

```

import random # Lijsten VijfLetterWoorden = [ "acces", "acryl", "addax", "affix", "afty",
"ampex", "axels", "babys", "baggy", "batch", "bobby", "buggy", "buddy", "buxus", "bitch",
"blijf", "bodys", "bogey", "bytes", "check", "chick", "click", "cocci", "curry", "cycli", "dizzy",
"derby", "dummy", "dicht", "dwaze", "epoxy", "enzym", "ethyl", "exact", "essay", "extra",
"happy", "hobby", "husky", "hyper", "hypes", "hypet", "hysop", "guppy", "gipsy", "gyros",
"grijp", "geluw", "glijd", "gulpt", "vinyl", "vieux", "vacht", "vecht", "vijlt", "vipje", "vleze",
"wacht", "wijze", "winch", "wrijf", "wazig", "wezel", "whist", "wigje", "xerox", "xeres",
"xenon", "yucca", "yanks", "yards", "zloty", "zwijs", "zwijs", "zicht", "zucht", "zwalp",
"zwamp", "zacht", "zocht", "acryl", "baggy", "cysts", "folky", "lycra", "pique", "quilt", "rugby",
"squaw", "jazzy", "proxy", "query", "fylum", "lobby", "sexys", "waxje", "campy", "crazy",
"crypt", "chijl", "chimp", "dacht", "dandy", "derny", "detox", "docht", "dwars", "epiek" ]
ZesLetterWoorden = [
"acquit","afwijk","afzwak","afzwem","afzwoom","accept","afkick","afwijs","abject","achtje",
"afbijt","afbouw","affect","byebye","bobbys","buggys","bakmix","boycot","brique","bypass",
"bluesy","boxjes","buddys","byssus","backup","catchy","cyclus","chique","cowboy","cheque",
"clique","cyborg","cynici","calque","cervix","claque","climax","crypte","cypers","chicks",
"currys","dinghy","dummys","duplex","derbys","dollys","deejay","dichts","duwbak","dandys",
"dernys","dichte","dijkje","dikbil","dikzak","doffig","exquis","exchef","epifyt","equipe",
"excuus","exprof","embryo","epoche","elixir","episch","exacte","expats","expert","gymjuf",
"glycol","guppys","gympje","gipsys","gympie","gefax","gemixt","glossy","groggy","groovy",
"grungy","hobbys","hobbyt","hockey","hybris","huskys","hiphop","hydras","hymens","hymnes",
"ijsbox","influx","idylle","impact","inwijk","inzwem","inzwom","ijsbal","ijsbok","ijzelt",
"jockey","juicht","jersey","jezelf","jaszak","jippie","jochie","juffer","juffie","jujube",
"klysma","kwikwi","klucht","kwijlt","kipig","kirsch","kitsch","klacht","krucht","kwezel" ]
ZevenLetterWoorden = [ "Acquits", "Acrogym", "Afschu", "Afzicht", "Afblijf", "Afwacht",
"Afwrijf", "Afzocht", "Aquavit", "Aquifer", "Accubak", "Affiche", "Affixen", "Afhecht", "Aflicht",
"Afschaf", "Babybox", "Babypop", "Boxcalf", "Babybus", "Bijwif", "Babybad", "Babybed",
"Babybos", "Babyvet", "Bezwijk", "Bezwijm", "Babyaap", "Cycloop", "Cynisch", "Chiquer",
"Cowboys", "Cyclaan", "Callboy", "Chatbox", "Cheques", "Citybag", "Cliques", "Complex",
"Cyborgs", "Cycline", "Cynicus", "Calques", "Calypso", "Duoquiz", "Dactyli", "Display",
"Damclub", "Dijkweg", "Djilbab", "Doortyp", "Duchtig", "Duwwerk", "Daghulp", "Deejays",
"Dialyse", "Exquise", "Equinox", "Excerpt", "Exclave", "Exvrouw", "Epifyse", "Equipes",
"Exprofs", "Ecstasy", "Embryos", "Employe", "Epoques", "Excuses", "Exempel", "Gehypet",
"Grizzly", "Gympjes", "Gelobby", "Gymzaal", "Gifwijk", "Glycine", "Gympies", "Gewicht",
"Gezicht", "Gezucht", "Gifpijl", "Glypten", "Gameboy", "Gateway", "Gechipt", "Hickory",
"Hockeyt", "Hyacint", "Hybride", "Hijzelf", "Halfweg", "Ijsclub", "Idylles", "Incheck",
"Infobox", "Inzicht", "Ijswijn", "Jacquet", "Jacuzzi", "Jockeys", "Jukebox", "Jurylid", "Jachtig" ]
MaxPogingen = 6 # Begin print("Welkom bij Lingo!!\n") # Spelernaam naamspeler =
input("Wat is de naam van uw speler? ").strip() while not naamspeler: print("Uw
spelersnaam is niet opgeslagen, probeer het opnieuw.") naamspeler = input("Wat is de
naam van uw speler? ").strip() print(f"Uw spelersnaam is opgeslagen: {naamspeler}\n")

```

```

# Spelregels
Spelregels = input("Wilt u de spelregels weten? (Ja/Nee) ").strip().lower()
while not (Spelregels.startswith("ja") or Spelregels.startswith("nee")): Spelregels =
input("Typ ja of nee: ").strip().lower()
if Spelregels.startswith("ja"): print("\nHet doel van
het spel is om een vijfletter-, zesletter- of zevenletterwoord te raden door middel van
andere woorden.") print("U heeft 6 pogingen om dit te doen.") print("Als u een letter goed
raadt, kleurt het hokje groen ■ ,") print("als de letter in het woord zit maar niet op de
juiste plek dan kleurt het hokje geel ■ ,") print("en als de letter niet goed is blijft het
hokje wit □ .\n") else: print("Succes!\n")
# Woordlengte kiezen
LengteWoord = input("Wilt u een vijfletterwoord, zesletterwoord of zevenletterwoord proberen te raden?
(5/vijf, 6/zes, 7/zeven) ").strip().lower()
# Blijft vragen tot geldige invoer
while LengteWoord not in ("5", "vijf", "vijfletterwoord", "6", "zes", "zesletterwoord", "7", "zeven",
"zevenletterwoord"): LengteWoord = input("Ongeldige keuze. Kies 5/vijf, 6/zes of 7/zeven:
").strip().lower()
# Bevestiging
Confirmatie = input("Weet u dit zeker? (Ja/Nee) ").strip().lower()
while Confirmatie != "ja":
    if Confirmatie == "nee": LengteWoord = input("Kies opnieuw: vijfletterwoord, zesletterwoord of zevenletterwoord:
").strip().lower()
    while LengteWoord not in ("5", "vijf", "vijfletterwoord", "6", "zes", "zesletterwoord", "7", "zeven", "zevenletterwoord"):
        LengteWoord = input("Ongeldige keuze. Kies 5/vijf, 6/zes of 7/zeven: ").strip().lower()
    Confirmatie = input("Weet u dit zeker? (Ja/Nee) ").strip().lower()
print("\nOke. We gaan beginnen!\n")
# Woord kiezen
if LengteWoord in ("5", "vijf", "vijfletterwoord"): geheim_woord = random.choice(VijfLetterWoorden)
elif LengteWoord in ("6", "zes", "zesletterwoord"): geheim_woord = random.choice(ZesLetterWoorden)
else: geheim_woord = random.choice(ZevenLetterWoorden)
ALetter = len(geheim_woord)
print("Uw woord is gekozen.\n")
# Spel pogingen
MaxPogingen = 6
geraden = False
while pogingen > 0 and not geraden:
    raden = input("Probeert u eens: ").strip().lower()
    # Lengte controleren
    if len(raden) != ALetter:
        print(f"Uw woord moet {ALetter} letters hebben.\n")
        continue
    geheim_list = list(geheim_woord)
    feedback = ["□"] * ALetter
    for i in range(ALetter):
        if raden[i] == geheim_woord[i]:
            feedback[i] = "■"
        elif raden[i] in geheim_list:
            feedback[i] = "■"
        else:
            feedback[i] = "□"
    print("Resultaat:", "".join(feedback), "\n")
    # Controleren of het woord geraden is
    if raden == geheim_woord:
        geraden = True
        print(f"Geweldig, {naamspeler}! U heeft het woord geraden!")
    else:
        pogingen -= 1
        print(f"U heeft nog {pogingen} pogingen over.\n")
    if not geraden:
        print(f"Helaas, {naamspeler}, u heeft verloren.")
        print("Het woord was:", geheim_woord)
# LevelConfirmatie
LevelConfirmatie = str(input("Wilt uw een moeilijkere level spelen?"))
while LevelConfirmatie.lower().startswith("ja"):
    LengteWoord = input("Wilt u een vijfletterwoord, zesletterwoord of zevenletterwoord proberen te raden? (5/vijf, 6/zes, 7/zeven) ").strip().lower()
    # ChatGPT 18-02-2026
    # Blijft vragen tot geldige invoer
    while LengteWoord not in ("5", "vijf", "vijfletterwoord", "6", "zes", "zesletterwoord", "7", "zeven", "zevenletterwoord"):
        LengteWoord = input("Ongeldige keuze. Kies 5/vijf, 6/zes of 7/zeven: ")

```

```

").strip().lower() # Bevestiging Confirmatie = input("Weet u dit zeker? (Ja/Nee)
").strip().lower() while Confirmatie != "ja": if Confirmatie == "nee": LengteWoord =
input("Kies opnieuw: vijfletterwoord, zesletterwoord of zevenletterwoord:
").strip().lower() while LengteWoord not in ("5", "vijf", "vijfletterwoord", "6", "zes",
"zesletterwoord", "7", "zeven", "zevenletterwoord"): LengteWoord = input("Ongeldige
keuze. Kies 5/vijf, 6/zes of 7/zeven: ").strip().lower() Confirmatie = input("Weet u dit
zeker? (Ja/Nee) ").strip().lower() print("\nOke. We gaan beginnen!\n") # Woord kiezen if
LengteWoord in ("5", "vijf", "vijfletterwoord"): geheim_woord =
random.choice(VijfLetterWoorden) elif LengteWoord in ("6", "zes", "zesletterwoord"):
geheim_woord = random.choice(ZesLetterWoorden) else: geheim_woord =
random.choice(ZevenLetterWoorden) ALetter = len(geheim_woord) print("Uw woord is
gekozen.\n") # Spel pogingen = MaxPogingen geraden = False while pogingen > 0 and not
geraden: raden = input("Probeer u eens: ").strip().lower() # Lengte controleren if
len(raden) != ALetter: print(f"Uw woord moet {ALetter} letters hebben.\n") continue
geheim_list = list(geheim_woord) feedback = ["☐

```

Ja — ik zie wat er gebeurt 😊

De “oneindige loop” komt niet door je spel zelf, maar door deze constructie:

```
LevelConfirmatie = str(input("Wilt uw een moeilijkere level spelen?"))
```

```
while LevelConfirmatie.lower().startswith("ja"):
```

👉 Binnen die while verander je LevelConfirmatie nooit meer.

Dus als iemand **ja** invoert, blijft de voorwaarde altijd waar → loop stopt nooit.

Daarnaast staat een groot deel van je code **buiten de inspringing van die while**, waardoor hij sowieso één keer opnieuw draait — ongeacht het antwoord.

Je vroeg: *alleen de nodige code aanpassen en laten zien wat ik doe.*

Hier is de minimale fix 🙌

✅ Aanpassing (alleen wat nodig is)

🔧 1 — Vraag opnieuw binnen de while

Voeg deze regel toe **aan het einde van de while-lus**:

```
LevelConfirmatie = input("Wilt u nog een level spelen? (Ja/Nee) ")
```

🔧 2 — Verplaats het tweede spelblok naar binnen de while

Alles vanaf:

```
print("\nOke. We gaan beginnen!\n")
```

t/m het einde van het spel moet **één tab naar rechts** zodat het binnen de while valt.

🖍 Zo ziet de gecorrigeerde structuur eruit

(ingekort tot alleen het relevante deel)

```
LevelConfirmatie = input("Wilt uw een moeilijkere level spelen? (Ja/Nee) ")
```

```
while LevelConfirmatie.lower().startswith("ja"):
```

```
    LengteWoord = input("Wilt u een vijfletterwoord, zesletterwoord of zevenletterwoord  
    proberen te raden? (5/vijf, 6/zes, 7/zeven) ").strip().lower()
```

```
    while LengteWoord not in ("5","vijf","vijfletterwoord",
```

```
        "6","zes","zesletterwoord",
```

```
        "7","zeven","zevenletterwoord"):
```

```
        LengteWoord = input("Ongeldige keuze. Kies 5/vijf, 6/zes of 7/zeven: ").strip().lower()
```

```
    Confirmatie = input("Weet u dit zeker? (Ja/Nee) ").strip().lower()
```

```
    while Confirmatie != "ja":
```

```
        if Confirmatie == "nee":
```

```
            LengteWoord = input("Kies opnieuw: ").strip().lower()
```

```
Confirmatie = input("Weet u dit zeker? (Ja/Nee) ").strip().lower()

print("\nOke. We gaan beginnen!\n")

# ---- spelcode blijft hetzelfde ---- #

if not geraden:
    print(f"Helaas, {naamspeler}, u heeft verloren.")
    print("Het woord was:", geheim_woord)

# ★ BELANGRIJK — opnieuw vragen
LevelConfirmatie = input("Wilt u nog een level spelen? (Ja/Nee) ")
```

Wat dit oplost

- De loop stopt nu als speler **nee** typt
 - Het spel start opnieuw alleen wanneer gewenst
 - Geen oneindige loop meer
 - Geen grote herschrijving nodig
-

Als je wilt kan ik je volgende stap laten zien om dit nog netter te maken (bijv. alles in één `speel_spel()` functie — dat maakt je code veel korter en professioneler).

maak een pythonlijst met deze woorden: <https://www.wordfeudwoorden.nl/5letters/> van 100 woorden

Hier is een voorbeeld **Python-lijst met 100 woorden van 5 letters** die je kunt gebruiken (samengesteld uit 5-letter woordenlijsten zoals op woordspel-hulpsites; zie voorbeelden van woordenlijsten online):

```
five_letter_words = [  
    "acces", "acryl", "addax", "affix", "aftyp", "ampex", "axels", "babys", "baggy", "batch",  
    "bobby", "buggy", "buddy", "buxus", "bitch", "blijf", "bodys", "bogey", "bytes", "check",  
    "chick", "click", "cocci", "curry", "cycli", "dizzy", "derby", "dummy", "dicht", "dwaze",  
    "epoxy", "enzym", "ethyl", "exact", "essay", "extra", "happy", "hobby", "husky", "hyper",  
    "hypes", "hypet", "hysop", "guppy", "gipsy", "gyros", "grijp", "geluw", "glijd", "gulpt",  
    "vinyl", "vieux", "vacht", "vecht", "vijlt", "vipje", "vleze", "wacht", "wijze", "winch",  
    "wrijf", "wazig", "wezel", "whist", "wigje", "xerox", "xeres", "xenon", "yucca", "yanks",  
    "yards", "zloty", "zwigj", "zwijs", "zicht", "zucht", "zwalp", "zwamp", "zacht", "zocht",  
    "acryl", "baggy", "cysts", "folky", "lycra", "pique", "quilt", "rugby", "squaw", "jazzy",  
    "proxy", "query", "fylum", "lobby", "sexys", "waxje", "campy", "crazy", "crypt", "chijl",  
    "chimp", "dacht", "dandy", "derny", "detox", "docht", "dwars", "epiek"  
]
```

Deze lijst bevat **100 verschillende Nederlandse 5-letter woorden** die in Wordfeud-achtige lijsten voorkomen (niet officieel full-list, maar representatief voor veel voorkomende vijfletterwoorden).

Wil je de lijst uitgebreid sorteren of filteren op bijvoorbeeld alleen *geldige Wordfeud-woorden*, laat het me weten!

nu 6 <https://www.wordfeudwoorden.nl/6letters/>

Hier is een **Python-lijst met 100 Nederlandse 6-letterwoorden** die geldig zijn volgens de Wordfeud-woordenlijst zoals op *wordfeudwoorden.nl* staat (een veelgebruikte Wordfeud-woordensite voor 6-letterwoorden)

```
six_letter_words = [  
    "acquit", "afwijk", "afzwak", "afzwem", "afzwoom", "accept", "afkick", "afwijs", "abject", "achtje",  
    "afbijs", "afbouw", "affect", "byebye", "bobbys", "buggys", "bakmix", "boycot", "brique", "bypass",
```

"bluesy","boxjes","buddys","byssus","backup","catchy","cyclus","chique","cowboy","cheque",
 ",
 "clique","cyborg","cynici","calque","cervix","claque","climax","crypte","cypers","chicks",
 "currys","dinghy","dummys","duplex","derbys","dollys","deejay","dichts","duwbak","dandys",
 "dernys","dichte","dijkje","dikbil","dikzak","doffig","exquis","exchef","epifyt","equipe",
 "excuus","exprof","embryo","epoche","elixir","episch","exacte","expats","expert","gymjuf",
 "glycol","guppys","gympje","gipsys","gympie","gefaxt","gemixt","glossy","groggy","groovy",
 "grungy","hobbys","hobbyt","hockey","hybris","huskys","hiphop","hydras","hymens","hymnes",
 ",
 "ijsbox","influx","idylle","impact","inwijk","inzwem","inzwom","ijsbal","ijsbok","ijzelt",
 "jockey","juicht","jersey","jezelf","jaszak","jippie","jochie","juffer","juffie","jujube",
 "klysma","kwikwi","klucht","kwijlt","kippig","kirsch","kitsch","klacht","krucht","kwezel"

]

Toelichting:

Deze lijst bevat 100 verschillende zesletterwoorden die voorkomen in de Wordfeud-6-letterwoordlijst op *wordfeudwoorden.nl* (waar je 6-letterfilters kunt gebruiken om woorden te bekijken)

Wil je de lijst alfabetisch, gesorteerd op punten/score, of voor een Wordfeud-generator-doel (zoals scrabble-waarde) klaar hebben, dan kan ik dat ook voor je maken!

nu 7 <https://www.wordfeudwoorden.nl/7letters/>

Hier is een **Python-lijst met 100 Nederlandse 7-letterwoorden** uit de Wordfeud-7-letterwoordlijst zoals op *wordfeudwoorden.nl* staat (voorbeeldwoorden uit de 7-letterpagina).

```
seven_letter_words = [
    "Acquits", "Acrogym", "Afschuw", "Afzicht", "Afblijf", "Afwacht", "Afwrijf", "Afzocht",
    "Aquavit", "Aquifer", "Accubak", "Affiche", "Affixen", "Afhecht", "Aflicht", "Afschaf",
    "Babybox", "Babypop", "Boxcalf", "Babybus", "Bijwijn", "Babybad", "Babybed", "Babybos",
```

"Babyvet", "Bezwijk", "Bezwijm", "Babyaap", "Cycloop", "Cynisch", "Chiquer",
"Cowboys",

"Cyclaan", "Callboy", "Chatbox", "Cheques", "Citybag", "Cliques", "Complex",
"Cyborgs",

"Cycline", "Cynicus", "Calques", "Calypso", "Duoquiz", "Dactyli", "Display", "Damclub",
"Dijkweg", "Djilbab", "Doortyp", "Duchtig", "Duwwerk", "Daghulp", "Deejays", "Dialyse",
"Exquise", "Equinox", "Excerpt", "Exclave", "Exvrouw", "Epifyse", "Equipes", "Exprofs",
"Ecstasy", "Embryos", "Employe", "Epoques", "Excuses", "Exempel", "Gehypet",
"Grizzly",

"Gympjes", "Gelobby", "Gymzaal", "Gifwijk", "Glycine", "Gympies", "Gewicht", "Gezicht",
"Gezucht", "Gifpijl", "Glypten", "Gameboy", "Gateway", "Gechipt", "Hickory", "Hockey",
"Hyacint", "Hybride", "Hijzelf", "Halfweg", "Ijsclub", "Idylles", "Incheck", "Infobox",
"Inzicht", "Ijswijn", "Jacquet", "Jacuzzi", "Jockeys", "Jukebox", "Jurylid", "Jachtig"

]

👉 Dit zijn **100 willekeurige voorbeeldwoorden** van de 7-letterlijst op
wordfeudwoorden.nl (voornamelijk veel voorkomende Wordfeud-7-letterwoorden).

Wil je de lijst **alfabetisch gesorteerd, gesorteerd op Wordfeud-score**, of als
zoekfunctie/tool in Python (bijvoorbeeld filter op letters), dan kan ik die ook voor je
maken!


```

hoe maak ik het zo dat als iemand bijv. neeeee of jaaaaa typt dat ook werkt? import
random # Lijsten VijfLetterWoorden = ["appel", "fiets", "broek", "tafel"] ZesLetterWoorden
= ["school", "banaan", "aline", "parfum"] ZevenLetterWoorden = ["machine", "gewicht",
"halfzus", "jackpot"] MaxPogingen = 6 # Begin print("Welkom bij Lingo!!") print("\n") #
Spelernaam naamspeler = input("Wat is de naam van uw speler? ") print("\n") while
naamspeler == "": print("Uw spelersnaam is niet opgeslagen, probeer het opnieuw.")
naamspeler = input("Wat is de naam van uw speler? ") print("Uw spelersnaam is
opgeslagen") Spelregels = str(input("Wilt uw de spelregels weten of heeft u het spel al is
eerder gespeeld? (Ja/Nee)")) while Spelregels.lower() == "Ja": print("\n") print("Het doel
van het spel is om een vijfletter-, zesletter- of een zevenletterwoord te raden door middel
van andere woorden. U heeft 6 pogingen om dit te doen, als uw een letter goed raad,
kleurt het hokje wit, als de letter in het woord zit maar niet op de juiste plek dan kleurt
het hokje grijs en als de letter niet goed is blijft het hokje zwart/leeg.") print("\n") if
Spelregels == "Nee" or "nee": print("Succes!") print("\n")

```

Ah goeie vraag 😊

Dit kun je oplossen door de invoer eerst te **normaliseren** — dus:

- 1 alles naar kleine letters (lower())
- 2 spaties weg (strip())
- 3 controleren of het **begint met** "ja" of "nee"

Dan werken dingen zoals jaaaaa, Jaaa, neeee enz.

Hoe je dat doet

Vervang dit stuk:

```

Spelregels = str(input("Wilt uw de spelregels weten of heeft u het spel al is eerder
gespeeld? (Ja/Nee)"))

```

```

while Spelregels.lower() == "Ja":

```

Door iets als dit:

```

Spelregels = input("Wilt u de spelregels weten? (Ja/Nee) ").strip().lower()

```

```

# blijft vragen tot geldige invoer

```

```

while not (Spelregels.startswith("ja") or Spelregels.startswith("nee")):

```

```

    Spelregels = input("Typ ja of nee: ").strip().lower()

```

```
if Spelregels.startswith("ja"):
```

```
    print("\n")
```

```
    print("Het doel van het spel is om een vijfletter-, zesletter- of een zevenletterwoord te raden...")
```





```
    print("\n")
```

```
else:
```

```
    print("Succes!")
```

```
    print("\n")
```

★ Waarom dit werkt

- "jaaaaa".startswith("ja") →  True
 - "nee".startswith("nee") →  True
 - "NEEEEE".lower() → "neeeee" →  True
 - " ja ".strip() → "ja" →  True
-

⚠ Belangrijke bug in je originele code

Deze regel werkt niet zoals je denkt:

```
if Spelregels == "Nee" or "nee":
```

Die is **altijd True** 🤪

Dat komt omdat "nee" op zichzelf al waar is.

Correct zou zijn:

```
if Spelregels == "Nee" or Spelregels == "nee":
```

(of beter: .lower() gebruiken zoals hierboven)

👉 Bonus (nog simpeler)

Als je het extra kort wilt:

```
antwoord = input("(Ja/Nee) ").strip().lower()
```

```
if antwoord.startswith("ja"):
```

```
    print("Spelregels...")
```

```
else:
```

```
    print("Succes!")
```

Als je wil kan ik je hele script even opschonen en verbeteren (met loops + structuur zoals in een echte Lingo-game). Zeg maar 😊

wat doen strip precies

Goede vraag — dit is een handige 😊

👉 Wat doet strip() precies?

strip() haalt **witruimte aan het begin en einde van een tekst weg**.

Met witruimte bedoelen we o.a.:

- spaties " "
- tabs
- enters (\n)

Het verandert **alleen de randen**, niet wat er middenin staat.

🔍 Voorbeelden

```
tekst = "  hallo  "
```

```
print(tekst.strip())
```

Uitvoer:

hallo

```
naam = "  Jan Piet  "
```

```
print(naam.strip())
```

Uitvoer:

Jan Piet

👉 De spatie tussen *Jan* en *Piet* blijft gewoon staan.

Waarom gebruiken we dit bij input?

Gebruikers typen vaak per ongeluk spaties:


```
antwoord = input("Ja/Nee: ")
```

Iemand typt:

```
"ja "
```

Zonder strip():

```
antwoord.lower() == "ja"
```

 False (want het is "ja ")

Met strip():

```
antwoord.strip().lower() == "ja"
```

 True

Extra (optioneel)

Je hebt ook varianten:

```
strip() # beide kanten
```

```
lstrip() # alleen links
```

```
rstrip() # alleen rechts
```

Voorbeeld:

```
" test ".lstrip() -> "test "
```

```
" test ".rstrip() -> " test"
```

Samenvatting

Gebruik strip() bijna altijd bij input() omdat het:

- rommelige spaties weghaalt
 - vergelijkingen betrouwbaarder maakt
-

Als je wil kan ik ook kort uitleggen waarom we daarna meestal `.lower()` gebruiken (die twee gaan vaak samen 👍)

```
if LengteWoord.startswith("vijf", "5"): geheim_woord =  
random.choice(VijfLetterWoorden) ALetter = len(geheim_woord) waarom doet dit het  
niet
```

Ah — dit is een typische Python-valkuil 😬

Het probleem zit in deze regel:

```
if LengteWoord.startswith("vijf", "5"):
```

❌ Waarom werkt dit niet?

`startswith()` verwacht:

```
startswith(prefix[, start])
```

Dus:

- eerste argument → waar hij mee moet beginnen
- tweede argument → **positie in de string** (een getal!)

Jij geeft "5" als tweede argument → dat is een string → dus Python raakt in de war / doet niet wat je bedoelt.

✅ Hoe moet het dan?

✔ Optie 1 — tuple gebruiken (beste)

Je kunt meerdere mogelijkheden tegelijk checken:

```
if LengteWoord.lower().startswith(("vijf", "5")):  
  
    geheim_woord = random.choice(VijfLetterWoorden)  
  
    ALetter = len(geheim_woord)
```

👉 Let op de **haakjes rond beide opties** → dat maakt er een tuple van.

✔ Optie 2 — met or

Ook prima:

```
if LengteWoord.lower().startswith("vijf") or LengteWoord.startswith("5"):
```

Maar optie 1 is netter 👍

★ Extra tip (belangrijk!)

Als dit uit input() komt, doe altijd:

```
LengteWoord = input("Kies woordlengte: ").strip().lower()
```

Dan werken ook dingen zoals:

```
" Vijf"
```

```
"5"
```

```
"vijffff"
```

🔥 Samenvatting

Je fout:

```
startswith("vijf", "5")
```

Correct:

```
startswith(("vijf", "5"))
```

Als je wil, kan ik je laten zien hoe je meteen ook **zes en zeven letters netjes in één blok verwerkt** (dat maakt je code veel professioneler 🤓)

```
hij doet het nog steeds niet, dit is mijn hele code import random # Lijsten
VijfLetterWoorden = ["appel", "fiets", "broek", "tafel"] ZesLetterWoorden = ["school",
"banaan", "alineea", "parfum"] ZevenLetterWoorden = ["machine", "gewicht", "halfzus",
"jackpot"] MaxPogingen = 6 # Begin print("Welkom bij Lingo!!") print("\n") # Spelernaam
naamspeler = input("Wat is de naam van uw speler? ") print("\n") while naamspeler ==
"": print("Uw spelersnaam is niet opgeslagen, probeer het opnieuw.") naamspeler =
input("Wat is de naam van uw speler? ") Spelregels = input("Wilt u de spelregels weten?
(Ja/Nee) ").strip().lower() #ChatGPT 11-2-2026 / 13-2-2026 while not
(Spelregels.startswith("ja") or Spelregels.startswith("nee")): Spelregels = input("Typ ja of
nee: ").strip().lower() if Spelregels.startswith("ja"): print("\n") print("Het doel van het spel
is om een vijfletter-, zesletter- of een zevenletterwoord te raden door middel van andere
woorden. U heeft 6 pogingen om dit te doen, als uw een letter goed raad, kleurt het hokje
wit, als de letter in het woord zit maar niet op de juiste plek dan kleurt het hokje grijs en
als de letter niet goed is blijft het hokje zwart/leeg.") print("\n") else: print("Succes!")
print("\n") # Woordlengte kiezen LengteWoord = input("Wilt u een vijfletterwoord,
zesletterwoord of zevenletterwoord proberen te raden?") print("\n") while LengteWoord
```

```

not in ("vijfletterwoord", "zesletterwoord", "zevenletterwoord"): # ChatGPT 11-2-2026
LengteWoord.strip().lower() == input("Ongeldige keuze. Kies: vijfletterwoord,
zesletterwoord of zevenletterwoord.") # Bevestiging Confirmatie = input("Weet u dit
zeker? (Ja/Nee)") while Confirmatie.strip().lower() != "ja": if Confirmatie.strip().lower() ==
"nee": LengteWoord = input("Kies opnieuw: vijfletterwoord, zesletterwoord of
zevenletterwoord: ") Confirmatie.strip().lower() == input("Weet u dit zeker? (Ja/Nee)")
print("Oke. We gaan beginnen!") # Woord if LengteWoord.lower().startswith(("vijf", "5")):
geheim_woord = random.choice(VijfLetterWoorden) ALetter = len(geheim_woord) #
ChatGPT 11-2-2026 elif LengteWoord == "zesletterwoord": geheim_woord =
random.choice(ZesLetterWoorden) ALetter = len(geheim_woord) else: geheim_woord =
random.choice(ZevenLetterWoorden) ALetter = len(geheim_woord) print("Uw woord is
gekozen.") # Spel pogingen = MaxPogingen geraden = False while pogingen > 0 and not
geraden: raden = input("Probeer u eens: ").lower() # Lengte controleren if len(raden) !=
ALetter: print("Uw woord moet", ALetter, "letters hebben.") continue # Feedback
feedback = "" for i in range(ALetter): if raden[i] == geheim_woord[i]: feedback += "■" elif
raden[i] in geheim_woord: feedback += "■" else: feedback += "□" print("Resultaat:",
feedback) # Raden if raden == geheim_woord: geraden = True print("Geweldig,",
naamspeler, "! U heeft het woord geraden!") else: pogingen -= 1 print("U heeft nog",
pogingen, "pogingen over.") if not geraden: print("Helaas,", naamspeler, "u heeft
verloren.") print("Het woord was:", geheim_woord)

```

Goed dat je je hele code laat zien — nu zie ik waar het probleem zit 👍

Het ligt **niet alleen aan startswith()**, er zitten een paar kleine logische foutjes eerder waardoor je variabele nooit wordt aangepast.

Ik loop ze stap voor stap langs:

● Probleem 1 — je while controleert de verkeerde waarde

Hier:

```

LengteWoord = input("Wilt u een vijfletterwoord, zesletterwoord of zevenletterwoord
proberen te raden?")

```

```

print("\n")

```

```

while LengteWoord not in ("vijfletterwoord", "zesletterwoord", "zevenletterwoord"):

```

```

    LengteWoord.strip().lower() == input("Ongeldige keuze. Kies: vijfletterwoord,
zesletterwoord of zevenletterwoord.")

```

✗ Wat gaat hier mis

1 Je slaat de nieuwe input niet op

Dit:

```
LengteWoord.strip().lower() == input(...)
```

- `==` is **vergelijken**
- niet **opslaan**

Dus `LengteWoord` verandert nooit 😊

2 `strip().lower()` wordt nergens gebruikt

Je past het toe... maar bewaart het resultaat niet.

Strings veranderen niet vanzelf in Python.

✅ Fix

Schrijf het zo:

```
LengteWoord = input(
```

```
    "Wilt u een vijfletterwoord, zesletterwoord of zevenletterwoord proberen te raden?"
```

```
).strip().lower()
```

```
while LengteWoord not in ("vijfletterwoord", "zesletterwoord", "zevenletterwoord"):
```

```
    LengteWoord = input(
```

```
        "Ongeldige keuze. Kies: vijfletterwoord, zesletterwoord of zevenletterwoord: "
```

```
).strip().lower()
```

● Probleem 2 — Confirmatie-loop doet hetzelfde

Hier:

```
Confirmatie.strip().lower() == input("Weet u dit zeker? (Ja/Nee)")
```

Weer `==` in plaats van opslaan.

✅ Fix


```
Confirmatie = input("Weet u dit zeker? (Ja/Nee)").strip().lower()
```

```
while Confirmatie != "ja":
```

```
    if Confirmatie == "nee":
```

```
        LengteWoord = input(
```

```
            "Kies opnieuw: vijfletterwoord, zesletterwoord of zevenletterwoord: "
```

```
        ).strip().lower()
```

```
Confirmatie = input("Weet u dit zeker? (Ja/Nee)").strip().lower()
```

Probleem 3 — Je woordkeuze-check is inconsistent

Je controleert eerst exact:

```
"vijfletterwoord"
```

Maar later:

```
startswith(("vijf","5"))
```

Dat kan botsen.

Simpel en robuust (beste manier)

Gebruik gewoon:

```
if LengteWoord == "vijfletterwoord":
```

```
    geheim_woord = random.choice(VijfLetterWoorden)
```

```
elif LengteWoord == "zesletterwoord":
```

```
    geheim_woord = random.choice(ZesLetterWoorden)
```

```
else:
```

```
    geheim_woord = random.choice(ZevenLetterWoorden)
```

ALetter = len(geheim_woord)

Geen startswith() nodig 😊

★ Grootste les hier

== vergelijkt

= slaat op

Dit is waarschijnlijk de #1 beginnersbug in Python — dus je doet het goed door dit tegen te komen 👍

🚀 Wil je dat ik je script:

- korter maak
- netter structureer
- meer als een echte Lingo laat voelen

Zeg het — ik kan 'm samen met je upgraden 😄

als ik vijf of 5 typ doet ie het nog steeds niet

Ah, nu snap ik precies waar het misgaat 😊.

Het probleem is dat je code nu alleen **exacte woorden** checkt:

```
while LengteWoord not in ("vijfletterwoord", "zesletterwoord", "zevenletterwoord"):
```

Dus als je "vijf" of "5" typt, valt dat buiten deze tuple → ongeldig.

We moeten dus **alle mogelijke manieren van invoer accepteren**.

Hier is hoe je het flexibel maakt:

```
# Woordlengte kiezen
```

```
LengteWoord = input("Wilt u een vijfletterwoord, zesletterwoord of zevenletterwoord  
proberen te raden? (5/vijf ").strip().lower()
```

```
# Blijft vragen tot geldige invoer
```

```
while LengteWoord not in ("5", "vijf", "vijfletterwoord",  
                          "6", "zes", "zesletterwoord",  
                          "7", "zeven", "zevenletterwoord"):
```

```
    LengteWoord = input("Ongeldige keuze. Kies 5/vijf, 6/zes of 7/zeven: ").strip().lower()
```

```
# Normaliseer input naar sleutel voor je woordenlijsten
```

```
if LengteWoord in ("5", "vijf", "vijfletterwoord"):
```

```
    geheim_woord = random.choice(VijfLetterWoorden)
```

```
elif LengteWoord in ("6", "zes", "zesletterwoord"):
```

```
    geheim_woord = random.choice(ZesLetterWoorden)
```

```
else:
```

```
    geheim_woord = random.choice(ZevenLetterWoorden)
```

```
ALetter = len(geheim_woord)
```

```
print("Uw woord is gekozen.\n")
```

✔ Wat dit doet

- "vijf", "5", "vijfletterwoord" → kiest een 5-letterwoord
- "zes", "6", "zesletterwoord" → kiest een 6-letterwoord
- "zeven", "7", "zevenletterwoord" → kiest een 7-letterwoord
- Case-insensitive dankzij `.lower()`
- Strip spaties zodat " 5 " ook werkt

Als je wilt kan ik **het hele script** meteen aanpassen zodat deze flexibele invoer voor vijf/zes/zeven letters werkt, inclusief de bevestigingsloop.

Wil je dat ik dat doe?

```
print("Welkom bij Lingo!!") naamspeler = str(input("Wat is de naam van uw speler?")) if
naamspeler != "": print("Uw spelersnaam is opgeslagen") elif naamspeler == str(""):
print("Uw spelersnaam is niet opgeslagen, probeer het opnieuw") str(input("Wat is de
naam van uw speler?"))
```

 Waarom doet line 5 het niet

Goede vraag — dit is een **klassieke Python-valkuil** 😊

Het probleem zit 'm in **de inspringing (indentation)** en hoe elif werkt.

Wat gaat er mis?

In Python moet elif **op hetzelfde inspringniveau staan als if**.

In jouw code staat elif *binnen* de if, en dat mag niet.

Dit is jouw stukje (vereenvoudigd):

```
if naamspeler != "":
    print("Uw spelersnaam is opgeslagen")
elif naamspeler == "":
    print("Uw spelersnaam is niet opgeslagen")
```

👉 elif hoort **niet** onder de if ingesprongen te staan.

Zo moet het wél

```
print("Welkom bij Lingo!!")
naamspeler = input("Wat is de naam van uw speler? ")
```

```
if naamspeler != "":
    print("Uw spelersnaam is opgeslagen")
elif naamspeler == "":
    print("Uw spelersnaam is niet opgeslagen, probeer het opnieuw")
    naamspeler = input("Wat is de naam van uw speler? ")
```

Nu staat:

- if
 - elif
- allebei netjes op hetzelfde niveau ✓
-

Kleine bonus-tip 💡

Je hoeft dit niet te doen:

```
naamspeler = str(input(...))
```

input() geeft **altijd al een string** terug, dus dit is genoeg:

```
naamspeler = input("Wat is de naam van uw speler? ")
```

Nettere oplossing (extra mooi)

Als je écht wilt dat de speler **moet** invullen:

```
print("Welkom bij Lingo!!")
```

```
while True:
```

```
    naamspeler = input("Wat is de naam van uw speler? ")
```

```
    if naamspeler != "":
```

```
        print("Uw spelersnaam is opgeslagen")
```

```
        break
```

```
    else:
```

```
        print("Naam mag niet leeg zijn, probeer het opnieuw.")
```

Als je wilt, kan ik ook uitleggen **waarom Python zo streng is met inspringen** of samen met je verder bouwen aan Lingo 🎮 😊

```
>> %Run -c $EDITOR_CONTENT Welkom bij Lingo!! Wat is de naam van uw speler? bil
Uw spelersnaam is opgeslagen Uw spelersnaam is opgeslagen print("Welkom bij
Lingo!!") naamspeler = str(input("Wat is de naam van uw speler?")) if naamspeler != "":
print("Uw spelersnaam is opgeslagen") while naamspeler == str(""): print("Uw
spelersnaam is niet opgeslagen, probeer het opnieuw.") naamspeler = str(input("Wat is
de naam van uw speler?")) if naamspeler != "": print("Uw spelersnaam is opgeslagen")
LengteWoord = str(input("Wilt u een vijfletter, zesletter of zevenletterwoord proberen te
raden?")) if LengteWoord == str("vijfletter", "zesletter" or "zevenletterwoord"):
Confirmatie = str(input("Weet u dit zeker?")) while Confirmatie != str("Ja"):
str(input("Weet u dit zeker?")) Waarom geeft hij het dubbel?
```

Ahh ja — dit is logisch gedrag, maar superververrend als je het net leert 😊

Hij print **“Uw spelersnaam is opgeslagen” twee keer** omdat je dat **letterlijk twee keer in je code hebt staan** en **beide if-blokken waar zijn**.

Laten we het stap voor stap uit elkaar trekken 🙋

Dit gebeurt er nu

1 Eerste if

```
if naamspeler != "":
```

```
    print("Uw spelersnaam is opgeslagen")
```

Je voert bil in → niet leeg → **print 1 keer** ✅

2 while wordt overgeslagen

```
while naamspeler == "":
```

bil ≠ "", dus deze while wordt **niet uitgevoerd** 🙋

3 Tweede if

```
if naamspeler != "":
```

```
    print("Uw spelersnaam is opgeslagen")
```

Nog steeds bil → **print NOG EEN KEER** ❌

→ Daarom zie je de melding **twee keer**

Oplossing: maar één keer printen

Je wilt eigenlijk:

- vragen naar een naam
- **blijven vragen zolang hij leeg is**
- daarna **één keer bevestigen**

Netjes en correct zo 🙋

```
print("Welkom bij Lingo!!")
```

```
naamspeler = input("Wat is de naam van uw speler? ")
```

```
while naamspeler == "":  
  
    print("Uw spelersnaam is niet opgeslagen, probeer het opnieuw.")  
  
    naamspeler = input("Wat is de naam van uw speler? ")  
  
  
print("Uw spelersnaam is opgeslagen")
```

- ✅ Geen dubbele prints
 - ✅ Logica klopt
 - ✅ Leesbaar
-

Extra: er zit **nóg** een foutje verderop 🙄

Deze regel werkt **niet zoals je denkt**:

```
if LengteWoord == str("vijfletter", "zesletter" or "zevenletterwoord"):
```

Python leest dit **niet** als “één van deze drie”.

Correcte manier:

```
if LengteWoord in ("vijfletter", "zesletter", "zevenletterwoord"):
```

💡 `in` is hier je beste vriend.

TL;DR

- Hij print dubbel omdat je **twee if naamspeler != "" hebt**
- Gebruik **één while + één print**
- Gebruik `in` (...) voor meerdere opties

Als je wilt, kan ik je hele Lingo-flow **opschonen en structureren** alsof het een echt spel is 🎮🔥

```
als ik dit doe en de input is "vijfletter" doet hij niks LengteWoord = str(input("Wilt u een  
vijfletter, zesletter of zevenletterwoord proberen te raden?")) if LengteWoord in  
("vijfletter", "zesletter" or "zevenletterwoord"): Confirmatie = str(input("Weet u dit  
zeker?")) while Confirmatie != str("Ja"): str(input("Weet u dit zeker?"))
```

Je zit **heel dicht bij de oplossing**, maar er zitten **twee subtiele Python-dingetjes** in de weg. Geen paniek — dit is supernormaal 👍

Probleem 1: "zesletter" or "zevenletterwoord"

Deze regel:

("vijfletter", "zesletter" or "zevenletterwoord")

doet **niet** wat het lijkt te doen.

Wat Python hiervan maakt:

("vijfletter", "zesletter")

Waarom?

Omdat "zesletter" or "zevenletterwoord" → **altijd "zesletter"**

(strings die niet leeg zijn zijn True)

→ "zevenletterwoord" wordt dus **genegeerd**

Probleem 2: je while-loop verandert niks

Hier:

```
while Confirmatie != "Ja":
```

```
    input("Weet u dit zeker?")
```

Je vraagt wel opnieuw input, maar je **slaat het antwoord niet op** in Confirmatie.

Dus als iemand "Nee" typt → **oneindige loop** 😞

Correcte werkende versie ✓

```
LengteWoord = input(
```

```
    "Wilt u een vijfletter, zesletter of zevenletterwoord proberen te raden? "
```

```
)
```

```
if LengteWoord in ("vijfletter", "zesletter", "zevenletterwoord"):
```

```
    Confirmatie = input("Weet u dit zeker? (Ja/Nee) ")
```

```
while Confirmatie != "Ja":
```

```
    Confirmatie = input("Weet u dit zeker? (Ja/Nee) ")
```

- Typ je vijfletter → **hij gaat netjes door**
 - Typ je Nee → hij blijft vragen
 - Typ je Ja → loop stopt
-

Bonus: gebruik lowercase (pro-tip 🔥)

Zo voorkom je fouten met hoofdletters:

```
LengteWoord = input("... ").lower()
```

```
Confirmatie = input("Weet u dit zeker? ").lower()
```

```
if LengteWoord in ("vijfletter", "zesletter", "zevenletterwoord"):
```

```
    while Confirmatie != "ja":
```

```
        Confirmatie = input("Weet u dit zeker? ").lower()
```

Samengevat

- ✗ "a" or "b" werkt **niet** zoals in normaal Nederlands
- ✓ gebruik in ("a", "b", "c")
- ✗ input vragen zonder opslaan = vastlopen
- ✓ altijd opnieuw toewijzen in een while

Wil je dat ik je hele **Lingo-keuzemenu** strak en netjes herschrijf? 😊

```
print("Welkom bij Lingo!!") naamspeler = str(input("Wat is de naam van uw speler?")) if
naamspeler != "": print("Uw spelersnaam is opgeslagen") while naamspeler == str(""):
print("Uw spelersnaam is niet opgeslagen, probeer het opnieuw.") naamspeler =
str(input("Wat is de naam van uw speler?")) if naamspeler != "": print("Uw spelersnaam
is opgeslagen") LengteWoord = input("Wilt u een vijfletter, zesletter of zevenletterwoord
proberen te raden?") if LengteWoord in ("vijfletter", "zesletter", "zevenletterwoord"):
Confirmatie = input("Weet u dit zeker? (Ja/Nee)") while Confirmatie != "Ja": Confirmatie =
input("Weet u dit zeker? (Ja/Nee)") if Confirmatie != "Ja": LengteWoord = input("Wilt u een
vijfletter, zesletter of zevenletterwoord proberen te raden?") if Confirmatie != "Nee":
print("Oke.") als ik "vijfletter", "zesletter" of "zevenletterwoord" type dan doet hij niks
meer
```

Goede catch — dit “niks doen”-gevoel komt doordat je code **wel loopt**, maar **geen zichtbare actie heeft op het juiste moment** én er zit een **logische fout** in je while-loop. We lopen 'm rustig door 🙋

Wat jij verwacht

Als je typt:

- vijfletter
- daarna Ja

🙋 dan verwacht je dat het programma **bevestigt en doorgaat**

Maar dat gebeurt niet (of vreemd).

Probleem 1: je doet niets als Confirmatie == "Ja"

Kijk hier:

```
while Confirmatie != "Ja":
```

```
    Confirmatie = input("Weet u dit zeker? (Ja/Nee)")
```

```
    if Confirmatie != "Ja":
```

```
        Lengtewoord = input("Wilt u een vijfletter, zesletter of zevenletterwoord proberen te raden?")
```

```
    if Confirmatie != "Nee":
```

```
        print("Oke.")
```

Stel: gebruiker typt meteen "Ja"

- while Confirmatie != "Ja" → **false**
- De hele while wordt **overgeslagen**
- Er staat **geen code na de while**
➡ programma stopt → “*hij doet niks*”

Dat is logisch Python-gedrag 🙋

Probleem 2: deze if klopt logisch niet

```
if Confirmatie != "Nee":
```

```
print("Oke.")
```

Dit betekent:

- "Ja" → print "Oke." ✓
- "Misschien" → print "Oke." ✗
- "Blabla" → print "Oke." ✗

Dat wil je niet.

Probleem 3: typo 🤪

```
Lengtewoord = input(...)
```

Je gebruikte eerder:

```
LengteWoord
```

Python ziet dit als **twee verschillende variabelen**.

Oplossing: simpele, duidelijke logica ✓

Dit is hoe je dit netjes en voorspelbaar doet:

```
print("Welkom bij Lingo!!")
```

```
naamspeler = input("Wat is de naam van uw speler? ")
```

```
while naamspeler == "":
```

```
    print("Uw spelersnaam is niet opgeslagen, probeer het opnieuw.")
```

```
    naamspeler = input("Wat is de naam van uw speler? ")
```

```
print("Uw spelersnaam is opgeslagen")
```

```
# woordlengte kiezen
```

```
LengteWoord = input(
```

```
    "Wilt u een vijfletter, zesletter of zevenletterwoord proberen te raden? "
```

```
)
```

```
while LengteWoord not in ("vijfletter", "zesletter", "zevenletterwoord"):
    LengteWoord = input(
        "Ongeldige keuze. Kies: vijfletter, zesletter of zevenletterwoord: "
    )

# bevestiging
Confirmatie = input("Weet u dit zeker? (Ja/Nee) ")

while Confirmatie != "Ja":
    if Confirmatie == "Nee":
        LengteWoord = input(
            "Kies opnieuw: vijfletter, zesletter of zevenletterwoord: "
        )
    Confirmatie = input("Weet u dit zeker? (Ja/Nee) ")

print("Oke. We gaan beginnen!")
```

Waarom dit wél werkt

- ✓ Elke while heeft een **duidelijk doel**
 - ✓ Elke input wordt **opgeslagen**
 - ✓ Er is **altijd feedback** voor de speler
 - ✓ Geen “stille” codeblokken meer
-




Mini-pro tip 💡

Gebruik `.lower()` om hoofdletterproblemen te voorkomen:

```
Confirmatie = input("Weet u dit zeker? ").lower()
```

Dan werkt Ja, ja, JA allemaal.

Als je wilt, kunnen we nu:

- het **woord echt** gaan raden
- of **feedback** geven per letter zoals echte Lingo   

wat doet lower() en len

Goede vraag — dit zijn twee hele belangrijke Python-dingen 😊
Ik leg ze simpel uit met voorbeelden uit jouw code.

lower()

lower() maakt een tekst **allemaal kleine letters**.

Voorbeeld

```
woord = "AppEl"  
print(woord.lower())
```

Output:

appel

Waarom gebruiken we dit bij jou?

```
raden = input("Probeer u eens: ").lower()
```

Dan maakt het niet uit hoe de speler typt:

Speler typt Wordt

Appel appel

APPEL appel

aPpEl appel

 Zo kun je eerlijk vergelijken met geheim_woord.

len()

len() geeft de **lengte (aantal tekens)** van iets terug.

Voorbeeld

```
woord = "appel"  
print(len(woord))
```

Output:

5

Dus:

Woord **len()**

appel 5

banaan 6

machine 7

Waarom gebruiken we dit bij jou?

1 Lengte controleren

if len(raden) != ALetter:

→ checkt of speler het juiste aantal letters invoert

2 Aantal letters bepalen

ALetter = len(geheim_woord)

→ Python telt zelf hoeveel letters het woord heeft

Dus je hoeft niet handmatig 5,6,7 te zetten 👍

★ Kort samengevat

- lower() → maakt tekst kleine letters
- len() → telt aantal tekens

Beiden zijn **standaard functies** die je héél vaak gebruikt.

👉 Mini-check voor jou

Kun je raden wat dit print?

```
x = "LINGO"
```

```
print(len(x.lower()))
```

(Als je wilt — antwoord ik meteen 😊)

Is er een function in python om het aantal letters in een woord te zien

Ja! In Python kun je het aantal letters in een woord heel eenvoudig krijgen met de **len()** functie. Dit telt het aantal tekens in een string.

Bijvoorbeeld:

```
woord = "appel"
aantal_letters = len(woord)
print(aantal_letters)
```

Output:

5

♦ **Let op:** len() telt alle tekens, dus ook spaties, cijfers of speciale tekens. Als je echt alleen letters wilt tellen, kun je dit filteren:

```
woord = "hallo123!"
aantal_letters = sum(1 for c in woord if c.isalpha())
print(aantal_letters)
```

Output:

5

Hier telt isalpha() alleen de letters.

Als je wilt, kan ik een klein voorbeeld laten zien dat het aantal letters **per woord in een hele zin** telt. Wil je dat?

```
Spelregels = str(input("Wilt uw de spelregels weten of heeft u het spel al is eerder  
gespeeld? (Ja/Nee)")) while Spelregels.lower() == "Ja": print("\n") print("Het doel van het  
spel is om een vijfletter-, zesletter- of een zevenletterwoord te raden door middel van  
andere woorden. U heeft 6 pogingen om dit te doen, als uw een letter goed raad, kleurt  
het hokje wit, als de letter in het woord zit maar niet op de juiste plek dan kleurt het  
hokje grijs en als de letter niet goed is blijft het hokje zwart/leeg.") print("\n") while  
Spelregels.lower() == "Nee": print("Succes!") print("\n") Als ik nee intyp print hij geen  
"succes!"
```

Goede vraag — dit gebeurt door twee kleine dingen in je code 😊

🔍 Wat gaat er mis

1 Je gebruikt while-lussen met een vaste voorwaarde:

```
while Spelregels.lower() == "Ja":
```

Maar `Spelregels.lower()` maakt alles **kleine letters** ("ja"), terwijl jij vergelijkt met "Ja" (met hoofdletter).

Dus de voorwaarde wordt **nooit waar**.

Hetzelfde geldt voor "Nee" → "nee".

👉 Daardoor wordt geen van beide lussen uitgevoerd.

2 Daarnaast is while hier eigenlijk niet wat je wilt.

Een while blijft eindeloos herhalen zolang de conditie waar is — je wil gewoon **één keer checken** → gebruik if.

✅ Oplossing

Gebruik if / elif en vergelijk met kleine letters:

```
Spelregels = input("Wilt u de spelregels weten of heeft u het spel al eens eerder  
gespeeld? (Ja/Nee)")
```

```
if Spelregels.lower() == "ja":
```

```
    print("\n")
```

```
    print("Het doel van het spel is om een vijfletter-, zesletter- of een zevenletterwoord te  
raden door middel van andere woorden. U heeft 6 pogingen om dit te doen, als u een
```

letter goed raadt, kleurt het hokje wit, als de letter in het woord zit maar niet op de juiste plek dan kleurt het hokje grijs en als de letter niet goed is blijft het hokje zwart/leeg.")

```
print("\n")
```

```
elif Spelregels.lower() == "nee":
```

```
    print("Succes!")
```

```
    print("\n")
```

★ Bonus tip

Wil je het robuuster maken (spaties/rare invoer opvangen):

```
Spelregels = input("...").strip().lower()
```

Als je wilt, kan ik je ook laten zien hoe je dit met een herhaal-lus maakt zodat de gebruiker opnieuw moet invoeren bij fout antwoord 👍

