IT Essentials

Hoofdstuk 3

Condities

DE HOGESCHOOL
MET HET NETWERK

Hogeschool PXL – Elfde-Liniestraat 24 – B-3500 Hasselt
www.pxl.be - www.pxl.be/facebook

Inhoud

0. Inleiding
1. Boolean expressies
   - Booleans, vergelijkingen, in operator, logische operatoren
2. Conditionele statements
   - Blokken code, inspringen, twee-weg beslissingen, stroomdiagrammen, meer-weg beslissingen, geneste condities

3.0 Inleiding

Een programma bestaat meestal uit:

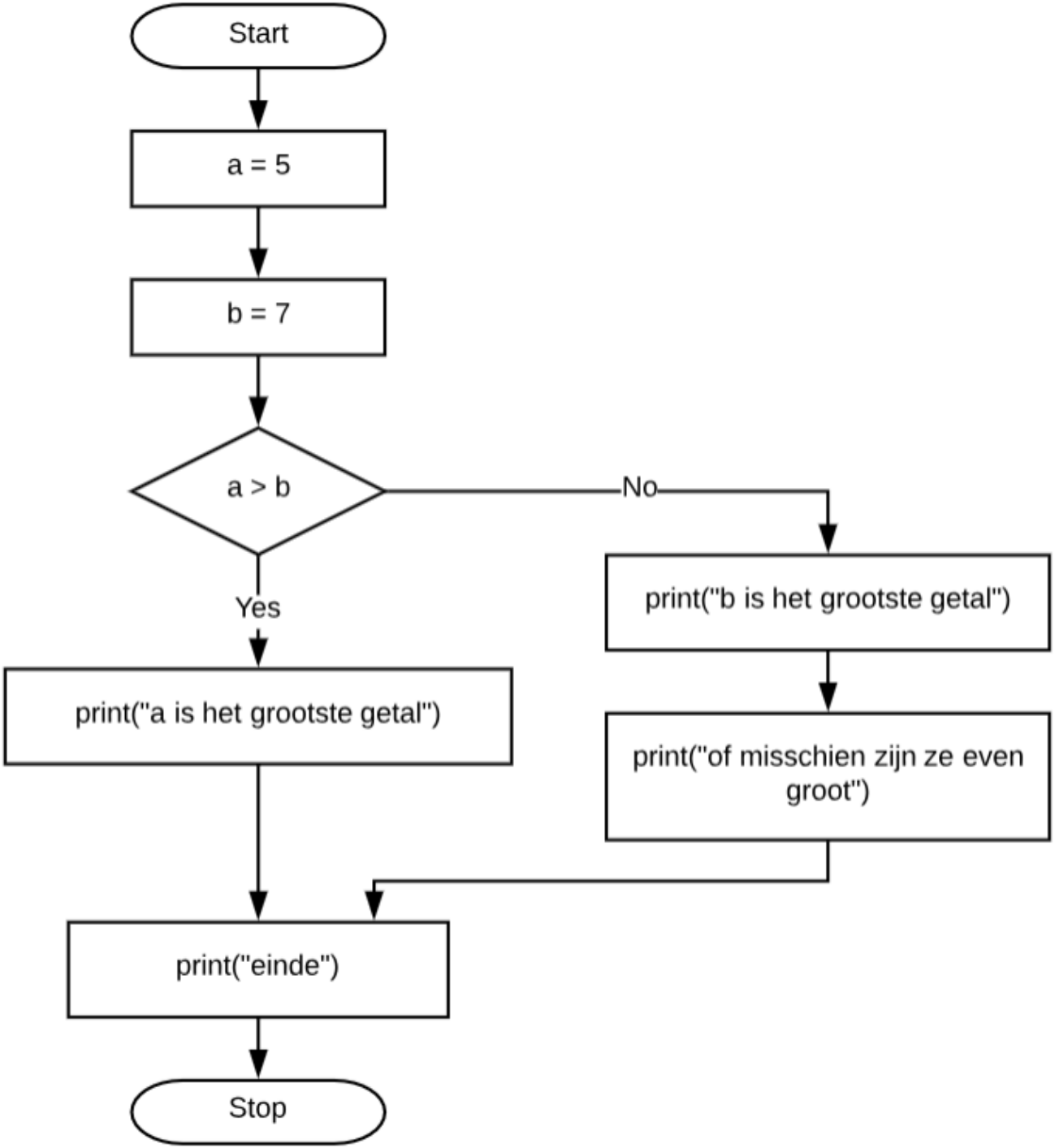
- Opeenvolgende instructies
- Keuzes: voorwaardelijke instructies (H3)
- Herhalingen: herhaaldelijke uitvoering van instructies (H4)
- Invoer: gegevens opvragen
- Uitvoer: gegevens ter beschikking stellen

Condities → keuzestructuur

Voorbeeld:

Gegeven: 2 getallen

Gevraagd: Druk het grootste af


Start
a = 5
b = 7
a > b
No
Yes
print("b is het grootste getal")
print("a is het grootste getal")
print("of misschien zijn ze even groot")
print("einde")
Stop

In Python

Output??

```
a = 5
b = 7
if a > b:
    print("a is het grootste getal")
else:
    print("b is het grootste getal")
    print("of misschien zijn ze even groot")
print("einde")
```

If = conditioneel statement

→ Een test (een boolean expressie)

→ 1 of meerdere acties

3.1 Boolean expressies

- Een test = een boolean expressie
- De acties worden alleen uitgevoerd als de test evalueert als zijnde “waar”

In Python:

Waar = True

Onwaar = False

3.1 Boolean expressies

3.1.1 Booleans

Een expressie die evalueert naar True of False is een “boolean expressie”.

True en False zijn Boolean waardes.

3.1 Boolean expressies

3.1.2 Vergelijkingen

Een vergelijking bestaat uit 2 waardes met een vergelijkingsoperator ertussen, vb a > b

Relationele operator	Betekenis
>	groter dan
>=	groter of gelijk aan
<	kleiner dan
<=	kleiner of gelijk aan
==	gelijk aan
!=	niet gelijk aan

Zowel voor getallen als voor tekst!

3.1 Boolean expressies

3.1.2 Vergelijkingen

Let op het verschil tussen = en ==

=	toekenning	vb a = 5
==	vergelijkingsoperator	vb if a == 5:

3.1 Boolean expressies

3.1.2 Vergelijkingen

Voorbeelden:

```
print("1. ", 9 < 10 )
print("2. ", 9 <= 10)
print("3. ", 9 > 10)
print("4. ", 9 == 9.0)
print("5. ", 9 == "9")
print("6. ", "Jan" == "jan")
print("7. ", "Jan" == "Jan ")
print("8. ", "Jan" > "Annita")
print("9. ", "Jan" > "annita")
print("10. ", "Jan" > "123")
```

Output??

1. True
2. True
3. False
4. True
5. False
6. False
7. False
8. True
9. False
10. True

3.1 Boolean expressies

3.1.2 Vergelijkingen

Voorbeelden:

```
print("5. ", 9 == "9")
```

Bij test op gelijkheid mag je tekst en getallen door elkaar gebruiken

```
print("11. ", "Jan" > 123)
```

ERROR!!

Bij test op ongelijkheid kan dat niet!!!!

Opgave 3.1

Wat is de uitkomst van:

```
7 < 21
"7" < "21"
```

Leg uit!

3.1 Boolean expressies

3.1.2 Vergelijkingen

Uitkomst van een boolean expressie toekennen aan een variabele:

```
antwoord = "x"
juist = antwoord == "X"
print(juist)
juist = antwoord > "X"
print(juist)
print(type(juist))
```

Output??

```
False
True
<class 'bool'>
```

Opgave 3.2

Schrijf code die test of volgende waarden gelijk zijn aan mekaar, of dat de eerste de grootste is of niet:

- 1/10 en 0.10
- 1/3 en 0.33
- (1/3)*3 en 1

3.1 Boolean expressies

3.1.3 in operator

test of een waarde voorkomt in een collectie
(voorlopig enkel de string)

Output??

```
print("e" in "IT essentials")          True
print("E" in "IT essentials")          False
print("ess" in "IT essentials")        True
print("ese" in "IT essentials")        False
print("e" not in "IT essentials")      False
```

Opgave 3.3

Schrijf code die test of volgende tekens of combinatie van tekens voorkomt in je volledige naam:

- e
- x
- Van

3.1 Boolean expressies

3.1.4 Logische operatoren

Twee of meer boolean expressies kunnen gecombineerd worden m.b.v. logische operatoren

Logische operator	Resultaat
and	true als beide expressies true zijn
or	true als 1 van beide expressies true is
not	is het tegenovergestelde

3.1 Boolean expressies

3.1.4 Logische operatoren

Stel: t = True
f = False

t and t	True
t and f	False
f and t	False
f and f	False
t or t	True
t or f	True
f or t	True
f or f	False
not f	True
not t	False

3.1 Boolean expressies

3.1.4 Logische operatoren

Let op bij het combineren van meerdere **and**s en **or**s!!

Gebruik van haakjes is aangewezen!

- Minder kans op fouten
- Maakt je code leesbaarder

3.1 Boolean expressies

3.1.4 Logische operatoren

Voorbeeld

```
x + 2 < y or y == 40 and a * 2 > 5 or b > 3
```

Beter:

```
x + 2 < y or (y == 40 and a * 2 > 5) or b > 3
```

3.1 Boolean expressies

3.1.4 Logische operatoren

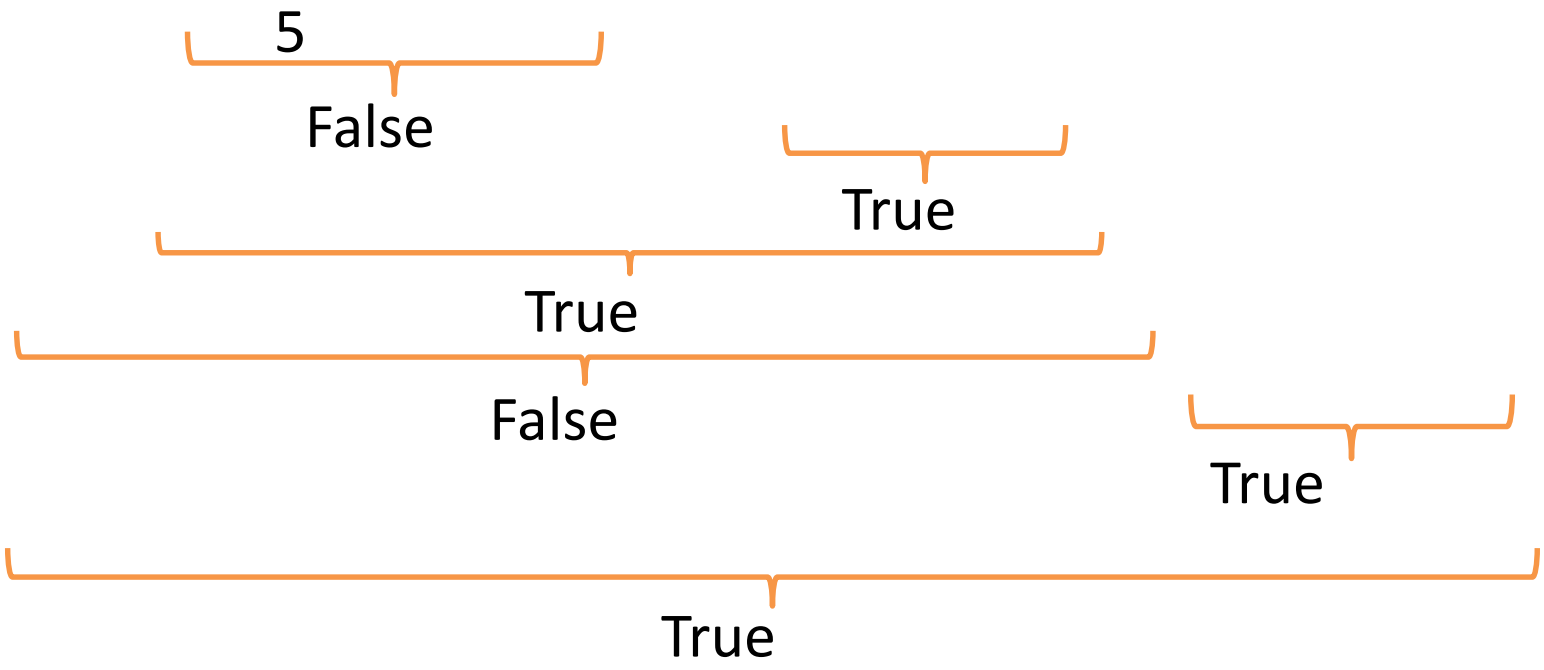
Let op bij het combineren van meerdere **and**s en **or**s!!

Prioriteitsregels: van hoog → laag

- haakjes ()
- numerieke operatoren
- vergelijkingsoperatoren
- not
- and
- or

Voorbeeld:

```
not(2 + 3 > 5 or 9 == 9) or 4 > 3
```



➔ Het resultaat van deze uitdrukking is True.

Opgave 3.4

Geef voor de code hieronder waardes voor a, b en c, die ertoe leiden dat de 2 expressies verschillende uitkomsten hebben:

a = # True of False?

b = # True of False?

c = # True of False?

```
print((a and b) or c)
print(a and (b or c))
```

3.1 Boolean expressies

3.1.4 Logische operatoren

Een logische expressie met enkel **and**s of enkels **or**s:

- Haakjes zijn niet nodig
- Expressie wordt van links naar rechts geëvalueerd.
  Python stopt de evaluatie op het moment dat de uitkomst bekend is!!

- Vb1: Bij de and-operator is de voorwaarde True als beide voorwaarden True zijn.

  a > 0 and b / a > 5

  Als de eerste voorwaarde False is, heeft het in dit geval geen zin om de 2e voorwaarde nog na te kijken. Als je de and-operator gebruikt, wordt de 2e voorwaarde alleen nagegaan als de 1e True is.

- Vb2: Bij de or-operator is de voorwaarde True als minstens 1 van beide voorwaarden True is.

  a < 0 or a > 100

  Bij de or-operator wordt de 2e voorwaarde alleen nagegaan als de 1e voorwaarde False is.

3.1 Boolean expressies

3.1.4 Logische operatoren

Voorbeeld

Stel x=30, y=40, a=0 en b=5

Zijn volgende uitdrukkingen True of False:

- x + 2 < y or a == 0 or b / a > 5 (nuldeling??)
- x + y < 100 and a == 9 and b > 3

Volgorde van je condities is in dit geval wel belangrijk!!

3.1 Boolean expressies

3.1.4 Logische operatoren

Voorbeeld

Stel x=30, y=40, a=10 en b=5

Zijn volgende uitdrukkingen True of False:

- x + 12 < y or not (y == 40 and a * 2 > 5)
- not (x + y < 10 or a == 9) and b > 3
- x + y != 70 or (a == 1 and b > 3)

3.2 Conditionele statements

conditioneel statement = if statement

→ Een test (een boolean expressie)

→ 1 of meerdere acties die enkel worden uitgevoerd als de test True oplevert

Voorbeeld

```
x = 5
if x == 5:
    print("x is 5")
```

3.2 Conditionele statements

```
if <boolean expressie>:
    <acties>
```

Let op:

- : achter de boolean expressie
- Inspringing!

3.2 Conditionele statements

3.2.1 Blokken code

Na de if, gaan de bijhorende <acties> inspringen

- Blok code = opeenvolgende statements die hetzelfde niveau van inspringing hebben

Voorbeeld

```
x = 5
if x == 5:
    print("dit wordt gedrukt als x 5 is")
    print("x is dus 5")
print("deze regel wordt altijd uitgevoerd")
```

3.2 Conditionele statements

3.2.1 Blokken code

Meerdere if-statements zijn uiteraard mogelijk

- In onderstaand vb is elke if onafhankelijk van de vorige
  Waarom?

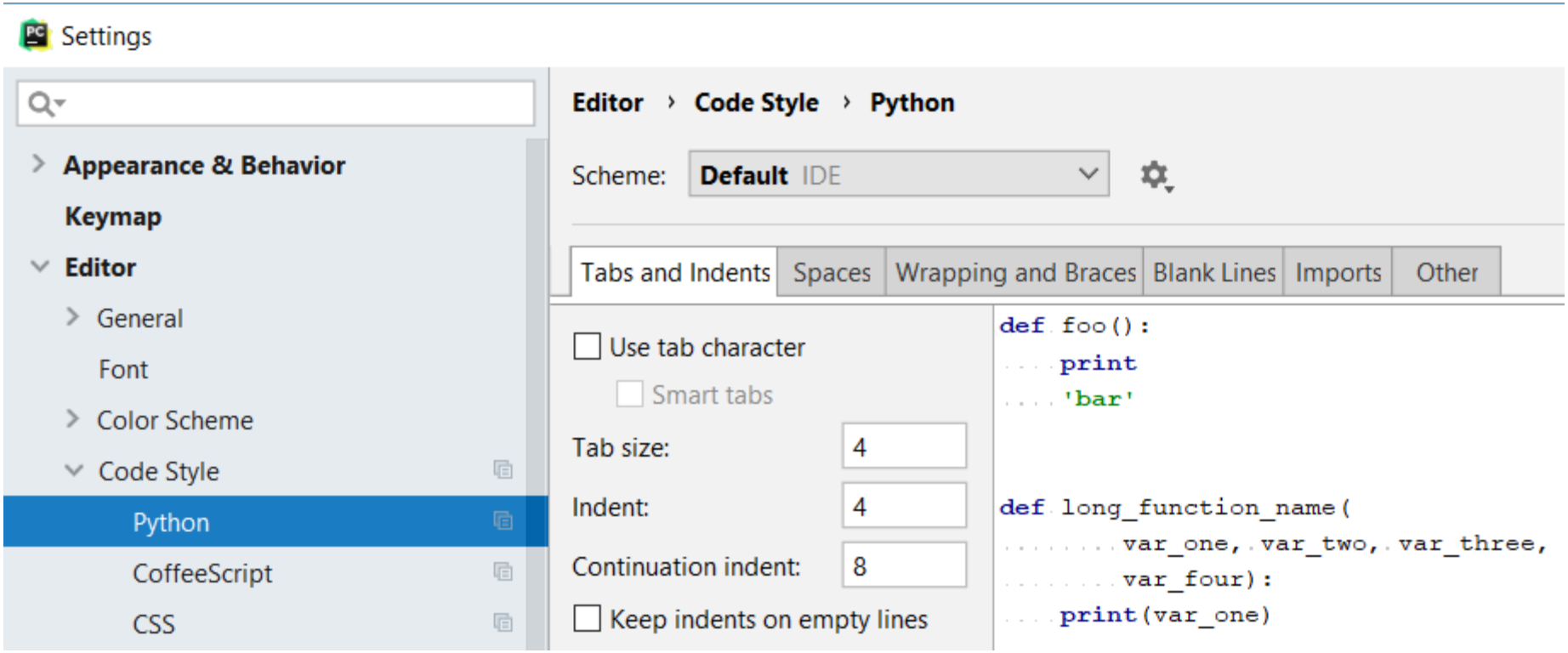
Voorbeeld

```
leeftijd = 24
if leeftijd < 25:
    print("als je nog studeert, kom je in aanmerking voor kindergeld")
if leeftijd >= 15:
    print("vanaf 15 mag je een vakantiejob uitoefenen")
if leeftijd >= 16:
    print("je mag bier drinken")
if leeftijd >= 18:
    print("je bent meerderjarig")
if leeftijd >= 65:
    print("je komt in aanmerking voor seniorenkorting")
```

3.2 Conditionele statements

3.2.2 Inspringen

- Van het allergrootste belang!!
- 4 spaties
- Meeste editors doen aan auto-indenting
- Tab wordt normaal automatisch vervangen door 4 spaties



Tabs and Indents

Item	Description
Use tab character	– If this checkbox is selected, tab characters are used: – On pressing the Tab key – For indentation – For code reformatting – When the checkbox is cleared, PyCharm uses spaces instead of tabs.

Opgave 3.7

```
# Deze code bevat tabulatie-fouten!
x = 3
y = 4
if x == 3 and y == 4:
    print("x is 3")
   print("y is 4")
if x > 2 and y < 5:
print("x > 2")
print("y < 5")
if x < 4 and y > 3:
    print("x < 4")
        print("y > 3")
```

Wat zijn de waarden van a en b na uitvoeren van volgende statements?

```
a = 5
b = 3
if a < b:
    a = 2 * a
    b = a + b
```

```
a = 3
b = 5
if a < b:
    a = 2 * a
    b = a + b
```

```
a = 5
b = 3
if a < b:
    a = 2 * a
b = a + b
```

```
a = 3
b = 5
if a < b:
    a = 2 * a
b = a + b
```

3.2 Conditionele statements

3.2.3 Twee-weg beslissingen

Voorbeeld

```
leeftijd = 19
if leeftijd < 18:
    print("je bent minderjarig")
else:
    print("je bent meerderjarig")
```

3.2 Conditionele statements

3.2.3 Twee-weg beslissingen

```
if <boolean expressie>:
    <acties>
else:
    <acties>
```

Let op:

- : achter de else
- else moet uitgelijnd zijn met de bijhorende if
- Sowieso wordt 1 vd actie-blokken uitgevoerd

Opgave 3.8

Schrijf code die vraagt om een integer en dan rapporteert of de integer even of oneven is

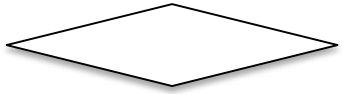
3.2 Conditionele statements

3.2.4 Stroomdiagrammen

= visuele weergave van alle statements in je programma

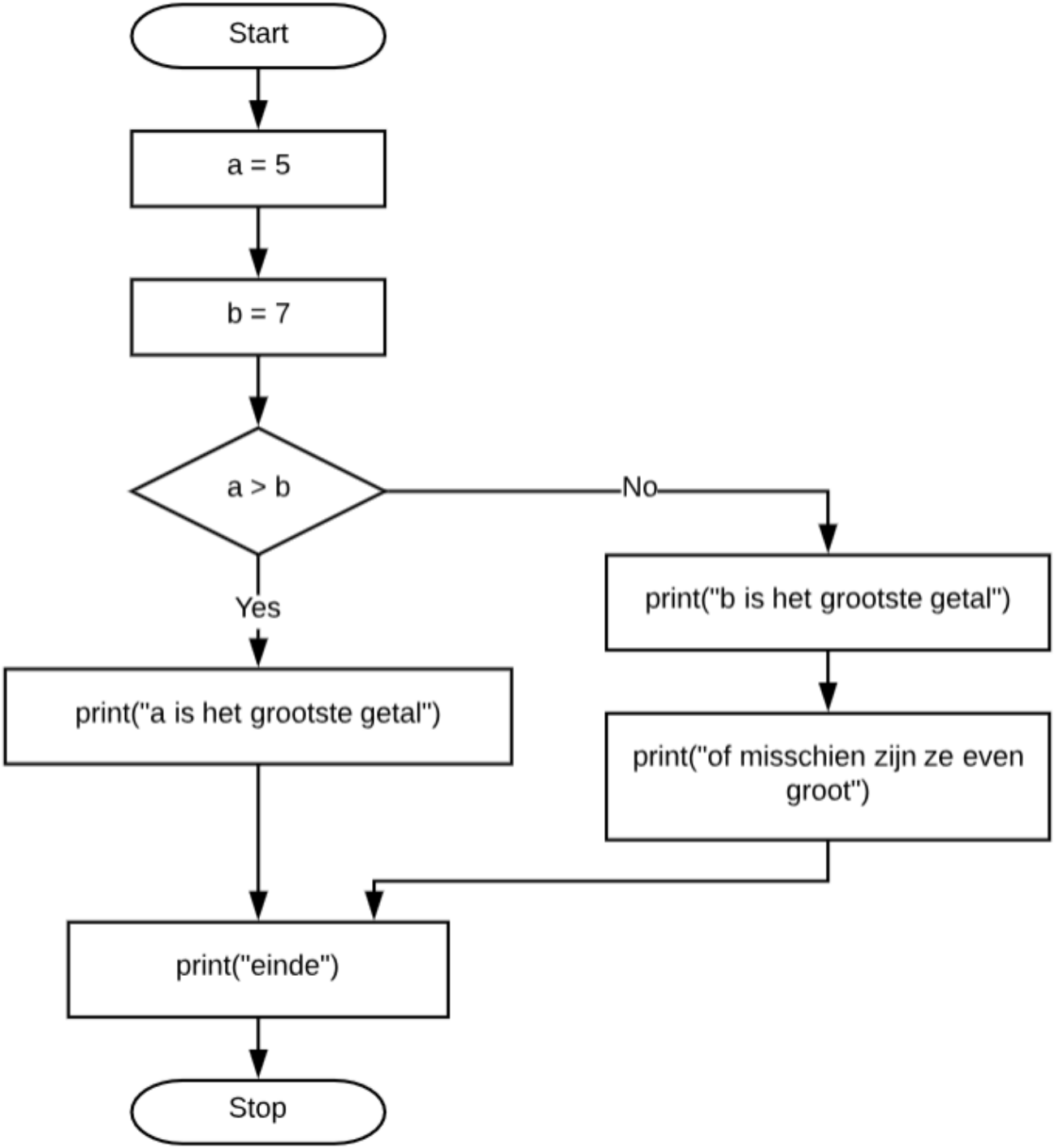
begin van het programma

uit te voeren instructies



een conditie die geëvalueerd wordt als True of False

einde van het programma

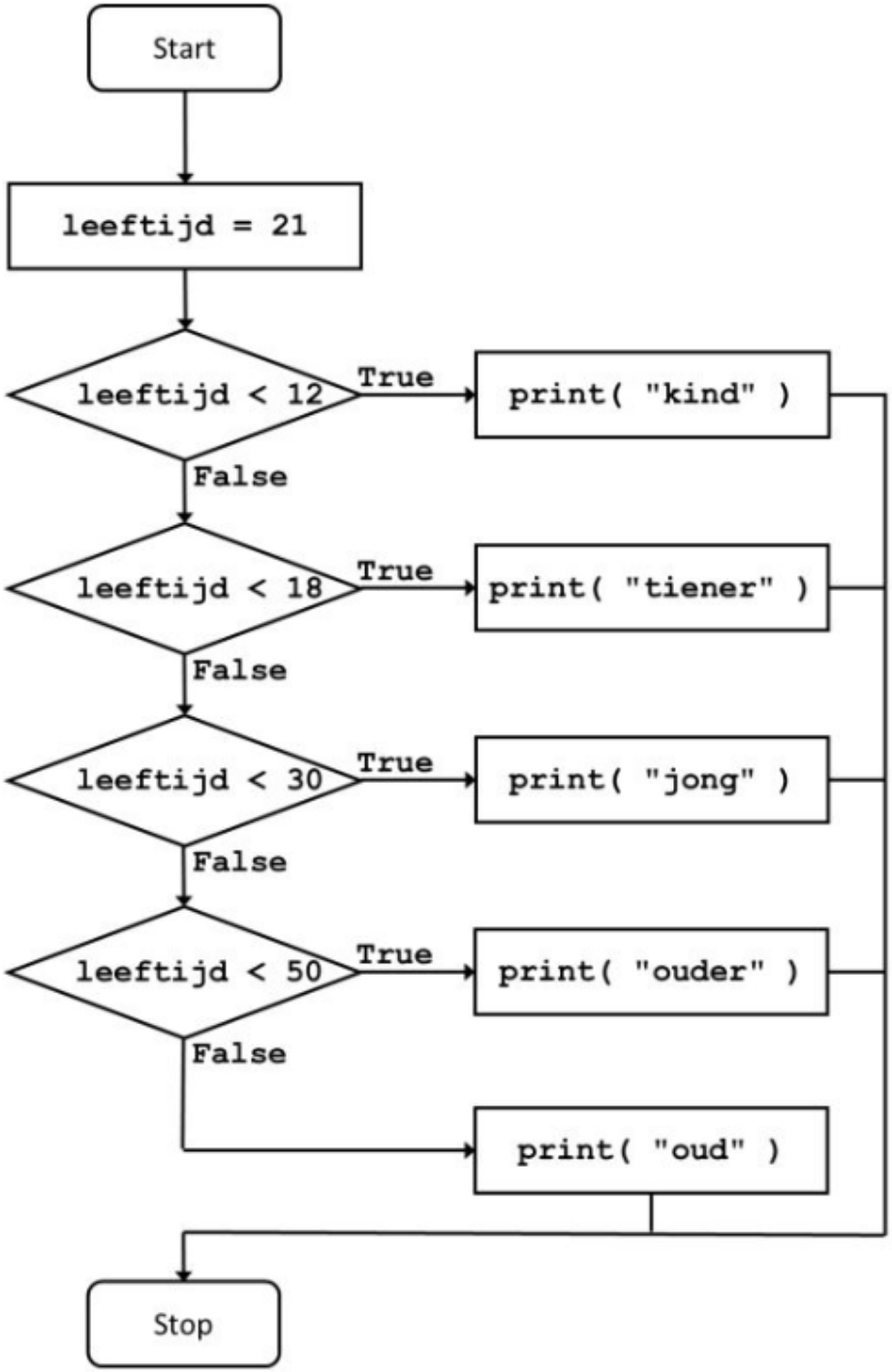

Start
a = 5
b = 7
a > b
No
Yes
print("b is het grootste getal")
print("a is het grootste getal")
print("of misschien zijn ze even groot")
print("einde")
Stop

3.2 Conditionele statements

3.2.5 Meer-weg beslissingen

Voorbeeld

```
leeftijd = 21
if leeftijd < 12:
    print("kind" )
elif leeftijd < 18:
    print("tiener")
elif leeftijd < 30:
    print("jong")
elif leeftijd < 50:
    print("ouder")
else:
    print("oud")
```


Start
leeftijd = 21
leeftijd < 12
True
print("kind")
False
leeftijd < 18
True
print("tiener")
False
leeftijd < 30
True
print("jong")
False
leeftijd < 50
True
print("ouder")
False
print("oud")
Stop

Opgave 3.9

- Bij welke waarden voor de variabele ‘leeftijd’ ga je als output ‘jong’ krijgen?
- Bij welke waarden voor de variabele ‘leeftijd’ ga je als output ‘oud’ krijgen?

3.2 Conditionele statements

3.2.5 Meer-weg beslissingen

```
if <boolean expressie>:
      <acties>
elif <boolean expressie>:
      <acties>
else:
      <acties>
```

Let op:

- In 1 if-statement kunnen meerdere elifs voorkomen. Volgorde!!
- Als alle boolean expressies False blijken, wordt de else uitgevoerd

Opgave 3.10

Schrijf een programma dat een variabele gewicht heeft. Als gewicht groter is dan 20 (kilo), print je: “Er moet een toeslag van €25 betaald worden voor bagage die te zwaar is.” Als gewicht kleiner is dan 20, print je: “Goede reis!”. Als gewicht precies 20 is, print je: “Poeh! Dat gewicht is precies goed!”.

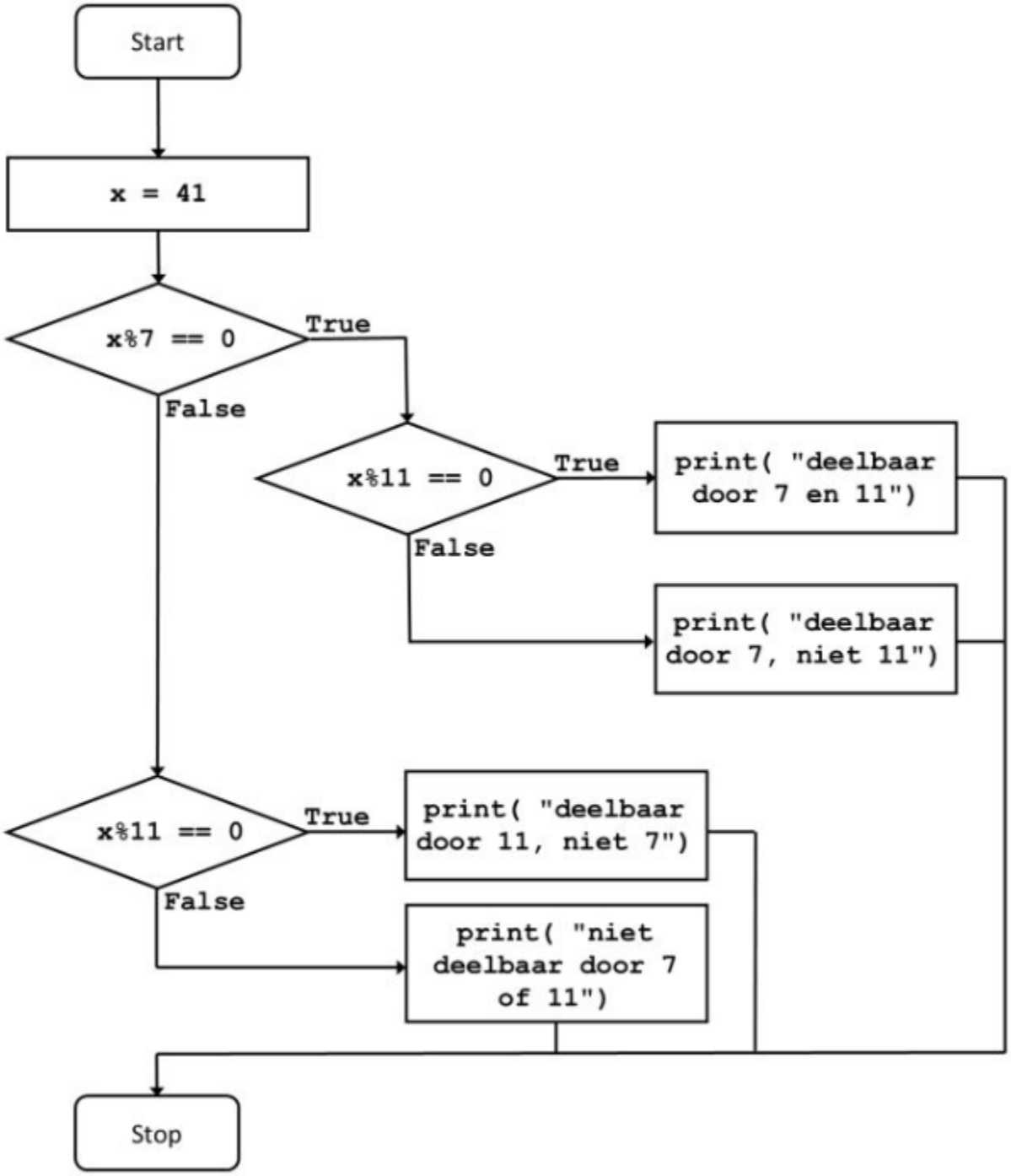
Wijzig de waarde van gewicht een paar keer om de code te testen.

3.2 Conditionele statements

3.2.6 Geneste condities

Voorbeeld

```
x = 41
if x % 7 == 0:
    # Hier begint een genest blok code
    if x % 11 == 0:
        print(x, "is deelbaar door 7 en 11.")
    else:
        print(x, "is deelbaar door 7, maar niet door 11.")
    # Hier eindigt een genest blok code
elif x % 11 == 0:
    print(x, "is deelbaar door 11, maar niet door 7.")
else:
    print(x, "is niet deelbaar door 7 of 11.")
```


Start
x = 41
x%7 == 0
True
False
x%11 == 0
True
print("deelbaar door 7 en 11")
False
print("deelbaar door 7, niet 11")
x%11 == 0
True
print("deelbaar door 11, niet 7")
False
print("niet deelbaar door 7 of 11")
Stop

3.2 Conditionele statements

3.2.6 Geneste condities

Afhankelijk van de situatie kan je beter een geneste if of een elif gebruiken

```
leeftijd = 21
if leeftijd < 12:
    print("kind")
else:
    if leeftijd < 18:
        print("tiener")
    else:
        if leeftijd < 30:
            print("jong")
        else:
            if leeftijd < 50:
                print("ouder")
            else:
                print("oud")
```

```
leeftijd = 21
if leeftijd < 12:
    print("kind")
elif leeftijd < 18:
    print("tiener")
elif leeftijd < 30:
    print("jong")
elif leeftijd < 50:
    print("ouder")
else:
    print("oud")
```

Afhankelijk van de situatie kan je beter een geneste if of een elif gebruiken

```
if leeftijd < 12:
    if kortingkaart == "ja":
        print("inkomgeld: 3 euro")
    else:
        print("inkomgeld: 5 euro")
else:
    if kortingkaart == "ja":
        print("inkomgeld: 7 euro")
    else:
        print("inkomgeld: 10 euro")
```

```
if leeftijd < 12 and kortingkaart == "ja":
    print("inkomgeld: 3 euro")
elif leeftijd < 12 and kortingkaart == "nee":
    print("inkomgeld: 5 euro")
elif leeftijd >= 12 and kortingkaart == "ja":
    print("inkomgeld: 7 euro")
else:
    print("inkomgeld: 10 euro")
```

Zijn volgende blokken code equivalent?

```
if x < 10:
    print("<10")
else:
    if x < 20:
        print("<20")
    else:
        print(">=20")
```

```
if x < 10:
    print("<10")
if x < 20:
    print("<20")
else:
    print(">=20")
```

```
if x < 10:
    print("<10")
    if x < 20:
        print("<20")
    else:
        print(">=20")
```

Zijn volgende blokken code equivalent?

```
if x == 1:
    a = x
if x == 2:
    b = x
```

```
if x == 1:
    a = x
else:
    if x == 2:
        b = x
```

Zijn volgende blokken code equivalent?

```
if a > 100:
    print("a > 100")
else:
    if a < 10:
        print("a < 10")
    else:
        print("a >= 10")
```

```
if a > 100:
    print("a > 100")
if a < 10:
    print("a < 10")
else:
    print("a >= 10")
```

Zijn volgende blokken code equivalent?

```
if x != 1:
   b = x
if x == 2:
   a = x
```

```
if x != 1:
   b = x
else:
    if x == 2:
       a = x
```

Opgave BMI

Maak een programma dat de **BMI** van een persoon berekent. Vraag de gebruiker naar zijn lengte en gewicht en bereken de BMI. Steek deze waarde in een afzonderlijke variabele. Geef vervolgens medisch advies.

lager dan 18: ondergewicht
18 tot 25: ok
25 tot 30: overgewicht
30 tot 40: obesitas
40 en hoger: ziekelijk overgewicht

$$BMI = \frac{\text{gewicht in kg}}{(\text{lengte in m}) * (\text{lengte in m})}$$

Opgave lidgeld

Schrijf 4 programma's voor volgende **vier situaties** waarbij het jaarlijkse lidgeld voor een sportvereniging dient berekend te worden. Invoer:

- De burgerlijke staat (dit is een cijfer van 1 tot en met 3: 1 = ongehuwd; 2 = gehuwd; 3 = weduw(e)(naar))
- de leeftijd

a) ongehuwd: 25 euro; gehuwd: 20 euro; weduw(e)(naar): 15 euro.

b) ongehuwd jonger dan 30: 25 euro; alle overige betalen 15 euro.

c) alle leden jonger dan 30 en alle ongehuwden: 25 euro; alle overige betalen 15 euro.

d) ongehuwd: 25 euro; gehuwd jonger dan 30: 20 euro; alle overige betalen 15 euro.