

Intro Kickstart-kursus i programmering 2023: 1

Daniel Spikol, The Team, & Martin Dybdal
ds@di.ku.dk

DIKU
Københavns Universitet

14.august 2023

Hello World!



Who are your Instructors

- Daniel
- Anders
- Casper
- Christian
- Jepper
- Laufey
- Nanna
- Nicolaj
- Sofie



Inclusive DIKU

- Welcome to DIKU. Our goal is to promote excellence in computer science education and research.
- To strive for a respectful, inclusive, diverse environment and encourage open and critical academic discussion.
- We strive to create a welcoming and respectful environment.



Velkommen!

- Presentation of Activities
- Why and what is the Kickstart course?



Overall Plan for the week

Day 1	5: 55m	Day 2	6: 00m	Day 3	6: 00m	Day 4	6: 00m	Day 5	6: 00m
09:00 Welcome and Introduction 18m		09:00 Hello and Intro to peer pro... 15m		09:00 Creative Coding & Conditionals 30m		09:00 Intro to State Machines 20m		09:00 Summary of week 15m	
09:15 Greeting Activity 15m		09:15 Intro to setup/draw + glob... 15m		09:30 Break 15m		09:20 Worksheets #5-6 & Project... 2h 25m		09:15 Project time 2h 30m	
09:30 Team Building and Icebreak... 1h 30m		09:30 Break 15m		09:40 Worksheet #4 15m		11:45 Lunch 45m		11:45 Lunch 45m	
11:00 Installation, wifi, setup, et... 45m		09:45 Worksheet #2 1h 00m		11:30 Peer Feedback Practice 15m		12:30 Project Time 2h 30m		12:30 Final Project Prep 1h 00m	
11:45 Lunch 45m		10:45 Worksheet #3 1h 00m		11:45 Lunch 45m		13:00 Project Time 1h 30m		13:30 Presentations 1h 15m	
12:30 Intro to Processing 10m		11:45 Lunch 45m		12:30 Projects and Brainstorming 30m		14:30 Peer Feedback 15m		14:45 Survey 15m	
12:40 Worksheet: #0 The First P... 1h 00m		12:30 Pair Programming Practice 15m		13:00 Project Time 1h 30m		14:45 Recap/Touchdowns 15m			
13:40 Worksheet: #1 1h 00m		12:45 Worksheet: #3 1h 15m							
14:40 Debrief 15m		14:00 Break 15m							
14:55		14:45 Peer Feedback 30m							
		14:45 Recap if needed 15m							
		15:00							

+ Add block

+ Add block

+ Add block

+ Add block



Monday's Plan

- kl. 9:00-11:00 - Team Building Activities, Mini-Lecture, Code, Code, and more Code
- KL 11:15-11:45 - Processing Introduction
- kl. 11:55 - Lunch - Bio Kantine
- kl. 13:00-13:30 - Aud.06 Intro to Processing
- kl. 13:30-14:45 - Team rooms - Intro to Processing
- kl. 14:45-15:00 - Check-in with Teams
- kl. 15:00-? - Relax for tomorrow or code more!



Intended Fun Outcomes

- Why IFOs and not Intended Learning Outcomes
 - Intended Learning Outcomes (ILOs) define what a learner will have acquired and will be able to do upon completing their courses and studies.



Intended Fun Outcomes:

- NDAB15009U Programmering og problemløsning (PoP)
 - Knowledge on:
 - Basic concepts within the imperative, object-oriented and functional programming paradigms: Functions and methods, variables, expressions, types, control structures, loops, block structure, classes and objects, object interaction, inheritance, recursion, polymorphism, abstraction, exceptions, pattern matching over recursive data types, etc.
 - Good programming practice: Documentation in the code, design patterns, testing, incl. unit testing, handling runtime errors, etc.
 - Techniques for problem-solving: Technical analysis of natural language problems, object-oriented design, modelling languages, etc.
 - <https://kurser.ku.dk/course/ndab15009u/>



Intended Fun Outcomes

- Meet people and collaborate
- Explore programming with a focus on creating games
- Have fun and take chances through creativity
- Take control of your education



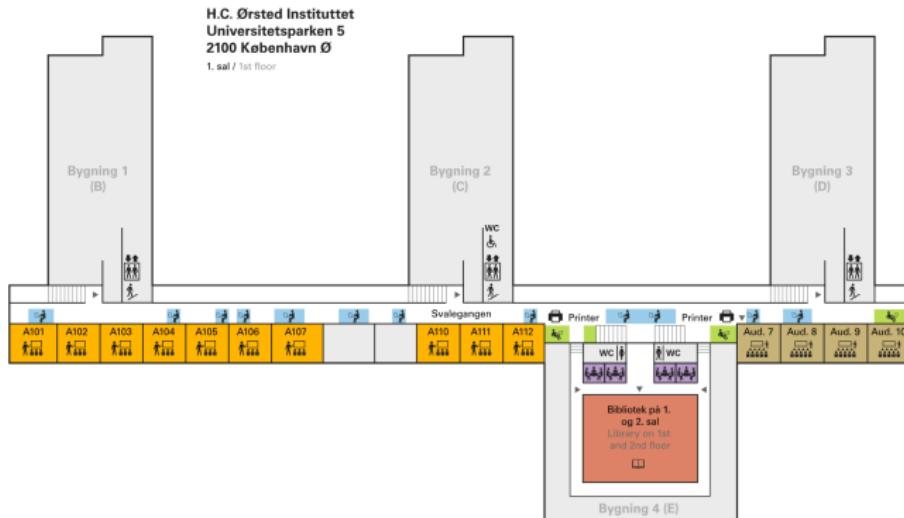
Meet people and collaborate

- Design Thinking
- Collaboration
- Reframing



Teams and Instructors

- **Team Alpha:** Christian and Casper, room A105
- **Team Bravo:** Nanna and Nicolaj, room A106
- **Team Charlie:** Anders and Laufey, room A107
- **Team Delta:** Sofie and Jeppe, room A110

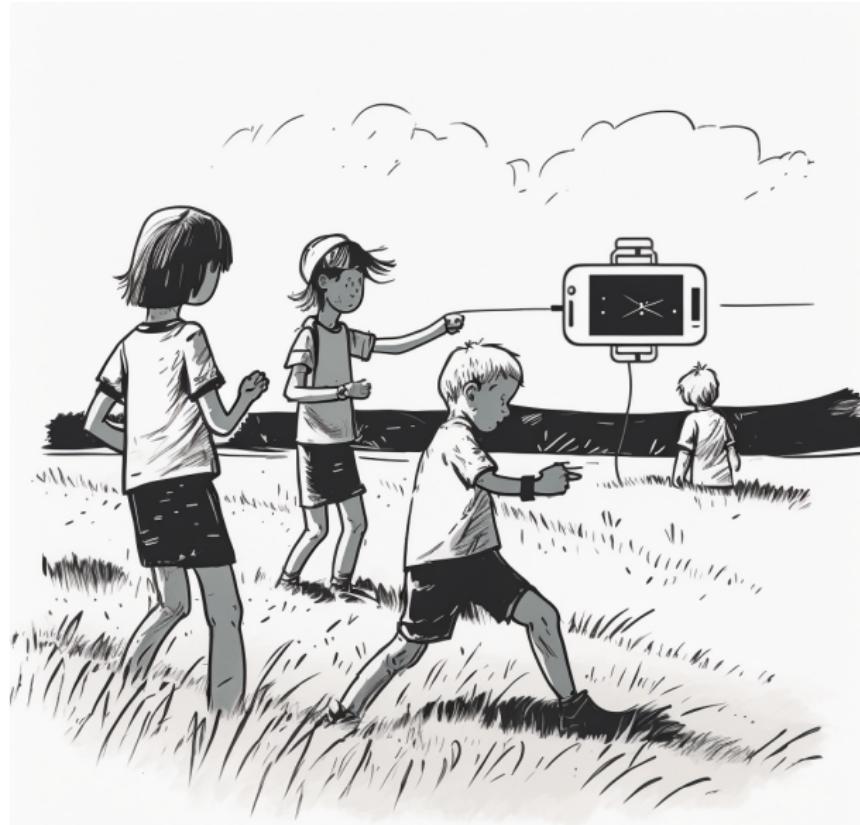


Challenge Time

- Design Thinking
- Collaboration
- Reframing



Time for Challenge



Marshmallow Challenge Reflection

- TED TALK Tom Wujec
- Constant prototyping as a problem-solving method.
- Getting the Design Right and the Right Design, Buxton, William (2007). Sketching user experiences: getting the design right and the right design. Amsterdam: Elsevier/Morgan Kaufmann



Internet Access

Aktiver KU-bruger

- Log på <http://mit.ku.dk> med NemID og find midlertidig pinkode.
- Gå til <http://kunet.dk> og tryk "Første gang du logger på".
- Log på med den midlertidige kode og CPR nummer uden bindestreg.
- Aflæs KU brugernavn (fx abc123) og angiv password.

eduroam WiFi

- Kræver KU bruger og password
- Log på med brugernavn: abc123@ku.dk

Alternativt: KU Guest WiFi

- Opret 24 timers konto ved at indtaste navn, mail, mobilnummer.
- Modtag kode via SMS.

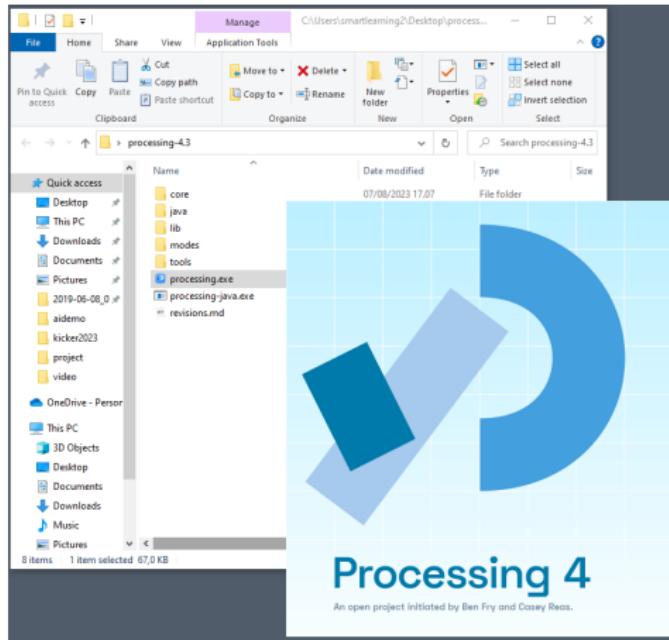


I. Installation af Processing.py

- Hent og installer Processing fra <http://processing.org/download/>
- Installer Python Mode via library manager
- SEE: <https://bit.ly/3SXOp4r>



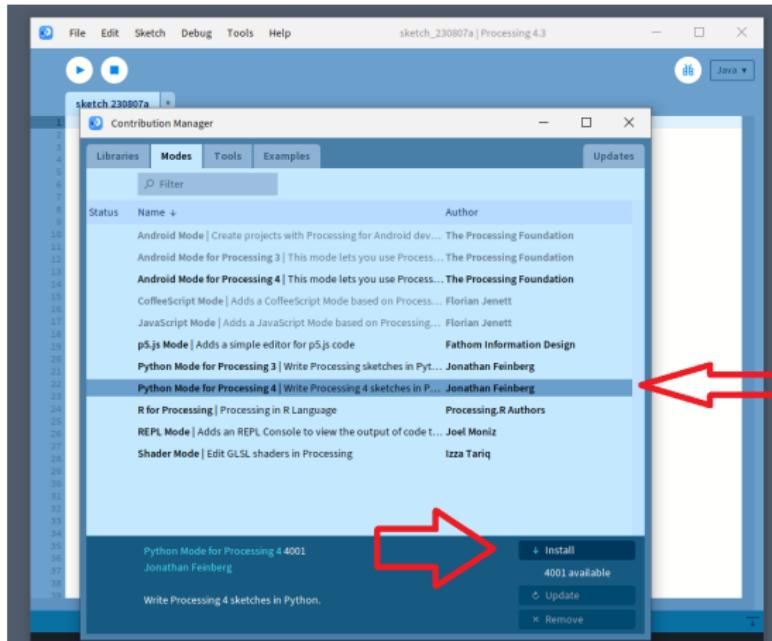
II Installation af Processing.py



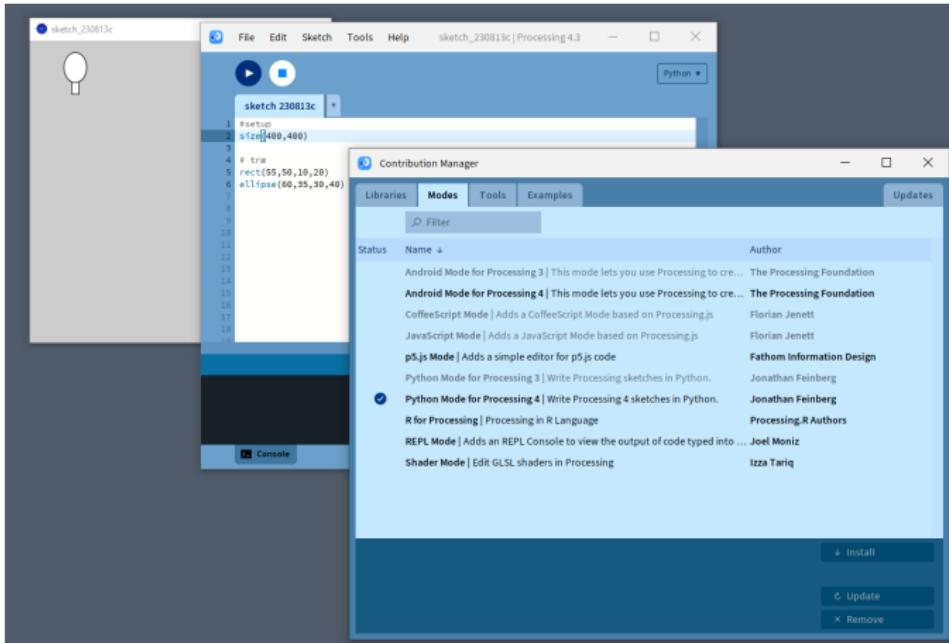
III. Installation af Processing.py



IV. Installation af Processing.py



V. Installation af Processing.py



Lunch



Processing History

- M.I.T. Media Lab Casey Reas & Ben Fry
- Design By Numbers - Maeda, John. - Design by numbers / John Maeda.. - 1999. - ISBN: 0262133547
- Creative Programming

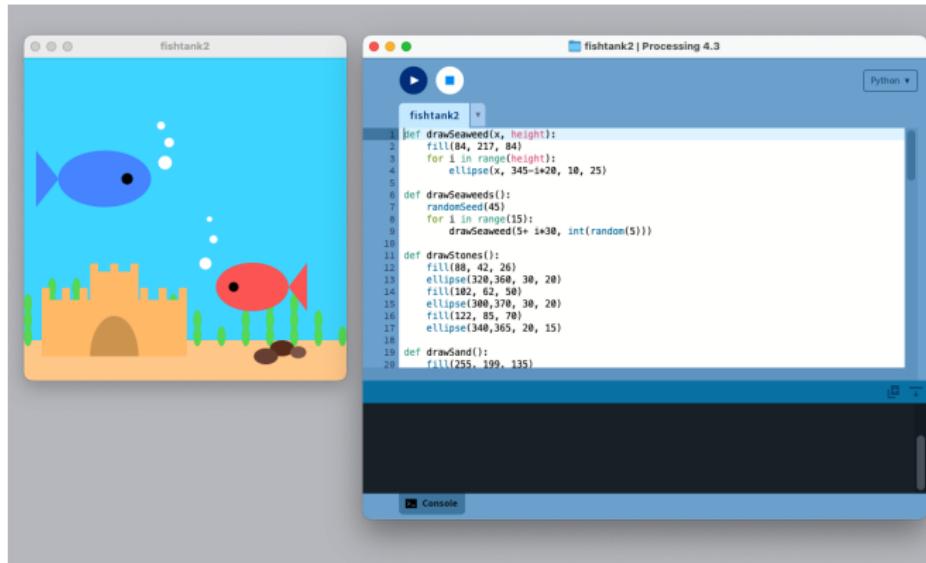


What is Processing

- Processing is a flexible software sketchbook and a language for learning code. Since 2001, Processing has promoted software literacy within the visual arts and visual literacy within technology. Thousands of students, artists, designers, researchers, and hobbyists use Processing for learning and prototyping.
- Processing Development Environment ↪ PDE
- Programs are called Sketches
- The PDE consists of a simple text editor for writing code, a message area, a text console, tabs for managing files, a toolbar with buttons for common actions, and a series of menus.



Processing Example



Processing Parts 1

1. Basics of Processing IDE:

- **Interface:** The Processing IDE has a simple interface. There's an area for writing code, buttons for running and stopping sketches, and a message area below for error messages and other notifications.
- **Sketch:** In Processing, each project is called a "sketch". A sketch is a combination of code, data, and output.

2. Language:

Processing uses a variant of the Java language. It's designed to be beginner-friendly, with simpler functions and setup to create visual and interactive projects quickly.

3. Structure of a Basic Sketch:

- **setup() function:** This is executed once when the sketch starts. It's commonly used to define initial environment properties such as screen size and background color.
- **draw() function:** This runs repeatedly after setup(). It's used for continuously running code, such as animation or checking for input.

4. Running the Sketch:

The sketch is compiled and run when you click the "Run" button (or press Ctrl+R/Cmd+R). The visual output is displayed in a separate window.



Processing Parts 2

1. **Libraries and Extensions:** Processing has a range of libraries that can be imported to add functionality, such as handling video, sound, or advanced graphics operations.
2. **Exporting:** You can export your sketches to standalone Windows, macOS, and Linux applications. Exporting projects for the web using the P5.js variant of Processing is also possible.
3. **Modes and Variants:**
 - **P5.js:** A JavaScript library with the essence of Processing but for web development. It brings the Processing approach to web artists and developers.
 - **Python Mode:** Allows you to write Processing sketches in Python.
 - **Android Mode:** This lets you create Android apps using the Processing API.
4. **Community:** One of the strengths of Processing is its active and supportive community. Numerous tutorials, forums, and resources are available to help users learn and troubleshoot.



Enough Lecture: Today's IFOs

- You will get a sense of the PDE shortly
- How to **draw and color** with Processing
- What are **Variables** as simple as they get

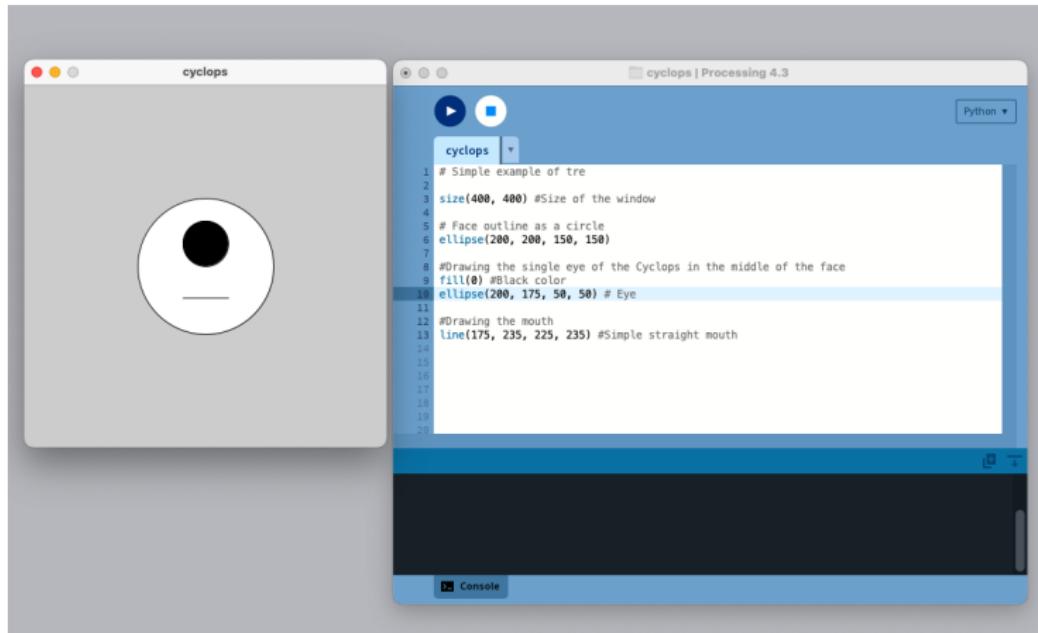


Enough Lecture: Today's IFOs

- You will get a sense of the PDE shortly
- How to **draw and color** with Processing
- What are **Variables** as simple as they get



Drawing



Variables

- At its core, a **variable in programming** is like a container or a storage box that holds data.
- **Variables** are names that hold **values**.
- Imagine you have a box in real life where you put an apple; you might label that box "fruit" so you know it contains a fruit. In programming, the box is the variable, the apple is the data, and the label "fruit" is the variable's name..
- Keep in mind that, unlike real life, if you put in orange into your variable, it will make the apple value disappear



Processing References

- Processing Reference
- Processing Tutorials



The screenshot shows a web browser displaying the Processing website's "Getting Started" tutorial. The page has a dark background with a geometric wireframe graphic at the top. The main content area is titled "Getting Started". It includes a brief introduction, instructions for Mac, Windows, and Linux, and a troubleshooting section. On the left, there is a sidebar with links to "Cover", "Reference", "Tutorials", "Examples", and "Bugs".

This tutorial is for Python Mode in Processing 3+. If you see any errors or have comments, please [let us know](#). This tutorial was adapted from the book, [Getting Started with Processing](#), by Casey Reas and Ben Fry / [O'Reilly / Make](#) 2010. Copyright © 2010 Casey Reas and Ben Fry. All rights reserved.

Welcome to Python Mode for Processing!

Start by visiting <http://processing.org/download> and selecting the Mac, Windows, or Linux version, depending on what machine you have. Installation on each machine is straightforward:

- On Windows, you'll have a .zip file. Double-click it, and drag the folder inside to a location on your hard disk. It could be Program Files or simply the desktop, but the important thing is for the processing folder to be pulled out of that .zip file. Then double-click processing.exe to start.
- The Mac OS X version is also a .zip file. Double-click it and drag the Processing icon to the Applications folder. If you're using someone else's machine and can't modify the Applications folder, just drag the application to the desktop. Then double-click the Processing icon to start.
- The Linux version is a .tar.gz file, which should be familiar to most Linux users. Download the file to your home directory, then open a terminal window, and type:
`tar xvfz processing-xxx.x.tgz`
(Replace xxx with the rest of the file's name, which is the version number.) This will create a folder named processing-2.0 or something similar. Then change to that directory:
`cd processing-xxx`
and run it:
`./processing`

With any luck, the main Processing window will now be visible. Everyone's setup is different, so if the program didn't start, or you're otherwise stuck, visit the [troubleshooting page](#) for possible solutions.



Coordinate System in Processing

