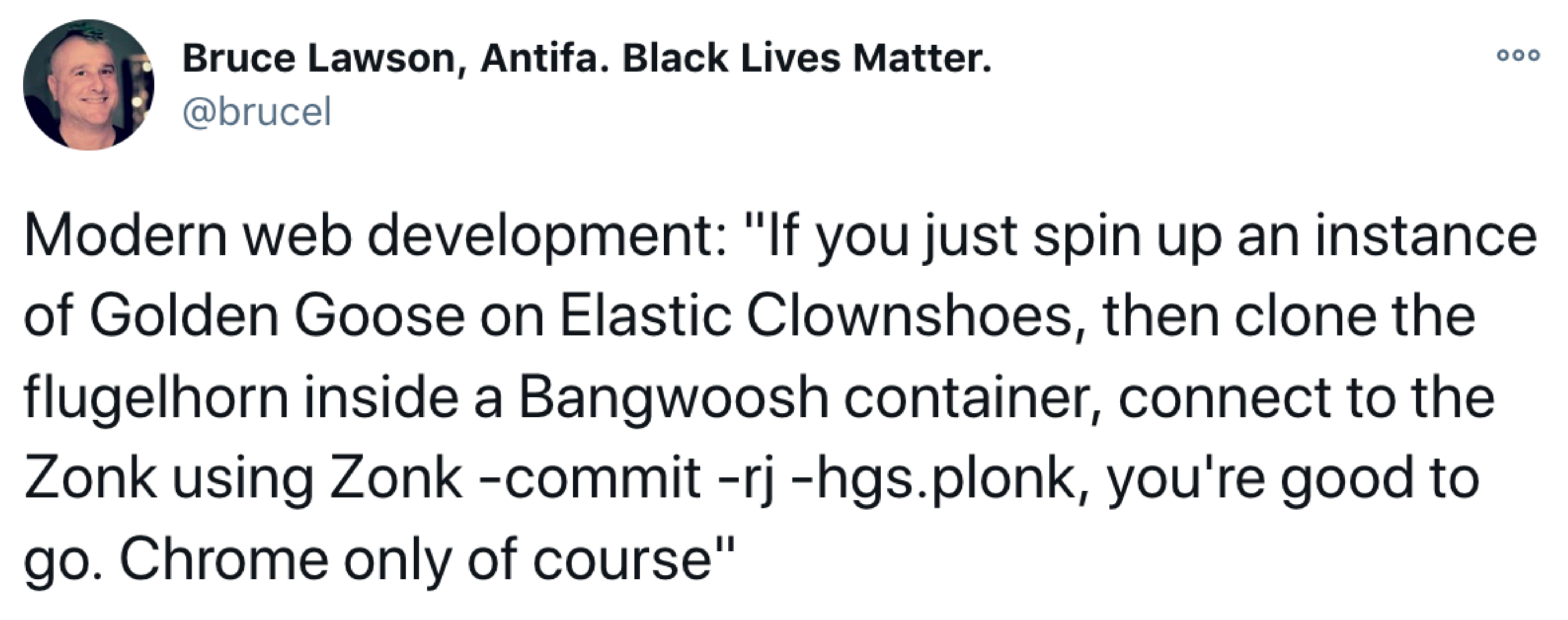
Browser Wars

- Internet Explorer vs. Netscape Navigator
  - Distribution
  - Costs
  - Bundles
- Websites designed for one browser only (proprietary tags, features)
- Coming back in “modern times” (e.g., WebKit)

What's so special with web applications?

- Fast changes in technology
- Accessible to "everyday people"
  - No training
  - Well-designed interface is a large competitive advantage
  - Direct access to large user base
  - **Societal responsibility!**
- World wide
- Common entry point to programming

It's getting complicated



Bruce Lawson, Antifa. Black Lives Matter.
@brucel

Modern web development: "If you just spin up an instance of Golden Goose on Elastic Clownshoes, then clone the flugelhorn inside a Bangwoosh container, connect to the Zonk using Zonk -commit -rj -hgs.plonk, you're good to go. Chrome only of course"

It's getting complicated

- There are a number of core languages/technologies
  - HTML, CSS, JavaScript
- And a lot of surrounding frameworks, libraries, methods
  - Frontend/Backend frameworks (e.g., React, Node.js)
  - Containerisation (e.g., Kubernetes, Docker)
  - Deployment and related services (e.g., AWS, Microsoft Azure)
  - Persistence solutions (e.g., MySQL, MongoDB, Firebase)
  - Development, testing and build tools (e.g., VSCode, ESLint, Babel, WebPack)

The Team

About me

- PhD in Software Engineering
- I do research on Software Engineering
  - Applied, with companies
  - Mainly processes, requirements, modelling
- I sometimes consult
  - E.g., government tenders

TAs

Birta Ósk Theodórsdóttir

Piotr Abramek

Hilmar Páll Stefánsson

Sunna Rún Þórarinsdóttir

?

Matthías Ólafur Matthíasson

Jóhann Rúnarsson

Sara Helena Bjarnad. Blöndal

Baldur Óli Barkarson (Aku)

Why are my slides so weird?

- More accessible to neurodiverse people
  - E.g., ADHD, Dyslexia, Autism Spectrum

- We provide all slides in this alternative style and a "traditional" style
  - The same for assignments
  - I use the alternative style in live streams

What are we covering here?

Language/Framework Focus

- Plain HTML, CSS, JavaScript
  - In particular: No frontend framework like React (Web Programming II)
- Backend framework Node.js + express.js
  - Because it uses JavaScript
- No persistence (other course)
- Some corresponding testing tools (Mocha/Chai)

Schedule is preliminary

Foundations:	Week 1: Introduction Week 1: Network basics and HTTP
Frontend:	Week 2: HTML Week 3: CSS Week 4: JavaScript
Backend:	Week 7/8: REST Week 7/8: Server-side JavaScript
Test & Debugging:	Week 5: Client side Week 10: Server side
Security:	Week 9: Web Security

Weeks 6, 11, 12:
Buffer

Learning Outcomes (LOs):

Knowledge and comprehension

LOs before each lecture!

- define and contrast client-side and server-side web applications
- summarise the content of HTTP requests and responses
- list different HTTP methods and explain their purpose
- explain the features of the different HTTP methods
- define and explain key language concepts of HTML, CSS, and JavaScript
- define accessibility for web applications and give examples for accessible/not accessible code
- predict the behaviour and look of a web application based on its source code
- predict the behaviour of asynchronous JavaScript code
- discuss web application testing and contrast different testing techniques
- summarise the different constraints of REST in relation to RESTful APIs
- discuss the correctness of HTTP response status codes for different endpoints of a RESTful API
- list and explain the most important web security threats according to the OWASP TOP 10

LOs: Application and analysis

- develop basic client-side web applications using HTML, CSS, and JavaScript
- make use of AJAX to enrich web applications with asynchronous behaviour
- debug and test basic client-side web applications
- analyse web application source code for errors
- choose the correct HTTP request method for different REST endpoints
- build a RESTful backend application
- make use of a database to persist data in a backend application
- analyse an existing RESTful API and point out shortcomings
- deploy a server-side application to an online cloud provider
- test and debug server-side applications
- develop tests for common web security threats
- inspect web application source code for potential security threats

LOs: Synthesis and evaluation

- propose improvements to web application source code
- improve existing web application source code
- assess existing code for errors and security vulnerabilities
- compare different testing techniques for web applications
- design a RESTful API according to given requirements
- convert a backend API so that it conforms to the REST style
- debate the importance of testing and debugging for web application development
- debate the societal role of web applications and the ethical impact this has on web development

Tools: VSCode as an IDE

- An IDE for different programming languages
  - By Microsoft
- Exists for Windows, Linux and Mac OS X
- Lots of useful extensions
  - syntax highlighting for different languages
  - linting
  - bracket pair colouring

Tools: Web Browser

- To display web pages
- We use Firefox and Chrome to test things!
- We expect you to do so as well
  - Let me know if you have any issues with this

Tools: Postman

- Easy way to test HTTP requests
- Recommended (especially when we get to REST)
  - Alternative: curl from the command line
- Necessary from Week 7 onwards

Tools: Node.js/npm

- JavaScript runtime environment
- Runs JavaScript outside of the browser
- npm used for dependency management

- Necessary from Week 7 onwards

Organisation

Time usage

- Lecture preparations (reading/watching recordings): 2 hours
- In-class session: 90 min
- Labs: 90 min
- Assignments/Practice: 6 hours

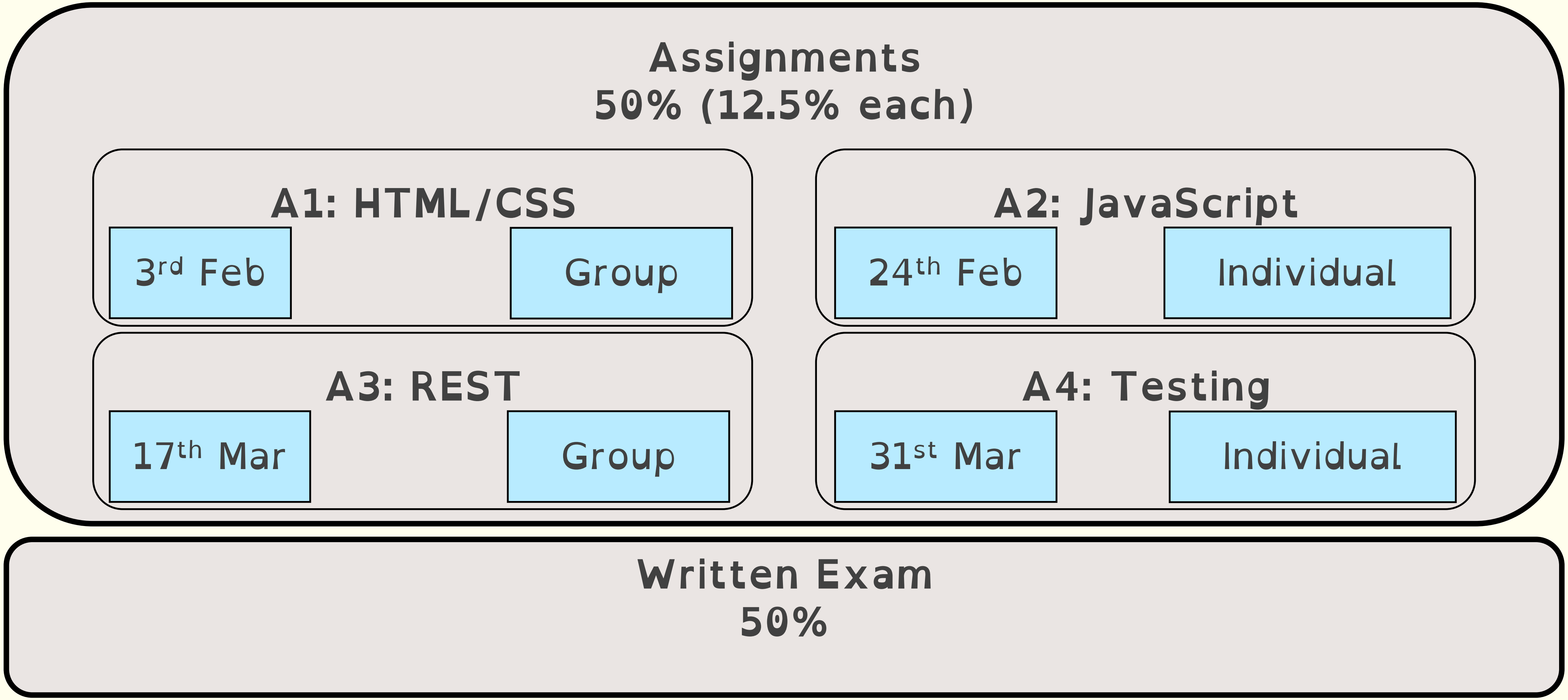
Pre-recordings vs. Lectures

- You watch the recordings before class
- In class, we only do repetition/examples
  - I assume that you watched the recording before
  - Quiz (ungraded) at the beginning
  - **Preferably questions/requests from Piazza - otherwise improvisation**

Literature

- Course book:
  Semmy Purewal,
  Learning Web App Development, O'Reilly
  - In addition to the slides
    (I will give chapters for each lecture)
  - Not a perfect overlap - just supplementary reading
- I will provide articles and tutorials

Course Overview



Assignments
50% (12.5% each)

A1: HTML/CSS

3rd Feb

Group

A2: JavaScript

24th Feb

Individual

A3: REST

17th Mar

Group

A4: Testing

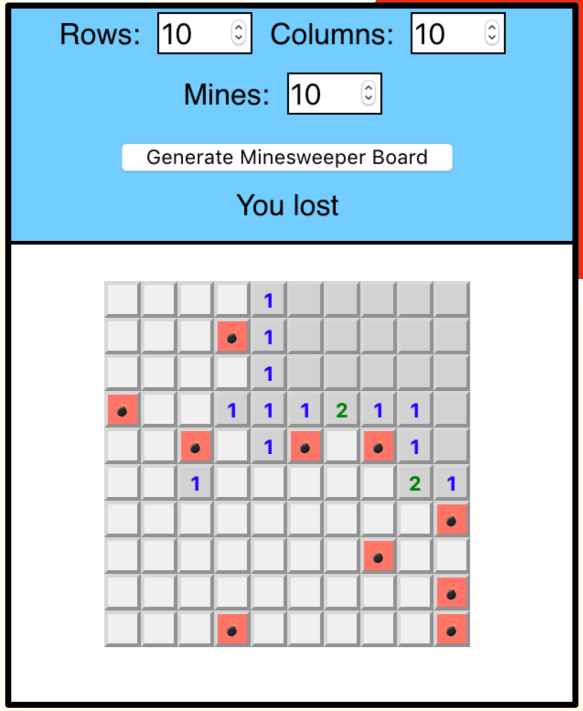
31st Mar

Individual

Written Exam
50%

Assignments

- Assignments are not straightforward
  - Not a step-by-step list of instructions
  - Rather a problem to solve
  - Understanding/breaking down the problem is part of the assignment (often the hardest part!)
- It's not uncommon that you have no idea how to start
- Start early, also if you have previous knowledge/experience
  - You might need discussions, support, etc.
  - **Especially the individual assignments**



Rows: 10 Columns: 10

Mines: 10

You lost

Difficulty

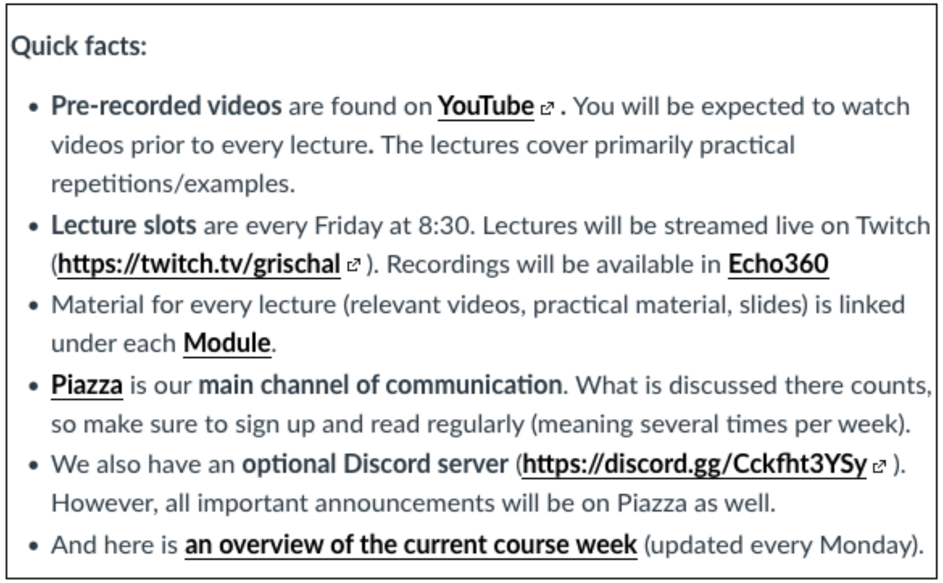
"[We need] more material on how to do things,
instead of making us google how to do every single assignment,
which in some cases doesn't even get us anywhere."

- Finding the right (and up-to-date) information is one of the essential skills for web development!
  - It's a large part of this course

- Problem solving is one (maybe the) key learning outcome of a BSc degree

Overview: Websites

- Canvas for overview and information only - links to everything
- Piazza for discussions, QnA
- YouTube for pre-recorded videos
- Twitch for live streams
- Echo360 for live stream recordings
- Discord for Dæmatímar & informal discussions



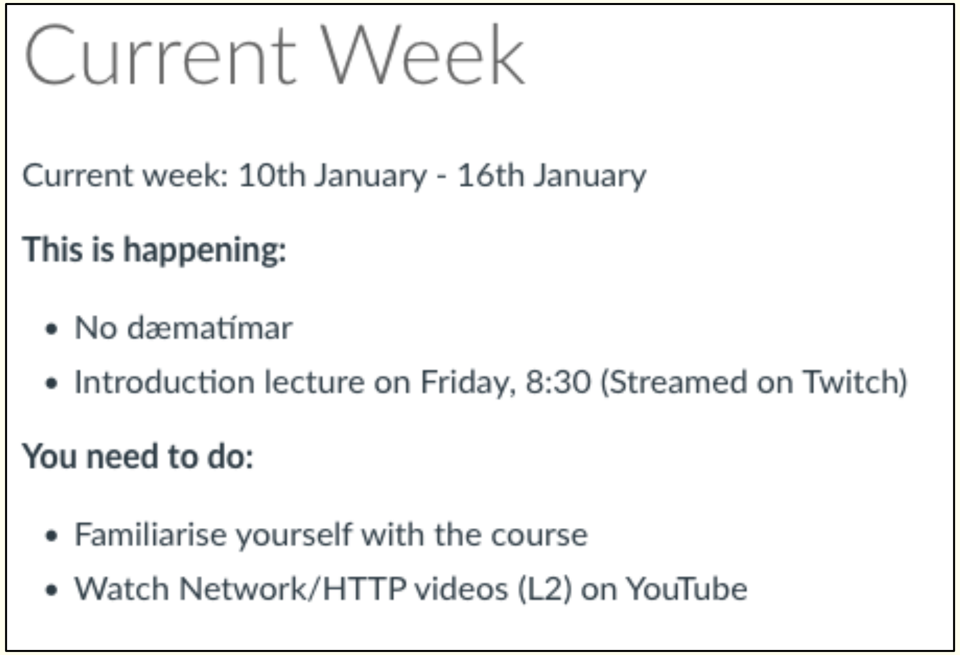
Quick facts:

- **Pre-recorded videos** are found on **YouTube** . You will be expected to watch videos prior to every lecture. The lectures cover primarily practical repetitions/examples.
- **Lecture slots** are every Friday at 8:30. Lectures will be streamed live on Twitch (**https://twitch.tv/grischal**). Recordings will be available in **Echo360**
- Material for every lecture (relevant videos, practical material, slides) is linked under each **Module**.
- **Piazza** is our **main channel of communication**. What is discussed there counts, so make sure to sign up and read regularly (meaning several times per week).
- We also have an **optional Discord server** (**https://discord.gg/Cckfht3YSy**). However, all important announcements will be on Piazza as well.
- And here is **an overview of the current course week** (updated every Monday).

Quick facts on Canvas course page

Overview: "Current Week"

- "Current week" page on Canvas shows a quick overview of the current week
- Updated every Monday



Current Week

Current week: 10th January - 16th January

This is happening:

- No dæmatímar
- Introduction lecture on Friday, 8:30 (Streamed on Twitch)

You need to do:

- Familiarise yourself with the course
- Watch Network/HTTP videos (L2) on YouTube

Canvas

- Canvas is used throughout the entire course
- All assignment hand-ins on Canvas

- No announcements through Canvas: All on Piazza

Piazza

- Heavy use of Piazza
  - You are expected to be active regularly
  - Important notifications there
  - Not only to talk to me, but to **clarify to each other**
  - What is being discussed publicly on Piazza counts!
- Avoid private messages to me if it concerns everybody
- Do not comment on Canvas assignments (i.e., grading results)
  - They get lost very easily

YouTube

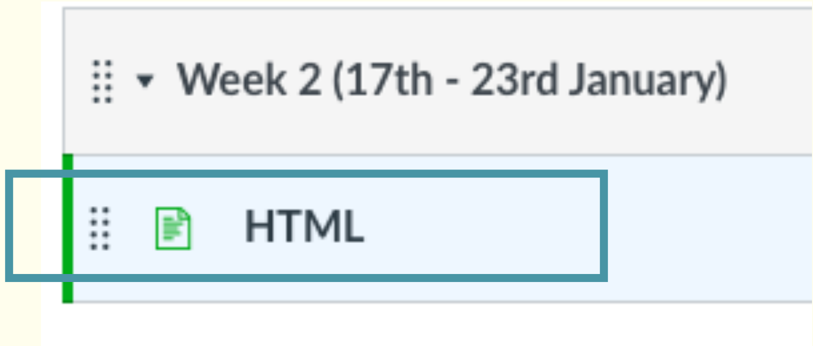
- Pre-recorded videos in YouTube playlist
  - https://www.youtube.com/playlist?list=PLCTWqbu-D5bP3oIuxW-re51lWaIB_RCaU

Twitch

- All live streams over Twitch (https://twitch.tv/grischal)
- To comment/chat: Verified email needed
    - To avoid spam
    - Please keep it professional
- Streams will be recorded and uploaded on Echo360
- Fridays, 8:30

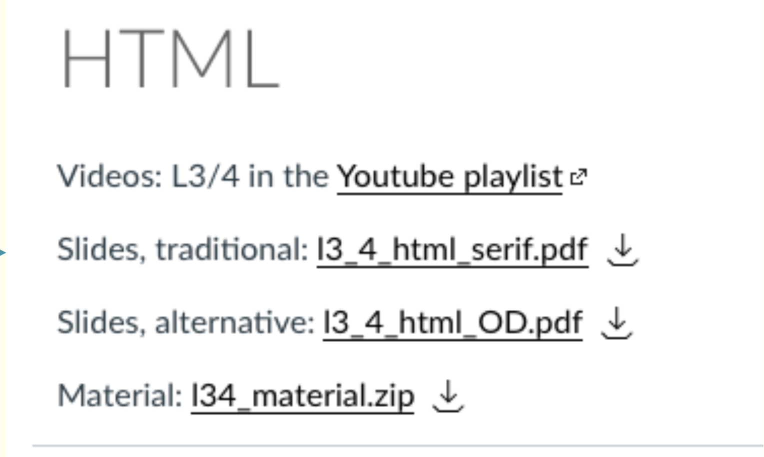
Slides/Material

- Slides to recordings are available in Modules
- Any material we use (e.g., example code) will be there as well
- Stream slides on Echo360 after each stream



Week 2 (17th - 23rd January)

HTML



HTML

Videos: L3/4 in the Youtube playlist

Slides, traditional: l3_4_html_serif.pdf

Slides, alternative: l3_4_html_OD.pdf

Material: l34_material.zip

Rules

- Late, corrupted hand-ins: 0 Points
  - Your responsibility to make sure it works/is there on time
- Unclear instructions: Ask, discuss
  - Don't just do something (and get angry at us when it's wrong)
- Every single assignment grade counts
  - No "Best X out of Y"
- Answers can take up to 2 working days - also from TAs
  - Emailing/calling does not speed this up

Expectations

I expect you

- to keep a professional tone
  (towards me, the TAs, and other students)
  - then you can expect the same from me
- to be informed (up-to-date on lectures, Piazza discussions)
  - The biggest problem for students with previous knowledge
- to know programming (T-111-PROG is a pre-requisite)

Other Resources

- This course borrows from two other courses
  - Web and Mobile Development (Gothenburg Uni, Fall 2018, Grischa Liebel & Joel Scheuner)
  - CSE1500: Web and Database Technology (TU Delft 2018/19, Claudia Hauff and Alessandro Bozzon, https://github.com/chauff/Web-Teaching)

Lots of additional resources on this page!

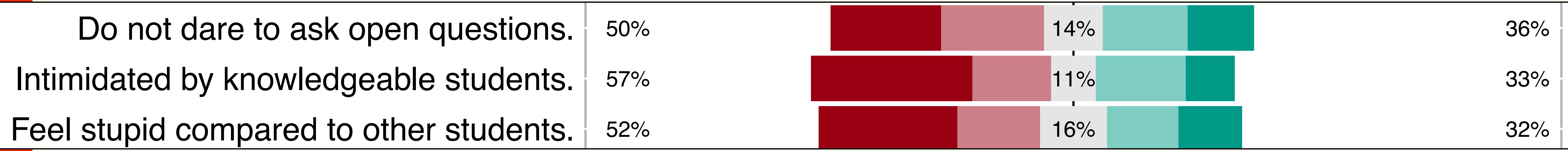
Plagiarism

- Copied code is a huge no-go
- I report every plagiarism case I find directly
  - And we talk among the teachers as well...
- But: There is a difference between plain copying, and discussion solution ideas
  - You are free to discuss your approach, steps taken, etc.
  - Just no "here is my solution - take a look"

Dæmatímar/'Labs'

- Starting course week 2 (Friday, 21st Jan)
- If you need to change section (time), contact the office (td@ru.is)
- Help with assignments
- Solutions/discussion of optional assignments
- On Discord!
  - We will make channels for each section/slot

Survey 2019 and 2020 (VEFF and GAG)



- Be welcoming and helpful - not everyone knows this stuff
- Try to keep tech smalltalk to a minimum in Twitch, Piazza, Discord

Next Lecture:

Networks/HTTP and HTML

- Networking basics
- HTTP
- HTML

Recordings on Youtube

Stream on Friday (Twitch)

HÁSKÓLINN Í REYKJAVÍK
REYKJAVÍK UNIVERSITY

Sources

Tim-Berners Lee photo: CC-BY-SA-4.0 Paul Clarke
Best-viewed-with: http://dougbelshaw.com
Wow: Designed by winkimages / Freepik
Education: Designed by Freepik
Exam: Designed by Freepik
Tools: Designed by Terdpongvector / Freepik
Confusion: Designed by Freepik

Assignment icons: Designed by ibrandify / Freepik
Chrome logo: Public domain
Firefox logo: Mozilla Public License Version 2, the Mozilla Foundation
VSCode logo: public domain
git logo: CC Attribution 3.0 Unported, Jason Long
GitLab logo: Copyright by GitLab, Inc.
Postman logo: Copyright by Postman, Inc.
Nodejs logo: Trademark by Joyent, Inc.
Megaphones: Designed by Freepik