

Programmering i Python

F1	Grunder i Python
F2	Loopar, moduler
F3	Funktioner, läsa data, dictionaries
F4	Dictionaries, API:er/requests
F5	?

Välkomna!

Studiecirkel

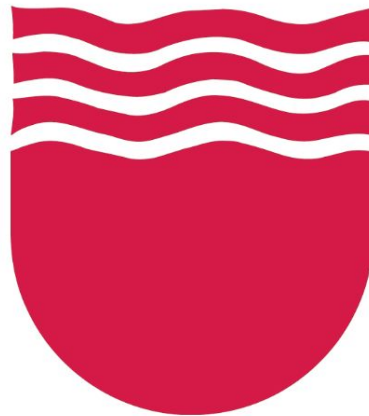
Gemensamt lärande

Varje tillfälle:

1. Genomgång
2. Paus
3. Uppgifter att arbeta med (med hjälp)

Det vi gör på kursen finns på en hemsida:

<http://github.com/mrkickling/pythonkurs>



Jag

Joakim Loxdal

Går datateknik på KTH

loxдалen@gmail.com

Vad är programmering?

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

Vad är Python?



- Ett kraftfullt men 'enkelt' språk (enkel grammatik)

- Exempel:

```
if person.age > 17:  
    print("Personen är välkommen in på klubben")
```

- Fungerar på i princip alla operativsystem
- Namnet från Monty Python
- Gavs ut 1991 första gången
- Python 3 kom 2008

Datatyper i python

- Heltal / Decimaltal (**integer**, **float**)
 - Heltal: -1, 1, 2, 3
 - Flyttal (decimaltal): -10.1, 1.3, 3.141593
- Boolska värden (**bool**)
 - Kan vara antingen `True` eller `False`
- Strängar (**string**)
 - `"Hej jag är en sträng"`
 - `""` (tom sträng är också en sträng)
- Listor (**list**)
 - `[10, 11, 12, 13]`
 - `["en", "lista", "med", "strängar", "och", 1, "heltal"]`
- Finns många andra datatyper

Operationer med heltal och flyttal i python

- Addition: `1 + 2` eller `4 + 5.6`
- 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`
- En int/float kan omvandlas till en sträng med funktionen `str()`
 - `str(7)` \Rightarrow `"7"`

Strängar i python

- Konkatenering (ihopsättning) 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 till en lista med ord
 - `("En sträng").split() => ["En", "sträng"]`
- Hämta en bokstav på en viss plats i strängen (räknar från 0)
 - `"En sträng"[0] => "E", "En sträng"[1] => "n"`
- Göra om en sträng till 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 skriver `dir(str)` eller på Pythons hemsida: <https://docs.python.org/3/library/stdtypes.html#string-methods>

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

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

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

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])`

`print()` är en funktion som visar en sträng för användaren

`#` betyder att vi skriver en kommentar python kan ignorera, kommentarer är för människor

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)  # OBS: print(fornamn + " " + efternamn)  går lika bra
```

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

Fråga

Vilka variabelnamn nedan är godkända av Python?

1. `1namn = "Karl"`
2. `namn2 = "Gustav"`
3. `mellan-namn = "XVI"`
4. `Min hälsning = "hej på dig"`

Variabler

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

```
a = 1  
b = 2  
a = 3
```

```
print(a, b)
```

Vad är värdet på A?

(Praktiskt att 'printa' fler värden kommaseparerade)

```
a = 1  
b = 2  
c = a + b
```

```
print(a, b, c)
```

Värdet på det till höger om likamedtecknet läggs i variabeln till vänster om likamedtecknet.

Listor i python

`[1, 2, 3, 4]` En lista med 4 element.

```
lista = [1, 2, 3, "hej"] # Läger en lista i variabeln 'lista'  
print(lista) # printa listan
```

```
lista[0] = 10 # ändra det första elementet i listan till 10  
lista[1] = 11 # ändra det andra elementet i listan till 11  
print(lista) # printa listan igen  
print(lista[0]) # printa bara första elementet i listan
```

Listor i python

```
greetings = ["God dag", "Hej"] # Lägger en lista i variabeln 'lista'  
print(greetings) # printa listan igen
```

```
greetings.remove("Hej") # tar bort "Hej" från listan  
print(greetings) # printa listan igen
```

```
greetings.append("Tjenare") # lägger till ett element i listan  
print(greetings) # printa listan igen
```

Fråga

```
lista = [1, 2, 3, 4, 5]
```

Hur ändrar vi det första elementet i listan till 10 istället för 1?

Hur lägger vi till siffran 6 i slutet av listan?

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)

Jämförelser mellan värden

- Jämför om två värden är samma
 - `1 == 1` `=> True`
 - `1 == "1"` `=> False`
 - `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`

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")
elif int(age) > 14:
    print("Du får iaf åka moppe")
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")
```

Fråga

```
age = int(input("Hur gammal är du?"))
```

Vad ska vi skriva i Python om vi vill kontrollera i fall age är större än 18 och mindre än 65?

If-satser

Det går att kontrollera om ett element finns i en lista med 'in'

```
familj = ["Erik", "Pierre", "Hivron"]
```

```
If "Erik" in familj:
```

```
    print("Erik är med i familj")
```

```
else:
```

```
    print("Erik är inte med i familj")
```

Fungerar även med strängar:

```
if "a" in "Joakim":
```

```
    print("ditt namn innehåller ett A")
```

Fråga

```
svenska_bokstaver = "åäö"  
bokstav = input()
```

Vad ska vi skriva om vi vill kontrollera ifall en bokstav vi får som input är en 'svensk' bokstav?

Uppgifter: (som hjälp använd föreläsningen eller fråga)

1. Skapa ett program (product.py) som tar två heltal som indata och printar ut produkten (multiplikation) av dessa

2. Skapa ett program (power.py) som tar två heltal som indata och printar ut det första talet upphöjt till det andra

3. 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”*

4. 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 (tips: du kan använda ‘in’ på en lista med vokaler/konsonanter)

5. Skapa ett program (max.py) som tar in tre tal som indata och printar ut det största av de tre värdena

6. Skapa ett program (append.py) som tar in tre heltal som indata och lägger till var och ett av dem i en lista om dom är större än 10

Fri uppgift: Skapa ett adventure-game där det händer olika saker beroende på vad användaren ger för