

Föreläsning 4

Programmeringsteknik

DD1310

- Felhantering
- Definiering av egna funktioner
- Parametrar
- Lokala och globala variabler
- Retursats
- None

Översikt

Felhantering

Program

Egen funktion

Logiskt fel

Global/ lokal

Sammanfattning

Felhantering

- `try` och `except` är reserverade ord som används för hantering av exekverings fel.

```
plista= ['s','v','m','mp','fp','c']
try:
    i=int(input("Ange index: "))
    parti = plista[i]
except:
    print ("Något fel inträffade")
```

Översikt

Felhantering

Program

Egen funktion

Logiskt fel

Global/ lokal

Sammanfattning

Syntax

```
try:
```

Kod som kan orsaka något typ av exekveringsfel

```
except:
```

Kod som exekveras om och endast om det blir något fel i kodblocket efter `try`

Översikt

Felhantering

Program

Egen funktion

Logiskt fel

Global/ lokal

Sammanfattning

Så här går det till

Kod i blocket mellan `try:` och `except:` börjar exekvera, men så fort ett fel uppstår i någon rad avbryts exekveringen omedelbart, resterande rader **i blocket** exekveras inte och koden som finns i blocket efter `except:` börjar exekvera i stället.

Översikt

Felhantering

Program

Egen funktion

Logiskt fel

Global/ lokal

Sammanfattning

Mer detaljerad

```

plista= ['s','v','m','mp','fp','c','kd']
try:
    i= int(input('Ange index: '))
    parti = plista[i]
except IndexError as indx:
    print ('ogilltigt index' , indx)
except ValueError:
    print ('index maste var ett tal')
except:
    print('uppstod något okänd fel!')

```

Översikt

Felhantering

Program

Egen funktion

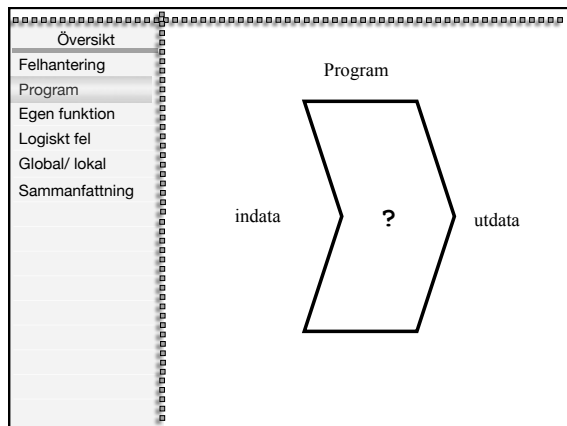
Logiskt fel

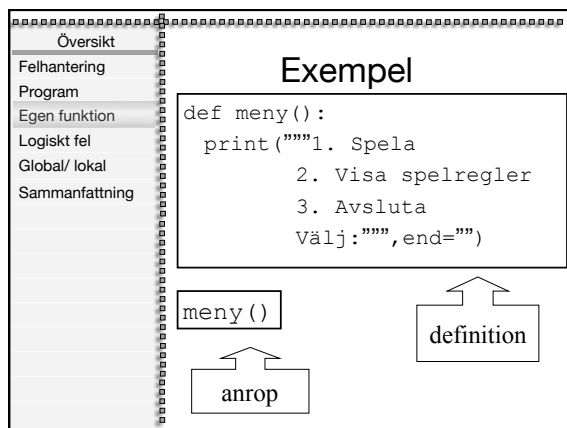
Global/ lokal

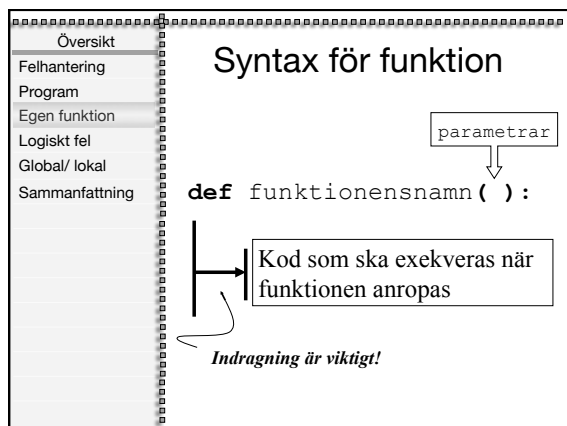
Sammanfattning

Olika typ av fel

Typ	Beskrivning
IOError	uppstår när man vill öppna en fil som inte finns.
IndexError	Uppstår vid tillgång till ett index i en lista som inte finns
KeyError	Uppstår när en nyckel inte finns i uppsalgslista
TypeError	Uppstår när en inbyggd operation eller funktion tillämpas på ett objekt av olämplig typ
ValueError	Uppstår när funktion används med ett parameter med korrekt typ men fel värde
ZeroDivisionError	Uppstår när divisor i en division är 0







Översikt
Felhantering
Program
Egen funktion
Logiskt fel
Global/ lokal
Sammanfattning

Funktioner

- Används för att dela upp ett program i **lätthanterliga** och **återanvändbara** delar
- En funktion tar oftast **indata** och ger **utdata**, men en funktion kan även utföra operationer
- Man kan undvika upprepning av kod genom att använda egna funktioner och parametrar
- Med hjälp av egna funktioner inför man abstraktion i sitt program

Översikt
Felhantering
Program
Egen funktion
Logiskt fel
Global/ lokal
Sammanfattning

Parametrar

En funktion med parametrar har större användningsområde än en funktion utan parametrar. Varför?

Vilken av följande har större användningsområde?

<pre>def calc(): print(2*3+3)</pre> <pre>calc() # 9</pre>	<pre>def calc(a,b): print(a*b+b)</pre> <pre>calc(2, 3) #9</pre> <pre>Calc(9, 5) #50</pre>
---	---

Översikt
Felhantering
Program
Egen funktion
Logiskt fel
Global/ lokal
Sammanfattning

return

En funktion med parametrar som **returnerar** ett värde har ännu större användningsområde.

<pre>def calc(a,b): print(a*b+b)</pre> <pre>calc(2,3)</pre> <pre>calc(5,4)</pre>	<pre>def calc(a,b): return a*b+b</pre> <pre>r = calc(2,3)</pre> <pre>print(r)</pre> <pre>q = calc(r,9)</pre>
--	--

Översikt
Felhantering
Program
Egen funktion
Logiskt fel
Global/ lokal
Sammanfattning

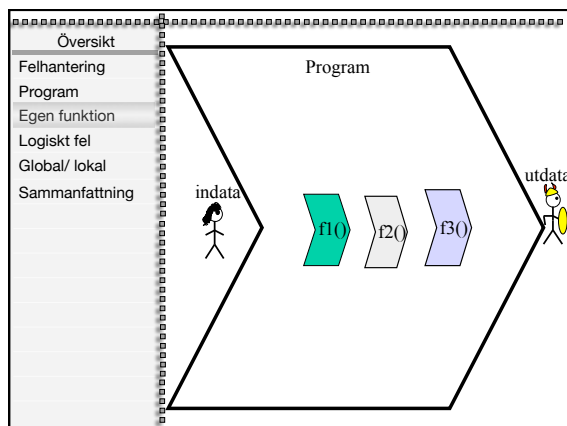
parametrar, return och None

- Indata skickas till funktioner via funktionens parametrar.
- Funktioner returnerar utdata med hjälp av `return`-satsen.
- Om en funktion inte har `return`-sats i kroppen kommer då funktionen att returnera `None`.
(*None* **vilket betyder ingenting i python**)

Översikt
Felhantering
Program
Egen funktion
Logiskt fel
Global/ lokal
Sammanfattning

Exempel

<pre>def calc(a,b): c = a*b + b return c</pre>	<pre>def calc(a,b): c = a*b + b</pre>
<pre>m=calc(2 , 3)</pre>	<pre>m=calc(2 , 3)</pre>
<i>Variabeln m kommer att få värdet 9 vilket är av typen <u>integer</u></i>	<i>Variabeln m kommer att få värdet <u>None</u></i>



Översikt
Felhantering
Program
Egen funktion
Logiskt fel
Global/ lokal
Sammanfattning

Logiskt fel

```
def BMI( height , weight):
    return weight/(height*height)
```

Vilket av följande anrop är fel?

BMI (1.70, 65)	BMI (65, 1.70)
22.49	0.000402

Översikt
Felhantering
Program
Egen funktion
Logiskt fel
Global/ lokal
Sammanfattning

Exekverings- och logiskt fel

- Exekveringsfel: det är ett fel som gör att programmet *kraschar* innan det hinner *avslutas normalt*
- Logiskt fel: det är då programmet *avslutas* med felaktig utdata. (denna typ av fel är oftast svårare att spåra)

Översikt
Felhantering
Program
Egen funktion
Logiskt fel
Global/ lokal
Sammanfattning

Globala och lokala variabler

En variabel kan antingen vara global eller lokal, kan alltså inte vara både samtidigt:

- Globala variabler: är variabler som lever tills programmet avslutas:
 - Globala variabler definieras med nyckelordet `global`
- Lokalvariabel: är variabler som oftast lever en kort stund.
 - Formella parameter
 - Lokala variabler som skapas i funktioner
 - Lokala variabler är kända endast i funktionen och har kortare livslängd än globala variabler

Översikt
Felhantering
Program
Egen funktion
Logiskt fel
Global/ lokal
Sammanfattning

Exempel

Följande kod är korrekt:

```
def change():
    global x
    x = x/2
x = 10
print(x)
change()
print(x)
```

Följande kod ger fel:

```
def change():
    x = x/2
x=10
change()
```

Översikt
Felhantering
Program
Egen funktion
Logiskt fel
Global/ lokal
Sammanfattning

Exempel

Följande kod ger fel:

```
def change(x):
    global x
    x = x/2
```

Följande kod är rätt:

```
def change(x):
    x = x/2
x=10
print(x)
change(x)
print(x)
```

Översikt
Felhantering
Program
Egen funktion
Logiskt fel
Global/ lokal
Sammanfattning

Sammanfattning

- Egna funktion leder till:
 - Högre abstraktion
 - Lättare att läsa och förstå
 - Mer överskådlig kod
 - Minskar upprepningar av kod
- Se till att dina funktioner är fristående och självständiga
- Beroenden mellan funktioner gör det svårt att modifiera programmet
- Undvik globala variabler
