

TDP002 – Imperativ programmering

Strängar och IO

Eric Elfving

Institutionen för datavetenskap (IDA)

Översikt

- Inläsning
- Strängformatering
- Utskrift
- Filhantering

Inläsning

- Program som gör en uträkning är bra, program som låter användaren "bestämma" är bättre
- Program blir ofta väldigt tråkiga om de gör samma beräkning varje gång
- I Python finns det "ett" sätt att låta användaren mata in data, `input`
- `input` ger alltid en sträng som resultat
- Till `input` kan man skicka en sträng som ska användas som ledtext till användare

Inläsning

input

```
name = input('Mata in ditt namn: ')\nprint("Hej " + name)
```

Mata in ditt namn: **Kalle**
Hej Kalle

```
price = input('Mata in pris: ')\nprice = int(price)\nproducts = input('Antal produkter: ')\nproducts = int(products)\nsum = price * products\nprint('Totalt värde: ' + str(sum))
```

Mata in pris: **12**
Antal produkter: **2**
Totalt värde: **24**

Inläsning

- Funktionerna `int` och `float` omvandlar en sträng till ett heltal respektive reellt tal
- Båda funktionerna kräver att strängen endast innehåller ett numeriskt värde (på rätt format)

Strängformatering

Metoden format

- Print är bra på att skriva ut saker men ibland vill man ha en snyggare utskrift.
- Då är `str.format()` bra att använda sig av.
- Man kan formatera strängar på flera sätt enligt <http://docs.python.org/py3k/library/string.html#format-string-syntax> och här tar vi upp en kortversion
- Liknar printf-språket en del men är väldigt sofistikerat!
- Tar formatbeskrivning på formatet `{namn:format}`

Strängformatering

format

```
name = input('Förnamn: ')
surname = input('Efternamn: ')
message = 'Hej {} {}!'.format(name, surname)
print(message)
```



```
Förnamn: Kalle
Efternamn: Svensson
Hej Kalle Svensson!
```

```
n = input('Förnamn: ')
s = input('Efternamn: ')
message = 'Hej {name} {surname}!'.format(
    name=n, surname=s)
print(message)
```

Strängformatering

formatering av tal

```
print('Tabell över tal i olika talbaser\n'+ '='*32)
print('  Bin  Oct  Dec  Hex')
for num in range(18):
    print('{0:5b}{0:5o}{0:5d}{0:5X}'.format(num))
```

```
pi = 3.141592
print('{}'.format(pi))
print('{:.2f}'.format(pi))
print('{:06.3f}'.format(pi))
```

3.141592
3.14
03.142

Tabell över tal i olika talbaser

Bin	Oct	Dec	Hex
0	0	0	0
1	1	1	1
10	2	2	2
11	3	3	3
100	4	4	4
101	5	5	5
110	6	6	6
111	7	7	7
1000	10	8	8
1001	11	9	9
1010	12	10	A
1011	13	11	B
1100	14	12	C
1101	15	13	D
1110	16	14	E
1111	17	15	F
10000	20	16	10
10001	21	17	11

Strängformatering

formatering av tal

```
print('Tabell över tal i olika talbaser\n'+ '='*32)
print('  Bin  Oct  Dec  Hex')
for num in range(18):
    formatstr = ''
    for f in 'bodX':
        formatstr += '{{0:5{c}}}'.format(c=f)

    print(formatstr.format(num))
```



Tabell över tal i olika talbaser

=====				
Bin	Oct	Dec	Hex	
0	0	0	0	
1	1	1	1	
10	2	2	2	
11	3	3	3	
100	4	4	4	
101	5	5	5	
110	6	6	6	
111	7	7	7	
1000	10	8	8	
1001	11	9	9	
1010	12	10	A	
1011	13	11	B	
1100	14	12	C	
1101	15	13	D	
1110	16	14	E	
1111	17	15	F	
10000	20	16	10	
10001	21	17	11	

Utskrift

```
>>> print(1, 2, 3)
1 2 3
>>> print(1, 2, 3,
          sep='x',
          end=' slut !')
1x2x3 slut !
```

- `print` används oftast för att skriva ut saker på skärmen
- `print` kan ta flera värden och skriver ut dem med ett mellanslag varje värde och avslutar med en ny rad (`' \n'`)
- Med hjälp av `sep` och `end` kan man ändra detta beteende

Filhantering

```
with open('minfil') as f:
    for line in f:
        for word in line.split():
            print(word, end='_')
        print()
#eller:
print(line.replace(' ', '_'))
```

- Ofta får man indata lagrad på fil
- Öppnas lättast med `with open (...)`
- Som standard går det endast att läsa från filen

Filhantering

moder

```
with open('minfil', 'r') as f:  
    for line in f:  
        print(line.strip())
```

- Om man vill ändra sättet man öppnar filen på (t.ex. för att kunna skriva till den) byter man ut 'r' mot något annat enligt nedanstående tabell:

Filmod	Egenskap
r	Öppna för läsning
w	Öppna för skrivning (trunkerar innehållet)
a	Öppna för skrivning till slutet av filen

Filhantering

teckenkodning

```
with open('minfil',  
          encoding='latin1') as f:  
    for line in f:  
        print(line)
```

- Vanligtvis öppnar python filen med den teckenkodning som är standard på ditt system
- Det kan man ändra med encoding-parametern

Filmod	Egenskap
r	Öppna för läsning
w	Öppna för skrivning (trunkerar innehållet)
a	Öppna för skrivning till slutet av filen
w+	Öppna för läsning och skrivning

Filhantering

Skriva till fil

```
with open('minfil', 'w') as f:  
    for val in range(3):  
        f.write(str(val))
```

```
with open('minfil', 'a') as f:  
    for val in range(3):  
        print(val,file=f)
```

- Med 'w' eller 'a' (write/append) kan man skriva till en fil
- Detta gör man med write eller med file-parametern till print



Linköpings universitet

expanding reality

www.liu.se