COS
Computersystemer

Lektion #3

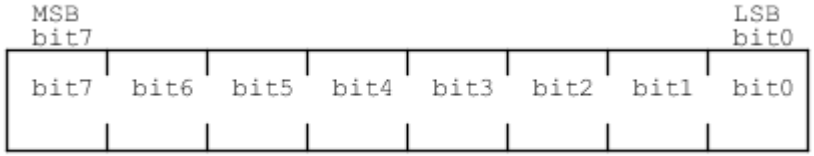
Programmering i Python.

Morten Hansen
moh@sdu.dk

Bits, bytes og main memory

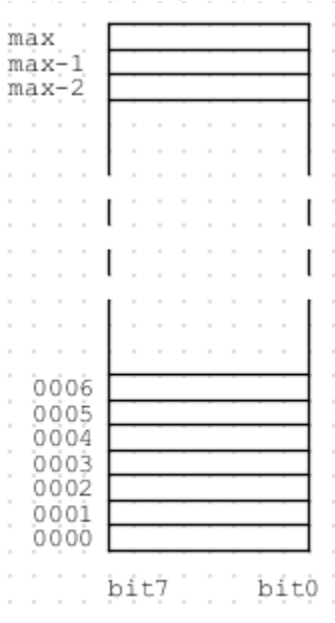
Bit:

- Mindst mulige mængde information.
- Kan antage 1 ud af 2 mulige værdier:
- 0 eller 1



Byte:

- Består af 8 bit.
- Kan antage 1 ud af 2^8 mulige værdier.
- =256 mulige, forskellige værdier, mønstre eller koder.



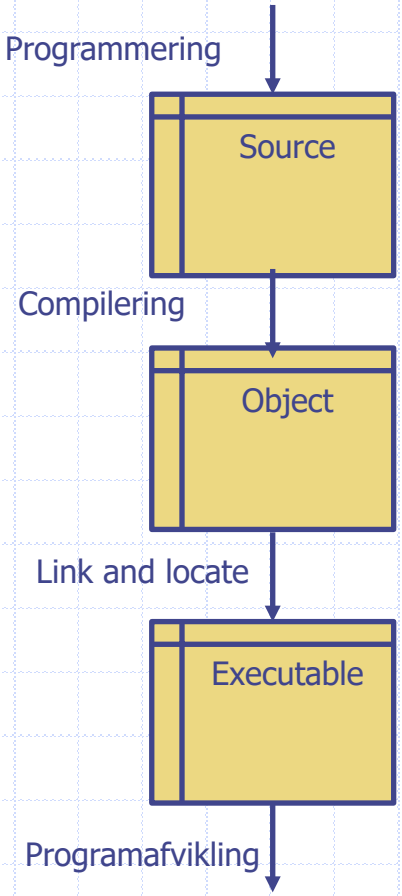
Main memory

- Består af en række celler
- Hver celle kan indeholde 8 bit information
- Hver celle har en adresse

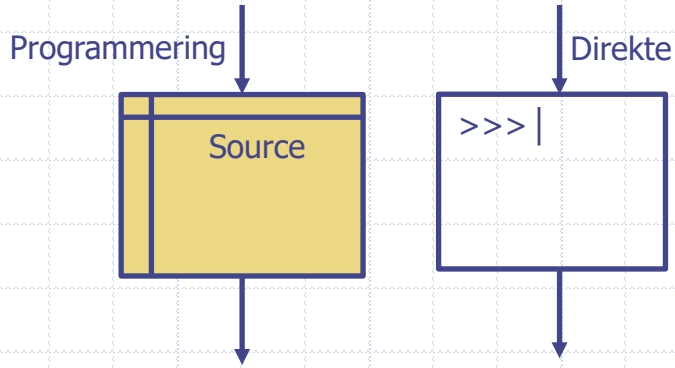
Morten Hansen
moh@sdu.dk

Afvikling af programmer på en computer.

Kompileret program



Fortolket program



Fortolkning:

- Linie for linie
  - Skab bytekode
  - Afvikling

Morten Hansen
moh@sdu.dk

Hvad er Python?

Morten Hansen
moh@sdu.dk

Python

- Python er et platformsuafhængigt programmeringssprog.
- Hvad bliver Python brugt til?
  - Data analyse
  - Maskine lærling
  - Kunstig inlinerens
  - Og meget mere
- Python er drivet meget af brugerne
  - Derfor findes der et omfattende community med dokumentation, biblioteker, tutorials, eksempler og meget andet
- Alt hvad der bliver gennemgået i dette kursus er i Python 3.x

Morten Hansen
moh@sdu.dk

Mit første Python Program

- Ved at skrive:

```
print('Hello, World!’)
```

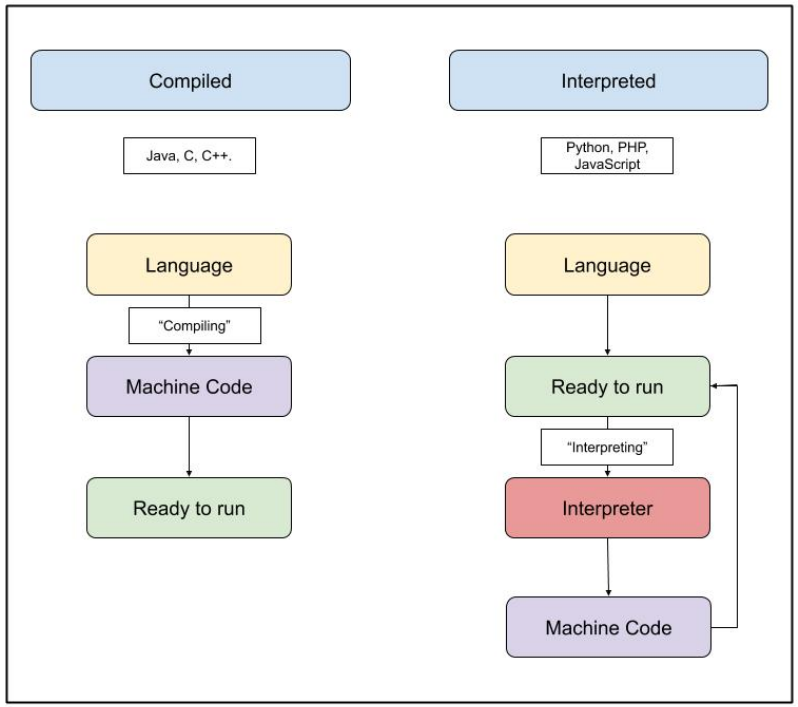
- Opnås:

```
Hello, World!
```


Morten Hansen
moh@sdu.dk

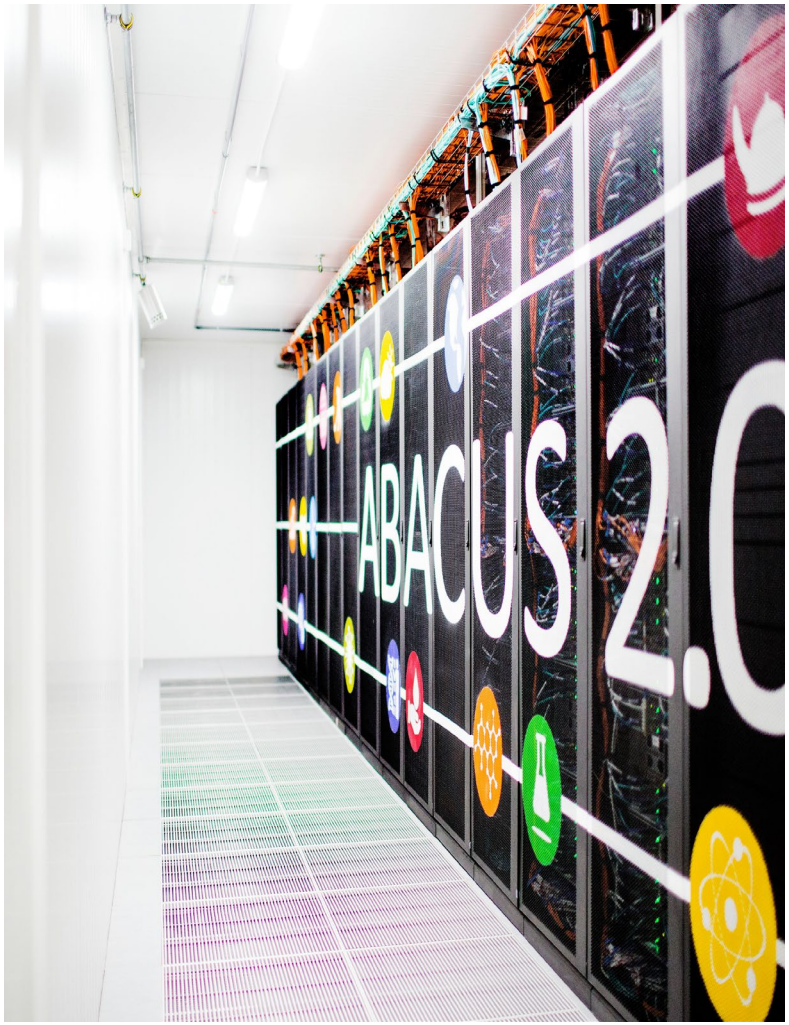
Fortolkning af Python kode

- Python kode køres fra toppen af filen/metode
- Python er et "*interpreted*" sprog
  - Bliver omdanet til bytecode og fortolket
  - En linje er fortolket ("interpreted") af gangen.
- I Python bruges "#" til at udkommenterer indhold og sætte kommentarer i koden



Morten Hansen
moh@sdu.dk

Python Syntaks



SDU

Variables

- **Variables:** name values for later use

```
my_integer = 5
my_floating_point = 26.2
my_boolean = True
my_string = 'characters’

my_integer_hex = 0xFF
my_integer_bit = 0b11010
```

- Har Python så ikke typer på variabler?

Morten Hansen
moh@sdu.dk

- Variabel navne
  - Case sensitive
  - A[a+1+_]
  - Meaningful, desciptive names
- Hexadecimal 0x prefix
- Indrykning

Morten Hansen
moh@sdu.dk

Python er case sensitivt.

Hvilken af disse vil blive printet?

```
size = "size"
Size = "Size"

print(size)
```

```
size
```


Morten Hansen
moh@sdu.dk

Print funktionen – Currency Exchange

```
# A converter for currency exchange.

USD_to_GBP = 0.66    # Today's exchange rate
GBP_sign = '\u00A3' # Unicode value for £
dollars = 1000       # Number dollars to convert

# Conversion calculations
pounds = dollars * USD_to_GBP


# Printing the results
print('Today, $' + str(dollars))
print('converts to ' + GBP_sign + str(pounds))
```


Morten Hansen
moh@sdu.dk

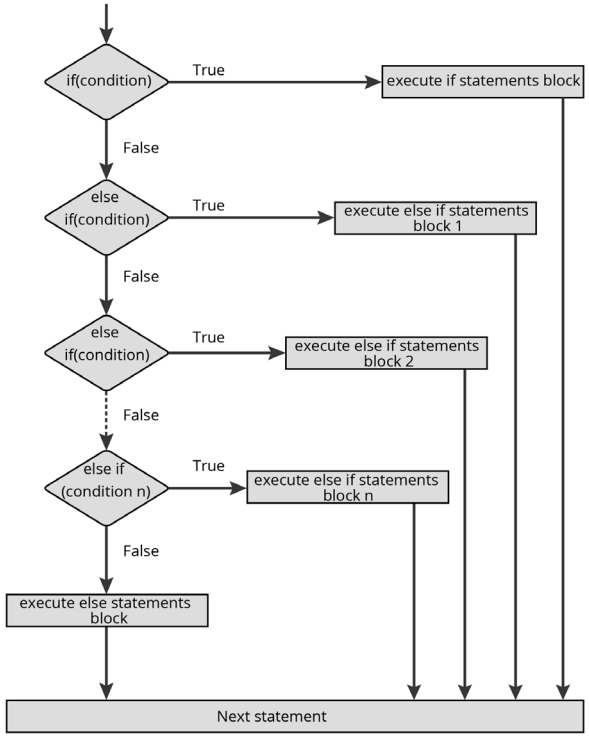
String formating

```
USD_to_GBP = 0.66    # Today's exchange rate
GBP_sign = '\u00A3' # Unicode value for £
dollars = 1000       # Number dollars to convert
pounds = dollars * USD_to_GBP

print('Today, $ %i'%dollars) #int input
#Output: "Today, $ 1000"
print('USD to GBP exchange rate %1.2f'% USD_to_GBP) #float input
#Output: "USD to GBP exchange rate 0.66" format with two decimals
print('Convert to %s %i'%(GBP_sign, pounds)) #str and int input
#Output: "Converts to £666"
```


Morten Hansen
moh@sdu.dk

if, else if, og else

- Ved brug af logiske sammenligninger kan vi styre eksekveringen af Python kode.
  - if
  - else if
  - else
- Kan sammensættes af flere logiske udtryk
  - and
  - or

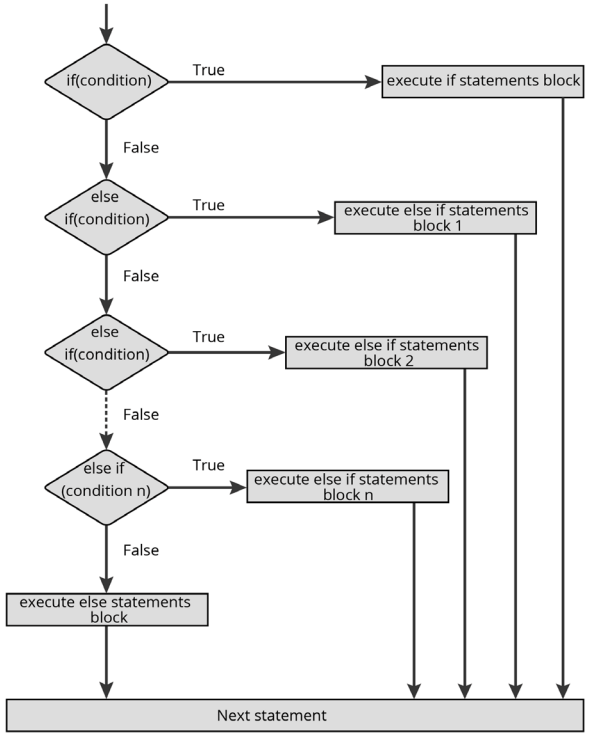



```
input = 1

if 1==input: #if
    print("input=1")
elif input<1: #else if
    print("input<1")
elif input==1: #else if
    print("input==1") #Vil aldrig ramme
elif input>1: #else if
    print("input>1")
else: #else
    print("Not at number")
```


Morten Hansen
moh@sdu.dk

if, else if, og else

- Ved brug af logiske sammenligninger kan vi styre eksekveringen af Python kode.
  - if
  - else if
  - else
- Kan sammensættes af flere logiske udtryk
  - and
  - or

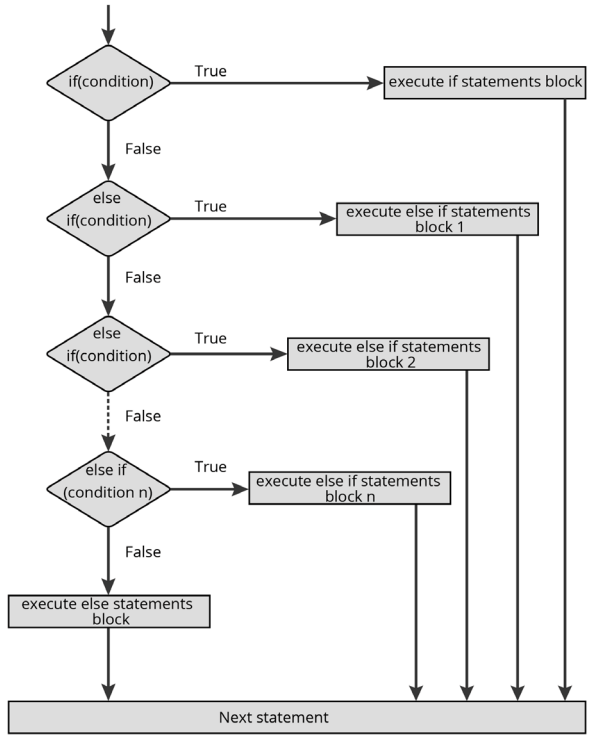



```
input = 1.0

if type(input) == int and bin(input) == "0b1":
    print("Input is is int and has the value 0b1")

if type(input) == int:
    if bin(input) == "0b1":
        print("Input is is int and has the value 0b1")
```


Morten Hansen
moh@sdu.dk

if, else if, og else

- Ved brug af logiske sammenligninger kan vi styre eksekveringen af Python kode.
  - if
  - else if
  - else
- Kan sammensættes af flere logiske udtryk
  - and
  - or




```
input = 1.0

if type(input) == int and bin(input) == "0b1":
    print("Input is is int and has the value 0b1")
```

```
if type(input) == int:
    if bin(input) == "0b1":
        print("Input is is int and has the value 0b1")
```

```
input = 1.0
if input < 1 or type(input) == float:
    print("Input < 1 or input is float")
```


Morten Hansen
moh@sdu.dk

Logiske udtryk

```
<       # Mindre end
<=      # Mindre eller lige
>       # Støre end
>=      # Støre eller lige
==      # Lig med.
        ( Virker ikke med objekter (e.g. klasser)
                brug i stedet ”is” ).
!=      # Forskellige fra
Is      # Objektidentitet
is not  # Objektidentitet forskellige fra
```

```
input = 1

if input != 1:
    print("!=")
elif input < 1:
    print("<")
elif input > 1:
    print(">")
elif input is None:
    print("None")
elif input is not None:
    print("not None")
```


Morten Hansen
moh@sdu.dk

Matematiske operationer

x + y	#plus
x - y	#minus
x * y	#gange
x / y	#divider
x // y	#divider rund ned.
x % y	# modulus finder restproduktet fra division
abs(x)	#naturlige tal
int(x)	#konverter til int
float(x)	# konverter til float
pow(x, y)	#potensen af x i y'erne
x ** y	#potensen af x i y'erne

```
x = 5
y = 3

x%y #2
x//y #1
x**y #5*5=25*5=125
```


Morten Hansen
moh@sdu.dk

Snak med din sidemand om:

Hvad bliver printet?

```
input = 2

if input>10:
    print("Større end 10")
elif input > 5:
    print("Større end 5")
elif input > 2:
     print("Større end 2")
else:
    print("Mindre end eller lig med 2")
```

```
input = 100

if input>10:
    print("Større end 10")
if input > 5:
    print("Større end 5")
elif input > 2:
     print("Større end 2")
else:
    print("Mindre end eller lig med 2")
```

SDU

Funktioner syntaks

- I Python kan der defineres funktioner:
  - Gøres ved brug af ordet **def**.
- Syntaks
  - def [navn_på_funktion] ([parameter]):
  - **return** bruges til at returnere indhold til kilderen.
- I Python bruges indentering til at viste nesting
  - Hvor man i Java bruger "{" "}".
- Eksempler på funktioner:

```
def demo_print(input):
    print(input)

demo_print("Demo")
```

```
def demo_return(input):
    return "Return: " + input

string_input = "Demo"

print(demo_return(string_input))
```


Morten Hansen
moh@sdu.dk

Bitvise operationer i Python ved brug af bin()

- Understøttet operationer:
  - & - And
  - | - OR
  - ^ - XOR
  - << - bit shift til venstre
  - >> - bit shift til højre
- Funktionen bin()
  - Bruges til at udskrive heltal til strenge i binær

```
bit_pattern_1 = 0b1101
bit_pattern_2 = 0b1010

result = bin(bit_pattern_1 ^ bit_pattern_2) #XOR

print("XOR:" + result)

result = bin(bit_pattern_1 & bit_pattern_2) #AND

print("AND:" + result)

result = bin(bit_pattern_1 | bit_pattern_2) #OR

print("OR:" + result)

result = bin(bit_pattern_1 >> 1) #bit shift Left

print("Shit:" + result)
```

```
XOR:0b111
AND:0b1000
OR:0b1111
Shit:0b110
```


Morten Hansen
moh@sdu.dk

Debugging – Typer af fejl

- *Syntax errors*

```
print(5 +)
SyntaxError: invalid syntax
```

```
pront(5)
NameError: name 'pront' is not defined
```

- *Semantic errors*
  - Forkerte udtryk som

✗
```
total_pay = 37 + extra_hours * pay_rate
```

✓
```
total_pay = (37 + extra_hours) * pay_rate
```

- *Runtime errors*
  - Fejl der opstår i runtime
  - Divide by zero

Morten Hansen
moh@sdu.dk

Debugging eksempel

```
x = 5

print("X=%i"%x)

y = 3
print("Y=%i"%y)

result = x+y
pront("%i+%i=%i"%(x,y,result))
```

```
>>> %Run debugging.py
X=5
Y=3
Traceback (most recent call last):
  File "C:\Users\jehs\OneDrive - Syddansk Universitet\Teaching\Computersystemer\E2020\Lektion 3 - Intro til programmering i Python\In class examples\debugging.py", line 9, in <module>
    pront("%i+%i=%i"%(x,y,result))
NameError: name 'pront' is not defined
```


Morten Hansen
moh@sdu.dk

Snak med din sidemand om:

Find fejl i følgende og identificer typen af fejl:

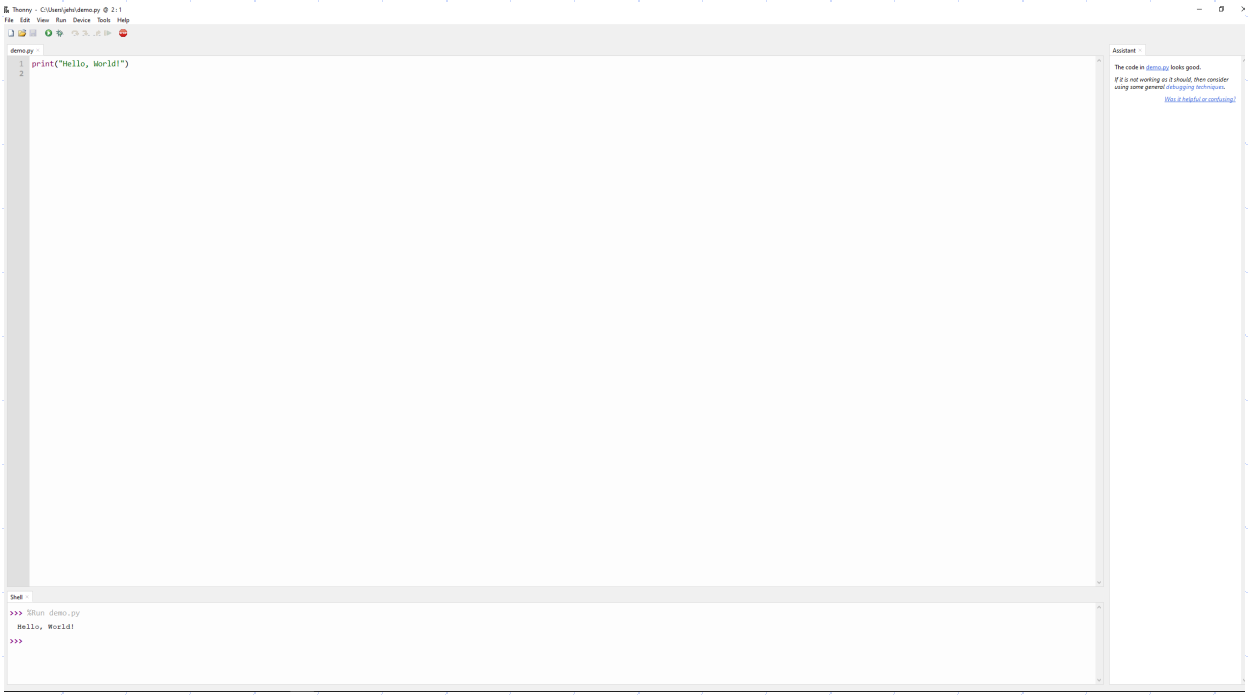
```
saldo = 1419.58
rentesats = 2.5
saldo = saldo * (rentesats / 100 + saldo)
print ('saldo: %.2f kr.' %saldo )
```

SDU

Thonny
Python editor

SDU

Editor



Morten Hansen
moh@sdu.dk

Summary

- Variabler
- If og Else struktur
- Funktioner — Nogle indbyggede funktioner
- Debugging

Morten Hansen
moh@sdu.dk

Spørgsmål?

Morten Hansen
moh@sdu.dk