

Uitwerking opdracht

Opdracht : *Double Dutch*
Weeknummer : *6*
Studentnummer : *s1060679*
Naam student : *Dion van den Berg*
Specialisatie : *MEDT*
Pogingnummer : *1*

1. Vraagstelling

Zet een tekst om naar een speciale geheim taal.

2. Specificatie

Invoer

De gebruiker voert een letter in die hij veranderd wil hebben naar de speciale notatie

Uitvoer

De gebruiker krijgt een lettergreep terug.

Verband tussen in- en uitvoer

De uitvoer wordt gemaakt door de ingevoerde letter

Beperkingen

Letters die niet ingevoerd zijn kunnen niet afgehandeld worden.

Voorbeelden (testscenario's)

Test 1

Invoer:

`Print(dutchLetter['s'])`

Uitvoer:

`'sus'`

Test 2

Invoer:

`Print(dutchLetter['q'])`

Uitvoer:

`'quack'`

3. Ontwerp

Hoe maak ik het bestand open?

Door het te openen met open() en daarna uit te lezen met .read()

Hoe ga ik alles omzetten?

Door het te splitsen op elke enter en spaties eruit te halen zodat ik ermee kan werken

Hoe laat ik alles uiteindelijk zien ?

in een print statement met een eigen regel

4. Pseudocode

Functie medeklinkers (bestandLocatie)

Open het bestand van de opgegeven locatie

Geef alle losse medeklinkers en klinkers apart in een array in

Ga door het bestand heen en strip alle nutteloze info eruit

Zet het complete woordenboek in de array en check of het al bestond

Functie vertaalwoord (woord, woordenboek)

Zet de ingevoerde letter om in de bijhorende lettergreep van het woordenboek

Als het meerdere letters zijn plak ze aan elkaar

Functie vertaal (zin, woordenboek)

Split de gemaakte zin

Loop door de zin heen per woord en voeg er een spatie aan toe.

5. Code

```
def medeklinkers(pad):

    # maak lijst aan van klinkers zoals het voorbeeld.
    klinkers = ['a', 'e', 'i', 'o', 'u']

    # maak de lijst aan van letters die wel kunnen
    medeklinkers = ['b', 'c', 'd', 'f', 'g', 'h', 'j', 'k', 'l', 'm', 'n',
                    'p', 'q', 'r', 's', 't', 'v', 'w', 'x', 'y', 'z']

    # Open het bestand dat wordt opgegeven in deze functie
    bestand = open(pad, 'r')

    # Split het geopende bestand in losse regels
    regels = bestand.read().strip().replace(" ", "").split("\n")
    bestand.close()

    # Maak een boolean aan die zegt als er iets fout is.
    testcase = False

    # Maak een woordenboek aan
    woordenboek = {}
    bestaandeKarakters = []

    # Split de regels in letters en vertalingen
    for regel in regels:
        verwijzing = regel.split("-")

        # Check of de letter een klinker is
        for klinker in klinkers:
            if verwijzing[0] == klinker:
                print(verwijzing[0] + " is een klinker")
                testcase = True

        # Check of de eerste letter van de verwijzing hetzelfde is als
        # de letter die ernaar verwezen wordt
        if not verwijzing[0] == verwijzing[1][0]:
            print(verwijzing[1] + " begint niet met " + verwijzing[0])
            testcase = True

        # Kijk of de letter al in de lijst voor komt
        if verwijzing[0] in bestaandeKarakters:
            print("we hebben deze al in de lijst")
            testcase = True

        bestaandeKarakters.append(verwijzing[0])

        # Voeg de nieuwe entry toe aan de dictionary
        woordenboek[verwijzing[0]] = verwijzing[1]

    # Check of de nieuwe dictionary alle medeklinkers bezit
    for letter in medeklinkers:
        if letter not in woordenboek.keys():
            print("er is een missende letter in het woordenboek")
            testcase = True

    # Geef een assertionerror als er iets fout is in het woordenboek
    if testcase:
        raise AssertionError("Ongeldige vertaling!")
```

```

# Anders return je woordenboek
else:
    return woordenboek

# Functie vertaalWoord met als parameters het woord en een dictionary
def vertaalWoord(woord, dictionary):

    # Maak een lege string aan die je later terug geeft
    returnWoord = ""
    letterCounter = 0
    klinkerUpper = False
    vorigeLetter = ""

    # Check alle letters in het woord
    for letter in woord:
        # kijk of het een hoofdletter is
        isUpper = letter.isupper()

        # Check of het letter voor komt in de dictionary
        letter = letter.lower()

        # Kijk of de letter in de dictionary voor komt
        contains = letter in dictionary.keys()

        # Als de letter in de dictionary staat, zoek het dan
        # daarin op
        if contains:
            nieuwToevoegen = dictionary[letter]
            letterCounter = 0
            klinkerUpper = False
        # Anders voeg je de letter toe aan de string
        else:
            nieuwToevoegen = letter
            if letter == vorigeLetter:
                letterCounter += 1

            if letterCounter == 0:
                if isUpper:
                    klinkerUpper = True
                else:
                    klinkerUpper = False

        # Als er twee klinkers achter elkaar staan
        if letterCounter == 1:
            nieuwToevoegen = "squat" + letter + "h"
            letterCounter = 0
            returnWoord = returnWoord[:-1]

        # Als het een hoofdletter was, zet de eerste letter naar
        # hoofdletters
        if isUpper or klinkerUpper:
            vervangWoord = nieuwToevoegen
            nieuwToevoegen = ""
            for i in range(0, len(vervangWoord)):
                if i == 0:
                    nieuwToevoegen += vervangWoord[i].upper()
                else:
                    nieuwToevoegen += vervangWoord[i]

```

```

        # Voeg de nieuwe letter toe aan het woord
        returnWoord += nieuwToevoegen

        # Zet de vorige letter
        vorigeLetter = letter

    # Return het woord
    return returnWoord

def vertaal(zin, dictionary):
    # Split de zin in woorden
    zin = zin.split(" ")

    # Lege string met de zin die terug gegeven wordt
    returnZin = ""

    # Ga ieder woord af in de zin
    for woord in zin:
        returnZin += vertaalWoord(woord, dictionary)
        returnZin += " "

    # Return de zin
    return returnZin

dutchLetter = medeklinkers('dutchLetters.txt')
print(dutchLetter['s'])
print(dutchLetter['q'])
print(dutchLetter['d'])
try:
    print(dutchLetter['e'])
except KeyError:
    print("KeyError")

print(vertaalWoord('took', dutchLetter))
print(vertaalWoord('BAMBOO', dutchLetter))
print(vertaalWoord('Yesterday', dutchLetter))

dutchLetter = medeklinkers('dutchLetters.txt')
print(vertaal('I took a walk to the park yesterday.', dutchLetter))

```

6. Test

Test 1

```
dutchLetter = medeklinkers('dutchLetters.txt')
print(dutchLetter['s'])
print(dutchLetter['q'])
print(dutchLetter['d'])
print(dutchLetter['e'])
print(vertaalWoord('took', dutchLetter))
print(vertaalWoord('BAMBOO', dutchLetter))
print(vertaalWoord('Yesterday', dutchLetter))
```

```
dutchLetter = medeklinkers('dutchLetters.txt')
print(vertaal('I took a walk to the park yesterday.', dutchLetter))
```

```
sus
quack
dud
KeyError
tutsquatohkuck
BubAMumBubSquatoh
Yubesustuterugdudayub
I tutsquatohkuck a wackalulkuck tuto tuthashe pubarugkuck yubesustuterugdudayub.
```