



Tekna

# CRASH COURSE

INF100

Sondre Bolland

# Overview

## Exceptions

- try / catch, raise
- ValueError, KeyError, IndexError, FileNotFoundError
- bruk for input-validering

## Import

- bruk av **import** med flere egne filer
- hvorfor bruker vi if `__name__ == "__main__"`

## Basics

- Matematiske uttrykk og operasjoner: +, -, \*, /, //, %, \*\*, ()
- Datatype: **int, float, str**
- Variabler holder data, assignment med =
- Bruk av variabler i uttrykk
- **print()**
- **f-strings** for enkelt printing av variabler: `f"A={a} B={b}"`
- **input()**
- **len()**
- Konvertering av datatyper med **int(), float(), str()**

## Programflyt

- Datatype **bool**: **True / False**
- Operasjoner for sammenligning: **==, !=, <, >, <=, >=**
- Kombinasjon av bool-verdier: **and, or, not**
- Betingelser: **if, elif, else**
- Løkker: **while, for**
- Styring av løkker: **break, continue**
- **range()** funksjon

## Funksjoner

- Definisjon av funksjoner: **def, return**
- Funksjonsargumenter
- Forskjellen mellom bruk av print / return
- Hvorfor bruker vi funksjoner? enklere programstruktur, gjenbruk, black-box
- bruk av **None**
- **abs, len, max, min, round**

## Sekvenser

- Datatype: **list, tuple, str**
- element-utvalg: `a[3]`, `a[-2]`
- slices: `a[3:7]`, `a[4:12:3]`
- **len()**
- operasjoner: +, \*
- bruk av **for**-løkker med sekvenser: `for i in [..., ..., ...]`
- bool-operasjoner: `x in liste`, `x not in liste`
- **enumerate(), zip()**
- **list.append()**
- **list.sort()**

## Dictionaries

- datatype **dict**, { 'a' : 'b', 'c' : 'd', ... }
- `d[...]`
- `dict.keys()`, `dict.values()`, `dict.items()`
- **if key in dict**

## Filhåndtering

- **with open("filename") as f**
- `f.read()`, `f.readlines()`, `for line in f`
- `f.write()`

<https://mitt.uib.no/courses/24077/pages/eksamensinnhold>

# Basics

- Datatyper
- Matematiske uttrykk og operatorer
- Variabler
- `print()`
- `input()`
- `len()`
- Konvertering av datatyper `int()`, `float()`, `str()`

# Dat typer

- String:

```
string_variable1 = "kake"  
string_variable2 = 'kake'
```

Tekst/En liste av chars

- Int:

```
int_variable = 100
```

Heltall

- Float:

```
float_variable = 100.0
```

Desimaltall

# Matematiske operatører

- + pluss
- - minus
- \* gange/multiplikasjon
- / vanlig deling
- // deling med nedrunding
- \*\* eksponent ( $5^{**}2 = 5^2 = 25$ )
- % modulo (resten etter deling)

# % - Modulo

$$5 / 2 = 2$$

-4

---

1

# Variabler

**print(), input() & len()**



# print(), input() & len()

- print() – skriver ut teksts
- input() – leser input fra en bruker
- len() – lengden av en liste

# Konvertere datatyper

- `float()`

- `int()`

- `str()`

# Programflyt

- Datatype **bool**: True / False
- Operasjoner for sammenligning: ==, !=, <, >, <=, >=
- Kombinasjonen av bool-verdier: **and**, **or**, **not**
- Betingelse: **if**, **elif**, **else**
- Løkker: **for**, **while**
- Styring av løkker: **break**, **continue**
- **range()** funksjonen

# Bool/Boolean

- True / False
- $x == y$  : er x og y lik?
- $x != y$  : er x og y ulik?
- $x > y$  : er x større enn y?
- $x < y$  : er x mindre enn y?
- $x >= y$  : er x større eller lik y?
- $x <= y$  : er x mindre eller lik y?

```
my_bool = True
```

# Boolean algebra (kombinasjon av bools)

## ■ and, or, not

```
bool1 and bool2
```

er True hvis både **bool1** OG **bool2** er True

```
bool1 or bool2
```

er True hvis **bool1** ELLER **bool2** er True

```
not bool1
```

er True hvis **bool1** er False

# Betingelser

- if
- elif        (else if)
- else

# Løkker

- **while**      loop'er basert på et boolsk uttrykk
- **for**          loop'er over en liste

# Styring av løkker

- **break**                      Bryt ut av loop'en
- **continue**                start neste iterasjon av loop'en



# range()

- Lager en liste av tall i et intervall

```
range(10) = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

- Tar inn 1 til 3 argumenter

- *Start*
- *Stop*
- *Step*

```
range(start, stop, step)
```

# Sekvenser

- Datatypene: `list`, `tuple`, `str`
- Elementutvalg: `a[3]`, `a[-2]`
- Slices: `a[3:7]`, `a[4:12:3]`
- Operasjoner: `+`, `*`
- Bruk av for-løkker i sekvenser
- Bool-operasjoner: `x in list`, `x not in list`
- `enumerate()`, `zip()`
- `list.append()`
- `list.sort()`

# tuple, list, str

- tuple: (1, 2, 3) immutable => hashable
- list: [1, 2, 3] mutable => not hashable
- str: "123" immutable => hashable

# Elementutvalg

0 1 2 3 4  
↓ ↓ ↓ ↓ ↓

- `my_list = [5, 6, 1, 0, 100]`
- `my_list[0] = 5`
- `my_list[4] = 100`
- `my_list[-3] = 1`

# Slices

- Delete opp listen din
- `my_list = [5, 6, 1, 0 , 100]`
- `my_list[1:3] = [6, 1]`
- `my_list[1:4:2] = [6, 0]`

hente ut elementene fra  
index 1 til index 3-1

hente ut elementene fra  
index 1 til index 4-1 hvor vi  
hopper over hver andre  
element

# Sekvens-operasjoner

- + concatenate
- \* repeat

for-løkker i sekvenser

# enumerate()

- Hvis du ønsker å iterere over en liste og samtidig ha indeksen til elementene i listen



# zip()

- Lage tupler av elementer fra flere lister

# `list.append()` og `list.sort()`

- `list.append()`      Legg til element i listen
- `list.sort()`      Sorter elementene i listen

# To-dimensjonelle lister

# Dictionaries

- En liste med indekser som ikke er heltall

key	value
"one"	5
"two"	17
"kake"	4
"inf100"	-40
«lælll"	10

if **key** in **dict**:

# Funksjoner

- Definisjonen av funksjoner: **def**, **return**
- Funksjonsargumenter
- Forskjellen mellom bruk av **print** og **return**
- Hvorfor bruke funksjoner?
- Bruk av **None**
- **abs**, **len**, **max**, **min**, **round**

# Definisjon av funksjon

# None

- Et objekt for å representere «ingenting»



# Vanlige funksjoner

- `abs()` Finner absolutt verdien av et tall
- `len()` Finner lengden av en sekvens
- `max()` Finner det største elementet i en liste
- `min()` Finner det minste elementet i en liste
- `round()` Runder av tall

# Filhåndtering

- **with open(«filename») as f**
- **f.read(), f.readlines(), for line in f:**
- **f.write()**

# file.open(<filename>)

- `file.open(«example.txt»)`

- `file.open(«example.txt», «r»)`

Åpner fil for lesing (*read*)

- `file.open(«example.txt», «w»)`

Åpner fil for å skrive ny tekst (*write*)

- `file.open(«example.txt», «a»)`

Åpner fil for legge til tekst (*append*)

# read(), readline(), readlines()

- `read()` Leser inn hele filen
- `readline()` Leser inn en linje av filen
- `readlines()` Leser inn alle linjene i filen

# write()

- `file.open(«example.txt», «w»)`

Åpner fil for å erstatte tekst (*write*)

- `file.open(«example.txt», «a»)`

Åpner fil for legge til tekst (*append*)

# Exceptions

- try, catch, raise
- ValueError, KeyError, IndexError, FileNotFoundError
- Input-validering

Try, catch

# Import

- Jeg ønsker å bruke kode som andre har skrevet
- Bibliotek/moduler inneholder kode som man kan importere



# Tidligere eksamen

- Vår 2020:

[https://mitt.uib.no/files/2785197/download?download\\_frd=1](https://mitt.uib.no/files/2785197/download?download_frd=1)