
Python for ingenøren

Who am I

NITO

Digitalisering

K o d i n g

3182 1586



<https://www.menti.com/alj6mx19imyz>

https://github.com/kbotnen/pythonkurs_h25

Her viser vi sessionlabprogrammet

En rolig start

Introduksjon til programering - Del 2

Øktens agenda

Noen flere datatyper

Funksjoner og moduler

Dokumentasjon og hjelp

Problemløsning med python

Virtuelle miljøer

Versjonshåndtering

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")
```

Navy

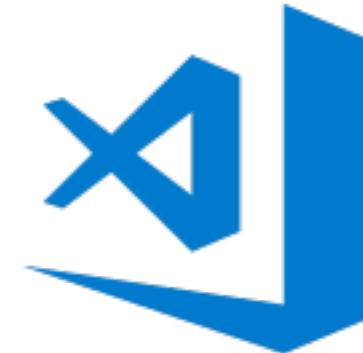
- > cryptservices
- > demonstrasjon
- > envtest
- > krus
- > lmlicenses
- > macdeploy
- > macintune
- > macprofiles
- > mdmenrollment
- > mdmservices
- > ms-graph-testing
 - > Adrian
 - > friedrich
 - > ingar
 - > python_examples
 - > __pycache__
 - auth_cache.txt
 - > code
 - __init__.py
 - > __pycache__
 - apitest_cache.sqlite
 - auth_cache.txt
 - graph_cache.sqlite
 - msgraphtest_clientsecret_createuser.py
 - msgraphtest_clientsecret_getusers.py
 - msgraphtest_msal.py
 - msgraphtest_msaltoken.py
 - msgraphtest_msgraphsdk.py
 - msgraphtest_userauth_getuserinfo.py
 - msgraphtest_userauth_getusermail.py
 - requirements.txt
 - > conf
 - > docs
 - graph_cache.sqlite
 - IT-KD-GraphAPI-Mac-01.cer
 - > profiles
 - README.md
 - README.md
 - > mwa3
 - > packages
 - > puppet
 - > pythonproject
 - > uib-klientinfo
 - > uibtools
 - > winston
 - > github
 - > panda3d

```
(base) kristianbotnen@l76xn0ryfv ~ % cd Utvikling/git/ms-graph-testing/python_examples/code  
(base) kristianbotnen@l76xn0ryfv code % ls  
__init__.py                      msgraphtest_msal.py  
__pycache__                         msgraphtest_msaltoken.py  
apistest_cache.sqlite                msgraphtest_msgraphsdk.py  
auth_cache.txt                     msgraphtest_userauth_getuserinfo.py  
graph_cache.sqlite                  msgraphtest_userauth_getusermail.py  
msgraphtest_clientsecret_createuser.py requirements.txt  
msgraphtest_clientsecret_getusers.py  
(base) kristianbotnen@l76xn0ryfv code %
```

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>



Websider - Django, Flask, Bottle, FastAPI og Pelican

Databaser - Sqlite, SQLAlchemy og Psycopg2

Spillutvikling - Pygame, Pygame Zero, Panda3D og Blender 3D

Maskinlæring - Scikitlearn, OpenCV, Tensorflow og Pytorch

Computervision - OpenCV, Tensorflow og Pytorch

IT-drift - Ansible

Datascience - Pandas, Matplotlib, Numpy og Scipy

GUI-applikasjoner - Tkinter, Kivy og PyQt



Typer

Variabler

Løkker

Tester

Datastrukturer

Samlinger

Abstraksjoner

Lese og skrive filer

Lese og skrive databaser

Nettverkskommunikasjon

Internettprotkoller

Noen flere datatyper

Syntax

```
:::python
def test():
    print("Hello world")
```

Variabler

```
 ::=python
# Define some variables
x = 5
name = "Kristian"

# Use our variables
print(x)
print(name)
```

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
```

Operatorer

```
:::python
Arithmetic operators      (+, -, *, /, %, **, //)
Assignment operators       (=, +=, -=, *=, /=, %=, //=, **=, &=, |=, ^=, >>=, <<=)
Comparison operators      (==, !=, >, <, >=, <=)
Logical operators         (and, or, not)
Identity operators        (is, is not)
Membership operators      (in, not in)
Bitwise operators          (&, |, ^, ~, <<, >>)
```

Python Operators and Booleans Cheat Sheet by Nouha_Thabet

Python Arithmetic Operators

| | | |
|----------------|---------------------|---------------------------|
| Addition | <code>9 + 2</code> | <code>>> 11</code> |
| Subtraction | <code>9 - 2</code> | <code>>> 7</code> |
| Multiplication | <code>9 * 2</code> | <code>>> 18</code> |
| Division | <code>9 / 2</code> | <code>>> 4.5</code> |
| Modulus | <code>9 % 2</code> | <code>>> 1</code> |
| Exponentiation | <code>3 ** 2</code> | <code>>> 81</code> |
| Floor division | <code>9 // 2</code> | <code>>> 4</code> |

Python Assignment Operators

| Operator | Example | Same As |
|----------|----------------------|-------------------------|
| = | <code>x = 2</code> | <code>x = 2</code> |
| += | <code>x += 2</code> | <code>x = x + 2</code> |
| -= | <code>x -= 2</code> | <code>x = x - 2</code> |
| *= | <code>x *= 2</code> | <code>x = x * 2</code> |
| /= | <code>x /= 2</code> | <code>x = x / 2</code> |
| %= | <code>x %= 2</code> | <code>x = x % 2</code> |
| //= | <code>x //= 2</code> | <code>x = x // 2</code> |
| **= | <code>x **= 2</code> | <code>x = x ** 2</code> |

Python Comparison Operators

| | |
|--------------------------|------------------------|
| Equal | <code>x == y</code> |
| Not equal | <code>x != y</code> |
| Greater than | <code>x > y</code> |
| Less than | <code>x < y</code> |
| Greater than or equal to | <code>x >= y</code> |
| Less than or equal to | <code>x <= y</code> |

Boolean Values

In programming you often need to know if an expression is True OR False. You can evaluate any expression in Python, and get the answer.

```
print(5 < 8)           >>> True
print(5 > 8)           >>> False
```

Python Logical Operators

and Returns True if both statements are true
`x < 5 and x < 10`

or Returns True if one of the statements is true
`x < 5 or x < 4`

not Reverse the result, returns False if the result is true
`not(x < 5 and x < 10)`

Python Identity Operators

is Returns true if both variables are the same object
`x is y`

is not Returns true if both variables are not the same object
`x is not y`

Python Membership Operators

in Returns True if a sequence with the specified value is present in the object
`x in y`

not in Returns True if a sequence with the specified value is not present in the object
`x not in y`

Python Bitwise Operators

| | | |
|----|----------------------|---|
| & | AND | Sets each bit to 1 if both bits are 1 |
| | OR | Sets each bit to 1 if one of two bits is 1 |
| ^ | XOR | Sets each bit to 1 if only one of two bits is 1 |
| - | NOT | Inverts all the bits |
| << | Zero fill left shift | Shift left by pushing zeros in from the right and let the leftmost bits fall off |
| >> | Signed right shift | Shift right by pushing copies of the leftmost bit in from the left, and let the rightmost bits fall off |

Int / Float / Complex

```
:::python
var_int = 1
var_float = 1.0
var_complex = 1j

print(var_int, var_float, var_complex, sep=', ')
print(type(var_int), type(var_float), type(var_complex), sep=', ')
```

String

```
:::python
print("Hello")
print('Hello')
print("""Lorem ipsum dolor sit amet,
consectetur adipiscing elit,
sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua.""")
```

Boolean

```
:::python
var_string = "Hello World!"
if ('ello' in var_string): # The values between ( and ) will evaluate to True or False, in this example True
    print("We found 'ello' in our string")
else:
    print("We did not find 'ello' in our string")
```

| <i>Index</i> | <i>Verdi</i> |
|--------------|--------------|
| 0 | «Henry» |
| 1 | «Ford» |
| 2 | «English» |

List

DUPLICATES
CHANGEABLE
ORDERED

| <i>Nøkkel</i> | <i>Verdi</i> |
|---------------|--------------|
| «f_Name» | «Henry» |
| «l_Name» | «Ford» |
| «language» | «English» |

Dictionary

NO-DUPLICATES
CHANGEABLE
ORDERED

Collections

| <i>Index</i> | <i>Verdi</i> |
|--------------|--------------|
| 0 | «Henry» |
| 1 | «Ford» |
| 2 | «English» |

Tuple

DUPLICATES
UNCHANGEABLE
ORDERED

| <i>Verdi</i> |
|--------------|
| «Henry» |
| «Ford» |
| «English» |

Set

NO-DUPLICATES
UNCHANGEABLE
UNORDERED

Lists

| | |
|------------------|--|
| :::python | |
| append() | Adds an element at the end of the list |
| clear() | Removes all the elements from the list |
| copy() | Returns a copy of the list |
| count() | Returns the number of elements with the specified value |
| extend() | Add the elements of a list (or any iterable), to the end of the current list |
| index() | Returns the index of the first element with the specified value |
| insert() | Adds an element at the specified position |
| pop() | Removes the element at the specified position |
| remove() | Removes the item with the specified value |
| reverse() | Reverses the order of the list |
| sort() | Sorts the list |

Dictionaries

```
:::python
```

| | |
|---------------------------|---|
| <code>clear()</code> | Removes all the elements from the dictionary |
| <code>copy()</code> | Returns a copy of the dictionary |
| <code>fromkeys()</code> | Returns a dictionary with the specified keys and value |
| <code>get()</code> | Returns the value of the specified key |
| <code>items()</code> | Returns a list containing a tuple for each key value pair |
| <code>keys()</code> | Returns a list containing the dictionary's keys |
| <code>pop()</code> | Removes the element with the specified key |
| <code>popitem()</code> | Removes the last inserted key-value pair |
| <code>setdefault()</code> | Returns the value of the specified key. If the key does not exist: insert the key, with the specified value |
| <code>update()</code> | Updates the dictionary with the specified key-value pairs |
| <code>values()</code> | Returns a list of all the values in the dictionary |

Tuples

```
:::python
count()    Returns the number of times a specified value occurs in a tuple
index()    Searches the tuple for a specified value and returns the position of where it was found
```

Sets

| | |
|--|--|
| <code>add()</code> | Adds an element to the set |
| <code>clear()</code> | Removes all the elements from the set |
| <code>copy()</code> | Returns a copy of the set |
| <code>difference()</code> | Returns a set containing the difference between two or more sets |
| <code>difference_update()</code> | Removes the items in this set that are also included in another, specified set |
| <code>discard()</code> | Remove the specified item |
| <code>intersection()</code> | Returns a set, that is the intersection of two other sets |
| <code>intersection_update()</code> | Removes the items in this set that are not present in other, specified set(s) |
| <code>isdisjoint()</code> | Returns whether two sets have a intersection or not |
| <code>issubset()</code> | Returns whether another set contains this set or not |
| <code>issuperset()</code> | Returns whether this set contains another set or not |
| <code>pop()</code> | Removes an element from the set |
| <code>remove()</code> | Removes the specified element |
| <code>symmetric_difference()</code> | Returns a set with the symmetric differences of two sets |
| <code>symmetric_difference_update()</code> | inserts the symmetric differences from this set and another |
| <code>union()</code> | Return a set containing the union of sets |
| <code>update()</code> | Update the set with the union of this set and others |

```
name = "Kristian"
print(f"Hello, {name}!")
print(f"{2*2}")
print(f"Hello, {name.upper()}!")
```

```
overskudd = 500000.987654321
print(f"Overskudd: {overskudd:.2f}")
```

F-strings

```
university = {"name": "UiB", "location": "Bergen" }
print(f"Enlisted at {university['name']}, campus {university['location']}")
```

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))
```

Funksjon uten parameter

Funksjoner

```
def funksjonsnavn():  
    print("Hei fra funksjon")
```

Funksjon med parameter

```
def funksjonsnavn(parameter):  
    print("Hei: " + parameter)
```

Funksjon med parameter og returverdi

```
def funksjonsnavn(parameter):  
    print("Hei")  
    return True
```

Funksjon uten parameter med returverdi

```
def funksjonsnavn():  
    print("Hei")  
    return True
```

```
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

Modul create

<https://docs.python.org/3/tutorial/modules.html>

Dokumentasjon og hjelp

```
"""Gets and echo out a given string.

Parameters
-----
name : string
    A string that is part of a greeting

Returns
-----
string
    a string that contains a greeting
"""

# Kommentar
def echo_name(name: str) -> str:
    """Gets and echo out a given string.

    Parameters
    -----
    name : string
        A string that is part of a greeting

    Returns
    -----
    string
        a string that contains a greeting
    """

    return(f"Hello {name}")
```

Dokumentasjon

<https://docs.python.org/3/library/typing.html>

<https://peps.python.org/pep-0257/>

```
# Kommentar
def echo_name(name: str) -> str:
    return(f"Hello {name}")
```

```
Type "help", "copyright", "credits" or "license" for more information.
>>> help()
Welcome to Python 3.12's help utility! If this is your first time using
Python, you should definitely check out the tutorial at
https://docs.python.org/3.12/tutorial/.
```

Enter the name of any module, keyword, or topic to get help on writing Python programs and using Python modules. To get a list of available modules, keywords, symbols, or topics, enter "modules", "keywords", "symbols", or "topics".

help()

Each module also comes with a one-line summary of what it does; to list the modules whose name or summary contain a given string such as "spam", enter "modules spam".

```
>>> help(print)
>>> import random
>>> help(random)
>>> help(random.randint)
>>> help("if")
>>> help("symbols")
>>> help("keywords")
>>> help("modules")
```

To quit this help utility and return to the interpreter, enter "q" or "quit".

Offisiell python dokumentasjon

<https://www.python.org/doc/>

Numpystyle docstrings

<https://numpydoc.readthedocs.io/en/latest/format.html#docstring-standard>

Realpython help()

<https://realpython.com/ref/builtin-functions/help/>

Ressurser

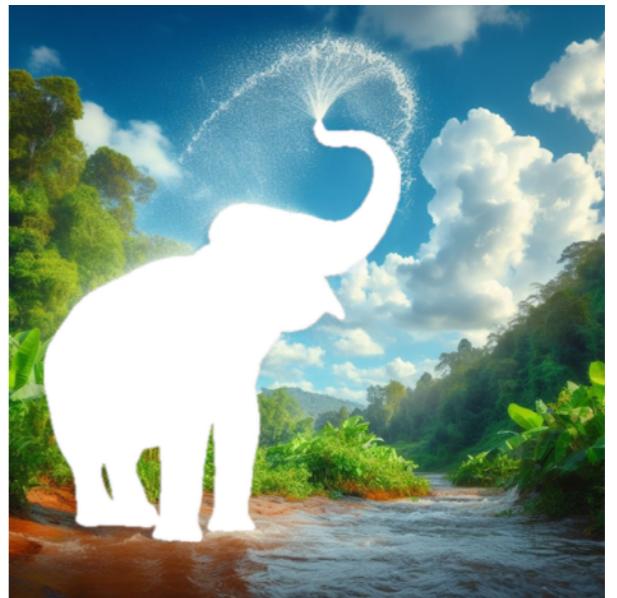
Oppgavetid

Problemløsning med python



=

+



"A happy elephant"

Laga av Bing bletskapar

Levert av DALL·E 3

STEG 1 - FORSTÅ PROBLEMET

```
# input = 2, 4  
# output = 6  
# function add_two_numbers(2, 4) ---> 6  
# function add_two_numbers(a, b) ---> print error
```

Problem: Lag et program som summerer to tall.

```
# input = 2, 4
```

```
# output = 6
```

```
# function add_two_numbers(2, 4) ---> 6
```

```
# function add_two_numbers(a, b) ---> print error
```

STEG 2 - LAG EN PLAN

```
# input = 2, 4  
# output = 6  
# function add_two_numbers(2, 4) ---> 6  
# function add_two_numbers(a, b) ---> print error
```

Problem: Lag et program som summerer to tall.

```
# Create a sum variable.  
# Create a function add_two_numbers that take two number as input parameters.  
# # function_add_two_numbers add the first parameter to the second parameter (+) and return the result.  
# Call the function add_two_numbers and store the result in the sum variable.  
# Print the sum variable.
```

```
# Create a sum variable.  
# Create a function add_two_numbers that take two number as input parameters.  
# # function_add_two_numbers add the first parameter to the second parameter (+) and return the result.  
# Call the function add_two_numbers and store the result in the sum variable.  
# Print the sum variable.
```

Problem: Lag et program som summerer to tall.

```
sum = 0  
  
def add_two_numbers(number_1, number_2):  
    return number_1 + number_2  
  
sum = add_two_numbers(2, 4)  
print(sum)
```

STEG 4 - EVALUER OG FORBEDRE

```
sum = 0

def add_two_numbers(number_1, number_2):
    return number_1 + number_2

sum = add_two_numbers(2, 4)
print(sum)
```

Problem: Lag et program som summerer to tall.

```
sum = 0

def add_two_numbers(number_1, number_2):
    return number_1 + number_2

print(add_two_numbers(2, 4))
```

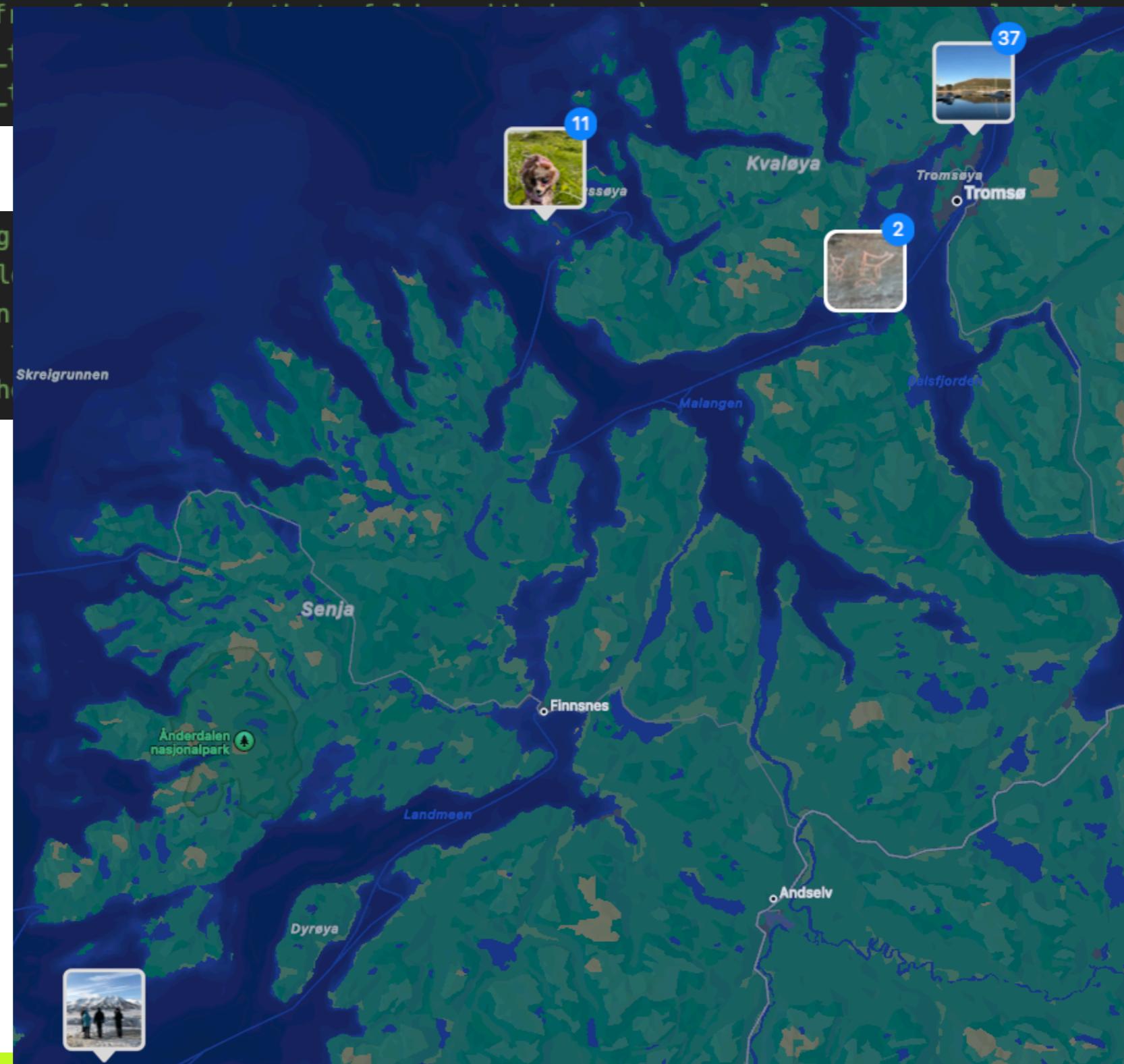
STEG 1 - FORSTÅ PROBLEMET

Problem: Geotag en mappe med bilder.

```
# input = folder containing images. foldername also contains information about location of the images.  
# output = all the images from the folder, now with a geotag / location.  
# function derive_location_from_file(  
# function read_folder(path_to_folder)  
# function read_file(path_to_file)
```

STEG 2 - LAG EN PLAN

```
# Read the folder containing  
# Create a function derive_l  
# # function derive_location  
# Create a function geo_tag  
# # function geo_tag open th
```



STEG 3 - UTFØR PLANEN

```
a.  
+ name, 'rb') as image_file:  
  
    ('/' + name, 'wb') as new_image_file:  
)  
  
ees, minutes and decimal seconds)  
lo')
```

```
try:  
    geo_tag_image(image_file, new_image_file, gps_array)  
except FileNotFoundError:  
    print(e)  
  
imagenames = os.listdir('../data/bilder/2024_04_oslo')  
for name in imagenames:  
    #print(name)  
    geo_tag(name, gps_array)
```

STEG 4 - EVALUER OG FORBEDRE

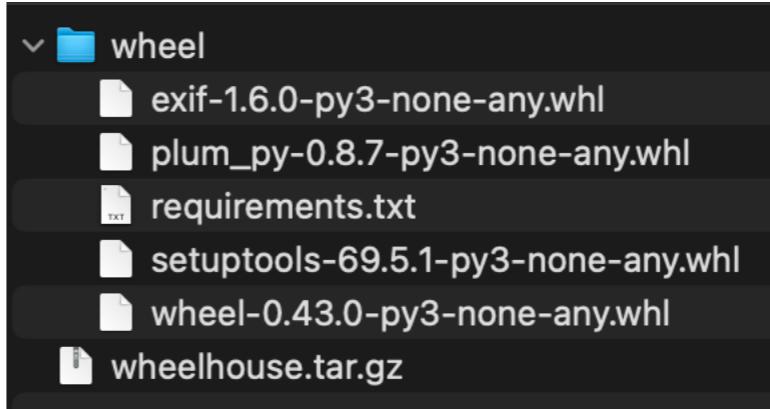
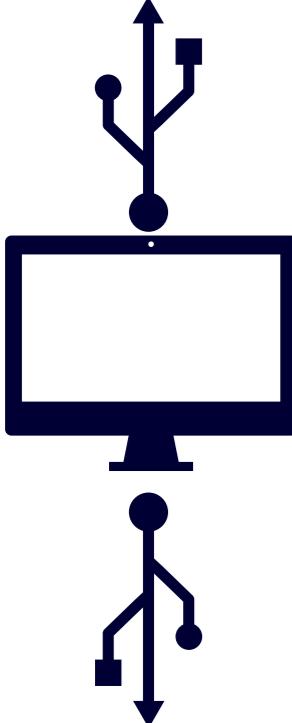
Problem: Geotag en mappe med bilder.

```
(tkinter)          kode % pip install clear
Collecting clear
  Downloading clear-2.0.0.tar.gz (2.1 kB)
    Installing build dependencies ... done
    Getting requirements to build wheel ... done
    Preparing metadata (pyproject.toml) ... done
Building wheels for collected packages: clear
  Building wheel for clear (pyproject.toml) ... done
Created wheel for clear: filename=clear-2.0.0-py3-none-any.whl size=2216 sha256=c1f036fc5e78af3ada4774c3ecc3f28e8871d46e6c4d5366c396a223e4c9b47b
  Stored in directory: /Users/kristianbotnen/Library/Caches/pip/wheels/79/91/60/042db9ddaba387b21c8ea278ab59b4b3abf320489511922558
Successfully built clear
Installing collected packages: clear
Successfully installed clear-2.0.0
```

Problem: Installere programmet mitt i et air-gapped system.

```
% pip freeze > requirements.txt  
% mkdir wheel  
% pip download -r requirements.txt -d wheel  
% mv requirements.txt wheel  
% tar -zcf wheelhouse.tar.gz wheel
```

```
% cat requirements.txt  
exif==1.6.0  
plum-py==0.8.7  
setuptools==69.5.1  
wheel==0.43.0
```

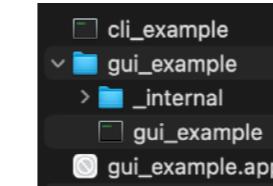


Problem: Installere programmet mitt i et air-gapped system.

```
% tar -zxf wheelhouse.tar.gz  
% pip install -r wheel/requirements.txt --no-index --find-links wheel
```

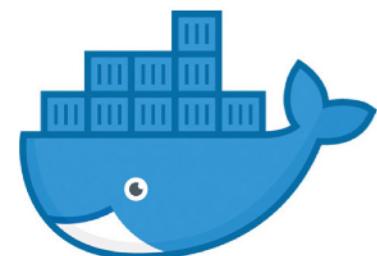


```
% pip install pyinstaller  
% pyinstaller cli_example.py --onefile  
% pyinstaller gui_example.py --noconsole
```



```
(base) dist % file cli_example.py  
cli_example.py: Mach-O 64-bit executable x86_64  
(base) dist % ./cli_example.py  
Your name: Kristian  
Hello, Kristian!  
(base) dist %
```

Problem: Installere programmet mitt i et air-gapped system.



```
FROM python:slim-bookworm  
  
MAINTAINER Kristian Botnen "kristian@botnen.org"  
  
ENV HOME /root  
ENV APPNAME webappinfo  
ENV APP_DIR /opt/botnen3d/webappinfo  
  
ENV PYTHONUNBUFFERED=1  
ENV LC_ALL en_US.UTF-8  
ENV LANGUAGE en_US.UTF-8  
ENV LANG en_US.UTF-8  
  
RUN apt-get update && apt-get install -y \  
    apache2-dev \  
    libffi-dev \  
    git \  
    && rm -rf /var/lib/apt/lists/*  
  
RUN echo "Europe/Oslo" > /etc/timezone  
RUN dpkg-reconfigure -f noninteractive tzdata  
  
COPY ./requirements.txt /tmp/requirements.txt  
RUN pip3 install --no-cache --upgrade pip setuptools && \  
    pip3 install --no-cache -r /tmp/requirements.txt && \  
    mkdir -p $APP_DIR  
  
COPY . $APP_DIR  
COPY ./run.sh $APP_DIR/run.sh  
  
WORKDIR $APP_DIR  
EXPOSE 8000/tcp  
  
CMD bash $APP_DIR/run.sh
```

```
#!/bin/bash  
  
/usr/local/bin/python $APP_DIR/$APPNAME.py
```

```
fastapi  
sqlalchemy  
psycopg2-binary  
uvicorn  
python-multipart  
python-jose  
cryptography  
passlib  
bcrypt
```

```
class Deltager:  
    def __init__(self, name: str, number: int, email: str):  
        self.name = name  
        self.number = number  
        self.email = email  
  
deltager_a = Deltager("Kristian", 2, "kristian@botnen.org")
```

```
from pydantic import BaseModel # pip install pydantic  
from pydantic import PositiveInt # pip install pydantic  
from pydantic import ValidationError # pip install pydantic  
from pydantic import EmailStr # pip install email-validator  
  
class Deltager(BaseModel):  
    name: str  
    antall: int  
    epost: EmailStr  
  
try:  
    deltager_a = Deltager(name='Kristian', antall=2, epost='kristian@botnen.org')  
except ValidationError as e:  
    print(e)
```

Problem: Lage påmeldingsskjema til sommerfesten på jobb.

```
from pydantic import BaseModel # pip install pydantic  
from pydantic import PositiveInt # pip install pydantic  
from pydantic import ValidationError # pip install pydantic  
from pydantic import EmailStr # pip install email-validator  
  
class Deltager(BaseModel):  
    name: str  
    antall: int  
    epost: EmailStr  
  
try:  
    deltager_navn = input("Navn: ")  
    deltager_antall = input("Antall: ")  
    deltager_epost = input("E-post: ")  
    deltager_a = Deltager(name=deltager_navn, antall=deltager_antall, epost=deltager_epost)  
    print(deltager_a)  
except ValidationError as e:  
    print(e)
```

Korrekt

Vedlikeholdbar

Lesbarhet

Ytelse

Sikkerhet

KORREKT

```
def divide(a, b):  
  
    try:  
        result = a / b  
    except:  
        result = None  
  
    return result
```

```
def divide(a, b):  
  
    result = None  
  
    try:  
        result = a / b  
    except ZeroDivisionError:  
        print("Type error: division by 0.")  
    except TypeError:  
        # E.g., if b is a string  
        print("Type error: division by '{0}'.".format(b))  
    except Exception as e:  
        # handle any other exception  
        print("Error '{0}' occurred. Arguments {1}.".format(e.message, e.args))  
    else:  
        # Executes if no exception occurred  
        print("No errors")  
    finally:  
        # Executes always  
        if result is None:  
            result = 0  
  
    return result
```

```
list = [1, 2, 3]  
numbers = list() # Error: TypeError: 'list' object is not callable
```

```
items = [1, 2, 3]  
numbers = list() # Ok
```

YTELSE

```
list_of_letters = ["A", "B", "C", "A", "D", "B"]  
check = "A" in list_of_letters
```

```
set_of_letters = {"A", "B", "C", "D"}  
check = "A" in set_of_letters
```

VEDLIKEHOLDBAR

```
f = open("file.txt", "r")
content = f.read()
1 / 0 # ZeroDivisionError
# never executes, possible memory issues or file corruption
f.close()
```

```
with open("file.txt", "r") as f:
    content = f.read()
    # Python still executes f.close() even though an exception occurs
    1 / 0
```

```
def get_code(password):
    if password != "automobile":
        return None
    else:
        return "42"

code = get_code("bicycle")

if code is None:
    print("Wrong password.")
else:
    print("The code is {}".format(code))
```

```
def get_code(password):
    if password != "automobile":
        raise ValueError
    else:
        return "42"

try:
    code = get_code("bicycle")
    print("The code is {}".format(code))
except ValueError:
    print("Wrong password.")
```

```
d = {'data': [{('a': 'b'), ('b': 'c'), ('c': 'd')}, 'texts': ['a', 'b', 'c']]}

for k, v in d.iteritems():
    if k == 'data':
        for i in v:
            for k2, v2 in i.iteritems():
                print(k2, v2)
```

```
data_dict = {
    'data': [{('a': 'b'), ('b': 'c'), ('c': 'd')}, 'texts': ['a', 'b', 'c']]
}

for key, value in data_dict.iteritems():
    if key == 'data':
        for data_item in value:
            # Do you know what are you iterating now?
            for data_key, data_value in data_item.iteritems():
                print(data_key, data_value)
```

```
number = None
```

```
if number == None:  
    print("This works, but is not the preferred PEP 8 pattern")
```

```
number = None
```

```
if number is None:  
    print("PEP 8 Style Guide prefers this pattern")
```

```
flag = True
```

```
# Not PEP 8's preferred pattern  
if flag == True:  
    print("This works, but is not the preferred PEP 8 pattern")
```

```
flag = True
```

```
if flag:  
    print("PEP 8 Style Guide prefers this pattern")
```

```
int_variable = "Hello, World!"  
int_variable / 2
```

```
variable = "Hello, World!"  
variable / 2
```

```
>>> import random  
>>> random.randrange(0, 7)  
1
```

```
>>> import random  
>>> random.seed(5)  
>>> random.random()  
0.6229016948897019  
>>> random.random()  
0.7417869892607294  
>>> random.random()  
0.7951935655656966  
>>> random.seed(5)  
>>> random.random()  
0.6229016948897019
```

```
>>> import secrets  
>>> secure_generator = secrets.SystemRandom()  
>>> secure_generator.randrange(0, 7)  
2
```

The [secrets](#) module is used for generating cryptographically strong random numbers suitable for managing data such as passwords, account authentication, security tokens, and related secrets.

In particular, [secrets](#) should be used in preference to the default pseudo-random number generator in the [random](#) module, which is designed for modelling and simulation, not security or cryptography.

Virtuelle miljøer

```
$ conda create --name envtest python  
$ conda activate envtest
```

```
$ python -m venv envtest  
$ source envtest/bin/activate
```

Environments

```
$ python -m venv envtest  
$ envtest\Scripts\activate.bat
```

```
$ python -m venv envtest  
$ envtest\Scripts\Activate.ps1
```

Versjonshåndtering

Intro

Hvorfor?

- Samarbeid
- Versjonshåndtering
- Gjenopprettning
- Dokumentasjon
- «Sikkerhetskopi»

Prosjekt = En samling med filer og mapper

Filer og mapper

```
n = int(input('Type a number, and its factorial will be printed: '))

if n < 0:
    raise non-negat
    n = int(input('Type a number, and its factorial will be printed: '))

factorial
for i in :
    raise non-negat
    n = int(input('Type a number, and its factorial will be printed: '))

    factorial
        if n < 0:
            raise non-negat
            n = int(input('Type a number, and its factorial will be printed: '))

            factorial
                if n < 0:
                    raise non-negat
                    n = int(input('Type a number, and its factorial will be printed: '))

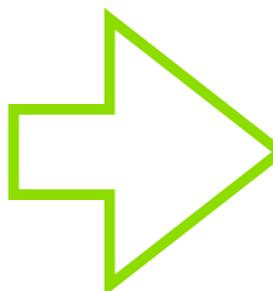
                    factorial
                        if n < 0:
                            raise non-negat
                            n = int(input('Type a number, and its factorial will be printed: '))

                            factorial
                                if n < 0:
                                    raise non-negat
                                    n = int(input('Type a number, and its factorial will be printed: '))

                                    factorial
                                        if n < 0:
                                            raise ValueError('You must enter a non-negative integer')

                                            factorial = 1
                                            for i in range(2, n + 1):
                                                factorial *= i

                                            print(factorial)
```



```
n = int(input('Type a number, and its factorial will be printed: '))

if n < 0:
    raise non-negat
    n = int(input('Type a number, and its factorial will be printed: '))

factorial
for i in :
    raise non-negat
    n = int(input('Type a number, and its factorial will be printed: '))

    factorial
        if n < 0:
            raise non-negat
            n = int(input('Type a number, and its factorial will be printed: '))

            factorial
                if n < 0:
                    raise non-negat
                    n = int(input('Type a number, and its factorial will be printed: '))

                    factorial
                        if n < 0:
                            raise non-negat
                            n = int(input('Type a number, and its factorial will be printed: '))

                            factorial
                                if n < 0:
                                    raise ValueError('You must enter a non-negative integer')

                                    factorial = 1
                                    for i in range(2, n + 1):
                                        factorial *= i

                                    print(factorial)
```

Repository = En samling med filer og mapper, som blir håndtert av versjonskontroll

```
n = int(input("Type a number, and its factorial will be printed: "))

if n < 0:
    raise non-negative

factorial = 1
for i in range(1, n + 1):
    factorial *= i
print(factorial)

commit 160d9cc0ed96c794542ed935dd8dc6668a0d5 (HEAD -> main, origin/main, origin/HEAD)
Author: rtrouton <rtrouton@yahoo.com>
Date: Tue May 16 11:28:01 2023 -0400
    Update README.md

commit 65555ec5ca0e70c05d04116c568a7f7705e61
Author: rtrouton <rtrouton@yahoo.com>
Date: Thu Nov 24 09:21:51 2022 -0500
    Update README.md
        Updated OS compatibility information.

commit 8d1c34cf40f126930461791d472bd92e259e4db
Author: rtrouton <rtrouton@yahoo.com>
Date: Fri Dec 17 09:28:52 2021 -0500
    Update README.md

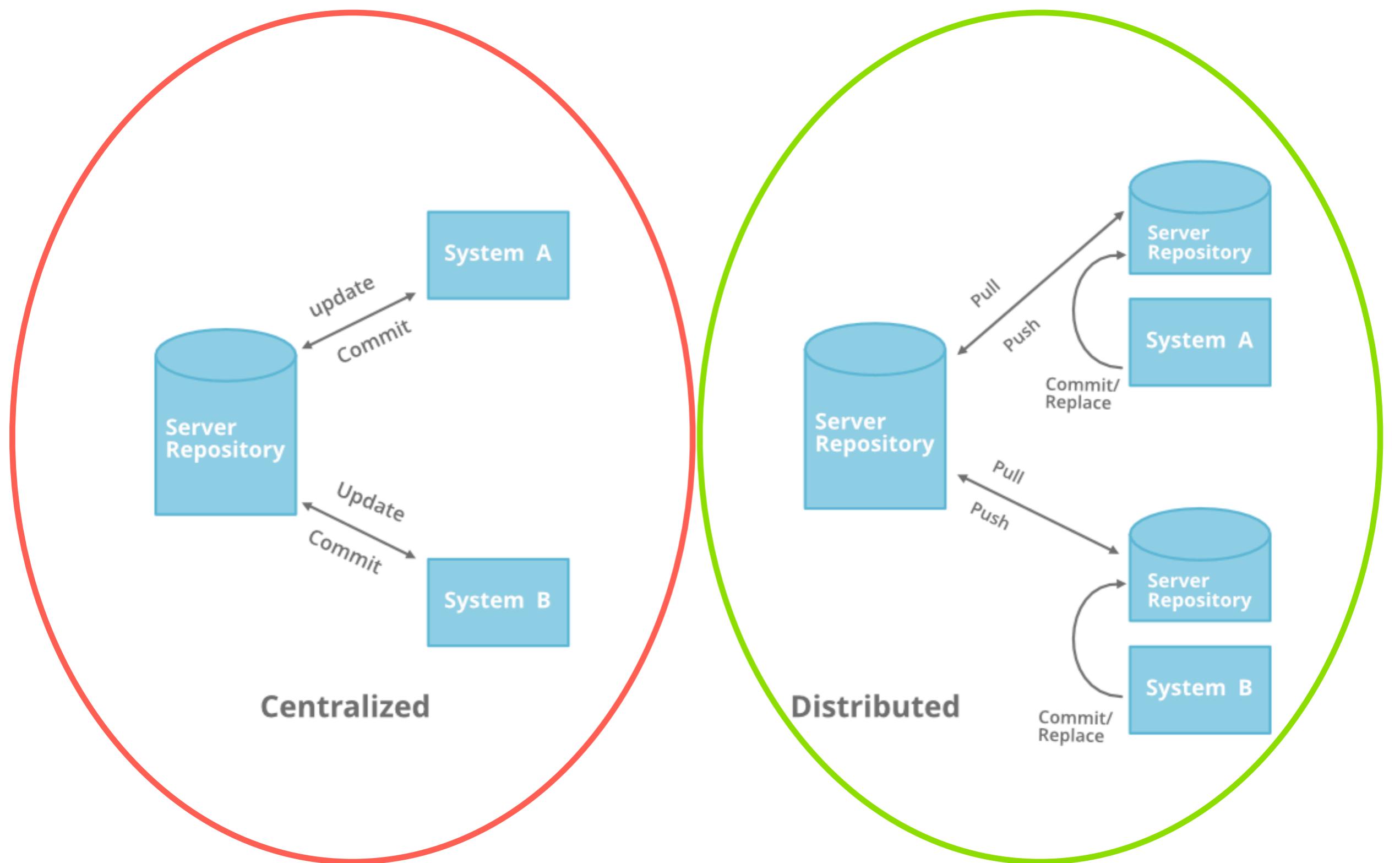
commit e5744def0a7d96173e4af188d9e6465f5cd8e3880
Author: rtrouton <rtrouton@yahoo.com>
Date: Fri Dec 17 09:25:56 2021 -0500
    Update README.md

commit 0ac915e339b01a400b0276743ed49091a099e
Author: rtrouton <rtrouton@yahoo.com>
Date: Tue Sep 28 10:50:30 2021 -0400
    Update README.md

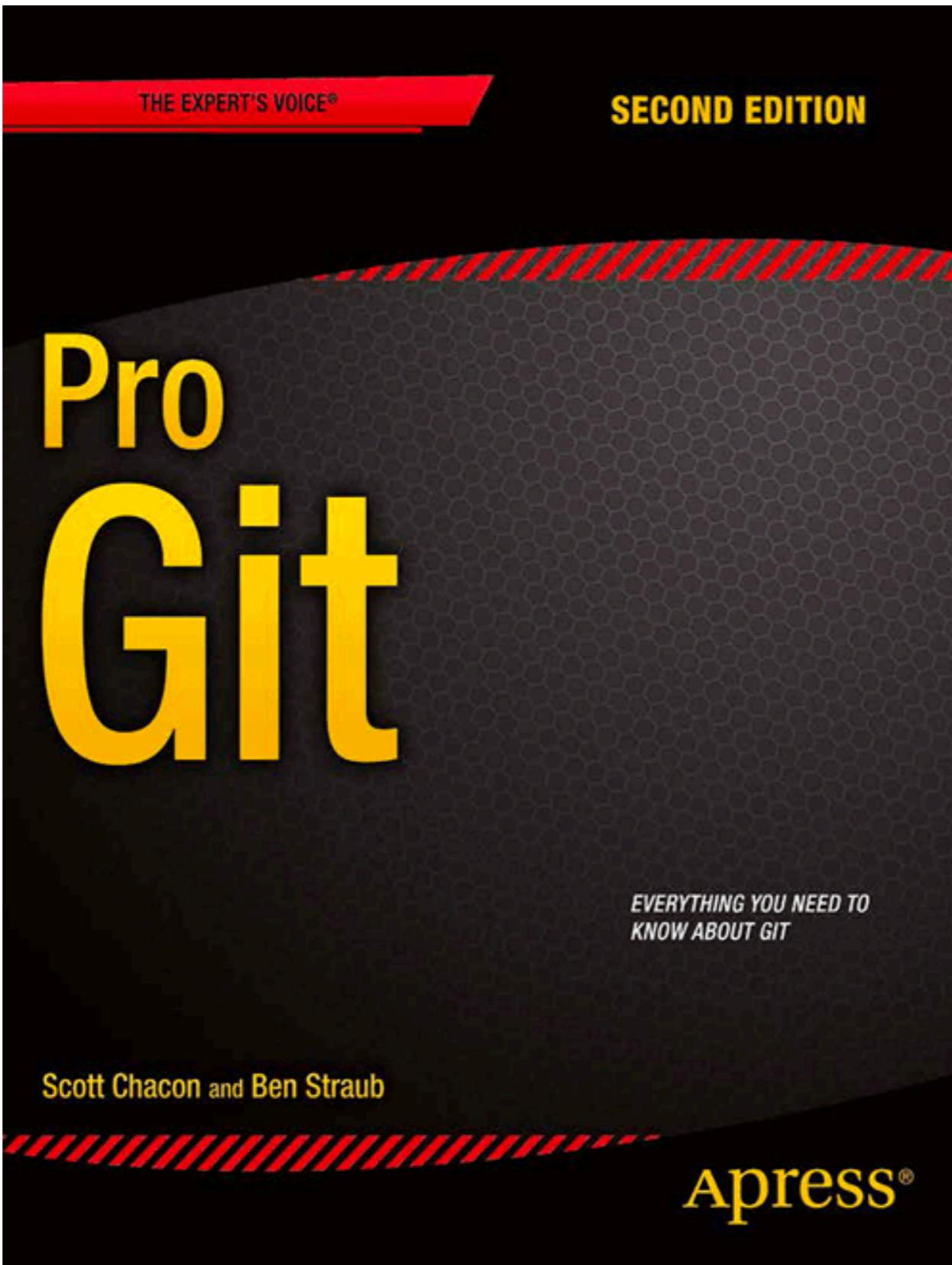
commit e5243d156522ad793d47ecd4bcf8aa345df8d25
Author: rtrouton <rtrouton@yahoo.com>
Date: Wed Jan 13 17:08:37 2021 -0500
    Update README.md
```

▼ TIMELINE create_macos_vm_install_dmg.sh

- Update create_macos_vm_install_dmg.sh Rich Trouton 2 yrs
- Update create_macos_vm_install_dmg.sh ... Rich Trouton 3 yrs
- Update create_macos_vm_install_dmg.sh ... Rich Trouton 4 yrs
- Uploading updated create_macos_vm_install_dmg.sh script ... Rich Trouton 6 yrs
- Uploading create_macos_vm_install_dmg script and README ... rtrouton 6 yrs



<https://www.geeksforgeeks.org/centralized-vs-distributed-version-control-which-one-should-we-choose/>



<https://git-scm.com/book/en/v2>

Versjonshåndtering i Kaggle, Google Colab og Github