Who am I

NITO

#Digitalisering

Koding

1185 2895

https://www.menti.com/altqoir9qdgf

https://github.com/kbotnen/pythonkurs_v25

Her viser vi sessionlabprogrammet

En rolig start

Introduksjon til programmering - Del 2

KRISTIAN BOTNEN, 2025

Øktens agenda

Noen flere datatyper

Funksjoner og moduler

Dokumentasjon og hjelp

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")
```

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

Python Assignment Operators

Operator	Example	Same As
=	`x = 2`	`x = 2`
+=	`x += 2`	`x = x + 2`
-=	`x -= 2`	`x = x - 2`
*=	`x *= 2`	`x = x * 2`
/=	`x /= 2`	`x = x / 2`
%=	`x %= 2`	`x = x % 2`
//=	`x //= 2`	`x = x // 2`
**=	`x **= 2`	`x = x ** 2`

Python Comparison Operators

Equal	`x == y`
Not equal	`x != y`
Greater than	`x > y`
Less than	`x < y`
Greater than or equal to	`x >= y`
Less than or equal to	`x <= y`

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")
```

Collections

Index	*Verdi*
0	«Henry»
1	«Ford»
2	«English»

DUPLICATES

CHANGEABLE

ORDERED

List

Nøkkel	*Verdi*
«f_Name»	«Henry»
«l_Name»	«Ford»
«language»	«English»

NO-DUPLICATES

CHANGEABLE

ORDERED

Dictionary

Index	*Verdi*
0	«Henry»
1	«Ford»
2	«English»

DUPLICATES

UNCHANGEABLE

ORDERED

Tuple

Verdi
«Henry»
«Ford»
«English»

NO-DUPLICATES

UNCHANGEABLE

UNORDERED

Set

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

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

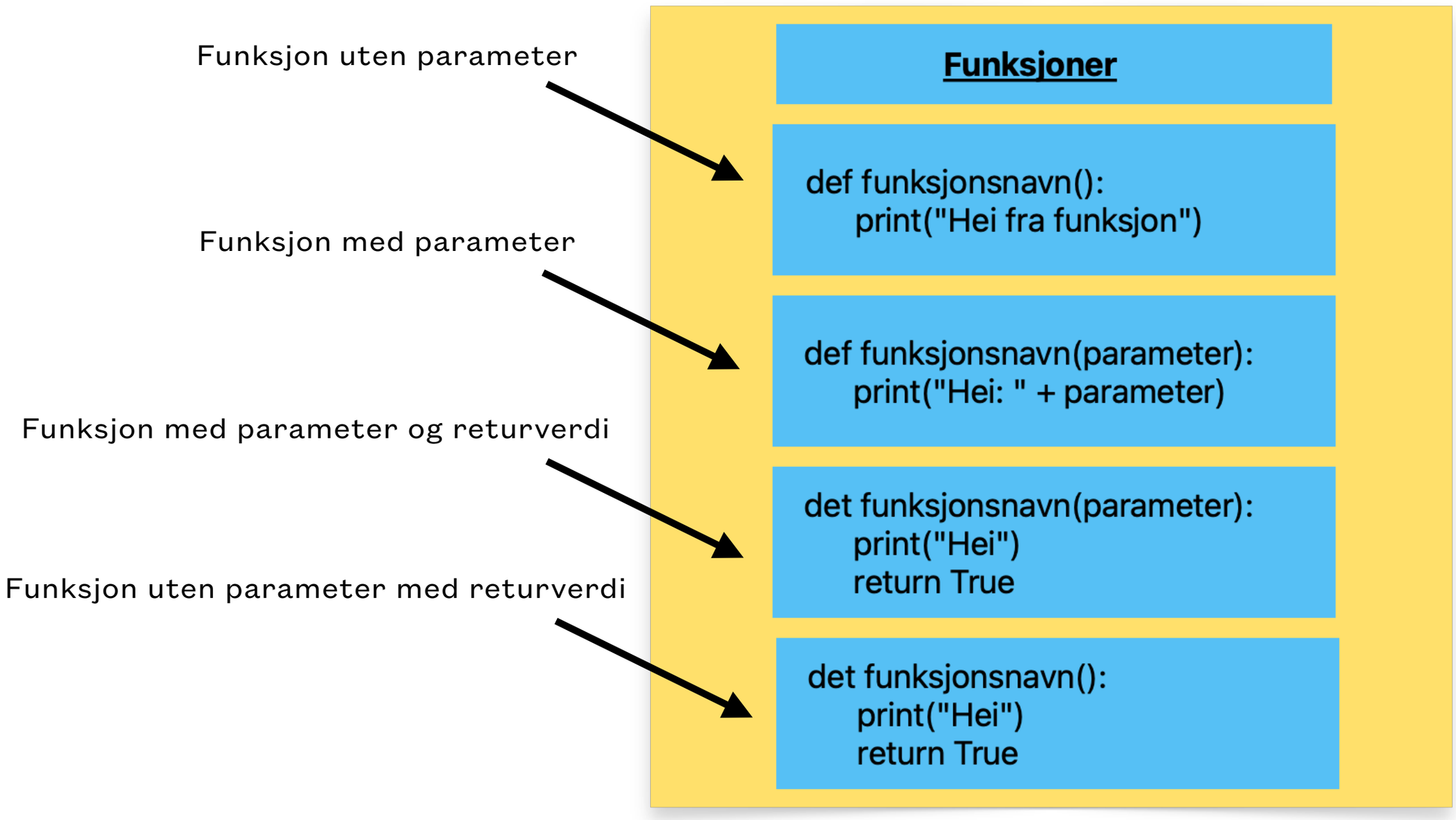
```python
def greet_function():
    print("Hello from a function")

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))
```


Funksjoner
Funksjon uten parameter
def funksjonsnavn():
print("Hei fra funksjon")
Funksjon med parameter
def funksjonsnavn(parameter):
print("Hei: " + parameter)
Funksjon med parameter og returverdi
det funksjonsnavn(parameter):
print("Hei")
return True
Funksjon uten parameter med returverdi
det funksjonsnavn():
print("Hei")
return True

Oppgavetid

```
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
Picke
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
"""
```

Dokumentasjon

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

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

```
# 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}")
```


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

help()

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

```
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".

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".

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

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

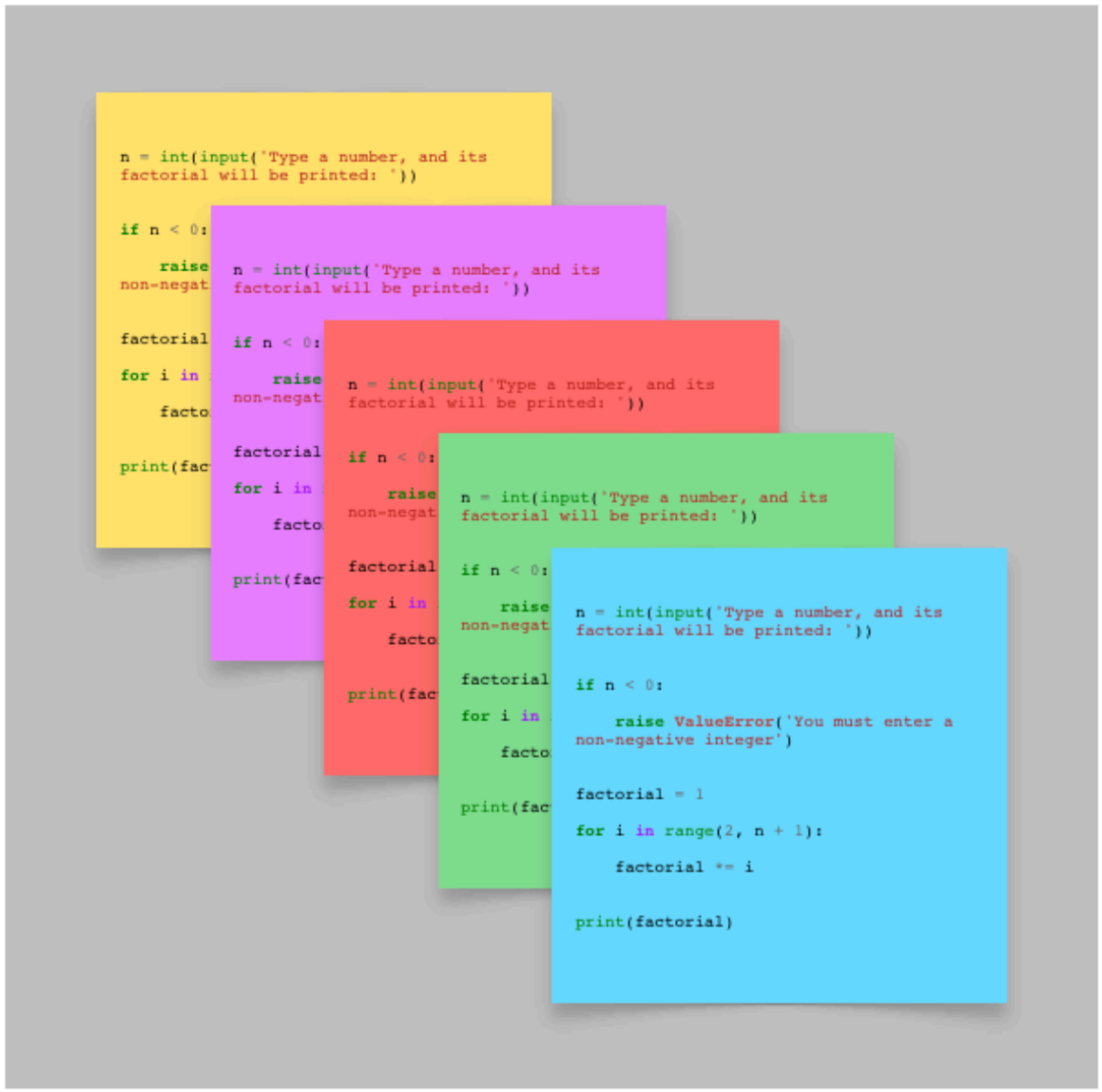
Intro

Hvorfor?

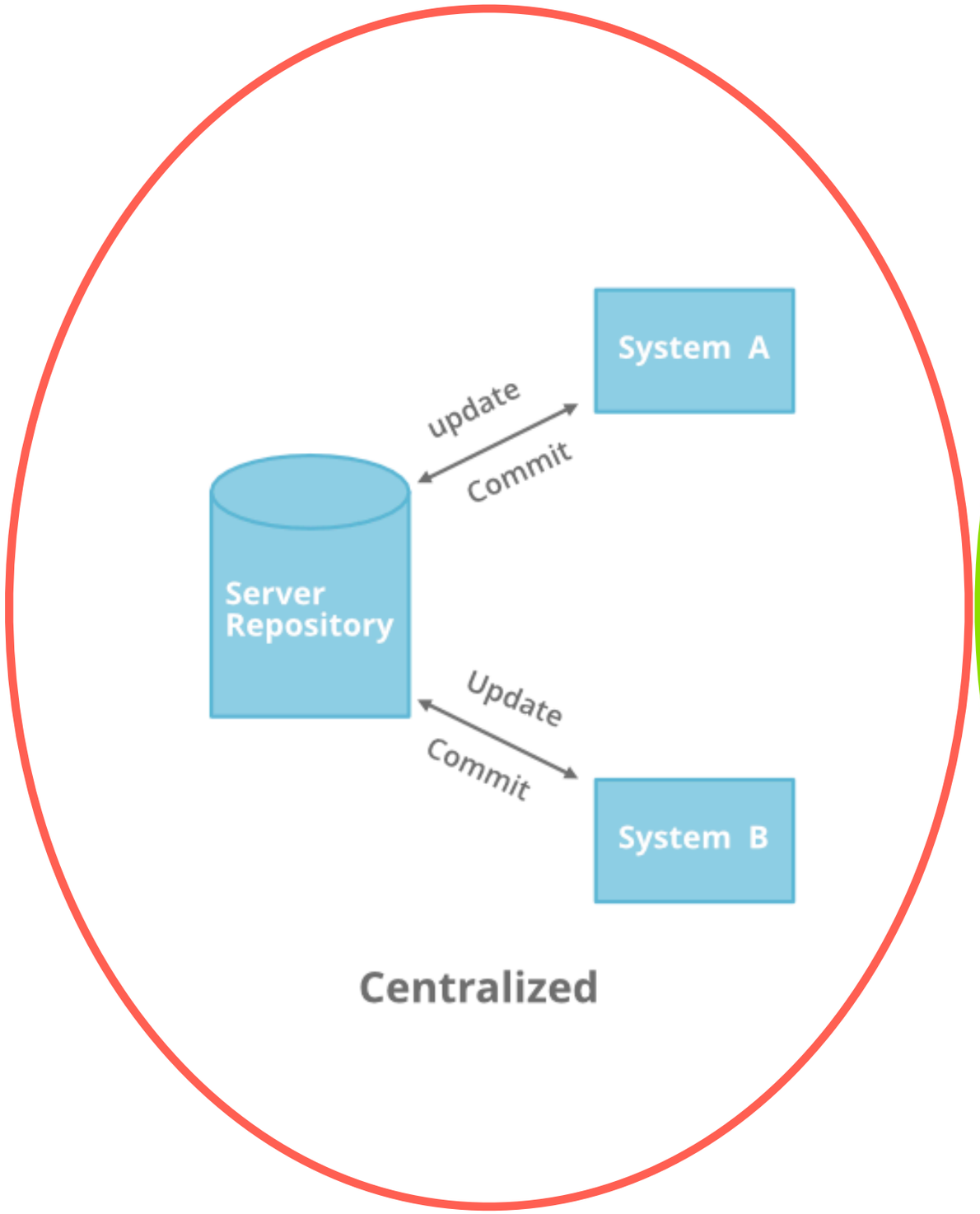
- Samarbeid
- Versjonshåndtering
- Gjenoppretting
- Dokumentasjon
- «Sikkerhetskopi»

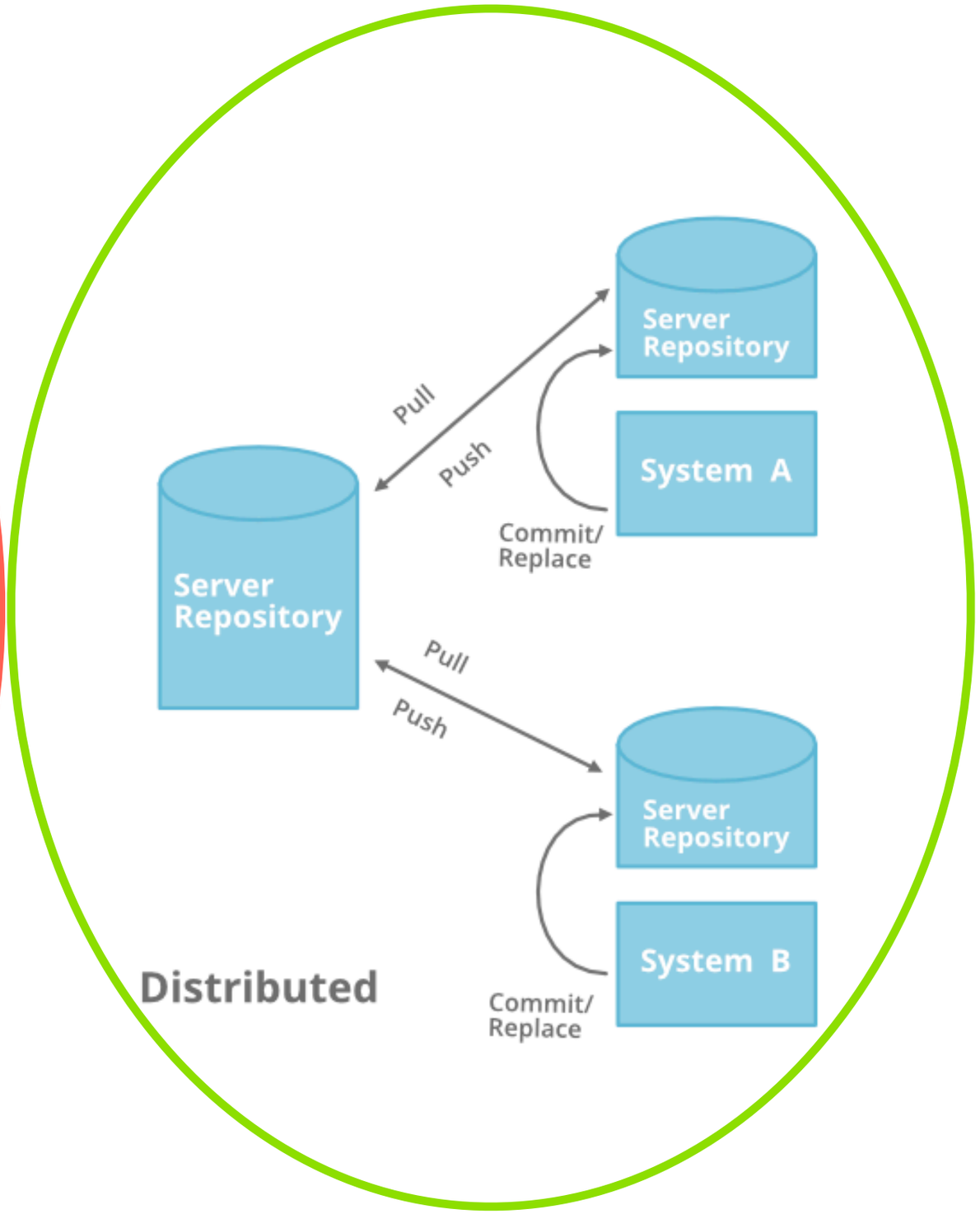
Filer og mapper

Prosjekt = En samling med filer og mapper

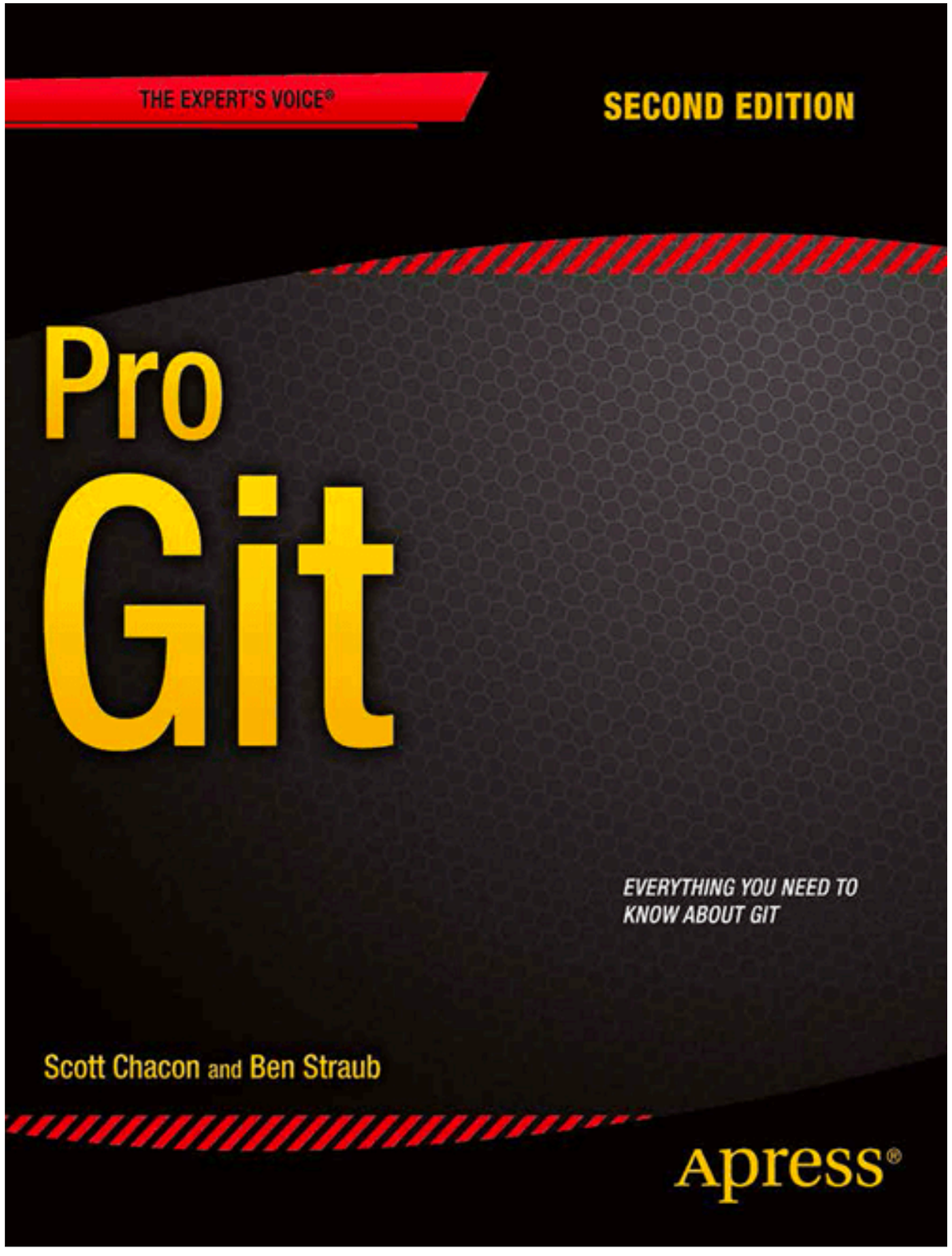


Repository = En samling med filer og mapper, som blir håndtert av versjonskontroll





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