**Graad 8 LESPLAN**

**Les 1**: Inleiding tot Python

Lesdoelwitte (Wat jy sal leer):

• Verstaan ​​wat Python is

• Leer hoekom ons dit gebruik

• Sien hoe dit in werklike toepassings gebruik word

1. Wat is Python?

• Python is 'n programmeertaal wat gebruik word om rekenaars instruksies te laat volg.

• Dit help ons om speletjies, toepassings, webwerwe te bou en selfs robotte te beheer!

• Dit is bekend daarvoor dat dit eenvoudig en maklik leesbaar is — ideaal vir beginners.

Python werk met:

• Teks/Opdragte – om die rekenaar te vertel wat om te doen

• Data – om name, getalle, antwoorde te stoor

• Logika – om besluite in 'n program te neem

2. Werklike voorbeelde van Python

• Speletjies soos Minecraft-mods of eenvoudige 2D-speletjies

• Sakrekenaars en ander klein programme

• Robotika – beheer van motors en sensors

• Webwerwe soos Instagram en YouTube gebruik Python agter die skerms

3. Probeer hierdie voorbeeld

print("Hallo, Python!")

Hierdie lyn sê vir die rekenaar om die boodskap te druk.

Probeer om die boodskap binne die aanhalingstekens te verander.

4. Oefeninge

Oefening 1: Verander die Boodskap

Verander hierdie deel van die kode om iets nuuts te sê:

print("Jou boodskap hier!")

Oefening 2: Sê Jou Naam

Skryf 'n skrip wat jou naam wys:

print("Hallo, my naam is [Jou Naam]!")

Oefening 3: Vra vir 'n Naam

Hierdie skrip vra die gebruiker vir hul naam en sê hallo:

name = input("Wat is jou naam?")

print("Hallo, " + naam + "! Welkom by Python.")

5. Lesopsomming

• Python is 'n beginner-vriendelike programmeertaal wat in speletjies, toepassings, webwerwe en robotika gebruik word.

Jy het geleer hoe om:

o Gebruik print() om 'n boodskap te wys

o Gebruik input() om die gebruiker vir iets te vra

o Verbind teks en veranderlikes met behulp van +

Vasvrae

1. Waarvoor word Python gebruik?

a) Prente teken

b) Rekenaars laat take verrig

c) Musiek speel

d) Stories skryf

2. Watter van die volgende is 'n werklike voorbeeld van Python?

a) Musiek maak in 'n orkes

b) Tik in Microsoft Word

c) 'n Robot beheer

d) Handmatig op Facebook plaas

3. Watter kode is korrek om 'n boodskap in Python te wys?

a) echo("Hallo!")

b) say("Hallo!")

c) print("Hallo!")

d) message("Hallo!")

4. Wat doen hierdie Python-kode?

print("Welkom by Python!")

a) Maak 'n venster oop

b) Maak 'n geluid

c) Wys 'n boodskap

d) Doen niks

5. Hoe vra jy iemand om hul naam in Python in te voer?

a) vra("Voer jou naam in")

b) prompt("Voer jou naam in")

c) invoer("Voer jou naam in")

d) naam("Voer jou naam in")

Antwoordsleutel:

1 – b

2 – c

3 – c

4 – c

5 – c

**Les 2**: Eenvoudige Wiskunde en Invoer in Python

Opsomming van Les 1:

Inleiding tot Python

• Python is 'n eenvoudige programmeertaal.

• Gebruik in speletjies, toepassings, webwerwe en robotika.

• Jy het geleer:

• print() – wys boodskappe

• input() – vra die gebruiker vir invoer

• + – verbind teks en veranderlikes

Lesdoelwitte (Wat jy sal leer):

• Gebruik wiskundige bewerkings in Python

• Kry getalle van die gebruiker

• Doen basiese berekeninge met invoer

1. Wiskunde in Python

• + Tel op

• - Trek af

• \* Vermenigvuldig

• / Deel

Voorbeeld:

python

Kopieer kode

a = 8

b = 2

print("Som:", a + b)

print("Deel:", a / b)

2. Invoer met Getalle

Wanneer jy vir invoer vra, is dit 'n string. Om wiskunde te doen, verander dit in 'n getal met behulp van int().

Voorbeeld:

python

Kopieer kode

num1 = int(input("Voer 'n getal in: "))

num2 = int(input("Voer nog 'n getal in: "))

print("Resultaat:", num1 + num2)

Probeer dit

# Probeer om Python as 'n sakrekenaar te gebruik

print(10 + 5)

print(12 - 3)

print(4 \* 2)

print(16 / 4)

Probeer met veranderlikes:

x = 8

y = 2

print("Som:", x + y)

print("Verskil:", x - y)

print("Produk:", x \* y)

print("Kwosiënt:", x / y)

En met gebruikersinvoer:

a = int(input("Voer 'n getal in: "))

b = int(input("Voer nog 'n getal in: "))

print("Die totaal is", a + b)

3. Oefeninge

Oefening 1: Vermenigvuldig Twee Getalle

Vra die gebruiker vir twee getalle en vermenigvuldig hulle.

Oplossing:

num1 = int(input("Voer 'n getal in: "))

num2 = int(input("Voer nog 'n getal in: "))

print("Resultaat:", num1 \* num2)

Oefening 2: Trek af en wys resultaat

Laat die gebruiker twee getalle invoer. Trek die tweede van die eerste af en wys die antwoord.

Oplossing:

num1 = int(input("Voer die eerste getal in: "))

num2 = int(input("Voer die tweede getal in: "))

print("Resultaat:", num1 - num2)

Gamifikasie-uitdaging:

Vra die gebruiker vir hul toetspunt uit 50. Wys hoeveel punte hulle benodig om 50 te bereik.

Sterre verdien: 1 ster vir die gebruik van invoer, 1 ster vir korrekte wiskunde!

Oplossing:

telling = int(input("Voer jou toetstelling uit 50 in: "))

punte\_benodig = 50 - telling

druk("Jy benodig", punte\_benodig, "meer punte om 50 te bereik.")

Opsomming

• Gebruik +, -, \*, / vir wiskunde

• Gebruik int(input(...)) om getalle te kry

• Jy kan berekeninge met gebruikersinvoer doen

Vasvra

1. Wat doen \* in Python?

a) Tel op

b) Deel

c) Vermenigvuldig

d) Trek af

2. Watter kode vra vir 'n getal en tel 10 by?

a) input("getal") + 10

b) int(input("Voer in: ")) + 10

c) ask("getal") + 10

d) print("getal") + 10

3. Wat sal dit druk?

python

Kopieer kode

a = 5

b = 3

druk(a - b)

a) 2

b) 8

c) a - b

d) Fout

4. Wat is die resultaat van 10 / 2?

a) 5

b) 20

c) 0

d) 2

Antwoordsleutel:

1 – c, 2 – b, 3 – a, 4 – a

**Les 3**: Voorwaardelike Verklarings (if, elif, else)

Opsomming:

In die vorige les het ons geleer hoe om basiese wiskundige bewerkings uit te voer, en ons het gesien hoe om die gebruiker vir getalle te vra met behulp van input(). Ons het ook int() gebruik om invoer in getalle vir berekeninge om te skakel.

Lesdoelwitte (Wat jy sal leer):

• Verstaan ​​hoe om besluite in Python te neem met behulp van if, elif, en else

• Leer hoe om waardes te vergelyk met behulp van logiese operateurs

• Oefen om eenvoudige voorwaardelike stellings te skryf om programvloei te beheer

1. Wat is Voorwaardelike Verklarings?

Voorwaardelike stellings laat ons besluite neem gebaseer op voorwaardes (waar of onwaar). Hulle help ons om die vloei van 'n program te beheer.

• as kontroleer 'n voorwaarde

• elif kontroleer 'n ander voorwaarde as die eerste vals is

• else word gebruik as alle vorige voorwaardes vals is

Voorbeeld:

ouderdom = 18

if ouderdom >= 18:

print("Jy is 'n volwassene.")

else:

print("Jy is 'n minderjarige.")

2. Gebruik van elif vir veelvuldige voorwaardes

Jy kan veelvuldige voorwaardes kontroleer deur elif te gebruik. Dit help as jy meer as twee besluite wil neem.

Voorbeeld:

temperatuur = 25

if temperatuur > 30:

print("Dis regtig warm buite.")

elif temperatuur > 20:

print("Die weer is lekker.")

else:

print("Dis 'n bietjie koud.")

3. Oefeninge

Oefening 1: Ouderdomskontroleerder

Skryf 'n program wat kontroleer of 'n persoon 'n kind (onder 12), 'n tiener (12-17) of 'n volwassene (18+) is.

ouderdom = int(input("Voer jou ouderdom in: "))

if ouderdom < 12:

print("Jy is 'n kind.")

elif ouderdom < 18:

print("Jy is 'n tiener.")

else:

print("Jy is 'n volwassene.")

Oefening 2: Getalkontroleerder

Vra die gebruiker om 'n getal in te voer. As die getal ewe is, druk "Ewe". As dit onewe is, druk "Onewe".

num = int(input("Voer 'n getal in: "))

if num % 2 == 0:

print("Ewe")

else:

print("Onewe")

Gamifikasie-uitdaging:

Vra die gebruiker vir hul ouderdom en gee hulle 'n boodskap gebaseer op hul ouderdomsgroep:

• As hulle jonger as 12 is, sê "Jy is jonk!"

• As hulle tussen 12 en 18 is, sê "Jy is 'n tiener!"

• As hulle ouer as 18 is, sê "Jy is 'n volwassene!"

Sterre verdien: 1 ster vir elke korrekte voorwaarde wat gebruik word!

Opsomming

• if kontroleer of 'n voorwaarde waar is

• elif kontroleer 'n ander voorwaarde as die if vals is

• else loop as geeneen van die voorwaardes waar is nie

• Ons gebruik voorwaardelike stellings om besluite in ons program te neem

Vasvra

1. Wat doen die if-stelling in Python?

a) Kontroleer 'n voorwaarde

b) Druk 'n boodskap

c) Loop deur 'n lys

d) Vra vir gebruikersinvoer

2. Wat is die korrekte sintaksis om te kontroleer of 'n getal groter as 10 is?

a) if number > 10:

b) if number > 10

c) if (number > 10)

d) if number >= 10:

3. Wat sal hierdie kode druk?

age = 15

if age >= 18:

print("Volwassene")

else:

print("Minor")

a) Volwassene

b) Minor

c) Niks

d) Fout

4. Wat doen elif in Python?

a) Dit kontroleer 'n ander voorwaarde as die "if"-voorwaarde vals is.

b) Dit kontroleer of 'n voorwaarde waar is.

c) Dit word gebruik om deur 'n lys te loop.

d) Dit voer kode uit na alle voorwaardes.

5. Hoe kontroleer jy of 'n getal onewe is in Python?

a) as num % 2 == 0:

b) as num % 2 != 0:

c) as num / 2:

d) as num == 1:

Antwoordsleutel:

1 – a, 2 – a, 3 – b, 4 – a, 5 – b

**Graad 9 LESPLAN**

**Les 1**: Python-funksies

Lesdoelwitte (Wat jy sal leer):

• Verstaan ​​wat 'n funksie in Python is

• Leer hoekom ons funksies gebruik

• Sien hoe funksies in die werklike lewe gebruik word

• Skryf eenvoudige Python-funksies

1️. Wat is 'n Python-funksie?

• 'n Funksie is 'n blok kode wat iets spesifiek doen.

• Dink daaraan soos 'n resep: jy skryf die stappe een keer neer en gebruik dit baie keer.

2️. Werklike voorbeelde van funksies

• Groet gebruikers – Wys 'n welkomsboodskap wanneer iemand 'n program oopmaak

• Wiskunde sakrekenaars – Tel getalle op of vermenigvuldig dit

• Kieslyste – Kies verskillende aksies (soos in 'n speletjie of toepassing)

• Robotika – Voer dieselfde motorkode oor en oor uit

3️. Voorbeelde

Voorbeeld 1 – Sê hallo

def sê\_hallo():

print("Hallo, welkom by Python!")

sê\_hallo()

Aktiwiteit: Verander die boodskap na: "Goeiemôre, Graad 9-leerders".

Voorbeeld 2 – Wys 'n Pasgemaakte Boodskap

def show\_message(message):

print(message)

show\_message("Welkom by Python!")

Voorbeeld 3 – Vermenigvuldig Twee Getalle

def multiply\_numbers(a, b):

result = a \* b

print("Die resultaat is:", result)

multiply\_numbers(5, 6)

4. Oefeninge

Oefening 1 – Tel Twee Getalle Op

Skep 'n funksie wat 5 + 10 bymekaar tel en die resultaat wys.

def add\_numbers(): num1 = 5 num2 = 10 som = num1 + num2 print(sum)

add\_numbers()

Oefening 2 – Groet iemand by naam

def greet(naam):

print("Hallo, " + naam + "!")

greet("John")

Gamifikasie:

Uitdaging van die dag:

Skep 3 funksies oor jouself:

• introduce() – druk jou naam

• favourite\_color() – druk jou gunstelingkleur

• hobby() – druk jou gunsteling stokperdjie

Verdien 1 ster vir elke funksie wat jy korrek voltooi

5. Lesopsomming

• Wat is 'n funksie?

'n Funksie is herbruikbare kode wat 'n spesifieke taak uitvoer.

• Waar word funksies gebruik?

Om gebruikers te groet, berekeninge te doen, herhaalde aksies te hanteer, ens.

• Voorbeelde wat jy geleer het:

Wys boodskappe, vermenigvuldig getalle en gebruik parameters.

Kwis: Python-funksies

1. Wat is die primêre doel van 'n funksie in Python?

a) Om beelde te teken

b) Om spesifieke take in kode te hergebruik

c) Om webwerwe te skep

d) Om stories te skryf

2. Watter van die volgende is 'n werklike voorbeeld van die gebruik van 'n Python-funksie?

a) Om jou naam een ​​keer te druk

b) Om dieselfde groet op verskeie plekke te herhaal

c) Om 'n Word-dokument te stoor

d) Om lettergrootte te verander

3. Hoe definieer jy 'n funksie in Python?

a) function greet() {}

b) def greet():

c) greet = function()

d) create function greet()

4. Wat sal hierdie kode doen?

def show\_message(message):

print(message)

show\_message("Welkom by Python!")

a) Vermenigvuldig getalle

b) Wys 'n boodskap

c) Wys 'n fout

d) Verander agtergrondkleur

5. Wat is die uitvoer van hierdie kode?

def add():

print(5 + 10)

add()

a) 10

b) 15

c) Fout

d) Niks

Antwoordsleutel:

1 – b

2 – b

3 – b

4 – b

5 – b

**Les 2**: Parameters en Teruggeewaardes

Lesdoelwitte (Wat jy sal leer)

• Verstaan ​​wat parameters in Python is

• Leer hoe om waardes aan funksies deur te gee

• Verstaan ​​wat 'n terugkeerwaarde is

• Skryf funksies wat resultate teruggee

Opsomming

Wat jy geleer het:

• 'n Funksie is 'n herbruikbare blok kode

• Funksies begin met die def-sleutelwoord

• Jy kan 'n funksie met sy naam aanroep

• Sommige funksies kan insette (parameters) aanvaar

Voorbeeld Opsomming:

def greet(naam):

print("Hallo, " + naam + "!")

greet("John")

1. Wat is parameters en terugkeerwaardes?

• Parameters is plekhouers vir waardes wat jy aan 'n funksie deurgee

• Argumente is die werklike waardes wat jy deurgee wanneer jy die funksie aanroep

• Teruggeewaardes laat 'n funksie 'n resultaat terugstuur

Voorbeeld: Funksie met parameters

def greet(naam):

print("Hallo " + naam + "!")

Jy kan nou enigiemand groet deur die argument te verander.

Voorbeeld: Funksie wat 'n waarde teruggee

def add(a, b):

return a + b

resultaat = add(5, 3)

print(resultaat)

Die funksie stuur die resultaat terug met behulp van return, en ons stoor dit in result.

2. Werklike voorbeelde van terugkeerwaardes

• 'n Sakrekenaar-app gee die resultaat van jou som terug

• 'n Speletjie kan jou telling na 'n vlak teruggee

• 'n Vorm kan kyk of jou invoer korrek is en "Ja" of "Nee" teruggee

3. Voorbeelde

Voorbeeld 1 – Tel twee getalle op

def add(a, b):

return a + b

print(add(4, 6)) # Uitsette: 10

Voorbeeld 2 – Kwadraat van 'n getal

def square(n):

return n \* n

resultaat = square(7)

print(resultaat) # Uitsette: 49

Voorbeeld 3 – Groet met 'n terugkeer

def get\_greeting(naam):

return "Hallo, " + naam + "!"

boodskap = kry\_groet("Graad 9")

druk(boodskap)

4. Aktiwiteite

Aktiwiteit 1 – Tel op en Gee Terug Skep 'n funksie tel\_getalle() wat twee getalle bymekaar tel en die resultaat teruggee. Stoor dit in 'n veranderlike en druk dit.

Oplossing:

def tel\_getalle(): getal1 = 10

getal2 = 15

gee terug getal1 + getal2

resultaat = tel\_getalle()

druk(resultaat) # Uitvoer: 25

Aktiwiteit 2 – Vermenigvuldig en Stoor Skryf 'n funksie vermenigvuldig(x, y) wat die produk teruggee. Gebruik dit saam met druk() om die resultaat te vertoon.

Oplossing:

def vermenigvuldig(x, y):

gee terug x \* y

druk(vermenigvuldig(4, 5)) # Uitvoer: 20

5. Oefeninge

Oefening 1 – Gee 'n Welkomsboodskap terug

def welkom(naam):

gee terug "Welkom by die klas," + naam + "!"

Roep dit met jou naam en druk die resultaat.

Oplossing:

def welcome(naam):

return "Welkom in die klas," + naam + "!"

message = welcome("Lerato")

print(message) # Uitvoer: Welkom in die klas, Lerato!

Oefening 2 – Gee die Kwadraat van 'n Getal terug Skryf 'n funksie square\_it(num) wat die kwadraat van die getal teruggee.

Oplossing:

def square\_it(num):

return num \* num

print(square\_it(9)) # Uitvoer: 81

Gamification:

Uitdaging van die Dag:

Skryf 'n funksie my\_info(naam, ouderdom) wat hierdie sin teruggee:

"My naam is [naam] en ek is [ouderdom] jaar oud."

Verdien 3 sterre:

• 1 ster vir korrekte funksie

• 1 ster vir die gebruik van return

• 1 ster vir die druk van die resultaat

Lesopsomming

• Funksies kan insette wat parameters genoem word, aanvaar

• Jy gee waardes aan daardie parameters deur wanneer jy die funksie aanroep

• Funksies kan resultate teruggee deur die return-sleutelwoord te gebruik

• Teruggegewe waardes kan gestoor en later gebruik word

Vasvra:

1. Wat is parameters in 'n funksie?

A. Die resultaat van die funksie

B. Waardes wat jy van invoer kry

C. Plekhouers in die funksiedefinisie

D. Kommentaar in die kode

2. Wat doen die return-sleutelwoord?

A. Herhaal 'n funksie

B. Stuur 'n waarde van die funksie terug

C. Beëindig die program

D. Tel getalle op

3. Wat word deur hierdie kode gedruk?

def add(a, b):

return a + b

print(add(2, 3))

A. a + b

B. 23

C. 5

D. Niks

4. Wat gebeur as jy nie return in 'n funksie gebruik nie?

A. Die funksie sal nie loop nie

B. Dit gee 'n fout terug

C. Dit gee niks terug nie (Geen)

D. Dit druk outomaties

5. Watter van die volgende is korrek?

A. def add(a + b):

B. def add(a, b): return a + b

C. add = def(a, b) return a + b

D. def add() = a + b

Antwoorde

1 – C

2 – B

3 – C

4 – C

5 – B

**Les 3**: Voorwaardelike Verklarings (if, elif, else)

Lesdoelwitte (Wat jy sal leer)

• Leer hoe om besluite in kode te neem met behulp van if, elif, en else

• Verstaan ​​hoe vergelykings in Python werk

• Oefen om funksies te skryf wat voorwaardes gebruik

• Gebruik werklike voorbeelde soos ouderdomskontroles, toetsresultate en spelbesluite

Opsomming

Wat jy geleer het:

• Funksies kan parameters neem en waardes teruggee

• return stuur 'n resultaat terug

• Jy kan teruggegewe resultate in veranderlikes stoor

Voorbeeld Opsomming:

def add(a, b):

return a + b

resultaat = add(3, 7)

print(resultaat) # Uitvoer: 10

1. Wat is Voorwaardelike Verklarings?

• Voorwaardelike stellings laat jou kode besluite neem

• Gebruik if, elif, en else om te beheer wat jou program doen

• 'n Voorwaarde kontroleer of iets Waar of Onwaar is

2. Werklike Voorbeelde van Voorwaardes

• As 'n student 50% of meer kry, slaag hulle

• As dit reën, bring 'n sambreel

• In 'n speletjie, as jou telling bo 100 is, styg jy na 'n vlak

• As 'n gebruiker onder 13 is, wys 'n "te jonk"-boodskap

3. Voorbeelde

Voorbeeld 1 – Slaag of Druip

def check\_slaag(punt):

if punt >= 50:

print("Jy het geslaag!")

else:

print("Jy het gedruip.")

Voorbeeld 2 – Ouderdomskontroleerder

def check\_age(ouderdom):

if age < 13:

print("Jy is te jonk.")

elif age < 18:

print("Jy is 'n tiener.")

else:

print("Jy is 'n volwassene.")

Voorbeeld 3 – Ewe of Onewe

def ewe\_of\_onewe(getal):

as getal % 2 == 0:

print("Ewe")

else:

print("Onewe")

4. Aktiwiteite

Aktiwiteit 1 – Puntekontroleerder

Skryf 'n funksie wat 'n punt neem en:

• As punt ≥ 80: print "Uitstekend"

• As punt ≥ 50: print "Goeie werk"

• Anders: print "Hou aan probeer"

Oplossing:

def graadkontroleerder(punt):

as punt >= 80:

print("Uitstekend")

elif punt >= 50:

print("Goeie werk")

else:

print("Hou aan probeer")

graadkontroleerder(85) # Uitvoer: Uitstekend

graadkontroleerder(60) # Uitvoer: Goeie werk

graadkontroleerder(40) # Uitvoer: Hou aan probeer

Aktiwiteit 2 – Getalkontroleerder

Skryf 'n funksie wat kontroleer of 'n getal positief, negatief of nul is, en druk die korrekte boodskap.

Oplossing:

def tjeknommer(nommer):

if nommer > 0:

print("Positief")

elif nommer < 0:

print("Negatief")

else:

print("Nul")

tjeknommer(5) # Uitvoer: Positief

tjeknommer(-2) # Uitvoer: Negatief

tjeknommer(0) # Uitvoer: Nul

5. Oefeninge

Oefening 1 – Kontroleer of 'n persoon kan stem

Skryf 'n funksie wat kontroleer of 'n persoon 18 jaar of ouer is en druk "Jy kan stem!" Andersins, druk "Te jonk om te stem."

Oplossing:

def kan\_stem(ouderdom):

if ouderdom >= 18:

print("Jy kan stem!")

else:

print("Te jonk om te stem.")

kan\_stem(20) # Uitvoer: Jy kan stem!

kan\_stem(16) # Uitvoer: Te jonk om te stem.

Oefening 2 – Wedstrydtelling Ranglys

Skep 'n funksie wat 'n telling neem en:

• As telling ≥ 100: druk "Goud rang"

• As telling ≥ 50: druk "Silwer rang"

• Anders: druk "Brons rang"

Oplossing:

def rang\_telling(telling):

as telling >= 100:

druk("Goud rang")

elif telling >= 50:

druk("Silwer rang")

anders:

druk("Brons rang")

rang\_telling(120) # Uitvoer: Goud rang

rang\_telling(70) # Uitvoer: Silwer rang

rang\_telling(30) # Uitvoer: Brons rang

Gamifikasie vir hierdie Les

Uitdaging van die Dag:

Skep 'n funksie student\_result(punt) wat teruggee:

• "Onderskeiding" vir 80 en hoër

• "Slaag" vir 50–79

• "Druip" vir onder 50

Verdien 3 sterre:

• 1 ster vir die gebruik van if

• 1 ster vir die gebruik van elif en else

• 1 ster vir die gebruik van return in plaas van print

Lesopsomming

• Gebruik if, elif, en else om keuses in jou kode te maak

• Voorwaardes vergelyk waardes met behulp van operateurs soos ==, >, <, >=, en <=

• Jy kan voorwaardes binne funksies gebruik om te beheer wat hulle doen

• Regte programme gebruik voorwaardes heeltyd—speletjies, toepassings, webwerwe, en meer!

Kwis:

1. Wat doen if in Python?

A. Herhaal die kode

B. Laat kode slegs loop wanneer 'n voorwaarde waar is

C. Stoor inligting

D. Stop die program

2. Wat is die uitvoer van hierdie kode?

def test():

if 5 > 3:

print("Ja")

else:

print("Nee")

test()

A. Ja

B. Nee

C. Fout

D. Niks

3. Watter sleutelwoord kyk vir die volgende voorwaarde?

A. anders

B. dan

C. elif

D. kontroleer

4. Wat druk hierdie kode?

def kontrolenommer(num):

if num == 0:

print("Nul")

else:

print("Nie nul")

kontrolenommer(0)

A. Nul

B. Nie nul nie

C. Fout

D. Niks

5. Wat is fout met hierdie kode?

def kontroleer():

if age >= 18:

print("Volwassene")

A. print moet in hoofletters wees

B. age is nie gedefinieer nie

C. if moet laaste wees

D. def kan nie hier gebruik word nie

Antwoorde

1 – B

2 – A

3 – C

4 – A

5 – B

**Graad 10 LESPLAN**

**Les 1**: Werk met Lyste en Woordeboeke in Python

Lesdoelwitte (Wat jy sal leer):

• Verstaan ​​wat lyste en woordeboeke in Python is

• Leer hoe om data te stoor en te bestuur deur hierdie strukture te gebruik

• Oefen toegang tot en wysiging van lyste en woordeboeke

• Gebruik werklike voorbeelde om programme met lusse en datastrukture te bou

1. Wat is 'n Lys in Python?

• 'n Lys is 'n versameling waardes wat in een veranderlike gestoor word.

• Lyste word georden en kan verander word (veranderlik).

Voorbeeld:

vrugte = ["Appel", "Piesang", "Kersie"]

druk(vrugte[1]) # Uitvoer: Piesang

Voeg by 'n lys:

vrugte.append("Lemoen")

druk(vrugte)

2. Wat is 'n Woordeboek in Python?

• 'n Woordeboek stoor data in sleutel-waarde pare.

• Dis soos 'n mini-databasis: jy benoem elke stukkie data met 'n naam (sleutel).

Voorbeeld:

persoon = {

"naam": "John",

"ouderdom": 30,

"werk": "Ontwikkelaar"

}

druk(persoon["naam"]) # Uitvoer: John

Voeg 'n nuwe sleutel-waarde-paar by:

persoon["stokperdjie"] = "Lees"

3. Werklike voorbeelde

Voorbeeld 1 – Lys van vrugte

vrugte = ["appel", "piesang", "kersie"]

vrugte.append("lemoen")

vir vrugte in vrugte:

druk(vrugte)

Voorbeeld 2 – Boekwoordeboek

boek = {

"titel": "Python vir beginners",

"outeur": "Jane Doe",

"jaar": 2021

}

boek["genre"] = "Programmering"

vir sleutel, waarde in boek.items():

druk(sleutel + ":", waarde)

Voorbeeld 3 – Lys van studentewoordeboeke

studente = [

{"naam": "John", "ouderdom": 16},

{"naam": "Jane", "ouderdom": 15},

{"naam": "Bob", "ouderdom": 17}

]

studente.append({"naam": "Alice", "ouderdom": 18})

vir student in studente:

druk(student["naam"], "is", student["ouderdom"], "jaar oud.")

4. Oefeninge

Oefening 1 – Kleurlys

Skep 'n lys van 5 kleure. Voeg nog een by. Gebruik 'n lus om alle kleure te vertoon.

kleure = ["rooi", "blou", "groen", "geel", "oranje"]

kleure.append("pers")

vir kleur in kleure:

druk(kleur)

Oefening 2 – Gunstelingfliekwoordeboek

Skep 'n woordeboek met titel, regisseur en jaar. Voeg genre by. Druk alle besonderhede.

fliek = {

"titel": "Inception",

"regisseur": "Christopher Nolan",

"jaar": 2010

}

fliek["genre"] = "Wetenskapfiksie"

vir sleutel, waarde in fliek.items():

druk(sleutel + ":", waarde)

Gamifikasie

Uitdaging van die Dag: Skep 'n mini-studentedatabasis met behulp van 'n lys van 3 woordeboeke. Elke woordeboek moet die volgende insluit:

• naam

• ouderdom

• onderwerp

Verdien 1 ster vir elke student wat jy byvoeg!

5. Lesopsomming

• Lys: 'n Versameling items. Jy kan meer byvoeg met behulp van .append() en deur hulle loop.

• Woordeboek: 'n Groep geëtiketteerde waardes met behulp van sleutel-waarde-pare. Jy kan hulle byvoeg en toegang daartoe verkry met behulp van sleutels.

Kwis: Lyste en Woordeboeke in Python

1. Wat is 'n lys in Python?

a) 'n Versameling sleutel-waarde-pare

b) 'n Versameling ongeordende waardes

c) 'n Versameling geordende waardes

d) 'n Teksboodskap

2. Hoe kry jy toegang tot die eerste element in 'n lys?

a) lys[1]

b) lys(0)

c) lys.eerste()

d) lys[0]

3. Wat is 'n woordeboek in Python?

a) 'n Funksie wat 'n taak uitvoer

b) 'n Versameling sleutel-waarde-pare

c) 'n Lys getalle

d) 'n Tipe lus

4. Wat doen die append()-metode?

a) Verwyder 'n item van 'n lys

b) Voeg 'n item aan die einde van 'n lys by

c) Dateer 'n woordeboeksleutel op

d) Vee die lys uit

5. Wat sal hierdie kode vertoon?

motor = {

"maak": "Tesla",

"model": "Model 3",

"jaar": 2021

}

motor["kleur"] = "blou"

druk(motor["kleur"])

a) blou

b) ongedefinieerd

c) Model 3

d) Tesla

6. Hoe kan jy deur 'n lys woordeboeke loop?

a) Gebruik 'n for-lus

b) Gebruik while

c) Gebruik forEach()

d) Al die bogenoemde

7. Wat is waar oor lyste in Python?

a) Lyste is onveranderlik

b) Lyste word deur snare geïndekseer

c) Lyste kan verskillende datatipes stoor

d) Lyste stoor slegs nommers

8. Wat druk hierdie kode?

kleure = ["rooi", "blou"]

kleure.append("groen")

druk(kleure[2])

a) rooi

b) blou

c) groen

d) Fout

9. Hoe kry jy toegang tot die waarde in 'n woordeboek met die sleutel "naam"?

a) woordeboek(naam)

b) woordeboek.naam

c) woordeboek["naam"]

d) woordeboek.kry(naam)

10. Wat is die uitvoer van hierdie kode?

studente = [

{"naam": "Alice", "ouderdom": 17},

{"naam": "Bob", "ouderdom": 18}

]

studente.append({"naam": "Charlie", "ouderdom": 16})

druk(studente[2]["naam"])

a) Charlie

b) Bob

c) Alice

d) Fout

Antwoordsleutel:

1 – c

2 – d

3 – b

4 – b

5 – a

6 – a

7 – c

8 – c

9 – c

10 – a

**Les 2**: Deur Lyste en Woordeboeke in Python Loop

Lesdoelwitte (Wat jy sal leer)

• Verstaan ​​hoe om for-lusse met lyste en woordeboeke te gebruik

• Leer hoe om lysitems met behulp van lusse te verkry en te manipuleer

• Gebruik lusse om woordeboekinhoud te vertoon en op te dateer

• Pas lusse toe op werklike programmeringsituasies

Opsomming

Wat jy geleer het:

• Lyste stoor geordende waardes

• Woordeboeke stoor geëtiketteerde data in sleutel-waarde-pare

• Jy kan .append() gebruik om by lyste te voeg en vierkantige hakies [] om toegang tot woordeboekwaardes te verkry

1. Deur 'n Lys Loop

Jy kan 'n for-lus gebruik om deur elke item in 'n lys te gaan.

Voorbeeld:

colors = ["rooi", "blou", "groen"]

for color in colors:

print(color)

2. Gebruik van range() en indekse

Gebruik range() om deur 'n lys volgens indeks te loop.

Voorbeeld:

vrugte = ["appel", "piesang", "kersie"]

vir i in range(len(vrugte)):

print("Vrugte", i + 1, "is", vrugte[i])

3. Deur 'n Woordeboek Loop

Gebruik .items() om deur beide sleutels en waardes te loop.

Voorbeeld:

persoon = {

"naam": "Alice",

"ouderdom": 17,

"stad": "Kaapstad"

}

vir sleutel, waarde in person.items():

print(sleutel + ":", waarde)

4. Deur 'n Lys Woordeboeke Loop

Dit is nuttig wanneer jy met groepe mense of voorwerpe werk.

Voorbeeld:

studente = [

{"naam": "Lerato", "ouderdom": 16},

{"naam": "Sipho", "ouderdom": 17},

{"naam": "Thando", "ouderdom": 15}

]

vir student in studente:

druk(student["naam"], "is", student["ouderdom"], "jaar oud.")

5. Aktiwiteite

Aktiwiteit 1 – Deur 'n Lys van Stede Loop

Skep 'n lys van 5 stede. Gebruik 'n for-lus om te druk:

“Ek wil eendag [stad] besoek.”

Oplossing:

stede = ["New York", "Parys", "Tokio", "Londen", "Sydney"]

vir stad in stede:

druk("Ek wil besoek", stad, "eendag.")

Aktiwiteit 2 – Deur 'n Studentewoordeboek Loop

Skep 'n woordeboek met jou naam, ouderdom, graad en gunstelingvak. Gebruik 'n lus om elke sleutel en sy waarde te druk.

Oplossing:

student = {

"naam": "Lara",

"ouderdom": 15,

"graad": "10de",

"gunsteling\_vak": "Wiskunde"

}

for key, value in student.items():

print(key + ":", value)

6. Oefeninge

Oefening 1 – Vermenigvuldiging van Getalle in 'n Lys

Skep 'n lys van 5 getalle. Gebruik 'n lus om elke getal met 2 te vermenigvuldig en druk die resultaat.

Oplossing:

getalle = [2, 4, 6, 8, 10]

for number in numbers:

print(number \* 2)

Oefening 2 – Vertoon Boeke uit 'n Lys Woordeboeke

Skep 'n lys met 3 boekwoordeboeke. Elkeen moet 'n titel en outeur hê. Gebruik 'n lus om te vertoon:

“[titel] is geskryf deur [outeur]”

Oplossing:

boeke = [

{"titel": "1984", "outeur": "George Orwell"},

{"titel": "To Kill a Mockingbird", "outeur": "Harper Lee"},

{"titel": "The Great Gatsby", "outeur": "F. Scott Fitzgerald"}

]

vir boek in boeke:

druk(boek["titel"], "is geskryf deur", boek["outeur"])

7. Gamifikasie:

Uitdaging van die Dag:

Skep 'n program wat die gebruiker vra om 3 vriende se name in te voer, hulle in 'n lys stoor en druk:

“Hallo, [naam]!” vir elke vriend.

• Verdien 1 ster vir die gebruik van 'n lys

• Verdien 1 ster vir die gebruik van 'n lus

8. Lesopsomming

• Gebruik for-lusse om deur lyste en woordeboeke te gaan

• Gebruik .items() om deur sleutel-waarde-pare in woordeboeke te loop

• Gebruik range() wanneer jy indeksgebaseerde toegang benodig

• Lusse help jou om aksies te herhaal en doeltreffend met data te werk

Vasvra: Loop deur lyste en woordeboeke

1. Wat doen 'n for-lus in Python?

A. Herhaal kode

B. Skep veranderlikes

C. Stop die program

D. Vee data uit

2. Wat laat range(len(lys)) jou toe om te doen?

A. Voeg by die lys

B. Loop met indeksnommers

C. Slaan items oor

D. Breek die lys

3. Watter metode word gebruik om deur beide sleutels en waardes van 'n woordeboek te loop?

A. .keys()

B. .values()

C. .items()

D. .all()

4. Wat is die uitvoer van hierdie kode?

name = ["Ava", "Ben", "Cara"]

vir naam in name:

druk("Hallo", naam)

A. Fout

B. Haai name

C. Haai Ava, Ben, Cara

D. Haai Ava (dan Haai Ben, Haai Cara op die volgende reëls)

5. Wat is die korrekte manier om alle ouderdomme van hierdie lys te druk?

mense = [

{"naam": "Neo", "ouderdom": 16},

{"naam": "Lira", "ouderdom": 17}

]

A. druk(mense.ouderdom)

B. vir ouderdom in mense["ouderdom"]

C. vir persoon in mense: druk(persoon["ouderdom"])

D. vir mense in ouderdom: druk(ouderdom)

Antwoorde

1. A

2. B

3. C

4. D

5. C

**Les 3**: Voorwaardelike Verklarings in Python

Lesdoelwitte (Wat jy sal leer):

• Verstaan ​​wat voorwaardelike stellings is

• Leer hoe om if, elif en else te gebruik om programvloei te beheer

• Oefen die skryf van voorwaardes deur vergelykings te gebruik

• Gebruik werklike voorbeelde om besluite in kode te neem

Opsomming

Wat jy in Les 2 geleer het:

• Hoe om for-lusse te skep en te gebruik om deur lyste en woordeboeke te gaan

• Deur elke item in 'n lys te loop

• Deur sleutel-waarde pare in 'n woordeboek te loop

1. Wat is 'n Voorwaardelike Verklaring?

• 'n Voorwaardelike stelling laat jou program besluite neem.

• Jy kan verskillende kodes uitvoer, afhangende van of iets waar of onwaar is.

Sleutelwoorde:

• if: Voer 'n blok kode uit as 'n voorwaarde waar is

• elif: Kontroleer 'n ander voorwaarde as die eerste vals is

• else: Voer uit as geeneen van die voorwaardes waar is nie

Voorbeeld:

age = 16

if age >= 18:

print("Jy is 'n volwassene.")

else:

print("Jy is nie 'n volwassene nie.")

2. Vergelykingsoperateurs

| **Operator** | **Betekenis** | **Voorbeeld** |
| --- | --- | --- |
| == | Gelyk aan | 5 == 5 |
| != | Nie gelyk aan | 5 != 3 |
| > | Groter as | 7 > 4 |
| < | Minder as | 3 < 10 |
| >= | Groter of gelyk aa | 10 >= 9 |
| <= | Minder of gelyk aan | 5 <= 5 |

3. Werklike voorbeelde

Voorbeeld 1 – Puntekontroleerder

grade = 75

if grade >= 50:

print("Jy het geslaag!")

else:

print("Jy het gedruip.")

Voorbeeld 2 – Temperatuurwaarskuwing

temperatuur = 35

if temperatuur > 30: print("Dis 'n warm dag!")

elif temperatuur > 20: print("Dis 'n lekker dag.")

else: print("Dis 'n bietjie koud.")

Voorbeeld 3 – Ewe of Onewe Getal

getal = 7

if getal % 2 == 0: print("Ewe getal")

else: print("Onewe getal")

4. Aktiwiteite

Aktiwiteit 1 – Slaag of Druip

Skep 'n program wat kontroleer of 'n student se telling 50 of meer is. Druk "Slaag" of "Druip".

Oplossing:

telling = int(invoer("Voer jou telling in: "))

if telling >= 50: print("Slaag")

else: print("Druip")

Aktiwiteit 2 – Kontroleer Bestuursouderdom

Vra vir iemand se ouderdom en kontroleer:

• As ouderdom >= 18: druk "Jy kan bestuur."

• Anders: druk "Jy is te jonk om te bestuur."

Oplossing:

ouderdom = int(input("Voer jou ouderdom in: "))

as ouderdom >= 18:

print("Jy kan bestuur.")

else:

print("Jy is te jonk om te bestuur.")

5. Oefeninge

Oefening 1 – Getalvergelyking

Vra die gebruiker vir 'n getal.

Kontroleer of dit groter as 10, gelyk aan 10 of minder as 10 is, en druk 'n boodskap vir elke geval.

Oplossing:

number = int(input("Voer 'n getal in: "))

if number > 10:

print("Die getal is groter as 10.")

elif number == 10:

print("Die getal is gelyk aan 10.")

else:

print("Die getal is minder as 10.")

Oefening 2 – Graadterugvoer Kontroleer die waarde van 'n graad en gee terugvoer:

• 80 en hoër: “Uitstekend”

• 60–79: “Goed”

• 50–59: “Redelik”

• Onder 50: “Druip”

Oplossing:

grade = int(input("Voer jou graad in: "))

if grade >= 80:

print("Uitstekend")

elif grade >= 60:

print("Goed")

elif grade >= 50:

print("Redelik")

else:

druk("Misluip")

Gamifikasie:

Uitdaging van die Dag:

Skep 'n program wat die gebruiker vra om 'n getal in te voer.

As die getal is:

• Ewe: druk "Ewe getal"

• Onewe: druk "Onewe getal"

(Bonus: Laat hulle weer probeer as die invoer nie 'n getal is nie)

Verdien Sterre:

• 1 ster vir die gebruik van if/else

• 1 ster vir die kontrolering van ewe of onewe

Lesopsomming

• Voorwaardelike stellings gebruik if, elif, en else om programbesluite te beheer

• Gebruik vergelykingsoperateurs om voorwaardes te toets

• Jy het geoefen om grade, ouderdom, getalle en meer te kontroleer

Vasvra: Voorwaardelike stellings in Python

1. Watter sleutelwoord word gebruik om 'n voorwaardelike in Python te begin?

A. vir

B. wanneer

C. if

D. lus

2. Wat beteken != in Python?

A. Is gelyk aan

B. Is nie gelyk aan nie

C. Opdrag

D. Word gedeel deur

3. Watter van hierdie is 'n geldige if-stelling?

A. as x == 5

B. as (x = 5)

C. as x !=

D. as dan x = 5

4. Wat sal hierdie kode uitvoer?

x = 3

as x > 5:

print("Groter")

else:

print("Kleiner of Gelyk")

A. Groter

B. Fout

C. Kleiner of Gelyk

D. Niks

5. Watter van hierdie is NIE 'n vergelykingsoperator nie?

A. ==

B. !=

C. //

D. <

Antwoordsleutel:

1 – C

2 – B

3 – A

4 – C

5 – C