

DAT650 Blockchain technology

Intro

Leander Jehl 2023

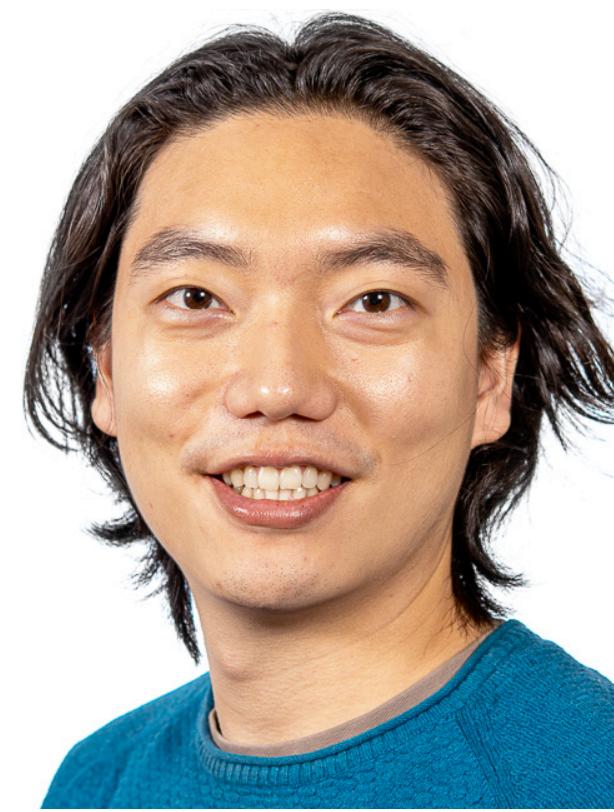
The team



Leander



Arian



Jungwon

Course info

<https://github.com/dat650-2023/info/>

The screenshot shows a web browser window with the following details:

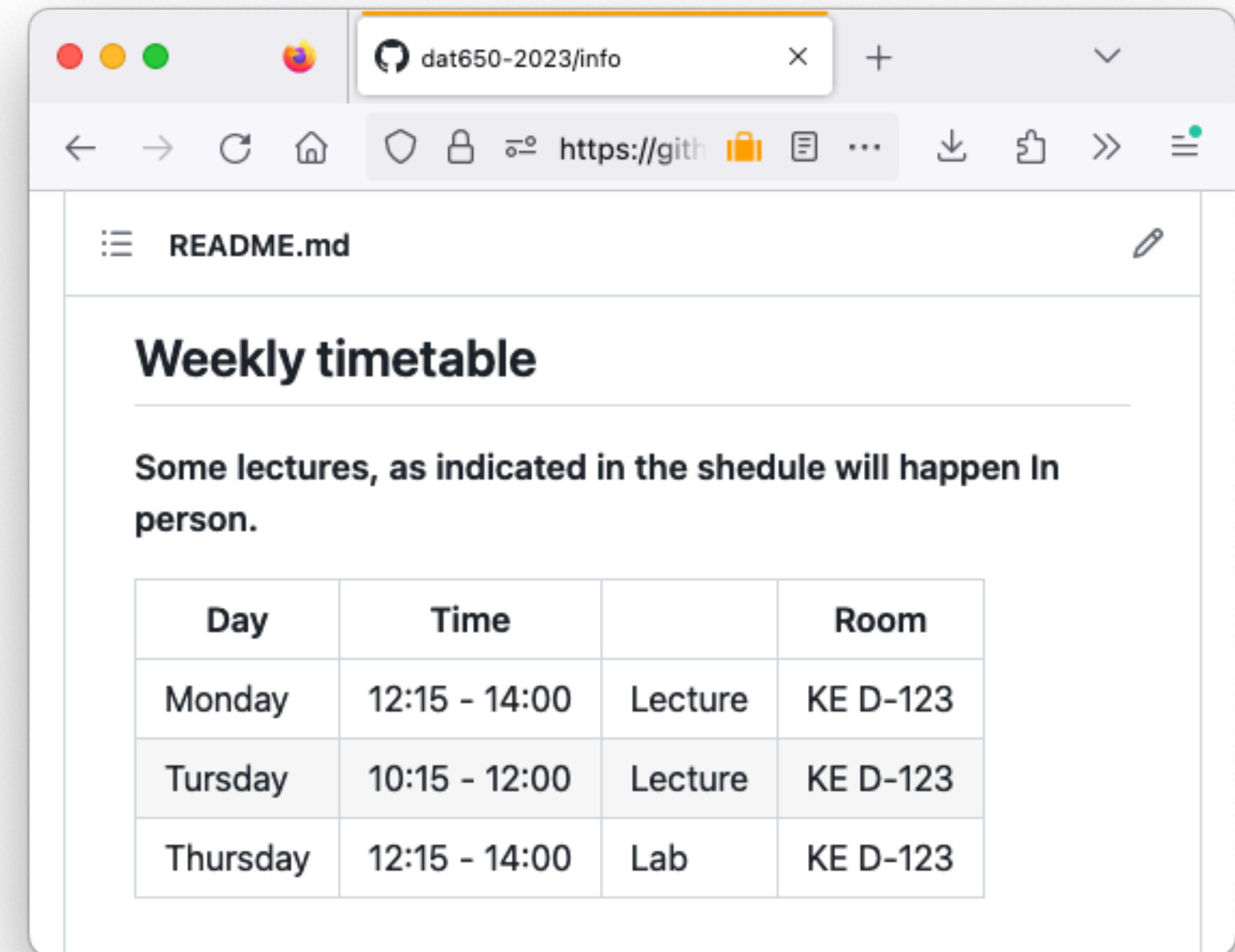
- Title Bar:** course-info/README.md at master
- Address Bar:** https://github.com/dat650-2023/info/ (highlighted with a blue border)
- Toolbar:** Includes standard browser icons for back, forward, search, and refresh.
- Content Area:**
 - ## DAT650 Blockchain technology
 - This repository contains useful information and common resources for the DAT650 course.
 - You can find here:

 - [Lecture Script](#)
 - [Lecture Schedule](#)
 - [Lab info](#)
 - [Project](#)

Course info

<https://github.com/dat650-2023/info/>

- Lectures
- Reading material
- Lecture script (will be updated)
- Slides (will be updated)



The screenshot shows a web browser window with the title bar "dat650-2023/info". The address bar shows the URL "https://github.com/dat650-2023/info/". The main content area displays a file named "README.md". The first section is titled "Weekly timetable" and contains the following text: "Some lectures, as indicated in the schedule will happen in person." Below this is a table showing the weekly schedule:

Day	Time		Room
Monday	12:15 - 14:00	Lecture	KE D-123
Tuesday	10:15 - 12:00	Lecture	KE D-123
Thursday	12:15 - 14:00	Lab	KE D-123

Course info

<https://github.com/dat650-2023/info/>

- Lectures
- Reading material
- Lecture script (will be updated)
- Slides (will be updated)

The screenshot shows a GitHub repository page for 'DAT650-21/course-info'. The main content is a README.md file containing a table of course information.

Table Headers:

Day	What	Format	Lecturer	Topic	Material
-----	------	--------	----------	-------	----------

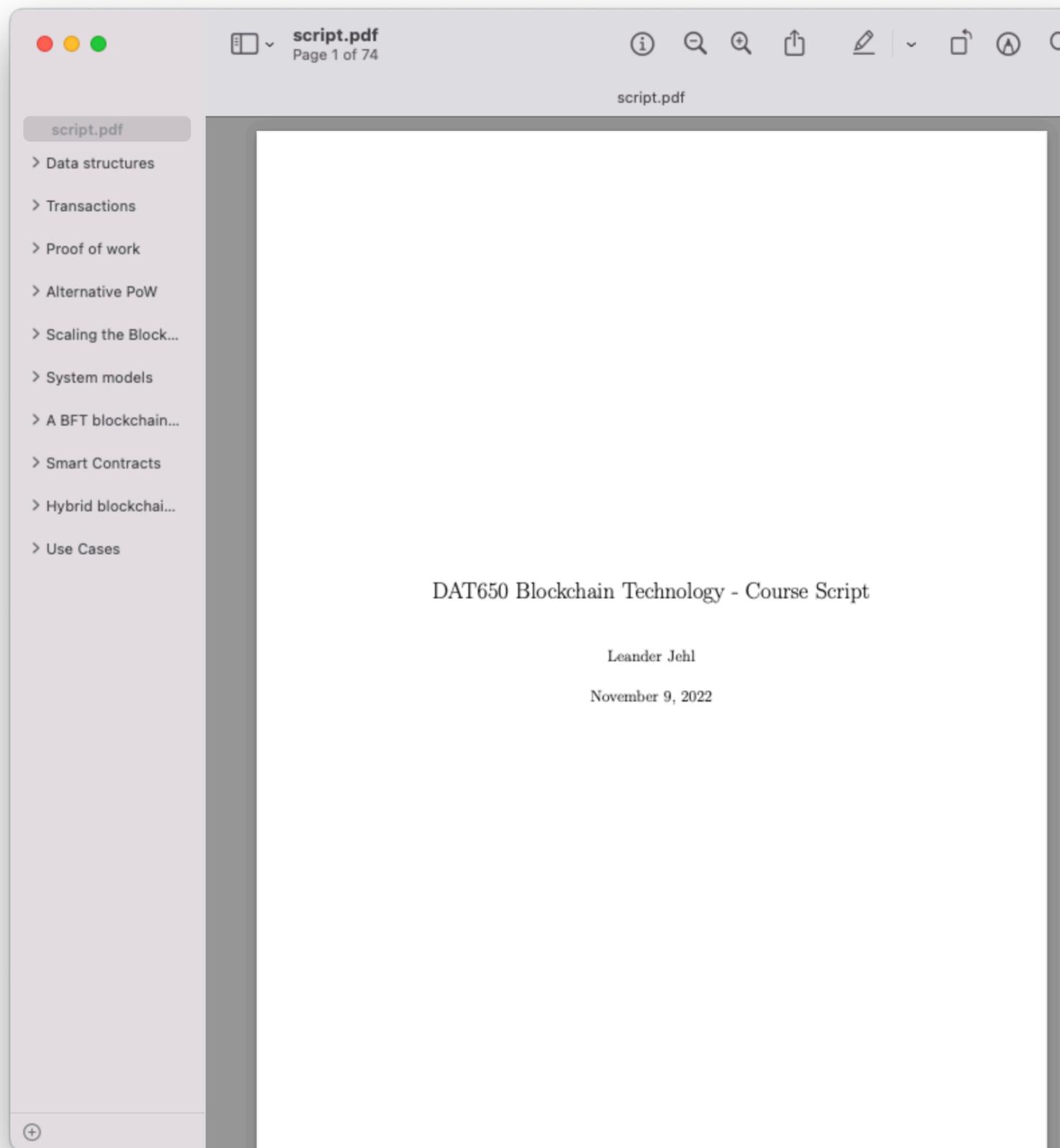
Table Data:

Tue 24.Aug	Lecture	Online	Leander	Hashes and hash chains and merkle trees	**Script** Chapter 1, Videos 1, 2, 3, also videos 7-10 from the same series.
Tue 24.Aug	Lab	?	Rodrigo	Lab 1	**Script** Chapter 2, Princeton book Chapter 1.4 and 1.5
Fri 27.Aug	Lecture	Online	Leander	Transactions and UTXO	**Script** Chapter 3.1 -
Tue					

Course info

<https://github.com/dat650-2023/info/>

- Lecture script



A screenshot of a GitHub repository page for 'dat650-2023/info'. The repository was created by 'leandernikolaus' 6 hours ago. It contains several files: 'notebooks', 'projects', 'slides', 'README.md', 'lecture_material.md', 'resources.md', and 'script.pdf'. The 'script.pdf' file is highlighted with a red box. Below the files, there is a section titled 'DAT650 Blockchain technology' with the text: 'This repository contains useful information and common resources for the DAT650 course.' There is also a 'Weekly timetable' section. A list of resources is provided, with the first item, 'Lecture Script', also highlighted with a red box.

File	Last modified
leandernikolaus more material ...	6 hours ago
notebooks	6 hours ago
projects	6 hours ago
slides	6 hours ago
README.md	6 hours ago
lecture_material.md	6 hours ago
resources.md	6 hours ago
script.pdf	6 hours ago

DAT650 Blockchain technology

This repository contains useful information and common resources for the DAT650 course.

You can find here:

- [Lecture Script](#)
- [Lecture Schedule](#)
- [Lab info](#)
- [Project](#)

Weekly timetable

Assignments

<https://github.com/dat650-2023/assignments>

- Send GitHub ID to: jungwon.seo@uis.no.

- Lab 1-2: Bitcoin basics in Go

- Lab 3 Smart Contracts
in Solidity

- Project

- No lectures during projects

Lab Schedule (subject to change)

Lab	Start	Topic	Deadline	Duration
1	24. AUG	Data structures and PoW	07. SEP	2 weeks
2	07. SEP	UTXO Transaction Model	21. SEP	2 weeks
3	21. SEP	Intro Smart Contracts	05. OCT	2 weeks
Project	05. OCT	Projects	09. NOV	4 weeks
4	19. OCT	Distributed Application	Optional	2 weeks

In class notebooks

<https://github.com/DAT650-2023/info/tree/main/notebooks>

- Jupyter notebooks

Lecture 1 hashes

```
In [30]: #Import statements
import hashlib as hasher

def hashbits(input):
    hash_obj = hasher.sha256()
    inputbytes = input.encode()
    print(type(inputbytes))
    hash_obj.update(inputbytes)
    hashbytes = hash_obj.digest()
    return ''.join(f'{x:08b}' for x in hashbytes)

def hash(input):
    hash_obj = hasher.sha256()
    inputbytes = input.encode()
    print(type(inputbytes))
    hash_obj.update(inputbytes)
    return hash_obj.hexdigest()
```

```
In [45]: print(hashbits("hello world!"))
print(hash("hello world!"))
```

```
<class 'bytes'>
01110101000100111100101101111011010000011000111011000101101001010
11101011000111111100100001101011101011000101101011011001000100110
001111110100000000111001100101110001010100001010101011010111000
1111011111000101100011101111100000100011100110110010101001
<class 'bytes'>
7509e5bda0c762d2bac7f90d758b5b2263fa01ccbc542ab5e3df163be08e6ca9
```

Exercise 1

Hashes look random but are deterministic:

Project

- September: Project topics published
- 3 Weeks no lecture to work on project
- Project paper presentation
- Hand in project report
- Project counts for 40% of the grade

Exam

- Oral exam
- 60% of the grade