

# Lecture VI - Computational Kindness

Programming with Python

Dr. Tobias Vlček

# Computational Kindness

## Computational Kindness

...

Question: **An idea what that is?**

...

- Idea of introducing empathy in algorithms
- **Consider human cognitive load and limitations**
- Making choices that **reduce mental burden** for others
- Creating systems that are **easy to understand and interact**

## Scheduling Meetings

- Propose specific times!
- Don't ask "when are you free?"
- **Reduces** cognitive load for others
- **Transforms** an open-ended problem

...

**!** Important

**Now a simple yes/no decision!**

## Communication

- Be explicit rather than implicit
- Provide **clear** options instead
- **Avoid** open-ended questions
- State your **preferences**

...

**!** Important

**Avoid** deferring ☒ "I'm fine with whatever"!

## The Cost of Flexibility

- “I’m free whenever” seems helpful
- Being too flexible can be **unhelpful**
- But it is not!
  - Forces others to **consider all possibilities**
  - **Increases** cognitive load
  - Makes decision-making **more complex**

## Principles

1. **Reduce Options:** Fewer choices lead to better decisions
2. **Be Explicit:** Clear constraints help others decide
3. **Consider Cognitive Load:** Design interactions minimizing mental effort for others
4. **Make Decisions:** Taking responsibility can be kind

# Key Concepts

## Topics from the Lecture

- **Optimal Stopping:** How to decide when to stop looking for better options.
- **Explore/Exploit Tradeoff:** Balancing between trying new things and sticking with known options.
- **Caching:** Storing and reusing information by sticking to a task to improve performance.
- **Scheduling:** Efficiently managing tasks and time.
- **Randomness:** Understanding and working with uncertainty.

# How to continue?

## How to continue learning?

- We have covered **a lot of topics**
- But there are **many more** to explore!

## Bayes' Rule

- Fundamental theorem in probability theory
- Updates probability of hypothesis based on new evidence
- Used in **statistics, machine learning, and decision-making**

...

! Important

Helps make **informed predictions and decisions** under uncertainty in the real world!

## Overfitting

- Occurs when a model learns the training data too well
- **Captures noise and outliers** rather than pattern
- Results in **poor generalization to new data**

...

! Important

Several ways to counter overfitting, e.g. cross-validation.

## Game Theory

- Strategic interactions among rational decision-makers
- Analyzes situations with **multiple agents** and their strategies

...

! Important

**Many applications** in economics, politics, and biology!

## Networking

- Study of **information exchange** over networks
- Includes understanding protocols and data transmission
- Optimizes **network resources**

...

! Important

**Important** in computer networks, the internet and social life!

## Relaxation

- Simplifies complex problems to make them more tractable
- Involves **relaxing certain constraints**
- Provides insights or **approximate** solutions

## How to continue programming?

- The best way to continue learning is to keep programming in the future
- Potentially, you will continue to do so during your studies
- Coding in your Thesis is another great way to improve
- Try to find a way to apply programming in your work
- **There are many interesting topics to explore!**

## Advent of Code

- [Advent of Code](#) is a fun way to keep programming
- Here you can solve programming puzzles during Advent
- It is **completely free** and ad-free and starts each year at 01.12.

## That's it for the Lecture Series!

- We now have covered the basics of Python
- I hope you enjoyed the lecture and found it helpful
- If you have questions or feedback, please let me know!
- I wish you all the best for your studies and your career!

# Q&A

# Literature

## Interesting literature to start

- Christian, B., & Griffiths, T. (2016). Algorithms to live by: the computer science of human decisions. First international edition. New York, Henry Holt and Company.<sup>1</sup>

## Books on Programming

- Downey, A. B. (2024). Think Python: How to think like a computer scientist (Third edition). O'Reilly. [Here](#)
- Elter, S. (2021). Schrödinger programmiert Python: Das etwas andere Fachbuch (1. Auflage). Rheinwerk Verlag.

...

### **i** Note

Think Python is a great book to start with. It's available online for free. Schrödinger Programmiert Python is a great alternative for German students, as it is a very playful introduction to programming with lots of examples.

## More Literature

For more interesting literature, take a look at the [literature list](#) of this course.

---

<sup>1</sup>The main inspiration for this lecture. Nils and I have read it and discussed it in depth, always wanting to translate it into a course.