NTNU
Det skapende universitet

TDT4110 Informasjonsteknologi grunnkurs:
Tema: Betingelser og logiske uttrykk

- 3rd edition: Kapittel 3

Professor Alf Inge Wang

if (be):
…
else (not_to_be):
…

NTNU
Det skapende universitet

Læringsmål og pensum

- Mål
  - Lære å bruke og forstå if-setninger
  - Lære å bruke og forstå sammenlikning av strenger
  - Lære å bruke og forstå nestede beslutningsstrukturer
  - Lære å bruke og forstå logiske operatorer
  - Lære å bruke og forstå boolske variabler
- Pensum: Decision Structures and Boolean Logic
  - Starting out with Python 3rd edition: Chapter 3

NTNU
Det skapende universitet

4

if-setningen

Kapittel 3.1

NTNU
Det skapende universitet

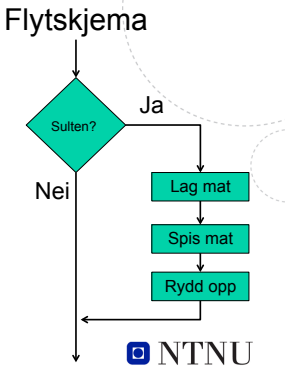
www.ntnu.no

5

If-setningen

- If-setninger brukes til å bestemme at spesielle deler av programmet utføres avhengig av verdier på variabler (kontrollstruktur).
- If-setning brukes typisk til å utføre en kodeblokk hvis noen betingelser er sanne.
- Kode:

```
if betingelse:
    uttrykk
    uttrykk
    etc.
```



NTNU
Det skapende universitet

www.ntnu.no

6

Betingelser i Python

- Python har spesielle tegn eller *operatorer* for å uttrykke betingelser
- På lik linje med de *aritmetiske* operatorene +, -, *, /
- Mest mulig lik tegnene vi kjenner fra matematikken
- Kalles også *relasjonsoperatorer*

Python	Matematikk	Forklaring
<	<	Mindre enn
>	>	Større enn
<=	≤	Mindre eller lik
>=	≥	Større eller lik
==	=	Er lik
!=	≠	Er ulik

NTNU
Det skapende universitet

www.ntnu.no

Eksempler på betingelser

Betingelse	Verdi
4 < 3	usann (False)
4==4	sann (True)
3!=3	usann (False)
3>=3	sann (True)

NTNU
Det skapende universitet

www.ntnu.no

OBS! Her er det lett å blande!

- Det er lett å blande sammen tilordning og evaluering av logiske uttrykk:
- Dette er tilordning: **`a=5`**
  - *Tilordner* variabelen *a* verdien 5 (lagrer tallet 5 i variabelen *a*)
- Dette er evaluering: **`a==5`**
  - *Tester* om variabelen *a* har verdien 5
- Python vil si ifra hvis du prøver å gjøre en tilordning når du skal teste en om en variabel har en verdi

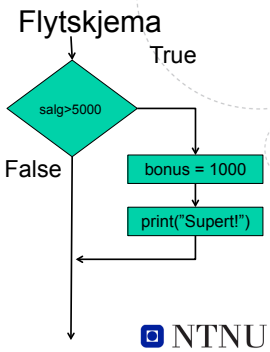
NTNU
Det skapende universitet

www.ntnu.no

If-setningen

- Eksempel på if-setning:

```
if salg>50000:
      bonus = 1000
      print("Supert!")
```



NTNU
Det skapende universitet

www.ntnu.no

10

Nøstede blokker

```
def main():
    # Ett innrykk
    poeng1 = int(input("Poeng spiller 1: "))
    poeng2 = int(input("poeng spiller 2: "))
    if poeng1 > poeng2:
        # To innrykk
        print("Spiller 1 vinner!!!!")

#Kall main-funksjonen
main()
```

blokken for main-funksjonen

blokken for if-setning

NTNU
Det skapende universitet

11

if-else uttrykk

Kapittel 3.2

NTNU
Det skapende universitet

12

if-else uttrykk

- Et if-else uttrykk vil kjøre en blokk av kode hvis betingelsen er sann (True) og en annen blokk av kode hvis betingelsen er usann (False).
- if-else uttrykk skal brukes i koden der det er to mulige alternativer av kode som skal utføres.

NTNU
Det skapende universitet

13

if-else generell kode

```
if betingelse:
      kode
      kode
      etc.
else:
      kode
      kode
      etc.
```

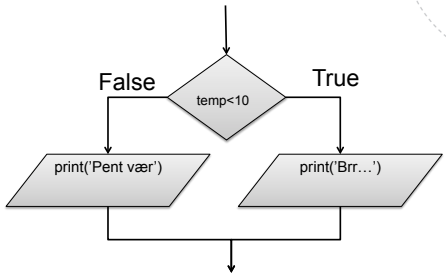
Denne kodeblokka blir utført hvis betingelsen er **sann**

Denne kodeblokka blir utført hvis betingelsen er **usann**

NTNU
Det skapende universitet

14

if-else flytskjema



NTNU
Det skapende universitet

15

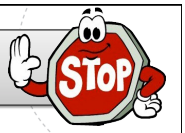
if-else kodeeksempel

```
temp = int(input("Temperaturen er :"))
if temp<10:
      # Betingelsen er sann
      # Innrykk for kodeblokka
      print("Brrr…")
else:
      # Betingelsen er usann
      print("Pent vær")
```

NTNU
Det skapende universitet

16

Oppgave: if else



- Skriv koden til følgende:
  - Spør brukeren om pulsen
  - Hvis puls er høyere eller lik 80:
    - Skriv til konsoll: *Ro deg ned!*
  - Hvis puls er lavere enn 80:
    - Skriv til konsoll: *Bare slapp av*

NTNU
Det skapende universitet

17

Sammenlikne strenger

Kapittel 3.3

NTNU
Det skapende universitet

18

Sammenlikning av to variabler som inneholder strenger

- Variabler som inneholder tekststrenger kan sammenlignes på lik linje med tall.
- Eksempel på å sjekke om to variabler er like:

```
navn1 = 'Peter'
navn2 = 'Pelle'
if navn1==navn2:
    print("Samme navn!")
else:
    print("Forskjellige navn")
```

NTNU
Det skapende universitet

19

Sammenlikning av en variabel og en tekststreng

- Man kan også sjekke om en variabel med tekststreng er lik en spesifisert tekst:

```
passord = input("Skriv inn passord: ")

if passord == "Nuff":
    print("Riktig passord")
else:
    print("Feil passord")
```

NTNU
Det skapende universitet

20

Sjekke om en streng er større enn en annen streng

- I Python kan du også sjekke om en streng er større (eller mindre) enn en annen streng.
  - Dvs. at en tekststreng har tegn som er representert med mindre eller større verdier enn i den andre strengen.
  - Alle tegn i Python representerer en tallverdi

```
if "A" < "B":
    print("Bokstaven A er mindre enn bokstaven B")
    # Bokstaven A representeres med tallet 65 og B med 66
```

NTNU
Det skapende universitet

21

ASCII tabellen – tegn representert som tall

1	☺	33	!	65	A	97	a	129	ü	161	í	193	┴	225	ß
2	☻	34	"	66	B	98	b	130	é	162	ó	194	┬	226	Γ
3	♥	35	#	67	C	99	c	131	â	163	ú	195	├	227	π
4	♦	36	$	68	D	100	d	132	ä	164	ñ	196	─	228	Σ
5	♣	37	%	69	E	101	e	133	à	165	Ñ	197	┼	229	σ
6	♠	38	&	70	F	102	f	134	å	166	ª	198	╞	230	µ
7	•	39	'	71	G	103	g	135	ç	167	º	199	╟	231	τ
8	◘	40	(	72	H	104	h	136	ê	168	¿	200	╚	232	Φ
9	○	41	)	73	I	105	i	137	ë	169	®	201	╔	233	Θ
10	◙	42	*	74	J	106	j	138	è	170	¬	202	╩	234	Ω
11	♂	43	+	75	K	107	k	139	ï	171	½	203	╦	235	δ
12	♀	44	,	76	L	108	l	140	î	172	¼	204	╠	236	∞
13	♪	45	-	77	M	109	m	141	ì	173	¡	205	═	237	φ
14	♫	46	.	78	N	110	n	142	Ä	174	«	206	╬	238	ε
15	☼	47	/	79	O	111	o	143	Å	175	»	207	╧	239	∩
16	►	48	0	80	P	112	p	144	É	176	░	208	╨	240	≡
17	◄	49	1	81	Q	113	q	145	æ	177	▒	209	╤	241	±
18	↕	50	2	82	R	114	r	146	Æ	178	▓	210	╥	242	≥
19	‼	51	3	83	S	115	s	147	ô	179	│	211	╙	243	≤
20	¶	52	4	84	T	116	t	148	ö	180	┤	212	╘	244	⌠
21	§	53	5	85	U	117	u	149	ò	181	╡	213	╒	245	⌡
22	▬	54	6	86	V	118	v	150	û	182	╢	214	╓	246	÷
23	↨	55	7	87	W	119	w	151	ù	183	╖	215	╫	247	≈
24	↑	56	8	88	X	120	x	152	ÿ	184	╕	216	╪	248	°
25	↓	57	9	89	Y	121	y	153	Ö	185	╣	217	┘	249	∙
26	→	58	:	90	Z	122	z	154	Ü	186	║	218	┌	250	·
27	←	59	;	91	[	123	{	155	ø	187	╗	219	█	251	√
28	∟	60	<	92	\	124	|	156	£	188	╝	220	▄	252	ⁿ
29	↔	61	=	93	]	125	}	157	Ø	189	╜	221	▌	253	²
30	▲	62	>	94	^	126	~	158	×	190	╛	222	▐	254	■
31	▼	63	?	95	_	127	⌂	159	ƒ	191	┐	223	▀	255	
32		64	@	96	`	128	Ç	160	á	192	└	224	Ó		

NTNU
Det skapende universitet

22

Sammenlikning av to strenger

- Hva skjer her? Hva sammenliknes?
  - Sjekker bokstav for bokstav!

M	a	r	y
77	97	114	121
↕	↕	↕	↕
M	a	r	k
77	97	114	107

```
navn1 = "Mary"
navn2 = "Mark"
if navn1 > navn2:
      print("Mary er større enn Mark")
else:
      print("Mary er ikke større enn Mark")
```

NTNU
Det skapende universitet

23

Eksempel: sjekk strenger

- Skal lage et program hvor bruker kan skrive inn to tekststrenger.
- Programmet skal så sjekke om strengene er like eller om en kommer før i alfabetet enn den andre og kommenterer resultatet til skjerm.

`streng_sjekk.py`

NTNU
Det skapende universitet

24

Nøstede betingelser og if-elif-else

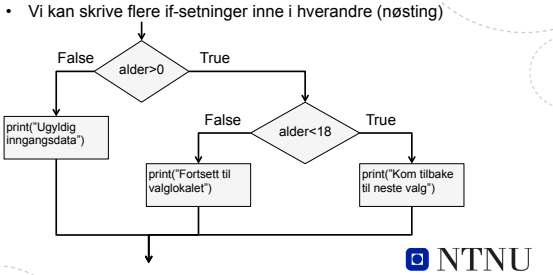
Kapittel 3.3

NTNU
Det skapende universitet

25

Flytskjema for nøstede av if-setninger

- Vi kan skrive flere if-setninger inne i hverandre (nøsting)



NTNU
Det skapende universitet

26

Nøsting av if-setninger

- Kodeeksempel på *nøste* if-setninger:

```
if (alder <0):
       print(”Ugyldige inngangsdata”)
else:
       if (alder <18):
              print(”Kom tilbake til neste valg”)
       else:
              print(”Forsett til valglokalet”)
```

- Merk at du må bruke *innrykk* for å si at de indre setningene hører til if-setningen. En if-setning avsluttes ved å fjerne innrykk.
- Kan ha flere nivåer med if-setninger inne i hverandre.

NTNU
Det skapende universitet

27

Bruk av if-elif-else

- Nøsting av setninger kan fort bli uoversiktlig
- Python har derfor `elif` for bedre lesbarhet.
  - elif er en forkortelse for **else if** (hvis ikke det ovenfor slår til, så….)

```
if (poeng>=89):
       karakter = ’A’
elif (poeng>=77):
       karakter = ’B’
elif (poeng>=65):
       karakter = ’C’
elif (poeng>=53):
       karakter = ’D’
elif (poeng>=41):
       karakter = ’E’
else:
       karakter = ’F’
```

- NB! Kun en av betingelsene vil slå til!

NTNU
Det skapende universitet

28

Logiske operatorer

Kapittel 3.5

NTNU
Det skapende universitet

29

Logiske uttrykk

- På samme vis som vi har sammensatte *aritmetiske* uttrykk kan vi sette sammen betingelser til vilkårlig store uttrykk
- Dette kaller vi *logiske uttrykk*
- Vi kaller "limet" som binder disse sammen for *logiske operatorer*
- Python definerer de følgende logiske operatorene slik:

Operator i Python	Forklaring
and	Logisk og
or	Logisk eller
not	Logisk ikke, eller negasjon

NTNU
Det skapende universitet

30

Logiske uttrykk (fortsettelse)

- Hva betyr `and`, `or` og `not` i praksis:
  - Et uttrykk med `and` blir True hvis begge sider er True:

```
False and True      gir  False
False and False     gir  False
True and True       gir  True
```

  - Et uttrykk med `or` blir True hvis minst en av sidene er True:

```
False or True       gir  True
True or True        gir  True
False or False      gir  False
```

  - `not` snur uttrykket:

```
not True            gir  False
not False           gir  True
```

NTNU
Det skapende universitet

31

Eksempel på logiske uttrykk

- Vanlig bruk er å sjekke at en verdi ligger i et intervall:

```
x >= 5 and x <= 10
```

- I Python kan man også sjekke intervaller på følgende måte (fungerer sjeldent i andre programmeringsspråk):

```
5<= x <= 10
```

- Vi lager større uttrykk og sjekke flere betingelser ved bruk av paranteser:

```
(i>=1 and i<=N) or(j>=1 and j<=N)
```

NTNU
Det skapende universitet

32

Resultatet av logiske utregninger

- En enkel eller sammensatt betingelse kalles et *logisk uttrykk*
- Resultatet av en logisk beregning (evaluering) er enten `True` eller `False` i Python
- … men IKKE `sann` eller `usann`!

NTNU
Det skapende universitet

33

For å få riktig betingelser husk Pythons Operatorhierarki:

```
1. ( )
2. **                 # Eksponent (opphøyd)
3. *, /, //, %        # heltallsdivisjon, rest
4. +, -
5. <, <=, >, >=, <>, !=, ==
6. not
7. and
8. or
9. if - else
10.Lik prioritet: fra venstre mot høyre
```

- Bruk parenteser for å få uttrykkene riktig

NTNU
Det skapende universitet

34

Oppgave: Dørvaktsimulator

- Det skal lages et dørvaktsimulator program som skal benytte seg av: print, input(), int(), if og else.
- Programmet skal skrive til skjerm ”Hei jeg er THE DOORMAN!” og så spørre etter navn, alder og om du er full (ja eller nei)
  - Hvis du er gammel nok (18+), så skal det skrives ut ”Du er gammel nok”.
  - Hvis du ikke er gammel nok, skal det skrives ut ”Du er for ung ”+ navnet
  - Hvis du er gammel nok, men er også full skal det skrives ut ”Du slipper ikke inn for du er full!”
  - Hvis du er gammel nok og ikke full, skal det skrives ut ”Velkommen inn ”+ navnet

doorman.py

NTNU
Det skapende universitet

35

Betingelser

- Oppgave: Er denne betingelsen **true** eller **false**?
- `4<7 and not(3>1 or 8>=9)`
- `True and not(True or False)`
- `True and not(True)`
- `True and False`
- `False`

- Begynn innenfra og jobb utover (inne i parenteser)

NTNU
Det skapende universitet

36

Boolske variabler

Kapittel 3.6

NTNU
Det skapende universitet

37

Boolske variabler

- EN boolsk variabel kan referere til en av to verdier: True eller False.
- Boolske variabler brukes typisk som flagg for å lagre at en spesiell betingelse er sann eller ikke.

```
riktig = False
svar = input('Skriv svar: ')
if (svar='tulling'):
    riktig = True
…
# sjekker om riktig == True lengre nede i programmet
```

NTNU
Det skapende universitet

38

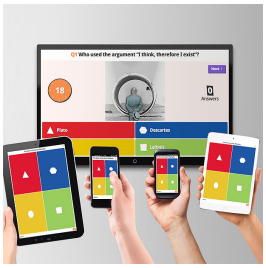
Avslutning if-setninger

- Unngå overflødig bruk av negasjoner (`not`) – tungt å lese
- Ved `if else`, skriv helst positiv utfall i `if` og negativt i `else`
- Vi kan ha flere setningen mellom `if` … `else`

NTNU
Det skapende universitet

39

Oppgave: True or False



- https://play.kahoot.it/#/k/1f6d063e-9f5e-40e1-[illegible]ec21d8f0fec9

NTNU
Det skapende universitet

40

Oppsummering

- Betingelser i Python: `<` , `>` , `<=` , `>=` , `==` , `!=`, `<>`
- Operatorer for logiske uttrykk: `and`, `or` , `not`
- Logiske uttrykk kan enten bli `False` eller `True`
- if-setninger:

```
if (<betingelse>):
      <utfør noe>          # HUSK INNRYKK!
elif (<betingelse):
      <utfør noe>          # HUSK INNRYKK!
else:
      <utfør noe annet>
```

- Vi kan også bruke nøstede if-setninger

NTNU
Det skapende universitet

41

Problemer med æ,ø, å på PyCharm Windows

- Kan fikses med å legge til dette i starten av programmet:

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
```

NTNU
Det skapende universitet