



**Peter M. Dahlgren**  
peter.dahlgren@jmg.gu.se

JU1203  
HT18

# Nu + senare












1. Vad Python är
2. Grundläggande koncept, med övningar direkt efter
3. Bygga nyhetsrobot

# Vad är Python?



Populärt programmeringsspråk för program och internet

- version 3+
- utvecklat 1991
- open source
- objekt-orienterat
- högnivåspråk
- tolkande
- journalister och forskare
- enkelt (jämfört med andra språk)

Language Rank	Types	Spectrum Ranking
1. Python	 	100.0
2. C	  	99.7
3. Java	  	99.5
4. C++	  	97.1
5. C#	  	87.7
6. R		87.7
7. JavaScript	 	85.6
8. PHP		81.2
9. Go	 	75.1
10. Swift	 	73.7

IEEE Spectrum ranking

# Vad betyder "enkelt"?

Python, högnivåspråk:

```
print("Hello, World!")
```

Assembler, lågnivåspråk:

```
org 100h  
mov dx,msg  
mov ah,9  
int 21h  
mov ah,4Ch  
int 21h  
msg db 'Hello, World!',0Dh,0Ah,'$'
```

**WAT.**



# Python-miljön

Python installerat på Mac + Linux, installation krävs på Win

- **Python** – tolkar och kör koden
- **pip** – hjälper till att installera bibliotek
- **Jupyter Notebooks** – verktyg för interaktiv kodning

Vi kommer använda **Anaconda** som innehåller allt vi behöver  
(allt kan dock installeras separat)

# Två sätt att köra Python-program

- Terminalen/kommandoprompten
- Interaktiv kodning

# Terminalen/kommandoprompten

Textfilen myprogram.py

```
print("Hello, World!")
```



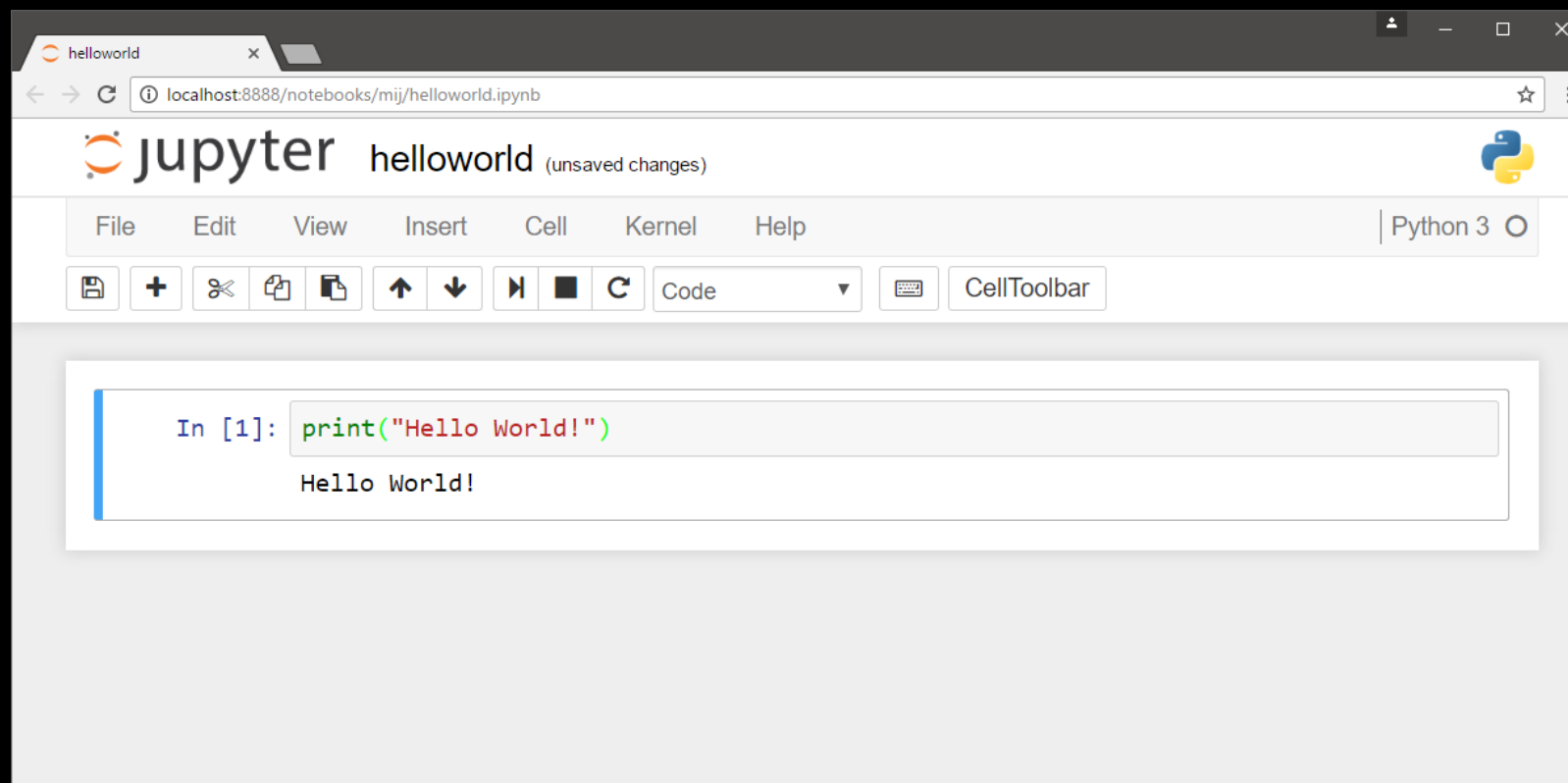
```
peter@ATLAS: ~  
peter@ATLAS:~$ python  
Python 2.7.6 (default, Oct 26 2016, 20:30:19)  
[GCC 4.8.4] on linux2  
Type "help", "copyright", "credits" or "license" for more information.  
>>> 10 + 5  
15  
>>> _
```

Skriv kod direkt i tolken, rad för rad

```
peter@ATLAS: ~  
peter@ATLAS:~$ python myprogram.py
```

Skicka textfil med kod till tolken

# Interaktiv kodning



Jupyter Notebook – skriv kod i webbläsaren



# Grundläggande koncept

- **Variabler** lagrar data (text, nummer, listor etc)
- **if** styr vad programmet gör
- **for**-loopar går igenom en lista, rad för rad
- Funktioner **def** (define) skapar kodblock som kan återanvändas

# Python som miniräknare

```
print(50 + 3)
print(50 - 3)
print(50 * 3)
print(50 / 3)
print((50 + 3) * (3 / 12))
```

Resultat:

```
53
47
150
16.666666666666668
13.25
```

# Variabler

## Nummer och strängar (text):

```
age = 25           # Heltal
height = 174.5     # Decimaltal
name = "John Doe"  # Sträng (valfri text)
married = True      # Booleansk (True/False)
```

Variabelnamn

Variabelvärde

Kommentarer (skriv vad du vill)

## Listor:

```
workdays = [1, 2, 3, 4, 5]      # Lista med nummer
employees = ["John", "Jane", "Joe"] # Lista med strängar
```

# Variabler

## Kombinera värden med +

```
age = 25 + 5           # 30
name = "John" + " " + "Doe" # John Doe
workdays = [1, 2, 3] + [4, 5] # 1, 2, 3, 4, 5
```

# Variabler

Kan inte blanda variabeltyper, måste konverteras:

```
age = 25  
text = "John Doe is " + str(age) + "years old"
```



str(x) gör om något till en textsträng

# Variabler

Lång text på flera rader:

```
biography = """Lång text över  
flera rader fungerar bra så länge det är tre fnuttar  
i början respektive slutet, i stället för bara en  
"""
```

# Variabler

- Använd **engelska** variabelnamn
- Variabelnamnen kan innehålla **A-Z**, **0-9** och **\_**

# Kommentera din kod

Skriv # följt av valfri text.

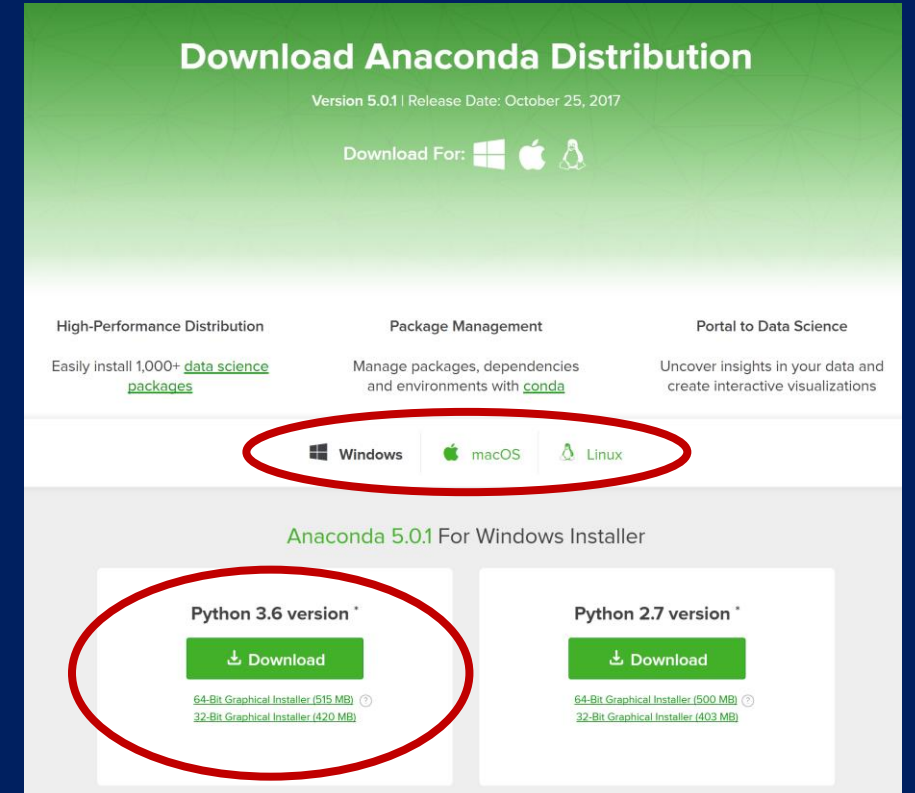
```
# Addera två nummer.  
print(50 + 3)  
  
print(50 - 3)    # Subtrahera två nummer.
```



# Installera Anaconda

Ladda ned och installera Anaconda: [continuum.io/downloads](https://continuum.io/downloads)

Starta [Navigator](#) → [Jupyter Notebooks](#)



# Övning 1 + 2

1. Starta Jupyter Notebooks
2. Gå till [github.com/peterdalle/ju](https://github.com/peterdalle/jupyter)
3. Leta upp Övning 1: Variabler
4. Därefter Övning 2: Strängar

# if-satser

`if`, `elif` och `else` styr vad programmet gör

- Bygger på booleansk logik (sant/falskt)
- Om uttrycket är sant, kör koden nedanför

```
if uttryck:  
    print("Kör denna koden om uttrycket är sant.")
```

# if-satser

**if** styr vad programmet gör

```
age = 25

if age == 25:
    print("Du är 25 år gammal")
```

# if-satser

**else** fångar upp allt annat

```
age = 20

if age == 25:
    print("Du är 25 år gammal")
else:
    print("Du är inte 25 år gammal")
```

# if-satser

**elif** är ytterligare if

```
age = 20

if age == 25:
    print("Du är 25 år gammal")
elif age >= 18 and age <= 25:
    print("Du är mellan 18 och 25 år gammal")
else:
    print("Du är inte 25 år gammal")
```

# if-satser och uttryck

## Jämförelser:

```
age = 18          # Skapar variabeln age med värdet 18

age == 25         # Lika med?
age != 25         # Är inte lika med?

age > 25          # Över 25?
age >= 25         # 25 eller över?
age < 25          # Under 25?
age <= 25         # 25 eller under?
```

## Kontrollera om jämförelsen är True/False:

```
print(age == 25)    # False
```

# if-satser och uttryck

## Jämför strängar:

```
name = "John Doe"           # Sätt variabeln name till "John Doe"

name == "Elvis Presley"     # Lika med?
name != "Elvis Presley"     # Inte lika med?

print(name == "John Doe")   # True
```



# if-satser och uttryck

Använd **or** och **and** för kombinationer

- **or** innebär att **någon** jämförelse måste vara sann
- **and** innebär att **alla** jämförelser måste vara sanna
- **not** negerer (något är inte sant)
- gruppera med (parenteser) för att styra ordningen

```
print(5 == 5 and 1 == 1)           # True
print(5 == 5 and 1 == 1 and 1 == 3) # False
print(5 == 5 or 1 == 3)            # True
print((1 == 1) and (1 == 2 or 3 == 3)) # True
```

# if-satser och uttryck

Använd **or** och **and** för kombinationer

```
age = 21
male = False

if age == 25 and male:
    print("You are male and 25 years old.")
elif age != 25 and not male:
    print("You are female and above or below 25 years old.")
```

Resultat:

```
You are female and above or below 25 years old.
```

# Indentera koden, tryck på Tab

Rätt:

Måste ha kolon

```
if 1 == 1:
    print("This line is indented and works fine.")
else:
    print("This too.")
```

Fel:

```
if 1 == 1:
print("This will not work.")
else:
print("Neither will this.")
```

# Övning 3

1. Starta Jupyter Notebooks
2. Gå till [github.com/peterdalle/ju](https://github.com/peterdalle/jupyter)
3. Leta upp Övning 3: if-satser

# For-loopar

Gå igenom en lista av någonting, rad för rad.

```
for i in range(3):  
    print(i)
```

Resultat:

```
0  
1  
2
```

Variabeln i skapas automatiskt



# For-loopar

Från 0 till 100, hoppa över 25 nummer varje steg.

```
for i in range(0, 101, 25):  
    print(i)
```

Resultat:

```
0  
25  
50  
75  
100
```

# For-loopar

Kan användas tillsammans med listor.

```
holidays = ["Saturday", "Sunday"]  
  
for day in holidays:  
    print(day)
```

Resultat:

```
Saturday  
Sunday
```

# For-loopar

Kan användas tillsammans med listor.

```
numbers = [6, 7632, 10]

for i in numbers:
    print(i)
```

Resultat:

```
6
7632
10
```

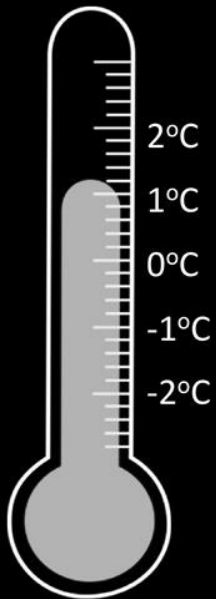


# Övning 4

1. Starta Jupyter Notebooks
2. Gå till [github.com/peterdalle/ju](https://github.com/peterdalle/jupyter)
3. Leta upp Övning 4: for-loopar

# Funktioner

Funktioner grupperar kod i logiska block. Sedan “kallar” man bara på funktionen. Då slipper man skriva koden flera gånger.



21 C



Funktion



69,8 F

1. Tar emot variabler

2. Magi

3. Returnerar ny variabel

# Funktioner: def

## Vi kan skapa egna funktioner

- kan ha valfritt namn
- kan ta emot variabler med valfria namn (**x** och **y** nedan)
- kan ge tillbaka (*returnera*) en variabel med **return**

```
def multiply(x, y):  
    return x * y
```

# Funktioner: def

Därefter kan vi använda vår nya funktion:

```
multiply(10, 3)
```

Resultat:

```
30
```

# Funktioner: def

## Fler exempel:

Ingenting returneras tillbaka eftersom denna funktion saknar return

```
def sayhello(name):  
    print("Hello " + name)  
  
sayhello("Peter")
```

Resultat:

```
Hello Peter
```

# Funktioner

Vi har redan stött på några funktioner

`range()` tar start- och slutvärden som variabler och returnerar en lista

```
for i in range(1, 10):  
    print(i)
```

`str()` tar en variabel som variabel och returnerar den som en sträng

```
message = "Du är " + str(20) + " år gammal"
```

`print()` tar valfri variabel och skriver det på skärmen (men returnerar ingenting)

```
print(message)
```

# Övning 5

1. Starta Jupyter Notebooks
2. Gå till [github.com/peterdalle/ju](https://github.com/peterdalle/jupyter)
3. Leta upp Övning 5: Funktioner

# Stoora program

Genom att kombinera enkla byggstengar som `if`, `for` och `def` kan vi bygga nästan vilket program som helst.

Viktigt att `# kommentera` koden!



# Övning 6

1. Starta Jupyter Notebooks
2. Gå till [github.com/peterdalle/ju](https://github.com/peterdalle/jupyter)
3. Leta upp Övning 6: Felhantering

# Lär dig mer

Kolla [github.com/peterdalle/ju](https://github.com/peterdalle/ju) längst ned för länktips

Bra att även lära sig **terminalen** eller kommandoprompten.

