



Lær Python dag 1 - modul 1

Introduktion, basis python

Jonas Bamse Andersen

Institut for Matematik og Datalogi - IMADA
Syddansk Universitet

Hvem er jeg?



Jonas Bamse Andersen

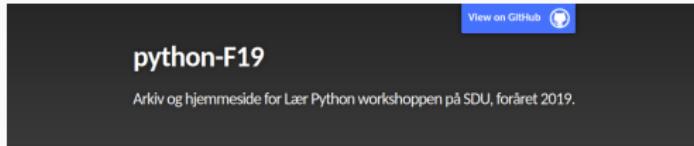
Datalogi ~ 4 år

Fritid: Rubiks terninger og Youtube

Hjemmeside

Kurset har en hjemmeside:

<https://jonan15.github.io/python-F19/>



The screenshot shows a GitHub page for the repository "python-F19". At the top right, there is a "View on GitHub" button with a GitHub icon. Below the button, the repository name "python-F19" is displayed in white text on a dark background. A subtitle below the name reads "Arkiv og hjemmeside for Lær Python workshoppen på SDU, foråret 2019." The main content area has a light gray background and contains several sections: "Lær Python", "Praktisk information", "Tid og Sted", and "Transport". Each section provides specific details about the course.

Lær Python

Velkommen til Lær Python workshoppen på SDU, foråret 2019.

Her vil du finde slides, links til opgaver, m.m. som løbende vil blive lagt op.

Praktisk information

Tid og Sted

Kurset finder sted de følgende søndage kl 11-16:

- maj 12
- maj 19
- maj 26

Det foregår på Syddansk Universitet, Campusvej 55, bygning 44, lokale U167.

Find nemt rundt på SDU ved at bruge [SDU Maps](#).

Transport

Her findes slides, opgaver og eventuelle tips & tricks.

Colaboratory

Online editor og interpreter (fortolker).

Kræver en Google Account.

Copy of Dag 1 Modul 1 - First Steps.ipynb

File Edit View Insert Runtime Tools Help

CODE TEXT CELL CELL

Table of contents Code snippets Files

General Strategi

De første skridt

Strange

Betinget udtryelse

Løkker

SECTION

Generel Strategi

Målet med disse opgaver er at gøre dit fortrolig med variabler og udtryk i Python. Den vigtigste egenhård du kan have er at være nysgerrig. Hver gang et program er givet kan du altså lave om i det, og se, hvad der så skeer. Prøv også at skrive forkerte ting, og se, hvilken fejmeddelelse du får.

De første skridt

Denne opgave er din første skridt i Python. I hver opgave er et lille stykke kode givet. Prøv at regne ud, hvad værdien output af programmet vil være. Kør det herefter ved at klikke i boksen og så på play' knappen i venstre side.

```
[ ] x=3  
y=4  
x+y  
print(x)
```

6

```
[ ] x=5  
xxx+1  
xxx*x  
print(x)
```

36

```
[ ] x=3  
y=4  
z=x*y  
print(z*y)
```

36

```
[ ] value=2  
result=3>value  
print(result)
```

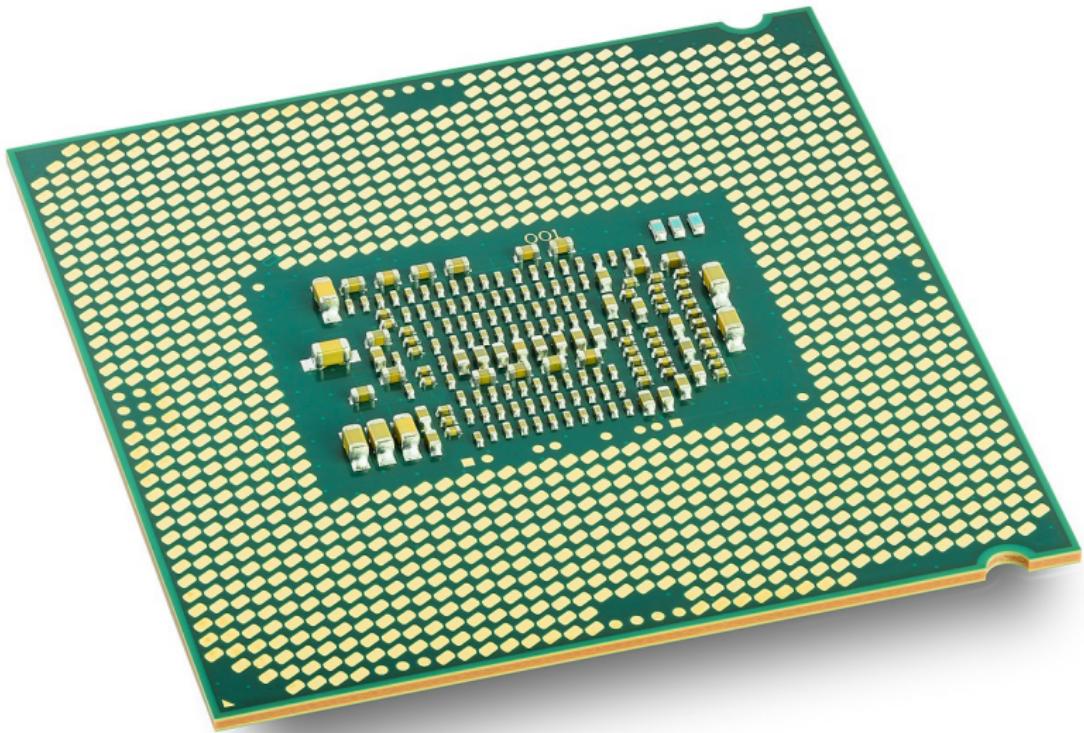
.....

```
[ ]  
NameError  
<ipython-input-3-6ff833f80043> in <module>()  
    1 value=2  
    2 result=3>value
```

Traceback (most recent call last)

Man kan også installere Python på sin egen computer.
IMADA - SDU

Hvad er Python?



Hvad er Python?

Hvad er en computer?

Hvad er Python?

Hvad er en computer?

Hvad er et program?

Hvad er Python?

Hvad er en computer?

Hvad er et program?

Program = sekvens af
instruktioner.

Hvad er Python?

Hvad er en computer?

Hvad er et program?

Program = sekvens af
instruktioner.

Computere er dumme! De
gør kun som de får besked
på...

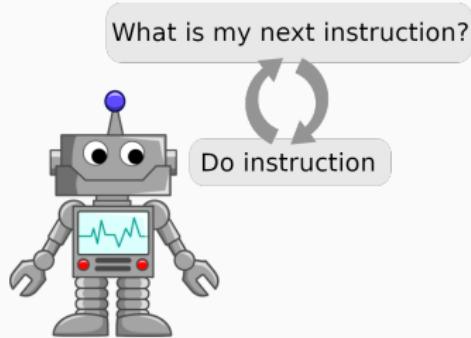
Hvad er Python?

Hvad er en computer?

Hvad er et program?

Program = sekvens af instruktioner.

Computere er dumme! De gør kun som de får besked på...



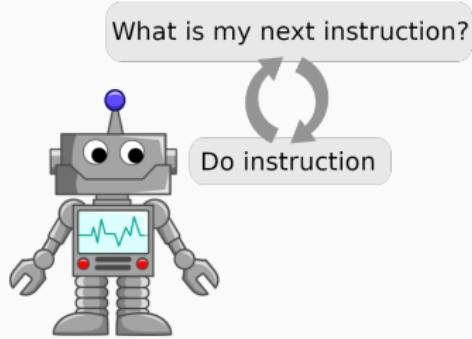
Hvad er Python?

Hvad er en computer?

Hvad er et program?

Program = sekvens af instruktioner.

Computere er dumme! De gør kun som de får besked på...



Tilgengæld er de lynende hurtige.

Hvad er Python?

Python er et programmeringssprog.

Sproget bestemmer hvilke instruktioner man kan skrive.

Men sproget skal oversættes/fortolkes.

Hvad er Python?

Python er et programmeringssprog.

Sproget bestemmer hvilke instruktioner man kan skrive.

Men sproget skal oversættes/fortolkes.

1 linje \approx 1 instruktion.

Programmet oversættes fra toppen og nedad.

Programmer

Når vi snakker om programmer kan vi bl.a. snakke om:

- Instruktioner: Hvad selve programmet består af.

Når vi snakker om programmer kan vi bl.a. snakke om:

- Instruktioner: Hvad selve programmet består af.
- Input: "Data" som programmet skal forholde sig til.

Når vi snakker om programmer kan vi bl.a. snakke om:

- Instruktioner: Hvad selve programmet består af.
- Input: "Data" som programmet skal forholde sig til.
- Output: Resultatet af programmet, f.eks. tekst/grafik på en skærm eller en ny fil.

Fordeler/Ulempel ved Python

Fordele

- Simpelt sprog - **nemt at lære**
- Stort standardbibliotek (allerede implementerede funktioner)
- Kan køre på mange platforme
- Udbredt brug i både industri, forskning og hobby projekter

Ulempel

- Langsommere end sprog som C/C++
- Typer er uskrevne (dynamisk)

Disclaimer

Programmering er en færdighed der skal øves, det tager tid.

Forvent

- ikke at blive programmør i løbet af kurset.
- at blive forvirret undervejs, det er helt naturligt.

Målet er at give færdigheder til at kunne gøre lidt og til at kunne lære mere.

Programmering i Python

Hello, World!

Det er tid til at se det første program køre.

Typer, variabler og udtryk

Datatyper - et programs enheder

Følgende er de mest brugte primitive typer:

Datatyper	Eksempel
String (streng)	"Hej"
Integer (heltal)	42
Float (kommatal)	42.0
Boolean	True

Derudover er der de mere avancerede typer som: list, tuple og dictionary

Mere om dem senere...

Datatyper - et programs enheder

En type kan tjekkes i python via:

```
type(<type_der_skal_tjekkes>)
```

Output:

Program:

```
type(42.0)
```

float

eller

<class 'float'>

Konvertering mellem typer (casting)

Nogle typer kan konverteres/tvinges til at blive andre typer.

Konvertering til kommatal:

`float(4)`

4.0

Konvertering til heltal:

`int(4.3)`

4

Konvertering til streng:

`str(4 + 3)`

"7"

Operatorer

Følgende beskriver de basale operatorer anvendt på tal:

Operation	Beskrivelse	Eksempel	Resultat
+	Læg to operander sammen	$2 + 2$	4
-	Træk to operander fra hinanden	$50 - 8$	42
*	Gang to operander	$3 * 4$	12
/	Division mellem to operander	$10 / 3$	3.333...
//	Heltalsdivision mellem to operander	$10 // 3$	3
**	Eksponentiering	$2 * * 3$	8

Operatorer

Følgende beskriver de basale operatorer anvendt på tal:

Operation	Beskrivelse	Eksempel	Resultat
+	Læg to operander sammen	$2 + 2$	4
-	Træk to operander fra hinanden	$50 - 8$	42
*	Gang to operander	$3 * 4$	12
/	Division mellem to operander	$10 / 3$	3.333...
//	Heltalsdivision mellem to operander	$10 // 3$	3
**	Eksponentiering	$2 * * 3$	8

Disse operatorer kan måske anvendes på andet?...

Streng-operatorer

Addition?:

```
print("hej" + " Per")
```

Streng-operatorer

Addition?:

```
print("hej" + " Per")
```

```
hej Per
```

Streng-operatorer

Addition?:

```
print("hej" + " Per")
```

```
hej Per
```

Subtraktion?:

```
print("hej" - "ej")
```

Streng-operatorer

Addition?:

```
print("hej" + " Per")
```

```
hej Per
```

Subtraktion?:

```
print("hej" - "ej")
```

```
Traceback (most recent call last):
File "<stdin>", line 1,
  in <module>
TypeError: unsupported operand type(s) for -: 'str' and 'str'
```

Streng-operatorer

Multiplikation?:

```
print("hej" * 3)
```

Streng-operatorer

Multiplikation?:

```
print("hej" * 3)
```

```
hejhejhej
```

Streng-operatorer

Multiplikation?:

```
print("hej" * 3)
```

```
hejhejhej
```

Division?:

```
print("hej" / 3)
```

Streng-operatorer

Multiplikation?:

```
print("hej" * 3)
```

```
hejhejhej
```

Division?:

```
print("hej" / 3)
```

```
Traceback (most recent call last):
File "<stdin>", line 1,
  in <module>
TypeError: unsupported operand type(s) for /: 'str' and 'int'
```

Variabler

En variabel er en "beholder" som kan gemme en værdi. I hukommelsen bliver der reserveret plads til den givne variabel.

En variabel har et navn:

Tilladte variabelnavne

x
et_navn
TEST
var2
_hej

Forbudte variabelnavne

1var
en.var
-var

Navnet bruges til at referere til værdien senere.

Giv meningsfyldte variabelnavne!

Variabler

Ud over fornævnte forbudte variabelnavne er der nogle reserverede nøgleord, som heller ikke må bruges.

Disse er:

and, exec, not, assert, finally, or, break, for, pass,
class, from, print, continue, global, raise, def, if,
return, del, import, try, elif, in, while, else, is, with,
except, lambda, yield

Hvad de forskellige nøgleord bruges til vil I løbende komme til at
forstå...

Variabler

En tildeling gemmer noget i den givne beholder/variabel.

Tildeling til en variabel sker på følgende vis:

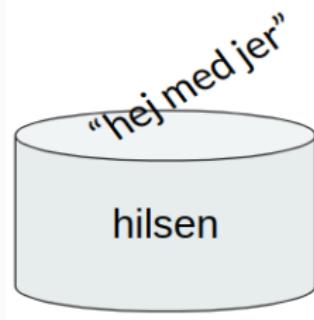
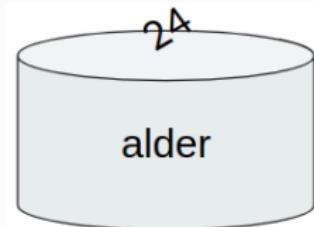
```
<navn> = <værdi>
```

Tildeling af et heltal:

```
alder = 24
```

Tildeling af en streng:

```
hilsen = "hej med jer"
```



Variabler

Man refererer til værdien gemt i en variabel ved at skrive variablenes navn. Navnet bliver erstattet af værdien den refererer til.

```
hilsen = "hej med jer"  
print( hilsen )
```

```
hej med jer
```

Variabler

Man refererer til værdien gemt i en variabel ved at skrive variablenes navn. Navnet bliver erstattet af værdien den refererer til.

```
hilsen = "hej med jer"  
print( hilsen )
```

```
hej med jer
```

Bemærk forskellen på at referere og på tekst-strenge

```
hilsen = "hej med jer"  
print( hilsen )  
print(" hilsen ")
```

```
hej med jer  
hilsen
```

Udtryk

Et udtryk er en kombination af værdier, variabler og operatorer

Eksempler på udtryk:

5

$x = 3$

$x + 5$

$x = 2$

$y = 3$

$(1 + x) * 5 - y$

Regnereglerne fra tal overholdes.

Print

Udskriv udtryk og variablers værdier med `print`-funktionen:

Program:

```
print("hej")
print(42)
x = 23
print(x)
```

Output:

```
hej
42
23
```

Print kan bruges til output fra et program, men også til fejlfinding af ens program (debugging).

Input fra brugeren

Input fra brugeren tages på følgende vis:

```
<variabel> = input(<Beskrivende streng>)
```

Eksempel 1:

```
navn = input("Hvad er dit navn?")
```

Input fra brugeren

Input fra brugeren tages på følgende vis:

```
<variabel> = input(<Beskrivende streng>)
```

Eksempel 1:

```
navn = input("Hvad er dit navn?")
```

Eksempel 2:

```
alder = input("Alder?")
print("Din alder om to år
      er")
print( alder+2)
```

Input fra brugeren

Input fra brugeren tages på følgende vis:

```
<variabel> = input(<Beskrivende streng>)
```

Eksempel 1:

```
navn = input("Hvad er dit navn?")
```

Eksempel 2:

```
alder = input("Alder?")
print("Din alder om to år
      er")
print(alder+2)
```

...

TypeError: must be str,
not int

Input fra brugeren

Vi bliver nødt til at lave teksten om:

```
alder = input("Alder?")  
print("Din alder om to år er"  
     )  
print(int(alder)+2)
```

Input fra brugeren

Vi bliver nødt til at lave teksten om:

```
alder = input("Alder?")  
print("Din alder om to år er"  
     )  
print(int(alder)+2)
```

```
Alder?24  
26
```

Input fra brugeren

Vi bliver nødt til at lave teksten om:

```
alder = input("Alder?")  
print("Din alder om to år er"  
      )  
print(int(alder)+2)
```

```
Alder?24  
26
```

Eller:

```
alder = int(input("Alder?"))  
print("Din alder om to år er")  
print(alder+2)
```

Input fra brugeren

Vi bliver nødt til at lave teksten om:

```
alder = input("Alder?")  
print("Din alder om to år er"  
      )  
print(int(alder)+2)
```

```
Alder?24  
26
```

Eller:

```
alder = int(input("Alder?"))  
print("Din alder om to år er")  
print(alder+2)
```

```
Alder?24  
26
```

Kommentarer

Kommentarer i koden er tekst som ignoreres når koden eksekveres.
Disse er kun til ære for den der læser koden.

```
print("hej") # en kommentar som beskriver denne  
instruktion  
  
# en fritstående kommentar som beskriver den følgende  
kode  
print("whatup")
```

Gode kommentarer hjælper når man ser på sin kode lang tid efter man har skrevet den.

Moduler

Mange ting er allerede implementeret i python af dygtige programmører. Disse funktioner er tilgængelige via moduler (aka. biblioteker).

Fx kan vi benytte `math`-biblioteket til at lave klassiske matematiske operationer:

Program:

```
import math  
x = 25  
y = math.sqrt(x)  
print(y)
```

Output:

```
5.0
```

Se dokumentation for alle matematikfunktioner:

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

Recap

Vi har nu snuset til alle disse ting:

- Print

Recap

Vi har nu snuset til alle disse ting:

- Print
- Typer, hvilke?

Recap

Vi har nu snuset til alle disse ting:

- Print
- Typer, hvilke?
- Værdier og operatorer

Recap

Vi har nu snuset til alle disse ting:

- Print
- Typer, hvilke?
- Værdier og operatorer
- Variabler

Recap

Vi har nu snuset til alle disse ting:

- Print
- Typer, hvilke?
- Værdier og operatorer
- Variabler
- Input

Recap

Vi har nu snuset til alle disse ting:

- Print
- Typer, hvilke?
- Værdier og operatorer
- Variabler
- Input
- Kommentarer

Recap

Vi har nu snuset til alle disse ting:

- Print
- Typer, hvilke?
- Værdier og operatorer
- Variabler
- Input
- Kommentarer
- Moduler

Recap

Vi har nu snuset til alle disse ting:

- Print
- Typer, hvilke?
- Værdier og operatorer
- Variabler
- Input
- Kommentarer
- Moduler

Og nu skal I (endelig) til at lave øvelser!