

## DD1310/DD1314/DA3009 PROGRAMMERINGSTEKNIK

### FÖRELÄSNING 3

- Kap 3-4 i Dawson
- Operatorer i villkor
- 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*

## OPERATORER I VILLKOR

Operator	Betyder	Om vi satt dag=20 blir
<code>==</code>	lika med	<code>dag==20</code> True
<code>!=</code>	skilt från	<code>dag!=20</code> False
<code>&gt;</code>	större än	<code>dag&gt;5</code> True
<code>&lt;</code>	mindre än	<code>dag&lt;5</code> False
<code>&gt;=</code>	större än eller lika med	<code>dag&gt;=5</code> True
<code>&lt;=</code>	mindre än eller lika med	<code>dag&lt;=20</code> True

## 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!")
```

## JÄMFÖRA STRÄNGAR

- Strängar kan jämföras med avseende på likhet:

```
if namn == "Linda":  
    print("Inte du nu igen...")
```
- 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`

<code>True and True</code>	<code>True</code>
<code>True and False</code>	<code>False</code>
<code>True or True</code>	<code>True</code>
<code>True or False</code>	<code>True</code>
<code>not True</code>	<code>False</code>
<code>not False</code>	<code>True</code>

## VILKET VILLKOR BLIR FALSE?

Givet `ålder=21`, `namn="Mireille"`; vilket av följande villkor blir False?

```
*ålder == 21 and namn == "Mireille"  
*ålder < 22 or namn == "Algot"  
*ålder == 5 or namn > "Mireille"  
*ålder >= 21 and namn > "Algot"
```

## ITERATION

- Iteration betyder upprepning.
- I Python:
  - `while`
  - `for`

## WHILE-SLINGAN

- En `while`-slinga upprepar ett antal satser så länge som ett villkor är uppfyllt.
- Så länge som kannan inte rinner över:
  - Fyll på mer vatten!
- Så länge som du inte har somnat:
  - Räkna ett för till!
- Så länge som du inte gissat rätt tal:
  - Gissa en gång till!

## WHILE - SYNTAX

```
while villkor:  
    sats1  
    sats2  
    ..
```

## 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 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!")
```

## OÄNDLIG SLINGA

- Om villkoret aldrig uppfylls får man en slinga som upprepas i all oändlighet.
- Kan yttra sig som att programmet "hänger sig" - inget händer
- Eller att massor av text rusar förbi på skärmen (om man har utskrift i slingan).
- Avbryt programmet genom att trycka **Ctrl-C** (Ctrl och C samtidigt).

## TALFÖLJDER

Funktionen `range()` ger en följd av heltal.  
`range(10)` ger 0,1,2,3,4,5,6,7,8,9

Man kan också ange både start och slut:  
`range(8,15)` ger 8,9,10,11,12,13,14


Och även steg:  
`range(44,55,3)` ger 44,47,50,53


Funktionen `range` ger en *iterator*.


## UPPGIFT: UDDA TAL


Vi vill ha alla udda tal mellan 0 och 10.

Hur ska vi anropa `range`?

 `range(0,9,1)`

 `range(0,10,2)`

 `range(1,9,2)`

 `range(1,10,2)`

## 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

## 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)
```

 15

 10

 5

 0

## 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