

Tafelpraat

Specifieke info

Gebruik de stringmethode `isalpha` om te controleren of een karakter een letter is. Deze methode herkent zowel hoofdletters als kleine letters.

```
>>> 'f'.isalpha()
True
>>> 'Y'.isalpha()
True
>>> '3'.isalpha()
False
>>> '.'.isalpha()
False
```

Stopcodons

Itereren over elementen én hun posities van sequentietypes

De ingebouwde functie `enumerate` kan gebruikt worden om een iterator op te vragen van een sequentietype (*sequence type*: strings, lijsten, tuples, bestanden, ...), die zowel de positie als de waarde van het volgende element van het type teruggeeft. Onderstaand voorbeeld geeft bijvoorbeeld aan hoe je tegelijkertijd de posities en de karakters op die posities kunt overlopen voor een string.

```
>>> for index, karakter in enumerate('abc'):
...     print(f'index: {index}')
...     print(f'karakter: {karakter}')
...
index: 0
karakter: a
index: 1
karakter: b
index: 2
karakter: c
```

Je kan dit bijvoorbeeld gebruiken als je synchroon de karakters van twee strings wil overlopen.

```
>>> eerste = 'abc'
>>> tweede = 'def'
>>> for index, karakter in enumerate(eerste):
...     print(f'{karakter}-{tweede[index]}')
...
a-d
b-e
c-f
```

In dat geval is het echter aangewezen om gebruik te maken van de ingebouwde functie `zip`, die net haar bestaansrecht haalt uit het feit dat ze een iterator teruggeeft voor twee of meer iterateerbare objecten.

```

>>> eerste = 'abc'
>>> tweede = 'def'
>>> for karakter1, karakter2 in zip(eerste, tweede):
...     print(f'{karakter1}-{karakter2}')
...
a-d
b-e
c-f

```

Pangrammatisch venster

Specifieke info

De eenvoudigste manier om de functie `venster` te implementeren, bestaat erin om alle mogelijke deelstrings van de gegeven tekst te overlopen, voor elk van die deelstring te bepalen of het al dan niet een pangram is, en van alle pangrammen het kortste pangram over te houden.

Je kan alle deelstrings van een gegeven string overlopen door alle mogelijk startposities van deelstrings te overlopen, en voor elke startpositie alle mogelijke stopposities van deelstrings te overlopen. Dit kan je implementeren aan de hand van een geneste `for`-lus. Onderstaande code schrijft bijvoorbeeld alle deelstrings van de string `'abc'` uit, waarbij we enkel de deelstrings met minimale lengte 1 in rekening brengen.

```

>>> s = 'abc'
>>> for start in range(len(s) - 1):
...     for stop in range(start, len(s)):
...         print(s[start:stop + 1])
...
a
ab
abc
b
bc

```

Als je er even over nadent, dan is het zelfs niet nodig om alle deelstrings van de gegeven tekst te overlopen om het kortste pangram te vinden. Om overbodige berekeningen te vermijden, kan je er even over nadenken hoe je kunt verhinderen dat Python overbodig werk moet doen.

Rövarspråket

Specifieke info

De oplossing van deze opgave bestaat erin dat je inziet dat je de verwerking van een groep medeklinkers moet uitstellen tot op het ogenblik dat je een niet-medeklinker tegenkomt in de string (waardoor je weet dat de groep medeklinkers compleet is). Dit wordt geïllustreerd aan de hand van het onderstaand voorbeeld, waarbij de string `abcdefg` wordt omgezet naar Rövarspråket.

Hierbij wordt de string die moet gecodeerd worden karakter per karakter overlopen. Als hierbij een medeklinker wordt tegengekomen, dan wordt die niet onmiddellijk toegevoegd aan de gecodeerde string, maar wordt die toegevoegd aan de groep opeenvolgende medeklinkers. Op het moment dat er een niet-medeklinker wordt tegengekomen, dan sluit deze de groep van opeenvolgende medeklinkers af. Indien er zich een groep gevormd

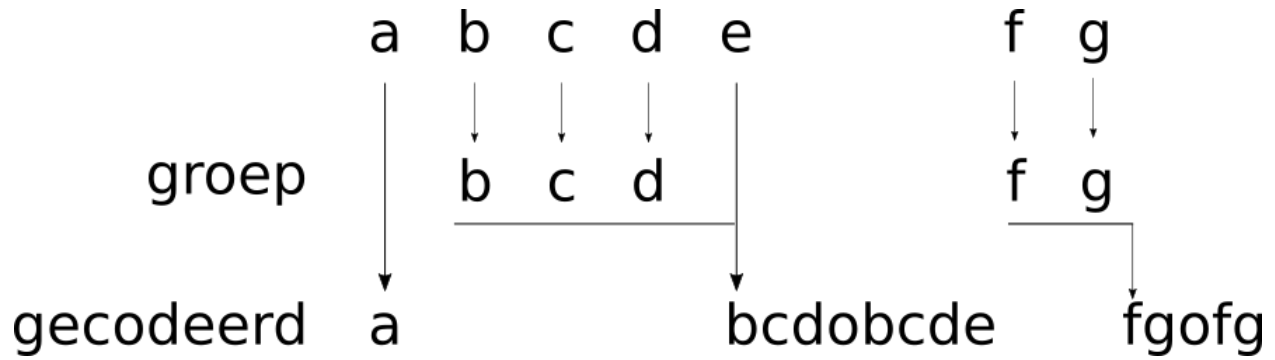


Figure 1: Rövarspråket

had, dan wordt die eerst aan de gecodeerde string toegevoegd, gevolgd door de letter o en nog een herhaling van de groep opeenvolgende medeklinkers. Pas daarna wordt de niet-medeklinker aan de gecodeerde groep toegevoegd, en wordt er een nieuwe groep medeklinkers gevormd.

Nadat alle karakters van de te coderen string overlopen zijn, kan het zijn dat er zich nog een groep van opeenvolgende medeklinkers gevormd heeft die niet aan de gecodeerde string werd toegevoegd. Dit is het geval wanneer de string eindigt op een medeklinker. In het voorbeeld is dit het geval voor de groep **fg**. Indien dit geval is, dan moet deze groep ook nog toegevoegd worden aan de gecodeerde string, gevolgd door de letter o en nog een herhaling van de groep opeenvolgende medeklinkers.