

Skolan för Datavetenskap och kommunikation

DD1310/DD1314/DA3009
Programmeringsteknik

Föreläsning 3

Kap 3-4 i Dawson

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

Iteration:

while

for

Slumptal

random

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 <code>dag=20</code> blir |
|--------------------|-----------------------------|--|
| <code>==</code> | lika med | <code>dag==20</code> True |
| <code>!=</code> | skilt från | <code>dag!=20</code> False |
| <code>></code> | större än | <code>dag>5</code> True |
| <code><</code> | mindre än | <code>dag<5</code> False |
| <code>>=</code> | större än eller lika med | <code>dag>=5</code> True |
| <code><=</code> | mindre än eller lika med | <code>dag<=20</code> 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 1.0

```
# Talgissning, version 1.1
tal = 17
gissning = int(input("Gissa mitt tal: "))
if gissning != tal:
    print("Det var fel")
else:
    print("Rätt!")
```


Talgissning 2.0

```
# 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!")
```

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 |

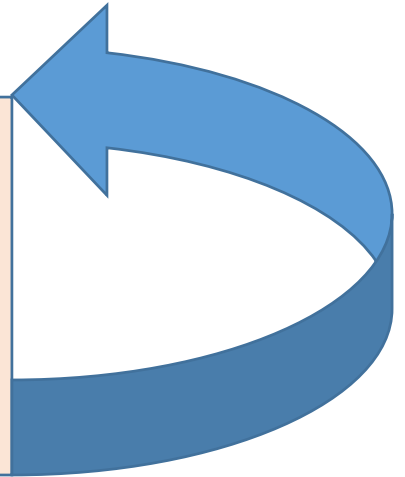
Algoritm för talgissning

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.

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!")
```



```
# Talgissning 4.0 med funktion
import random
```

```
def gissatal(tal = 17):
    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
    return n
```

```
n = gissatal()
print("Bravo, du gissade rätt!")
print("...på bara", n, "försök!")
```

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

for-slingan

En for-slinga upprepar ett antal satser för varje element i en följd.

Exempel:

- För varje tal i en följd (*range*):
 - Beräkna kvadraten!
- För varje bokstav i en sträng:
 - Skriv ut bokstaven!
- För varje telefonnummer på listan:
 - Ring upp personen!

följd = sequence

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

Vad blir ditt namn
baklänges?

Algoritm

1. Skapa en tom sträng
2. Gå igenom varje bokstav i namnet...
3. ...och lägg till i början på nya strängen

kontrollera algoritmen

| | |
|-------|-----------|
| bokst | bakfram |
| t | t |
| u | u+t |
| n | n+ut |
| s | s+nut |
| t | t+snut |
| r | r+tsnut |
| ö | ö+rtsnut |
| m | m+örtsnut |

funktion

```
def bakfram(namn):  
    """Returnerar namnet baklänges"""  
    bakfram = ""  
    for bokst in namn:  
        bakfram = bokst + bakfram  
    return bakfram
```

Läxa till torsdag: läs
Kap 5 i Dawson
list, Dictionary

