

# Flik 7: Moduler



# Moduler - varför och hur

- Organisera koden i logiska block – moduler är gjorda för att importeras
- Importera funktionalitet från olika moduler
- Varje modul är en egen fil - med filändelse *.py*
- Filnamnet är också modulens namn
- Moduler söks i följande ordning
  - Katalog i vilket programmet finns (`pwd/cwd/cd`)
  - Miljövariabeln `PYTHONPATH` (om den är satt)
  - Förkompilerade sökvägar, t.ex. `/usr/local/lib/python2.5/site-packages`
- Sökvägen finns i `sys.path` och kan ändras (importera först `import sys`)
- Innehållet i en modul exekveras när den laddas
- En modul laddas bara en gång

# Rekommenderad fillayout

```
#!/usr/bin/env python
```

(1) Första raden (Unix)

```
"This is my module"
```

(2) Dok-sträng för modulen

```
import sys  
import os
```

(3) Import av moduler

```
debug = True
```

(4) Globala variabler

```
class FooClass:  
    pass
```

(5) Klassdeklarationer

```
def my_func():  
    pass
```

(6) Funktionsdeklarationer

```
if __name__ == '__main__':  
    my_func()
```

(7) "main" body

# Exempel

## ■ Enkelt exempel - *mymod.py*

```
"""Modulen mymod är till för att...."""
__version__ = "0.1"
__author__ = "Göran Andersson <initgoran@gmail.com>"
my_global = 37
class MyFancyClass:
    """Klassen MyFancyClass är...."""
    pass
def my_func():
    """Funtionen my_func är...."""
    return True
```

## ■ Importera och använda *mymod.py*

```
#import standardlibrarymodules
#import 3rd party modules
import mymod

mfc = mymod.MyFancyClass()
mymod.my_global += 1
t = mymod.my_func()
```

# import vs from

- ***import*** - Skapar separat namnrymd

```
import mymod
```

- ***from*** - I vissa fall vill vi ladda in i samma namnrymd (undvik)

```
from mymod import *
```

```
# Nu kan jag stället för att skriva
```

```
# mfc = m.MyFancyClass()
```

```
# skriva
```

```
# mfc = MyFancyClass()
```

```
#
```

```
# Risk för namnkonflikter - "namespace pollution"!!
```

- **Ladda enbart specifika element från modul**

```
from mymod import MyFancyClass
```

```
# Fortfarande risk för namnkonflikter!!
```

# Specialvariabler

## ■ Specialvariabler

<code>_xxx</code>	Importera inte med <code>'from module import *'</code>
<code>__xxx</code>	Privat variabel i klasser (se nästa flik)
<code>__xxx__</code>	Systemdefinierat namn
<code>__all__</code>	Lista med namn som ska ingå i <code>import *</code>
<code>__author__</code>	Författare
<code>__version__</code>	Versionsnummer

- **ovn0701/aritmetik.py** Skapa en modul med namnet `aritmetik` som innehåller funktionen `addera`. Funktionen ska ta två tal som argument och returnera summan av argumenten.
- **ovn0702/importera.py** Skriv ett program som importerar modulen `aritmetik`. Programmet ska anropa `addera(2, 3)` och skriva ut resultatet.
- Lägg till variablerna `__author__` och `__version__` samt dokumentationssträngar i modulen och i funktionen. Kör kommandot `pydoc` (eller `pydoc3`) på modulens namn (utan ändelsen `.py`).