

En rolig start

Introduksjon til programmering

KRISTIAN BOTNEN, 2025

Introduksjon

Who are we

K o d i n g

PROGRAMMERING I GRUNNSKOLEN - NY LÆREPLAN

	Matematikk	Naturfag	Musikk	Kunst og håndverk
2	lage og følgje reglar og trinnvise instruksjonar i leik og spel			
3	lage og følgje reglar og trinnvise instruksjonar i leik og spel knytte til koordinatsystemet		Eksperimentere med rytmer, melodier og andre grunnelementer, sette sammen mønstre til komposisjoner, også ved bruk av digitale verktøy, og beskrive arbeidsprosesser og resultater	
4	lage algoritmar og uttrykkje dei ved bruk av variablar, vilkår og lykkjer			
5	lage og programmere algoritmar med bruk av variablar, vilkår og lykkjer	Utforske, lage og programmere teknologiske systemer som består av deler som virker sammen	Bruke teknologi og digitale verktøy til å skape, øve inn og bearbeide musikk	Bruke programmering til å skape interaktivitet og visuelle uttrykk
6	bruke variablar, lykkjer, vilkår og funksjonar i programmering til å utforske geometriske figurar og mønster			
7	bruke programmering til å utforske data i tabellar og datasett			
8	utforske korleis algoritmar kan skapast, testast og forbetrast ved hjelp av programmering	Utforske, forstå og lage teknologiske systemer som består av en sender og en mottaker Bruke programmering til å utforske naturfaglige fenomener	Skape og programmere musikalske forløp ved å eksperimentere med lyd fra ulike kilder	Utforske hvordan digitale verktøy og ny teknologi kan gi muligheter for kommunikasjonsformer og opplevelser i skapende prosesser og produkter
9	simulere utfall i tilfeldige forsøk og berekne sannsynet for at noko skal inntreffe ved å bruke programmering			
10	utforske matematiske eigenskapar og samanhengar ved å bruke programmering			

Hvem er vi

Hvem er "dere"?

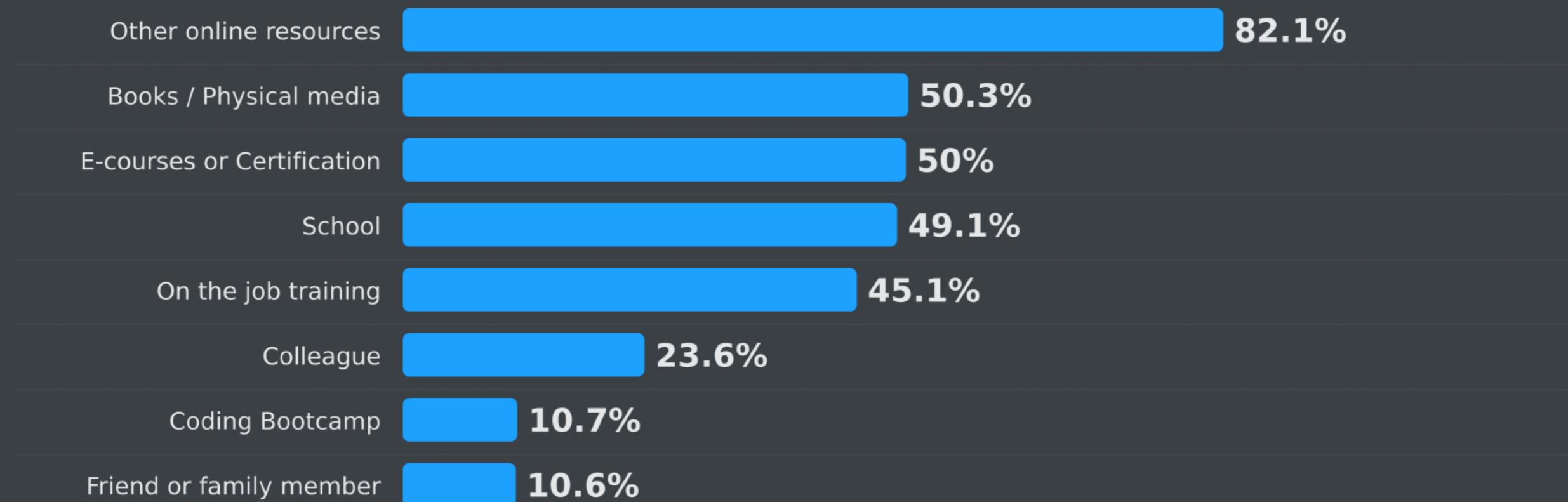


<https://survey.stackoverflow.co/2024/>

Hvem er "dere"?

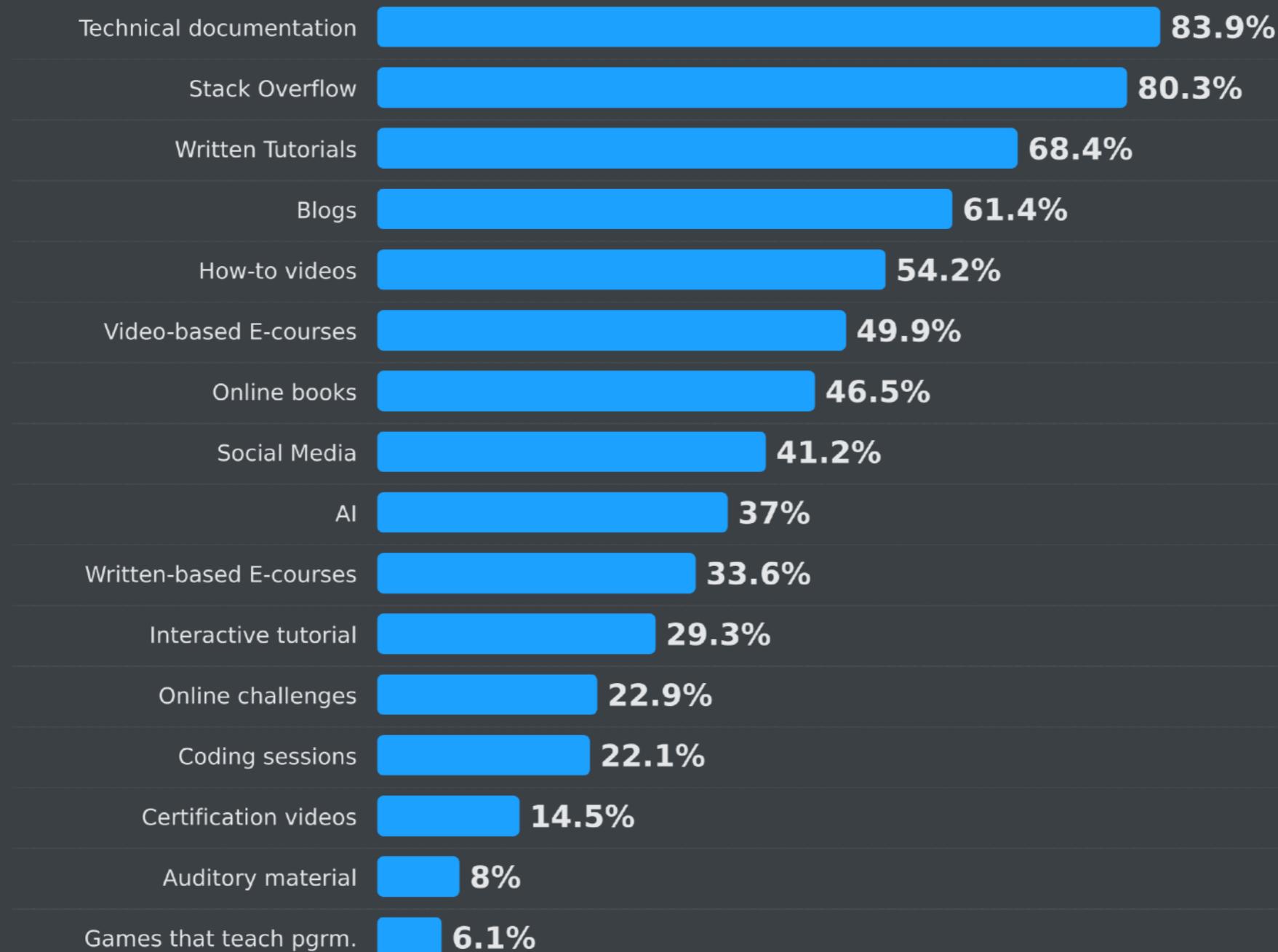
Learning to code / All Respondents

Learning to code



Learning to code / All Respondents

Online resources to learn how to code



Hvem er "dere"?



AI tools in the development process

76% of all respondents are using or are planning to use AI tools in their development process this year, an increase from last year (70%). Many more developers are currently using AI tools this year, too (62% vs. 44%).

?

Do you currently use AI tools in your development process? *

Accuracy of AI tools

Similar to last year, developers remain split on whether they trust AI output: 43% feel good about AI accuracy and 31% are skeptical. Developers learning to code are trusting AI accuracy more than their professional counterparts (49% vs. 42%).

?

How much do you trust the accuracy of the output from AI tools as part of your development workflow?

Benefits of AI tools

81% agree increasing productivity is the biggest benefit that developers identify for AI tools. Speeding up learning is seen as a bigger benefit to developers learning to code (71%) compared to professional developers (61%).

?

For the AI tools you use as part of your development workflow, what are the MOST important benefits you are hoping to achieve? Please check all that apply.

Are AI tools a threat to your job

70% of professional developers do not perceive AI as a threat to their job.

?

Do you believe AI is a threat to your current job?

Hvem er dere?



<https://www.menti.com/alw8a1qyqg1c>

Hjem er dere?

Hva er motivasjonen din for å delta på pythonkurs?

Hvorfor lære python?

458 responses

Hvem er dere?

En subjektiv vurdering av eget nivå

Tidligere erfaring med programering?



Hjem er dere?

Forventninger til kvelden?

310 responses



Dagens agenda

Hva er python

Noen kule datatyper

Variabler + oppgave 1

Kodeblokker & løkker + oppgave 2

If - then - else + oppgave + oppgave 3

Funksjoner og moduler + bonusoppgave

Oppsummering og veien videre

Kursmateriell



https://github.com/kbotnen/python_geovitenskap

Hva er Python

Python elevator pitch

Easy Efficient Interpreted Readable

Free as in beer

Community



Extensible Cross-plattform Portable

Well documented Open Source

KOMPILERT KODE

- Raskere kode
- Klar til å kjøres når brukeren får programmet
- Et ekstra steg i utviklingen
- Låst til plattform
- Kildekoden er lukket
- Eks: C++, Java, Swift

TOLKET KODE

- Enklere å utvikle, enklere å teste
- Ikke låst til plattform
- Brukeren trenger en tolker for å kjøre programmet
- Kildekoden er åpen
- Eks: Ruby, Matlab, Python

The image displays a grid of 15 code editors, each showing a "Hello, World!" program in a different programming language. The languages and their corresponding code snippets are:

- Go**: package main; import "fmt"; func main() { fmt.Println("Hello World") }
- lolo**: "Hello, world!" println
- Java**: class helloWorld { public static void main(String args[]) { System.out.println("Hello World"); } }
- JavaScript**: alert('Hello World');
- Objective-C**: #import <stdio.h> #import <Foundation/Foundation.h> int main(void) { NSLog(@"Hello, world!\n"); return 0; }
- Pascal**: Program HelloWorld; Begin Writeln ('Hello World'); End.
- Perl**: #!/usr/bin perl -w print ("hello world");
- PHP**: <?= "Hello world\n" ?>
- Postscript**: %! /Helvetica 20 selectfont 100 100 moveto (Hello World) show showpage
- Prolog**: main :- write('Hello, world!'), nl.
- Python**: print('Hello World')
- React**: ReactDOM.render(<h1>Hello, world!</h1>, document.getElementById('root'));
- REXX**: say "Hello World"
- Ruby**: #!/usr/local/bin/ruby -w puts "Hello world"
- Scala**: object HelloWorld extends App { println("Hello World") }
- Smalltalk**: Transcript show: 'Hello, world!'.
Transcript show: 'Hello, world!'.
- Standard ML**: print "Hello World\n";
- Swift**: println("Hello, world")

PEP 8 – Style Guide for Python Code

Author: Guido van Rossum <guido at python.org>, Barry Warsaw <barry at python.org>, Nick Coghlan <ncoghlan at gmail.com>

Status: Active

Type: Process

Created: 05-Jul-2001

Post-History: 05-Jul-2001, 01-Aug-2013

► Table of Contents

Introduction

This document gives coding conventions for the Python code comprising the standard library in the main Python distribution. Please see the companion informational PEP describing [style guidelines for the C code in the C implementation of Python](#).

This document and [PEP 257](#) (Docstring Conventions) were adapted from Guido's original Python Style Guide essay, with some additions from Barry's style guide [2].

This style guide evolves over time as additional conventions are identified and past conventions are rendered obsolete by changes in the language itself.

Many projects have their own coding style guidelines. In the event of any conflicts, such project-specific guides take precedence for that project.

A Foolish Consistency is the Hobgoblin of Little Minds

One of Guido's key insights is that code is read much more often than it is written. The guidelines provided here are intended to improve the readability of code and make it consistent across the wide spectrum of Python code. As [PEP 20](#) says, "Readability counts".

A style guide is about consistency. Consistency with this style guide is important. Consistency within a project is more important. Consistency within one module or function is the most important.

However, know when to be inconsistent – sometimes style guide recommendations just aren't applicable. When in doubt, use your best judgment. Look at other examples and decide what looks best. And don't hesitate to ask!

In particular: do not break backwards compatibility just to comply with this PEP!

Lokalt



Nettbasert

Python shell

```
(base) [kbo041@isfjell ~]$ python
Python 3.12.1 | packaged by Anaconda, Inc. | (main, Jan 19 2024, 15:51:05) [GCC 11.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello world")
Hello world
>>> exit()
(base) [kbo041@isfjell ~]$ cat helloscript.py
#!/home/kbo041/miniconda3/bin/python
print("Hello world")
(base) [kbo041@isfjell ~]$ python helloscript.py
Hello world
(base) [kbo041@isfjell ~]$ █
```

iPython shell

```
(jupyter) [kbo041@isfjell ~]$ ipython
Python 3.12.2 | packaged by Anaconda, Inc. | (main, Feb 27 2024, 17:35:02) [GCC 11.2.0]
Type 'copyright', 'credits' or 'license' for more information
IPython 8.20.0 -- An enhanced Interactive Python. Type '?' for help.
```

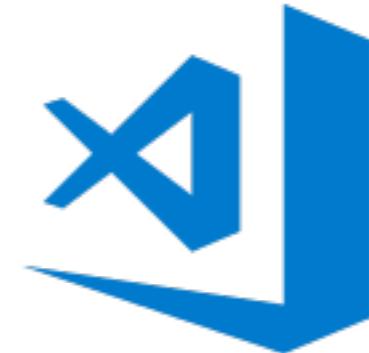
```
In [1]: print("Hello world")
Hello world
```

```
In [2]: print("Hello world")
```

Python utviklingsmiljø



<https://www.anaconda.com/docs/getting-started/miniconda/install>



Visual Studio Code

<https://code.visualstudio.com/Download>



<https://www.jetbrains.com/pycharm/>



<https://www.spyder-ide.org>

Sett inn noen skjermbilder som viser forskjellige måter å aksessere filer på.

Noen kule typer

1 = Tall
1.0 = Tall med desimal
a = Bokstav
hei = Ord
hei du = Setning

VERDIER

1 = int
1.0 = float
a = string
hei = string
hei du = string

TYPER

Typer

```
:::python
Text Type:           str
Numeric Types:      int, float, complex
Sequence Types:     list, tuple, range
Mapping Type:       dict
Set Types:          set, frozenset
Boolean Type:       bool
Binary Types:       bytes, bytearray, memoryview
None Type:          NoneType
```

Variabler

Variabelnavn

1

Variabelnavn

a

Variabelnavn

Kristian

variabel_a

1

variabel_b

"a"

variabel_c

"Kristian"

Pythonkode

```
variabelnavn_a = 1  
variabelnavn_b = "a"  
variabelnavn_c = "Kristian"
```

Pythonkode

```
variabelnavn_a = 1
```

```
variabelnavn_b = "a"
```

```
variabelnavn_c = "Kristian"
```

```
variabelnavn_a = "Python er gøy"
```

Oppgavetid

Kodeblokker & løkker

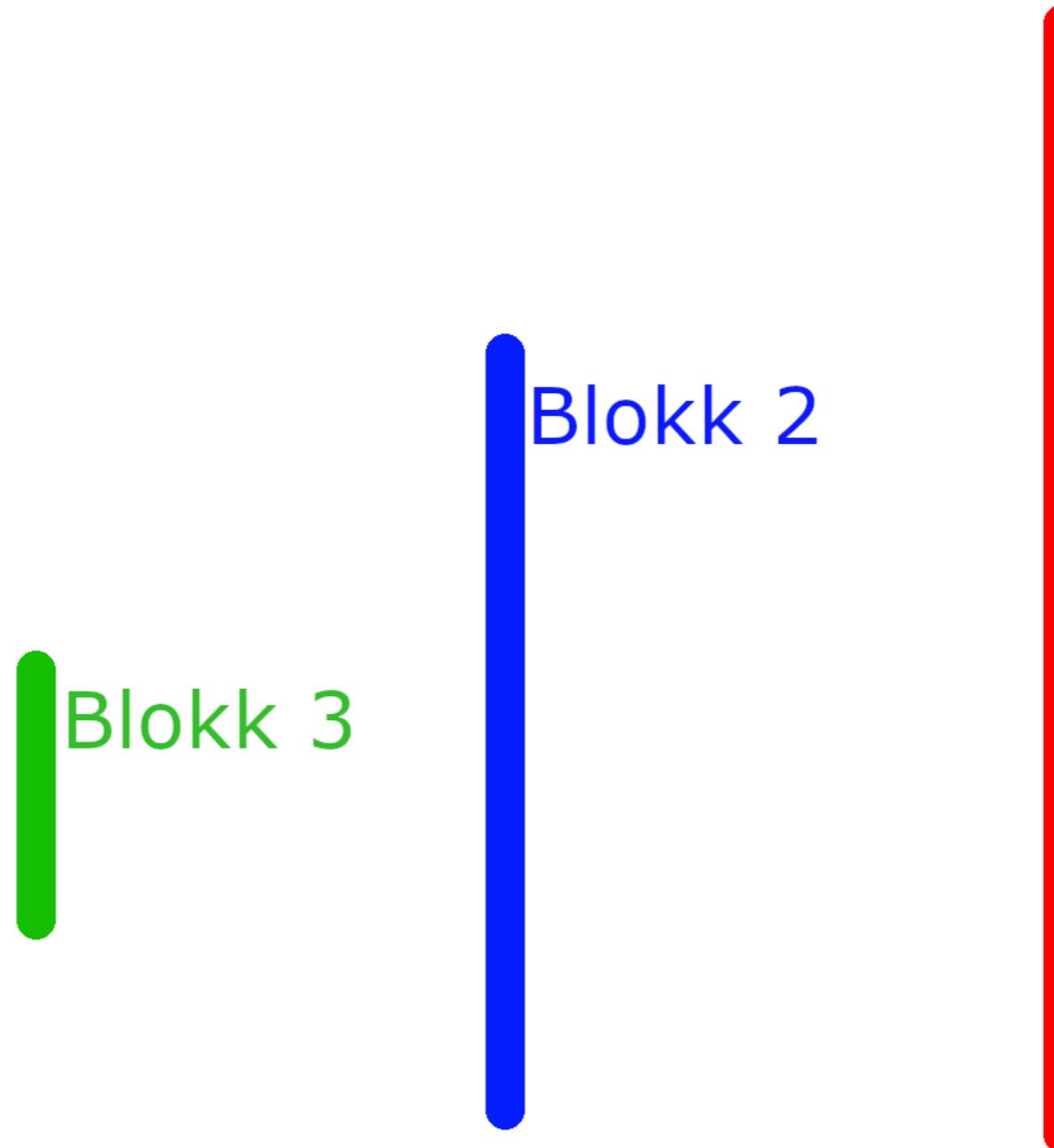
Kodeblokk

kodelinje

Blokk 3

Blokk 2

Blokk 1



Kodeblokk

kodelinje

Blokk 2

Blokk 3

Blokk 1

Tilstandsløkke

Ta på lue så lenge det snør ute.

Så lenge det snør ute så ta på lue

Om det ikke snør ute, ikke ta på lue



Tilstandsløkke

```
Linje 1     i = 0
Linje 2     while (i < 5):
Linje 3         print(i)
Linje 4         i = i + 1
```

Telleløkke

```
for i in range(0, 5):  
    print(i)
```

Oppgavetid

If - then - else

If - then - else

```
alder = 18
if alder > 18:
    print("Du er myndig")
else:
    print("Du er ikke myndig")
```

If - then - else

<u>Operator</u>	<u>Navn</u>	<u>Eksempel</u>
<code>==</code>	Equal	<code>x == y</code>
<code>!=</code>	Not Equal	<code>x != y</code>
<code>></code>	Greater than	<code>x > y</code>
<code><</code>	Less than	<code>x < y</code>
<code>>=</code>	Greater than or equal to	<code>x >= y</code>
<code><=</code>	Less than or equal to	<code>x <= y</code>

Funksjoner og moduler

```
::::python
def greet_function():
    print("Hello from a function")

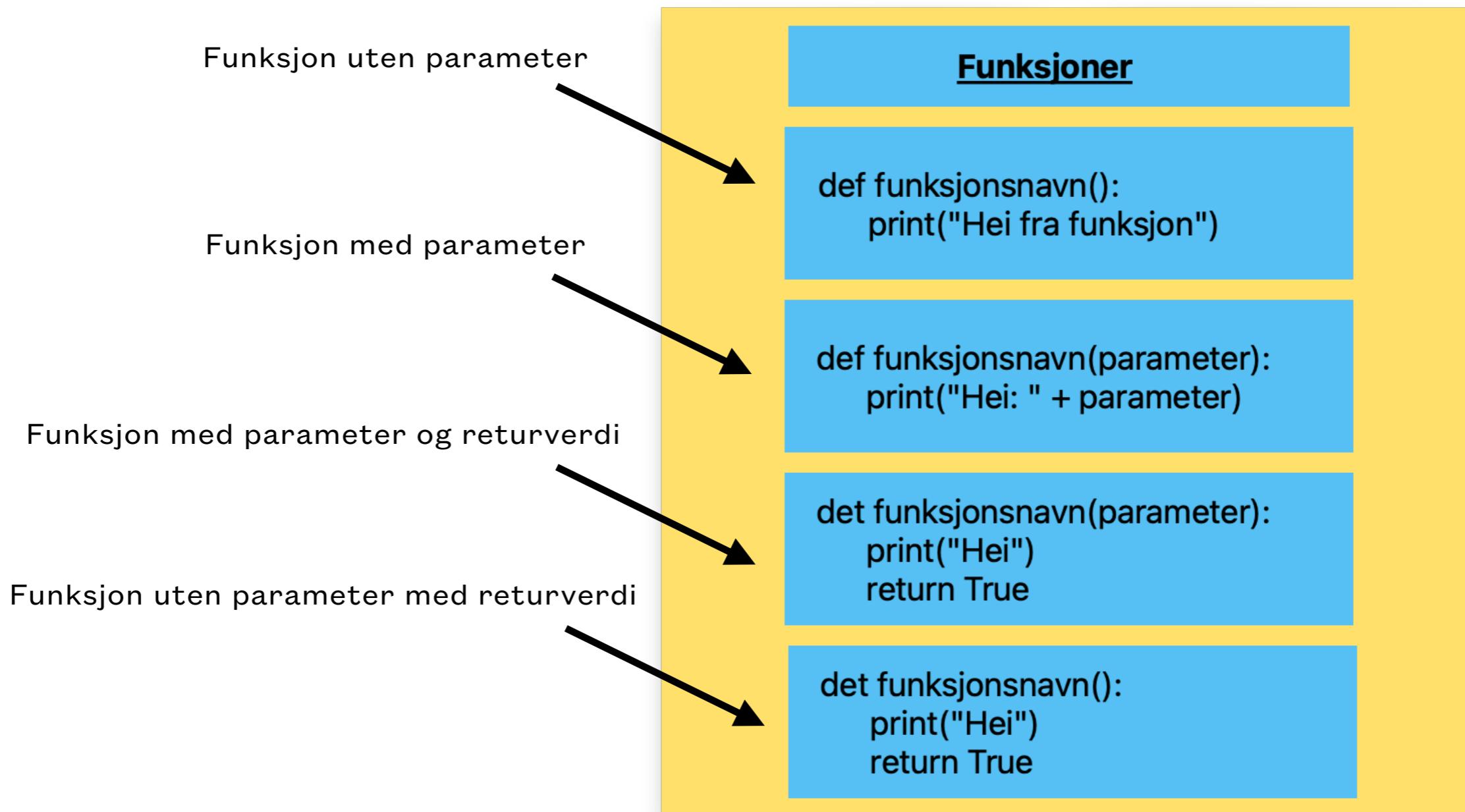
greet_function()
```

Funksjoner

```
::::python
def fibonacci(n):
    if n <= 1: # If the number is 0, then the answer is 0. If the number is 1, then the answer is 1.
        return n
    else:
        return fibonacci(n - 1) + fibonacci(n - 2) # Each successive fibonacci number is found by adding up the two numbers before it.

print('Fibonacci sequence:')
for i in range(5):
    print(fibonacci(i))
```

Funksjoner



Moduler

```
import random  
  
for i in range(10):  
    print(random.randint(1, 25))
```

Moduler

```
import numpy as np  
  
x = np.array([1, 2, 3])  
print(x)
```

Modul sikkerhet

https://docs.python.org/3/library/security_warnings.html

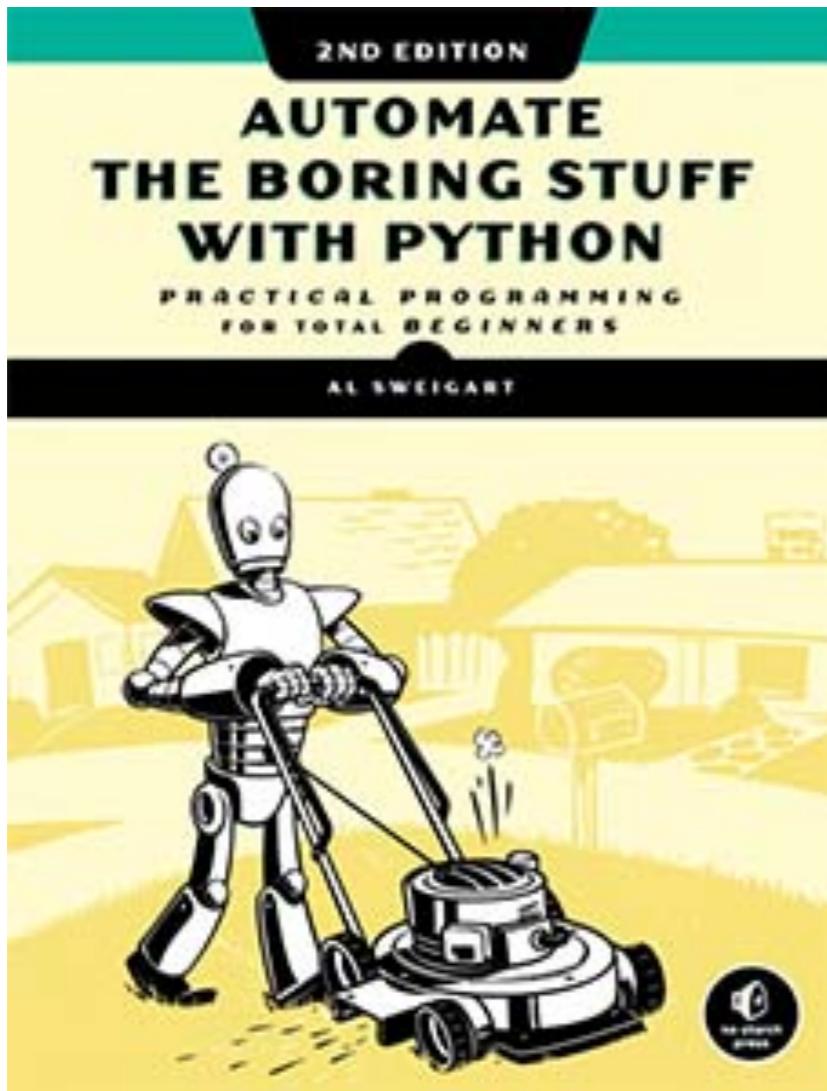
<https://app.opencve.io/cve/?vendor=python>

<https://blog.phylum.io/a-pypi-typosquatting-campaign-post-mortem/>

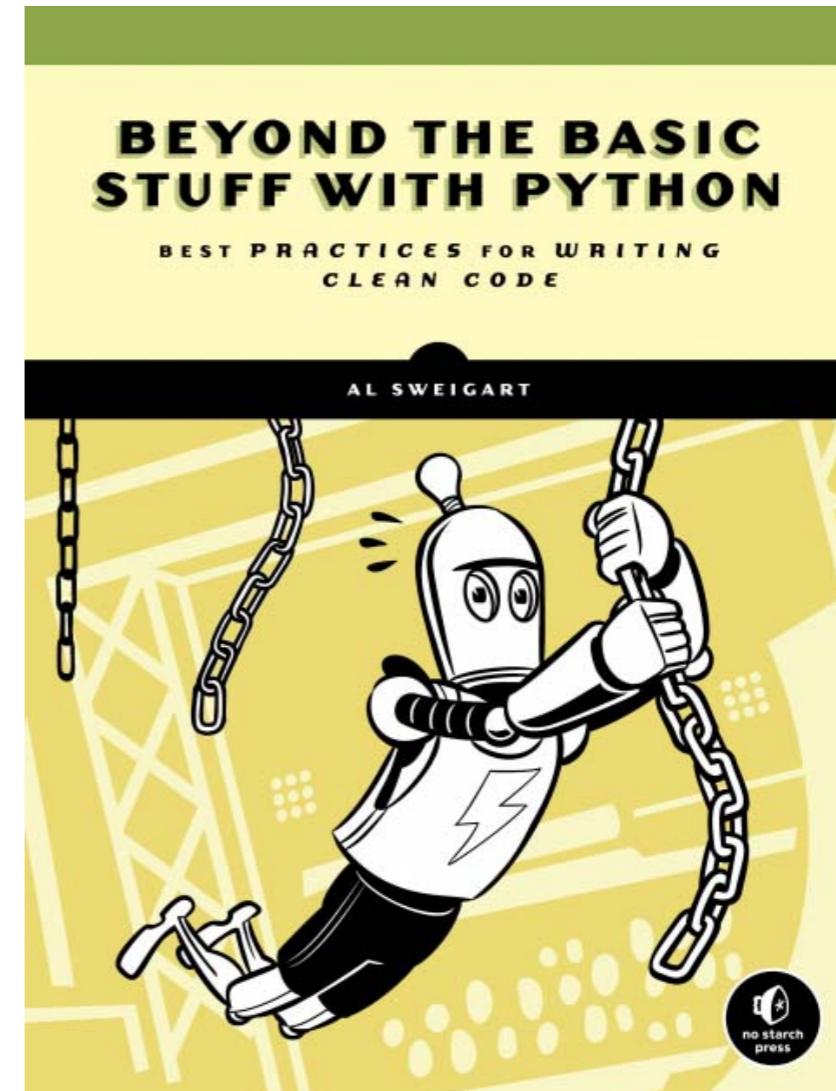
Base64
Hashlib
http.server
Logging
Multiprocessing
Pickle
Random
Shelve
Ssl
Subprocess
Tempfile
Xml
Zipfile

Oppgavetid

Veien videre



<https://automatetheboringstuff.com>



<https://inventwithpython.com/beyond/>

exercism

Learn Discover Contribute More Insiders 🔜

Python 477,283 students

Overview Practice About Python Practice Mode 352 contributors 6239 mentors

Explore the Python exercises on Exercism

Unlock more exercises as you progress. They're great practice and fun to do!

Search by title

All Exercises 140 Completed 0 In Progress 1 Available 139 Locked 0

Hello World
in-progress 1 iteration
Exercism's classic introductory exercise. Just say "Hello, World!".

Guido's Gorgeous Lasagna
Recommended Learning Exercise
Learn the basics of Python by cooking Guido's Gorgeous Lasagna.

Ghost Gobble Arcade Game
Available Learning Exercise
Learn about bools by setting up the rules for the Ghost Gobble arcade game.

Currency Exchange
Available Learning Exercise
Learn about numbers by solving Chandler's currency exchange conundrums.

Meltdown Mitigation
Available Learning Exercise
Learn about conditionals and avoid a meltdown by developing a simple control system for a Nuclear Reactor.

Black Jack
Available Learning Exercise
Learn about comparisons by implementing some Black Jack judging rules.

Little Sister's Essay
Available Learning Exercise
Learn about string methods while improving your little sister's school essay.

Little Sister's Vocabulary
Available Learning Exercise
Learn about strings by helping your little sister with her vocabulary homework.

<https://exercism.org/>

python



Norsk



Alle (171)

Programmering (155)

Vitenskap og teknologi (142)

Utdanning (81)

Studier og undervisning (53)

Spillutvikler (41)

Boter (38)

Samarbeid (37)



Python

We're a large community focused around the Python programming language.
We believe that anyone can learn to code.

48 818 pålogget • 392 425 medlemmer



World of Coding | Programming, Code, Infosec, Hacking, Tech,
Linux, Cybersec, Python, Java, math

WoC provides some of the best programming help on discord and provides a
place for developers to find friends and relax!

8 299 pålogget • 63 003 medlemmer

<https://support.discord.com/>

-
- <https://github.com/kbotnen/pythonkveldskurs>
 - <https://oppgaver.kidsakoder.no/python>
 - <https://www.w3schools.com/python/default.asp>
 - <https://www.learnpython.org>
 - <https://replit.com/learn/100-days-of-python>
 - <https://www.reddit.com/r/Python/>
 - <https://www.pythondiscord.com/resources/>
-