

Algemeen

De stringmethode `index`

De stringmethode `index` kan gebruikt worden om de meest linkse positie van een karakter in een string te bepalen.

```
>>> string = 'mississippi'
>>> teken = 'i'
>>> string.index(teken)
1
```

Dit codefragment vindt dus het eerste voorkomen van de letter `i` op positie 1 in de string `mississippi`. Let hierbij op het feit dat Python de posities van de karakters in een string indexeert vanaf 0, en niet vanaf 1.

Het valt wel op te merken dat deze methode niet altijd zeer efficiënt werkt. Om de positie te vinden van een karakter dat voor het eerst opduikt op plaats p , moet de functie de eerste $p - 1$ posities van de string doorlopen. Voor een karakter dat niet voorkomt in de string moet de stringmethode `index` zelfs de volledige string aflopen vooraleer er kan vastgesteld worden dat het karakter er niet gezien werd.

Verwijderen van witruimte vooraan en/of achteraan

In Python kan je gebruikmaken van de stringmethode `strip` om witruimte (spaties, tabs, regeleindes) vooraan en achteraan te verwijderen. Indien je enkel de witruimte vooraan wilt verwijderen, dan kan je gebruikmaken van de stringmethode `lstrip`. Indien je enkel de witruimte achteraan wilt verwijderen, dan kan je gebruikmaken van de stringmethode `rstrip`. Je kan aan deze stringmethode ook een argument meegeven, waarmee je aangeeft welke karakters er vooraan en/of achteraan moeten verwijderd worden. Voor de details hierover verwijzen we naar The Python Standard Library.

```
>>> tekst = ' Dit is een tekst '
>>> tekst.strip()
'Dit is een tekst'
>>> tekst.lstrip()
'Dit is een tekst '
>>> tekst.rstrip()
' Dit is een tekst'
>>> tekst2 = '===Dit is een tekst==='
>>> tekst2.lstrip('=')
'Dit is een tekst==='
```

Atbash

Letters omzetten naar hun getalwaarde

In Python heeft elk karakter een getalwaarde (integer). Zo heeft het vraagteken (`?`) als getalwaarde 63. Deze waarde kan bekomen worden met behulp van de ingebouwde functie `ord`.

```
>>> ord('?')
63
```

Het interessante aan deze getalwaarden is dat de opeenvolgende letters van het alfabet ook opeenvolgende getalwaarden hebben. Dit geldt zowel voor kleine letters als voor hoofdletters. Zo heeft de letter `a` als getalwaarde 97, waaruit we kunnen afleiden dat de letter `b` als getalwaarde 98 moet hebben. Met die wetenschap kunnen we dus op eenvoudige manier de positie van een letter in het alfabet bepalen.

```
>>> letter = 'd'
>>> ord(letter)
100
```

```
>>> ord('a')
97
>>> ord(letter) - ord('a') + 1    # d is de 4e letter van het alfabet
4
>>> ord('z') - ord('a') + 1      # z is de 26e letter van het alfabet
26
```

Stringmethoden `islower()` en `isupper()`

De stringmethode `islower` kijkt of *alle* tekens van een string kleine letters zijn. De stringmethode `isupper` controleert of *alle* letters van een string hoofdletters zijn.

```
>>> 'ABCD'.islower()
False
>>> 'ABCD'.isupper()
True
>>> 'abcd'.islower()
True
>>> 'abcd'.isupper()
False
>>> 'AbCd'.islower()
False
>>> 'AbCd'.isupper()
False
```

Wepe sprekeken p

Specifieke info

Een mogelijke strategie die je voor deze opgave kunt gebruiken is de volgende. Je itereert over de posities van de regel en je werkt met 2 variabelen. De eerste variabele zal de uiteindelijke gedecodeerde tekst bevatten en in de tweede variabele houd je de klinkergroep bij. Wanneer je dan een **p** tegenkomt en je hebt een *niet lege klinkergroep*, kun je die klinkergroep verwerken en enkele posities verder springen.

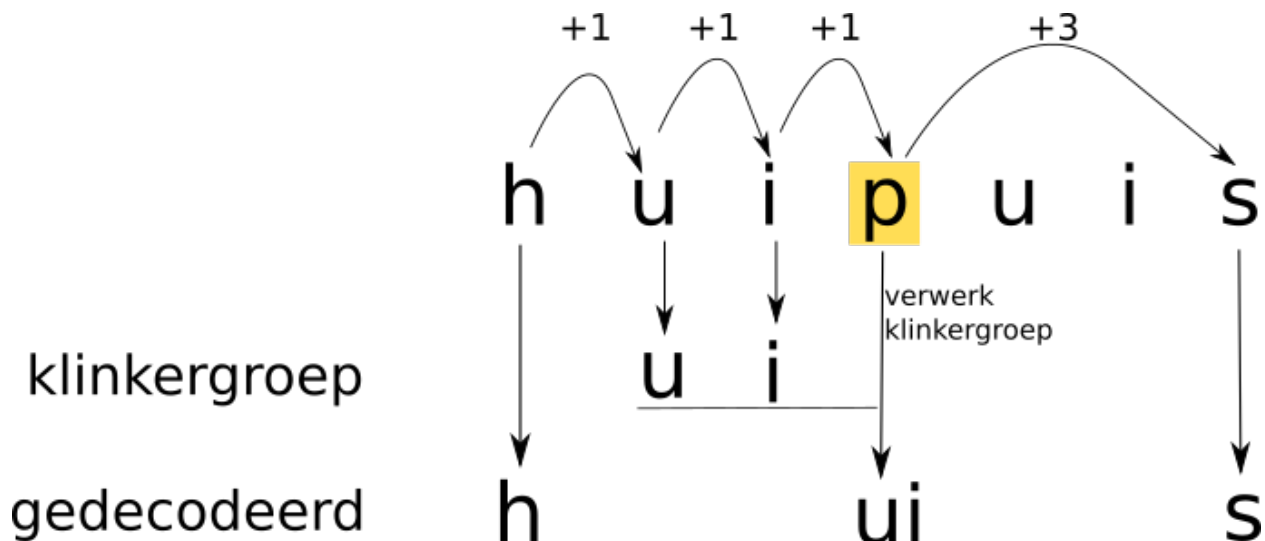


Figure 1: Wepe sprekeken p