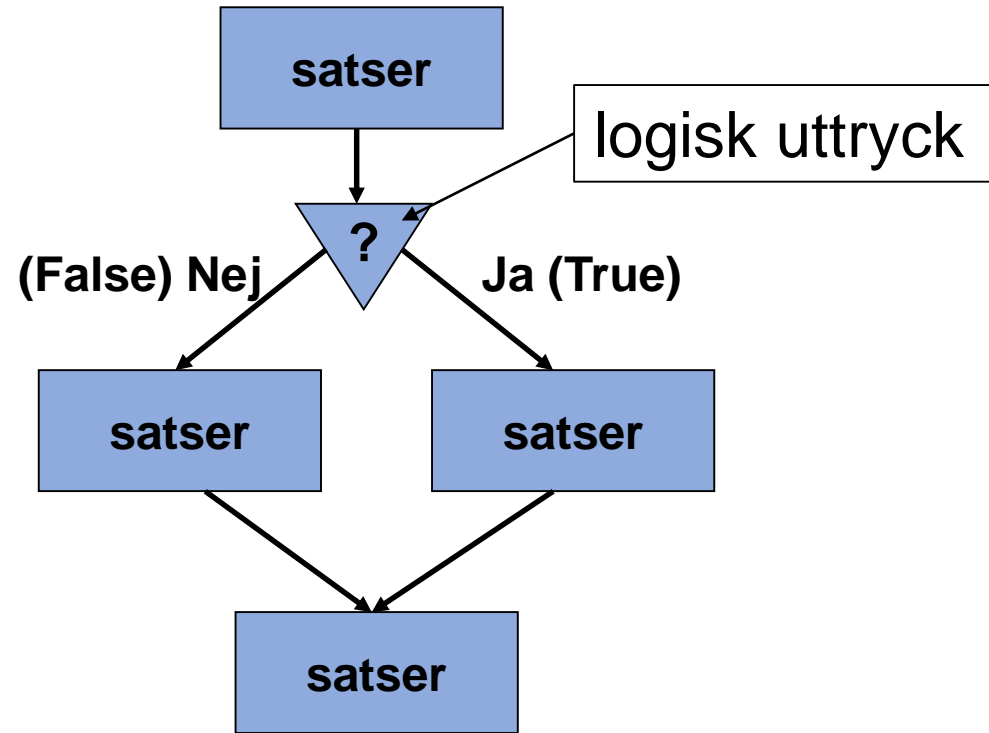


Villkor

Programmet väljer väg...

Om ... gör programmet så
annars gör det så



jämförelseoperator

Logiskt uttryck: $(x > 2)$ and $(x < 7)$ Är True eller False

logisk operator

if-satsen

if med else:

```
ålder = int(input('Ålder?'))  
if ålder >= 18:  
    print('Du är myndig')  
else:  
    print('Du är inte myndig')  
print('Efter if-satsen...')
```

Ålder?20

Du är myndig
Efter if-satsen...

Ålder?16

Du är inte myndig
Efter if-satsen...

if utan else:

```
ålder = int(input('Ålder?'))  
if ålder >= 18:  
    print('Du är myndig')  
print('Efter if-satsen...')
```

Ålder?23

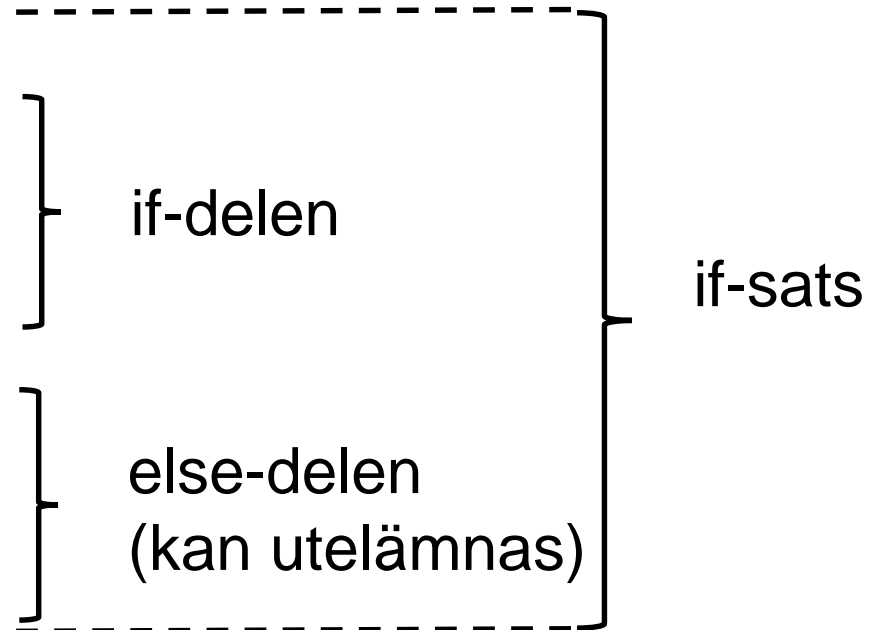
Du är myndig
Efter if-satsen...

Ålder?15

Efter if-satsen...

Kan lösas med två if-satser i sekvens också...

```
ålder = int(input('Ålder?'))
if ålder >= 18:
    print('Du är myndig')
    # Här kan man ha flera satser
    # ...
else:
    print('Du är inte myndig')
    # Här kan man ha flera satser
    # ...
# Sats efter if-satsen
```



if-delens och else-delens satser måste dras in, kallas *indentering*, annars fås ett felmeddelande.

Notera `if ålder >= 18:`

Är alltid ett logiskt uttryck,
som är `True` eller `False`

elif (else if)

Om man har fler alternativ än två.

```
ålder = int(input('Ålder?'))
if ålder >= 18:          # Villkor 1
    biljettPris = 50
elif ålder >=12:       # Villkor 2
    biljettPris = 30
elif ålder >=7:        # Villkor 3
    biljettPris = 20
else:                  # Annars om ingen av villkoren 1, 2, el 3 är True
    biljettPris = 0
print('Biljettpris = ', biljettPris)
```

Om man matar in `ålder = 25` testas ett villkor innan `biljettPris` får sitt värde.

Om man matar in `ålder = 9` testas tre villkor innan `biljettPris` får sitt värde.

Nästlade if-satser

Beräkna max av tre variabler: x, y, z. Antag att de har blivit tilldelade varsitt värde.

Alt 1:Nästlade if-satser

```
if x > y:
    if x > z:
        max = x
    else:
        max = z
elif y > z:
    max = y
else:
    max = z
```

Är något alternativ bättre än de andra?

Alt 2

```
if (x > y) and (x > z):
    max = x
if (y > x) and (y > z):
    max = y
if (z > x) and (z > y):
    max = z
```

Alt 3


```
max = x      # En hypotes
if y > max:
    max = y
if z > max:
    max = z
```

Logiska uttryck byggs upp av jämförelseoperatorer och logiska operatorer

Jämförelseoperatorer

		Exempel
==	Lika med	$x == 4$
!=	Icke lika med	$x != 4$
<	Mindre än	$x < 4$
<=	Mindre än eller lika med	$x <= 4$
>	Större än	$x > 4$
>=	Större än eller lika med	$x >= 4$

Logiska operatorer

		Exempel
and	och	$(x >= 1) \text{ and } (x <= 10)$
or	eller	$(x < y) \text{ or } (x > z)$
not	icke	$\text{not } (x != 4)$ 

Typen bool

Ett logiskt uttryck har typen `bool`. Värdet kan vara `True` eller `False`.

```
q = 4
print(q>9)           # skriver ut: False
print(2<q<5)        # skriver ut: True

x = q > 9            # x får värdet False (typen bool)
y = 7                # y får värdet 7 (typen int)
z = x and (y==7)    # z får värdet False (typen bool)
print(x,y,z)        # skriver ut: False 7 False
```

Variabler kan vara av typen bool.

```
ålder = 19  
myndig = ålder >= 18  
# myndig får värdet True
```

```
tonåring = (ålder >= 13 and (ålder <= 19))  
# tonåring får värdet True
```

```
x = 3  
likaMedTre = x == 3  
# likaMedTre får värdet True
```


Kortform av if-sats "one liners"

```
x = 14
```

```
y = 15
```

```
print('a') if x > y else print('=') if x == y else print('b')
```