

# Programmering i Python

**27 oktober**

Grunder i Python

**10 november**

Slingor, funktioner, moduler

**17 november**

Klasser, metoder

**1 december**

Praktisk tillämpning

**8 december**

?

# Kursen

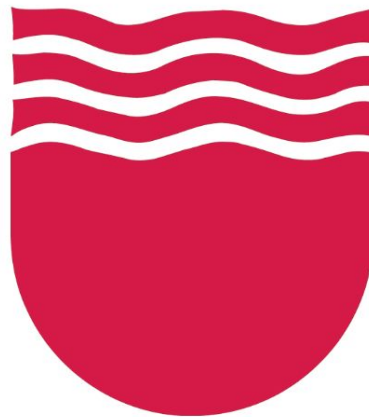
Studiecirkel

Gemensamt lärande

Varje tillfälle:

1. Genomgång
2. Bensträckare/Kaffe
3. Uppgifter att arbeta med (hjälp från mig och varandra)

Skickar länk till det vi gick igenom om man vill titta på det hemma



# Jag

Joakim Loxdal

Går tredje året datateknik på KTH

072 - 512 91 25 / loxdalen@gmail.com

# Vad är programmering?

- Instruktioner till en dator
- Olika språk är bra på olika saker
  - Snabbare (men krångligare grammatik)
  - Enklare grammatik (men långsammare)
  - Smarta inbyggda funktioner
  - Fungerar på alla operativsystem

# Vad är Python?



- Ett kraftfullt men enkelt språk
- Fungerar på i princip alla operativsystem
- Namnet från Monty Python
- Gavs ut 1991 första gången
- Python 3 kom 2008, ganska annorlunda från tidigare versioner
- Objektorienterat (frivilligt)

# Vanliga datatyper i python

- Heltal / Flyttal (int, float)
  - Heltal: -1, 1, 2, 3
  - Flyttal (decimaltal): -10.1, 1.3, 3.141593
- Boolska värden (bool)
  - True / False
- Strängar (str)
  - "Hej jag är en sträng"
  - "" (tom sträng är också en sträng)
- Listor (list)
  - [10, 11, 12, 13]
  - ["en", "lista", "med", "strängar", "och", 1, "heltal"]
- Tupler (tuple)
  - (0, 1), ("Jocke", 22)

# Operationer med heltal och flyttal i python

- Addition: `1 + 2`
- Subtraktion: `1 - 2`
- Multiplikation: `1 * 2`
- Vanlig division: `1 / 2`  $\Rightarrow$  `0.5`
- Heltalsdivision: `1 // 2`  $\Rightarrow$  `0` (avrundas nedåt till närmaste heltal)
- Upphöjt till: `1 ** 2`
- Modulo: `1 % 2`  $\Rightarrow$  `1` (resten när man heltalsdividerar 1 med 2)
- En int/float kan omvandlas till en sträng med funktionen `str()`
  - `str(7)`  $\Rightarrow$  `"7"`

# Strängar i python

- Konkatenering av strängar:
  - `"hej" + " " + "allihopa" => "hej allihopa"`
  - `"Hej, jag vill ha " + str(2) + " hamburgare"`
- "Multiplikation" med en sträng:
  - `"hej" * 3 => "hejhejhej"`
- Längden av en sträng
  - `len("En sträng") => 9`
- Dela upp en sträng i ord
  - `("En sträng").split() => ["En", "sträng"]`
- Se på en bokstav i strängen
  - `"En sträng"[0] => "E", "En sträng"[1] => "n"`
- Att behandla en sträng som ett heltal
  - `int("10") - 3 => 7`
  - `int("7") + 3 => 10`
  - `int("inte en siffra") => ERROR!`



# Strängar i python

Fler metoder som man kan använda på strängar. Alla går att hitta om man printar

`dir(string)` . Funkar även med `dir(int)`

`"sträng".upper()`  $\Rightarrow$  `"STRÄNG"`

`"STRÄNG".lower()`  $\Rightarrow$  `"sträng"`

`"sTrÄnG".swapcase()`  $\Rightarrow$  `"StRäNg"`

`"sträng".find("g")`  $\Rightarrow$  `5`

# Jämförelser mellan element

- Jämför om två element är samma
  - `1 == 1` `=> True`
  - `10 == 100` `=> False`
  - `"Hej" == "hej"` `=> False`
  - `"halloj" == "halloj"` `=> True`
- "Jämför om ett element är större/mindre än ett annat"
  - `3 > 5` `=> False`
  - `5 > 3` `=> True`
  - `len("Hej") < 5` `=> True`
  - `"Sträng" < 100` `=> ERROR`

# Vårt första program

Ett program som skriver ut strängen "Hej världen!"

```
# Detta program skriver ut strängen "Hej världen!"  
print("Hej världen!")
```

Skapa en fil (hejvarlden.py) med denna kod i IDLE och kör programmet

Vi kan också "print":a andra värden än strängar

Exempel: `print(42)` eller `print([1, 2, 3])`

# Variabler

För att vi ska kunna spara värden i vårt program till senare så måste vi ha någonstans att lägga dom. Detta är vad vi kallar variabler.

Exempel:

```
fornamn = "Karl"  
efternamn = "Gustav"  
print(fornamn + " " + efternamn)
```

Vi kan döpa variablerna till ungefär vad vi vill, så länge de börjar med en bokstav och är alfanumeriska. Understreck ( \_ ) är också godkänt.

Vilka variabelnamn nedan är godkända/inte godkända av Python?

```
lnamn = "Karl"  
namn2 = "Gustav"  
mellan-namn = "XVI"
```

# Variabler

Vi kan skriva över värdet i en variabel med ett nytt värde

```
a = 1
b = 1
a = 2

print(a + b)
```

variables1.py

```
a = 1
b = 1
c = a + b
a = b
b = c
c = a + b

print(c)
```

variables2.py

# Indata

Ofta vill vi använda information som användaren ger till programmet, så kallad indata/input.

En funktion som hämtar indata från användaren är `input()`.

Vi kan spara indatan i en variabel och göra något med den

```
name = input("Vad heter du?")  
name_caps = name.upper()  
print(name_caps)
```

Testa programmet (input.py)

# If-satser

För att göra olika saker beroende på värden i vårt program så använder vi if-satser.

Exempel (club.py):

```
age = input("Hur gammal är du? ")

if (int(age) > 17):
    print("Välkommen in")

else:
    print("Tyvärr, du är inte gammal nog")
```

# If-satser

Ibland vill man kontrollera fler saker samtidigt i en if-sats

```
age = 23
```

```
name = "John"
```

```
if age > 17 and name == "John":
```

```
    print("Du är vuxen och heter John")
```

```
else:
```

```
    print("Du är inte en vuxen person som heter John")
```



# Uppgifter: (som hjälp använd föreläsningen eller fråga, länk: [shorturl.at/evAZ3](https://shorturl.at/evAZ3))

1. Skapa ett program (product.py) som tar två siffror som indata och printar ut produkten av dessa  
-----
2. Skapa ett program (power.py) som tar två siffror som indata och printar ut det första talet upphöjt till det andra  
-----
3. Skapa ett program (even.py) som tar in ett heltal (int) och som printar Ja! om talet är jämnt och Nej! om talet är udda. Tips: modulo (%)  
-----
4. Skapa ett program (greeting.py) som tar in ett förnamn och ett efternamn och åldern och printar ut en hälsning som använder sig av den informationen, t.ex. *"Hej Förnamn Efternamn, du är 18 år"*  
-----
5. Skapa ett program (vowel.py) som tar en bokstav som indata och printar Ja! Om bokstaven är en vokal och Nej! Om bokstaven är en konsonant  
-----
6. Skapa ett program (max.py) som tar in tre tal som indata och printar ut det största av de tre värdena