

CS-A1113 Basics in Programming Y1

Kick-Off Lecture 6.9.2022



Why Study Programming?

- Digitalization!
- Computers will be part of everything in the society
 - Transportation, finance, media, culture...
- You will have a software project in your career
- 21:st century is the Century of Software

What do You Get out of this Course?

- A start on programming
 - The principles will work on any language
- An understanding on how computers and software work
 - This will be useful for all of us
- You can make utility programs
 - E.g., read a file in one format and output it in another format
- You can program your future house to take advantage of the market price of electricity

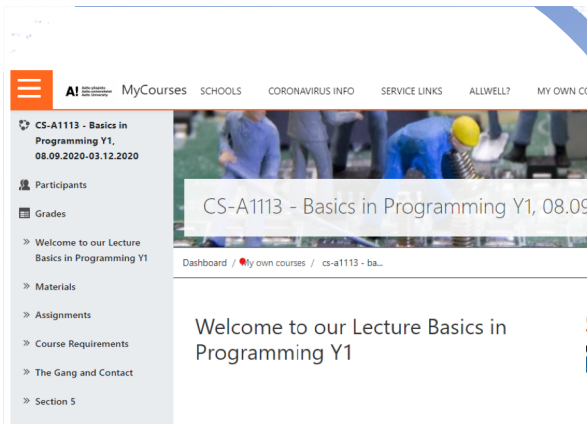
Staffing and Related Course

- Lecturer Timo Kiravuo, DSc
- 10 friendly teaching assistants to help you with exercises

- This course is similar in content to the Finnish language CS-A1111 Ohjelmoinnin peruskurssi Y1
- Same exercises
- You can not take both courses

Organisation

- Live lectures (Tuesdays 14:15 at T1 in CS building)
- 4 Exercise sessions every week (starting this evening)
 - Tuesdays 8:15 - 10:00 Online at Zoom, send your zoom link to the Slack channel #exercises
 - Tuesdays 16:15 - 18:00 Y342a
 - Wednesday 8:15 - 10:00 Y342a
 - Wednesday 16:15 - 18:00 U256
- Lecture material online on A+
- Discussion and Questions on Slack
- Exercise deadlines Thursday 11:00 AM! Mandatory! Starting next week!
10 submissions possible per exercise

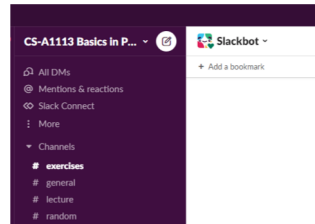


Main Entry Point!

- Information and Links
- On-site Exercise Booking
- Questionnaires

MyCourses

Organisation & Tools



Main Discussions

- Give and get Help on Exercises
- All kinds of questions
- It is what you make it

Slack

A+



Main Material

- Course Material
- Exercises

Exercise Sessions

- 1 Online session, 3 on-site sessions each week
- For asking guidance and help, participation is not mandatory
- For online
 - Create a Zoom meeting
 - Put your code to Codevault (if you are asking about code)
 - Post the Zoom link to Slack
 - Wait
- For on-site
 - Go to the lab room
 - Sign up at A+ queue system
 - Wait

Exercises & Grade

Grade:

- 50 % of the exercise grade
- 50 % of the exam grade
- If either of the grades is 0 the course is failed
- Exercises:
 - 8 rounds mandatory, deadline each week Thursday at 11:00
 - each round **must** be passed (minimal number of points for that week reached)
 - at most 3 exercises can be substituted -> amount to minimal numbers of points for said round (no gain in grade possible with substituted exercises)
 - Exercise 9 voluntary, if not done max grade from exercises is limited to 2
 - Exercise 9 can be substituted in a separate substitution exercise

Examination

- Computerized exam available at your convenience 29.11. - 16.12.
- You need to sign up at Sisu a week before, latest 22.11.
 - Signup opens a month before
- Uses PyCharm, so you should get experience with it

Plagiarism and co-operation

- Plagiarism is not tolerated!
- You may co-operate with others and discuss the exercises and your code
- You may **not** copy the answers from others
- There is a powerful artificial intelligence keeping track of you exercise answers
- Do your own work!
 - You are here to learn and the only way to learn programming is to program
- You may give hints to others and help them, but do not share your code
- Plagiarism can lead to failing the course for this year

Some Guidelines

- You learn programming by programming
- Learn the features of the PyCharm IDE (Integrated Development Environment)
 - Breakpoints, debugging
- Work in small steps and check what your changes do
- When nothing is working, go to bed and sleep over it (seriously, this works)
- To have time to sleep, start early on all exercise problems and get them into your mindspace for processing
- Learn to break problems to independent smaller problems
- Remember: the computer does what you tell it to do, not what you want it to do



TIPS &

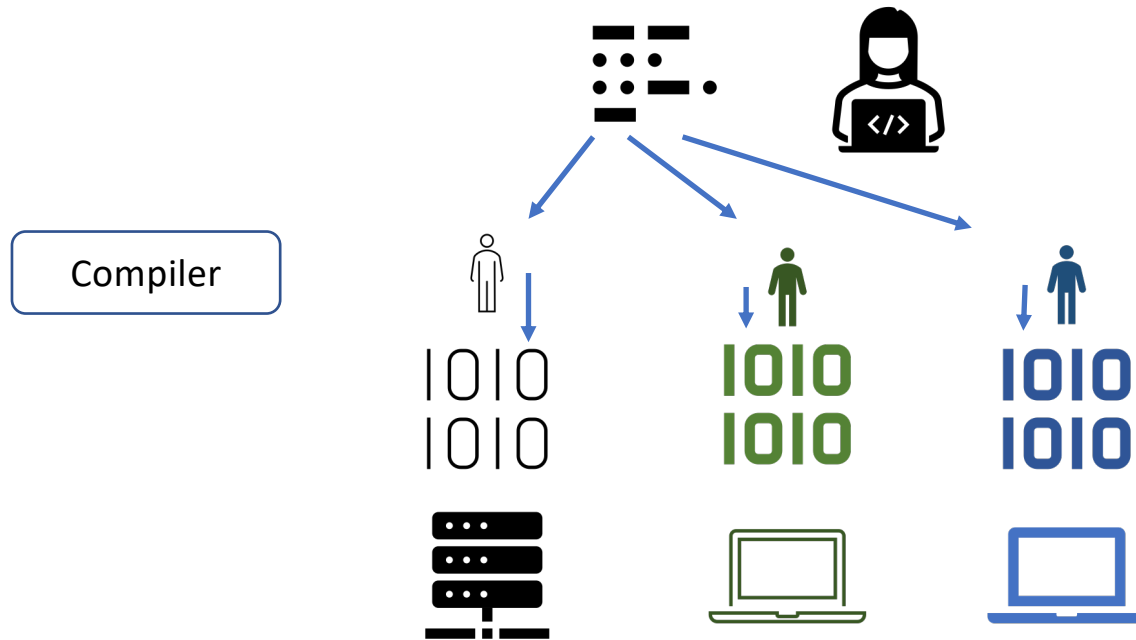
Do not despair!

- If not done yet: Start today with the installation of PyCharm
- Do use Slack for questions and answers
- Start the exercises as early as possible, do NOT wait until the last day before the deadline! I repeat: Do NOT wait until the last day before the deadline!
- Keep up with the lecture.

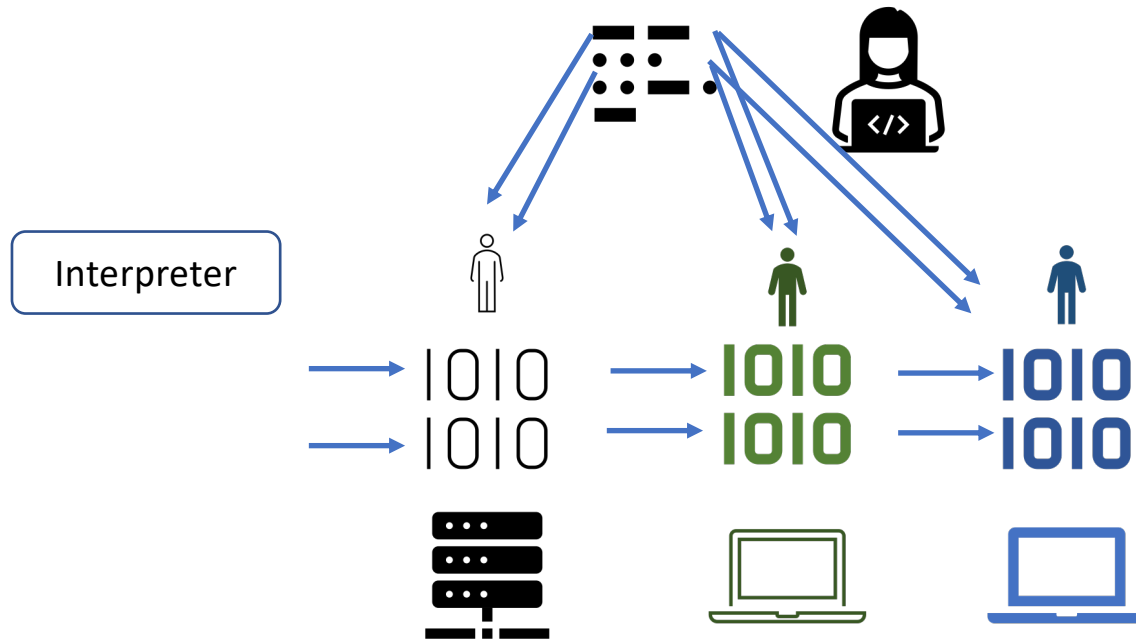
TRICKS



Computers and Computer Programs



Computers and Computer Programs (Interpreted)



Structure of a Computer (simplified)

CPU

- Central Processing Unit
- Executes commands from RAM
- Operates on data in RAM

Input/Output system (I/O)

- Display and keyboard
- etc...

RAM

- Random Access Memory
- Volatile memory
- Holds programs and data when powered
- Loses contents without power
- Fast

Storage, Disk

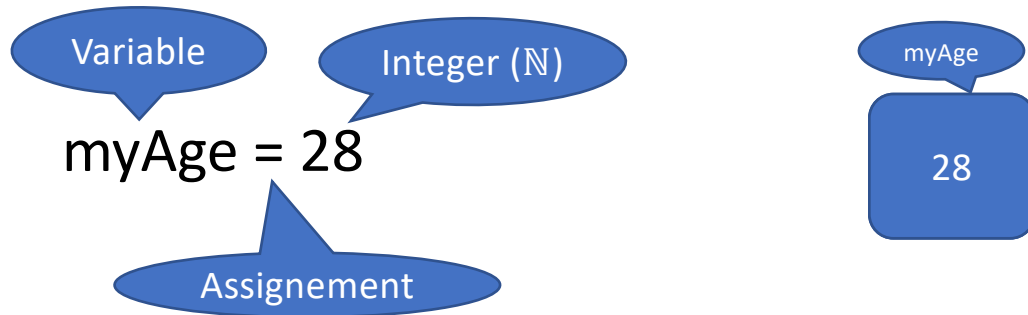
- Non-volatile memory
- Holds programs and even unpowered
- Slow

- There is more than this, but this is enough for now



What is with this “=” sign

It is not the mathematical equality sign, but it means assignement



What is with this "=" sign

It is not the mathematical equality sign, but it means assignement

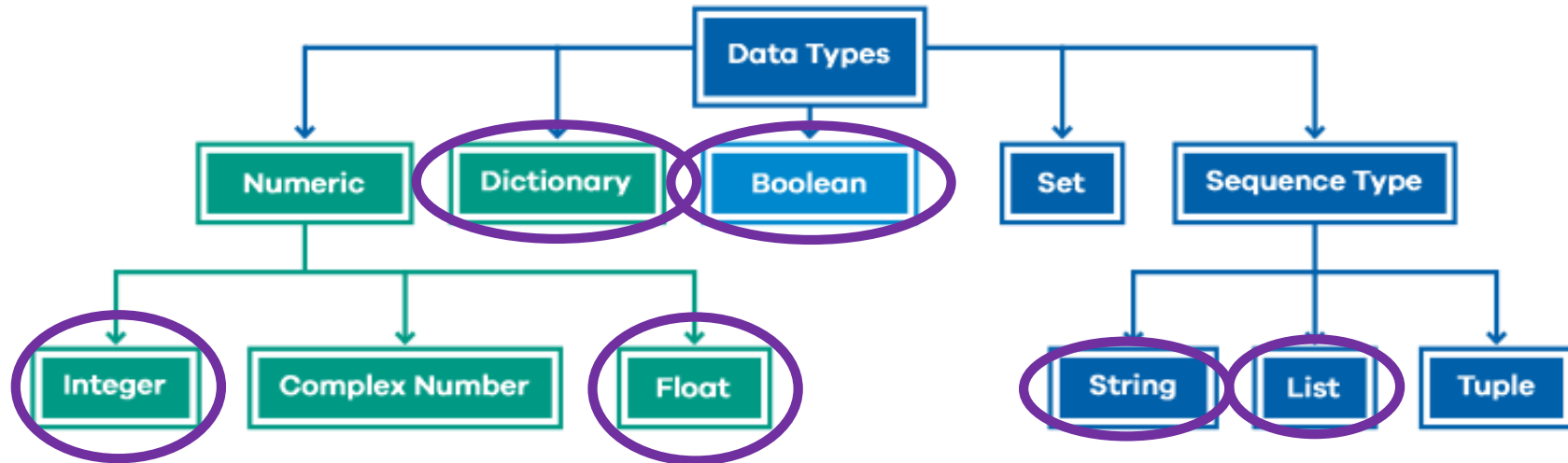
myAge = 28

myAge = myAge - 2

myAge = myAge - 4

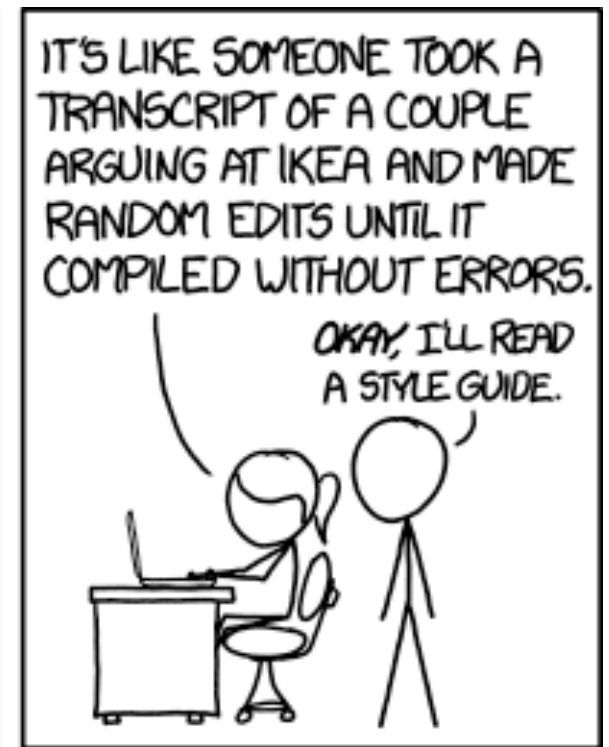
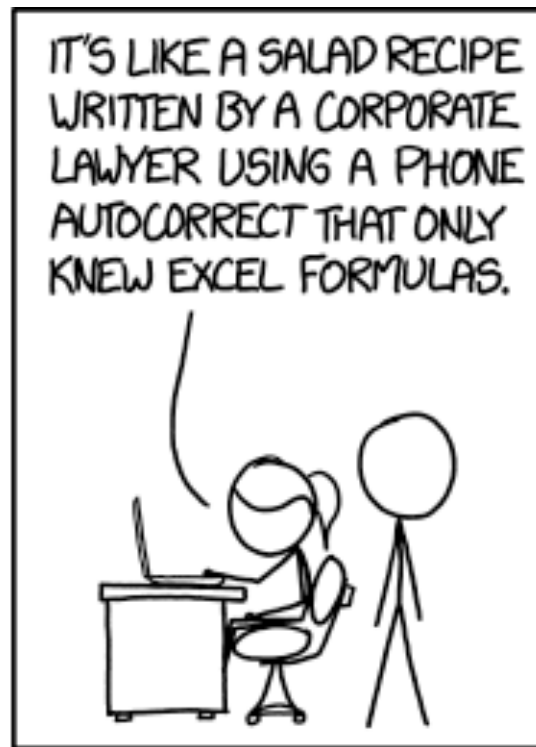
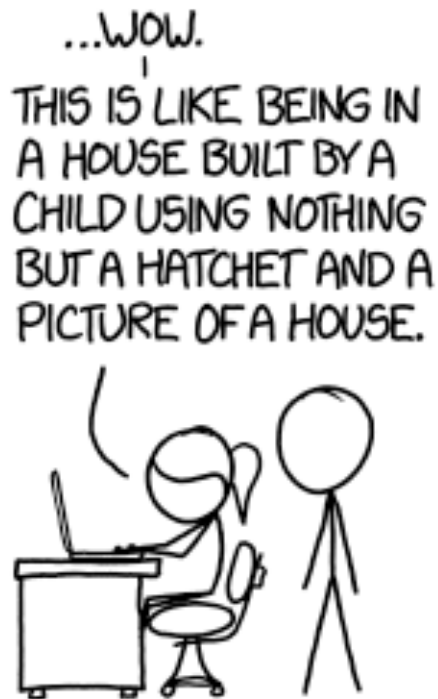


Data Types in Python



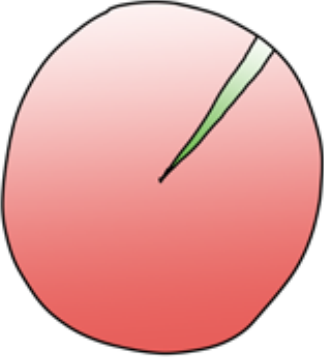
Dremendo

Coding Style (XKCD)




Think before you Act: Pseudocode

**THE USAGE OF PSEUDOCODE
IN REAL LIFE**



- DESCRIBING AN ALGORITHM
- A TOOL THAT FRESHMAN COMPUTER SCIENCE STUDENTS THAT JUST STARTED TO LEARN PROGRAMMING USES TO EXPRESS THEIR DUMB ACTIONS




Derp Johnson
@DerpyJohn

```
while (!morning){  
  alcohol++  
  dance++  
} #mylife #partyhard
```

↩ ↻ ⭐ ...

RETWEETS 48 FAVORITES 213



ctp200.com

Pseudo-Code

Calculate Average Grade

Calc_avg:

grades = input_in_data()

average = Calculations(grades)

print(average) Output

Pseudo-Code

Calculate Average Grade

Calc_avg:

```
grades = read_in_data()
```

```
average = calc_avg(grades)
```

```
print(average)
```


Pseudo-Code

Calculate Average Grade

```
Calc_avg:  
    # read input  
    file = open("gradesPython.csv","r")  
    grades = file.readline().split(",")  
  
    average = calc_avg(grades)  
  
    print(average)
```

Pseudo-Code

Calculate Average Grade

```
Calc_avg:  
    # read input  
    file = open("gradesPython.csv","r")  
    grades = file.readline().split(",")  
  
    average = total/nofStudents  
  
    print(average)
```

Pseudo-Code

Calculate Average Grade

Calc_avg:

```
# read input
file = open("gradesPython.csv","r")
grades = file.readline().split(",")
```

```
#calculate average
total = 0
nofStudents = 0
for grade in grades:
    total += grade
    nofStudents += 1
average = total/nofStudents
```

```
print(average)
```

Pseudo-Code

Calculate Average Grade

```
Calc_avg:
    # read input
    file = open("gradesPython.csv","r")
    grades = file.readline().split(",")

    #calculate average
    total = 0
    nofStudents = 0
    for grade in grades:
        total += grade
        nofStudents += 1
    average = total/nofStudents

    print(average)
```

Pseudo-Code

Calculate Average Grade

Calc_avg:

```
# read input
file = open("gradesPython.csv","r")
grades = file.readline().split(",")

#calculate average
total = 0
nofStudents = 0
for grade in grades:
    total += grade
    nofStudents += 1
average = total/nofStudents

#print average
print("Our {} students had an average grade of {}.".format(nofStudents,average )
```

TIPS &

Naming, Naming, Naming

- variables: use reasonable and self-describing names, not too long
- index variables: i,j,k
- x,y are usually used for axes in a plot

Comment your code

What does your code do?

What does it expect as input, which format?

Write your code for someone else

(you will be someone else in a few months ;))

Try not to swear or be inappropriate ;)

Always code as if the person who ends up maintaining your code is a violent psychopath who knows where you live.

TRICKS

