

Skolan för Datavetenskap och kommunikation

DD1314

Programmeringsteknik

Föreläsning 3

- **Iteration:**

 - while

 - for

- **Operatorer i villkor** ==, < , ..., and, or, not
(operators in conditions)

- **Slumptal**

 - random

while-slingan

Upprepar satser så länge villkoret är uppfyllt.

```
while villkor:
```

```
    sats1
```

```
    sats2
```

```
    sats3...
```

```
antal = 0  
while antal != 10:  
    print(5)  
    antal += 1
```

for-slingan

Upprepar satser för varje element i en sekvens.

for variabel in sekvens:

sats1

sats2

sats3...

```
for antal in range(10):  
    print(antal)
```

Uppgift: Vad blir summan?

```
summa = 0
for i in range(5):
    summa += i
print(summa)
```

A. 15

B. 10

C. 5

D. 0

E. Inget av
alternativen

range

Funktionen **range** kan anropas med en, två eller tre parametrar, start, stopp och steg.

<code>range (n)</code>	n anger stopp	<code>0,1,2,...,n-1</code>
<code>range (m, n)</code>	m anger start	<code>m,m+1,m+2,...,n-1</code>
<code>range (m, n, s)</code>	s anger steg	<code>m,m+s,m+2s,...,n-1</code>

Uppgift: udda tal

Vi vill ha alla udda tal
mellan 0 och 10.

Hur ska vi anropa range?

A. `range(0,9,1)`

B. `range(0,10,2)`

C. `range(1,9,2)`

D. `range(1,10,2)`

E. Inget av anropen

Villkor

- Ett villkor har värdet `True` eller `False`.
- Datatypen kallas *boolean*.
- En *boolesk* variabel kan sättas direkt till ett villkorsvärde.
- Exempel: `spara = True`

jämföra=compare
villkor=condition

Operatörer i villkor

Operator	Betyder	Om vi satt dag=20 blir
==	lika med	dag==20 True
!=	skilt från	dag!=20 False
>	större än	dag>5 True
<	mindre än	dag<5 False
>=	större än eller lika med	dag>=5 True
<=	mindre än eller lika med	dag<=20 True

Jämföra strängar

- Strängar kan jämföras med avseende på likhet:

```
lösenord = input("Ge lösenordet: ")  
if lösenord == "qwerty":  
    print("Du är inloggad.")
```

- Och även med alfabetisk ordning:

```
if "elefant" < "elmätare":  
    print("elefant kommer före\  
        elmätare i ordlistan")
```

kombinera villkor

Villkor kan kombineras med operatorerna and, or, not

True and True	True
True and False	False
True or True	True
True or False	True
not True	False
not False	True

Givet `ålder=21`, `namn="Mireille"`;

vilket/vilka av följande villkor blir False?

A. `(ålder == 21) and (namn == "Mireille")`

B. `(ålder < 22) or (namn == "Algot")`

C. `(ålder == 5) or (namn > "Mireille")`

D. `(ålder >= 21) and (namn > "Algot")`

E. Inget av villkoren

Talgissning

```
# Talgissning, version 1
tal = 17
gissning = int(input("Gissa mitt tal: "))
if gissning != tal:
    print("Det var fel")
else:
    print("Rätt!")
```

Talgissning med funktion

```
# Talgissning, version 1.2 med funktion

def gissa():
    tal = 17
    gissning = int(input("Gissa mitt tal: "))
    if gissning != tal:
        print("Det var fel")
    else:
        print("Rätt!")

gissa()
```

Talgissning med parametrar

```
# Talgissning, version 1.3 med parametrar

def gissa(tal, gissning):
    if gissning != tal:
        print("Det var fel")
    else:
        print("Rätt!")

tal = 17
gissning = int(input("Gissa mitt tal: "))
gissa(tal, gissning)
```

Talgissning med returvärde

```
# Talgissning, version 1.4 med returvärde

def gissa(tal, gissning):
    if gissning != tal:
        return False
    else:
        return True

tal = 17
gissning = int(input("Gissa mitt tal: "))
rätt = gissa(tal, gissning)
if rätt == False:
    print("Det var fel")
else:
    print("Rätt!")
```


Talgissning med feedback

```
# Talgissning, version 2.0
tal = 17
gissning = int(input("Gissa mitt tal: "))

if gissning > tal:
    gissning = int(input("Lägre:"))
elif gissning < tal:
    gissning = int(input("Högre:"))

if gissning != tal:
    print("Det var fel")
else:
    print("Rätt!")
```

Algoritm för talgissning

En algoritm är en stegvis beskrivning av vad programmet ska göra.

Exempel:

1. Slumpa ett tal.
2. Låt användaren göra en gissning
3. Så länge som gissningen är fel:
 - Om gissningen är för hög : uppmana användaren att ge ett lägre tal och läs in ny gissning.
 - Om gissningen är för låg : uppmana användaren att ge ett högre tal och läs in en ny gissning.
4. När gissningen är rätt – skriv ut beröm.

Random-funktioner

Funktion	Resultat
<code>random.randint(1, 6)</code>	Slumpar ett heltal: 1,2,3,4,5 eller 6
<code>random.randrange(1, 6, 2)</code>	Slumpar ett udda heltal: 1,3 eller 5
<code>random.random()</code>	Slumpar ett decimaltal mellan 0.0 och 0.999...
<code>namn="Kit", "Nour", "Robin"</code> <code>random.choice(namn)</code>	Slumpar ett av namnen

Talgissning 3.1

```
# Talgissning, version 3.1
import random
tal = random.randint(1,100)
gissning = int(input("Gissa mitt tal: "))
n = 1
while gissning != tal:
    if gissning > tal:
        gissning = int(input("Lägre:"))
    elif gissning < tal:
        gissning = int(input("Högre:"))
    n += 1
print("Bravo, du gissade rätt!")
print("...på bara", n, "försök!")
```

