Gruppetime

osjohans@uio.no

Agenda

1. Parameteroverføring
2. Skop
3. Oppg / LAB

Funksjoner

Funksjonskallet evaluerer til returverdien

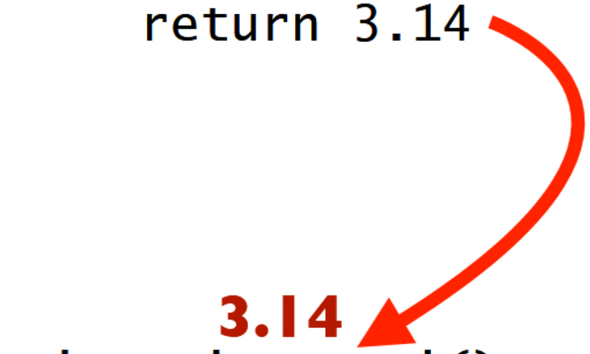
```
def gi_meg_pi():
    return 3.14
```

```
pi = gi_meg_pi()
print(pi)
```

Funksjoner

Funksjonskallet evaluerer til returverdien

```
def gi_meg_pi():
    return 3.14
```



3.14

```
pi = gi_meg_pi()
print(pi)
```

Funksjoner

Funksjonskallet evaluerer til returverdien

```
def gang_med_to(tall):

    return tall*2



dusin=13
toDusin = gang_med_to(dusin)
print(toDusin)
```

Funksjoner

Funksjonskallet evaluerer til returverdien

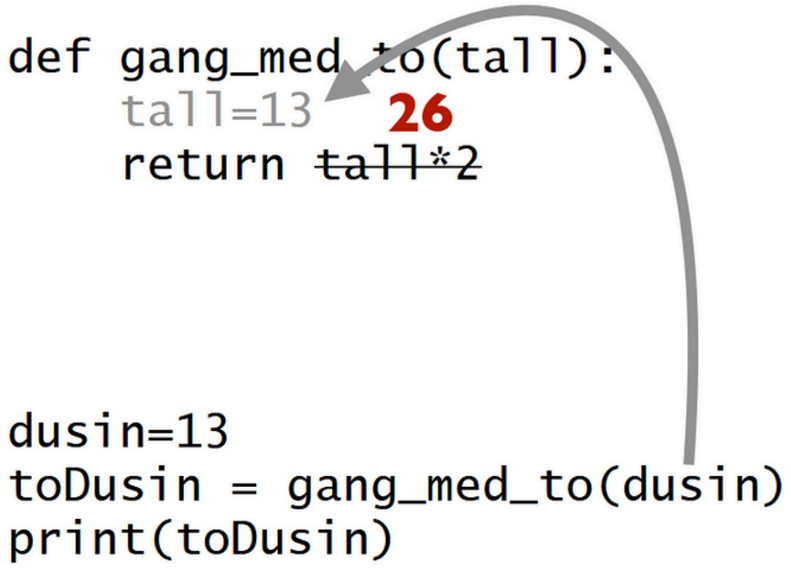
```
def gang_med_to(tall):
    tall=13
    return tall*2



dusin=13
toDusin = gang_med_to(dusin)
print(toDusin)
```

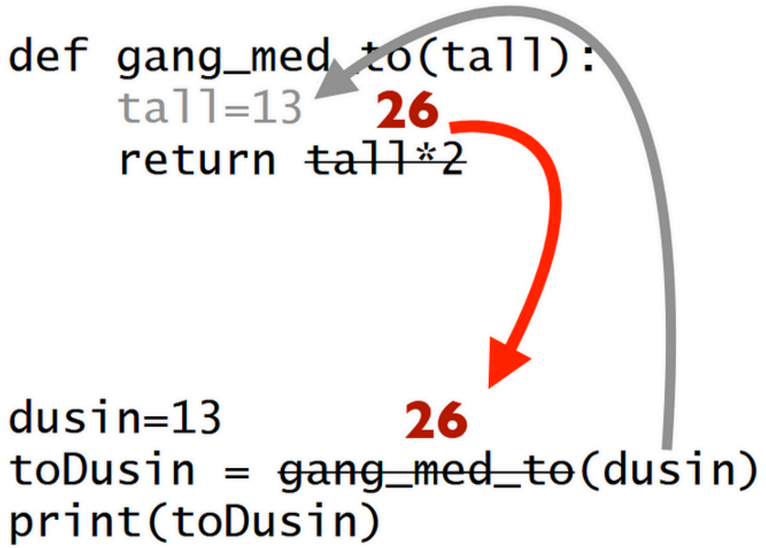
Funksjoner

Funksjonskallet evaluerer til returverdien



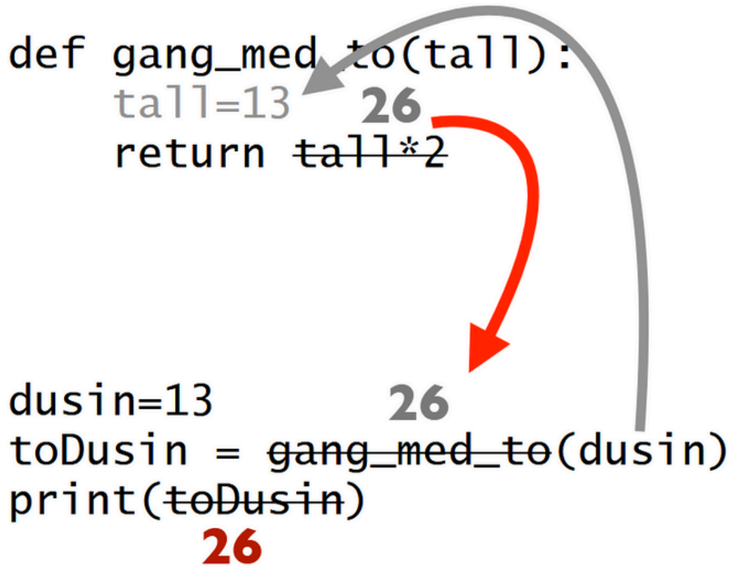
Funksjoner

Funksjonskallet evaluerer til returverdien



Funksjoner

Funksjonskallet evaluerer til returverdien



Skop

Vi har noen forskjellige skop i programmering:

1. Skop for variabler
2. Det globale skopet

Skop for variable

- Variablene som tilhører én funksjon er ikke tilgjengelige i en annen funksjon

- **Skopet** til en variabel er hvor i koden denne variabelen er tilgjengelig - hvilken funksjon den tilhører

- Vi overfører verdier mellom funksjoner gjennom parametre og returverdier

Globale skopet

- Variabler som **ikke** er del av en funksjon kalles globale

- Disse globale variablene er spesielle, fordi de i prinsippet er tilgjengelige inni alle funksjoner i filen

Globale skopet

I IN1000 unngår vi globale variable!

Hvordan?

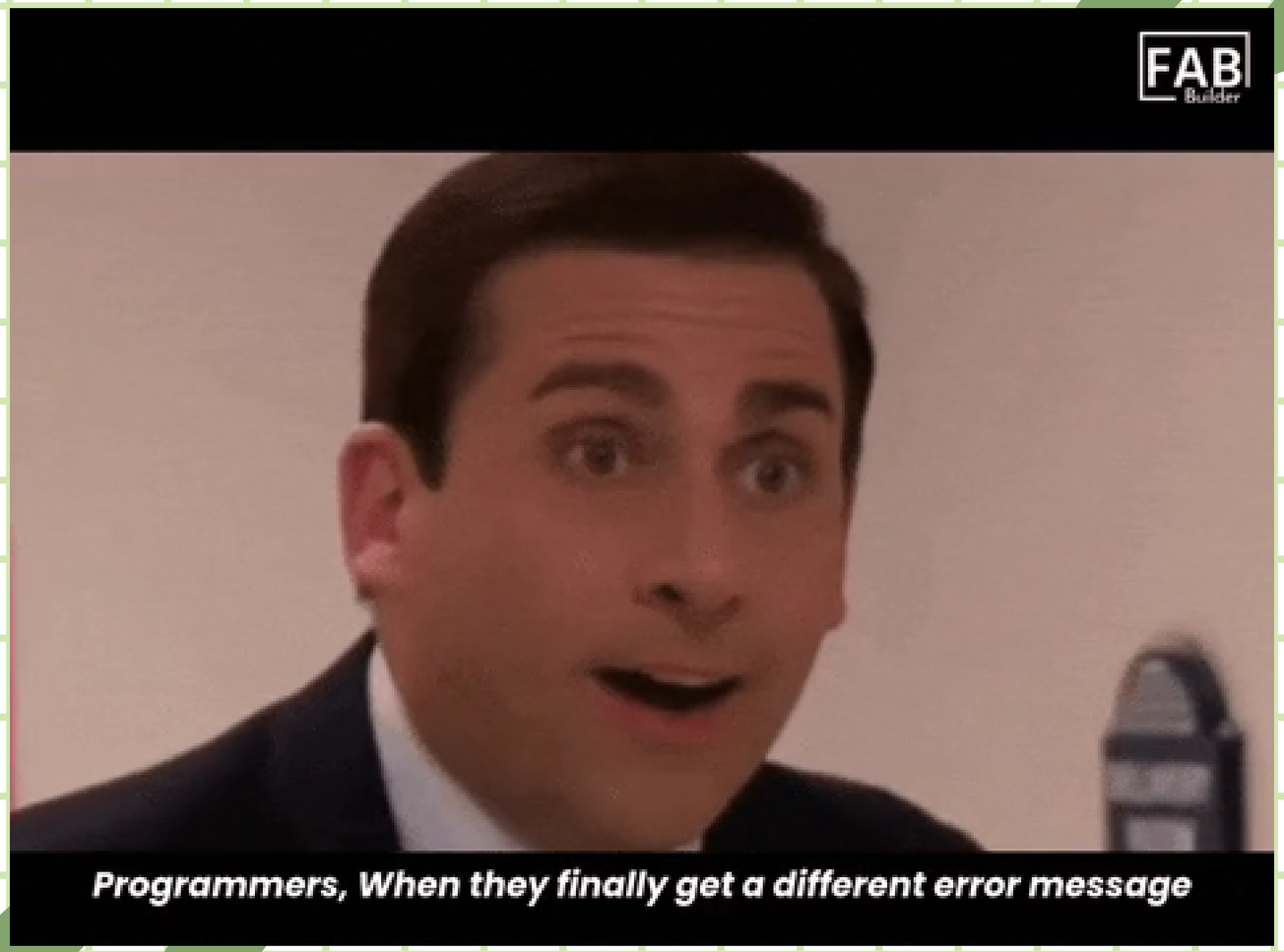
Flytt den ytre koden til en
funksjon (**hovedprogram**), slipper man mulig
forvirring rundt det lokale og globale skopet

PAUSE 15 M

osjohans@uio.no

OPPG / LAB

osjohans@uio.no


FAB
Builder
Programmers, When they finally get a different error message