Tekst ABC

Skrevet av: Ole Kristian Pedersen, Kodeklubben Trondheim

Oversatt av: Stein Olav Romslo

Kurs: Python
Tema: Tekstbasert
Fag: Programmering
Klassetrinn: 8.-10. klasse

Introduksjon

I denne oppgåva skal me gjere enkle operasjonar på tekst, som å endre storleiken på bokstavane og telje ord.

I Python lagrar me tekst til ein **variabel** slik som dette:

```
streng = "teksten er her"
```

Variabelen kallast då **tekststreng** eller berre **streng**, som er typen til variabelen.

Store og små bokstavar

Ein måte å endre tekst på er ved hjelp av funksjonar. Det er fleire slike funksjonar innebygd i Python, Gitt at me har tekst i ein variabel kalla `s`, så kan me bruke desse funksjonane: `s.lower()`, `s.upper()`, `s.title()`, `s.swapcase()` og `s.capitalize()`.

Legg merke til at funksjonen kallast **på** strengen - `s.lower()` - i staden for å **gi** strengen til funksjonen - `lower(s)`.

Her er nokre døme på korleis funksjonane brukast (legg merke til kva bokstavar som er store og små i utskrifta):

```
>>> s = "Per og Ada"

>>> s.upper() # store bokstavar
'PER OG ADA'

>>> s.lower() # små bokstavar
'per og ada'

>>> s.capitalize() # fyrste bokstav er stor
'Per og ada'

>>> s.title() # fyrste bokstav i kvart ord er stor
'Per Og Ada'

>>> s.swapcase() # byttar store og små bokstavar
'pER oG aDA'
```

Her er nokre døme på kva funksjonane kan brukast til:

- `s.capitalize()` brukast når me ynskjer stor forbokstav berre i starten av teksten:

  ```
  >>> sentence = "dENne sETNinGa har IKKJE riKTige bokSTAVstOrLeIKAR."

  >>> sentence.capitalize()
  'Denne setninga har ikkje riktige bokstavstorleikar.'
  ```

- `s.title()` kan brukast når me skal skrive filmtitlar:

  ```
  >>> movie_title = "star wars: a new hope"

  >>> movie_title.title()
  'Star Wars: A New Hope'
  ```

- `s.upper()` og `s.lower()` kan brukast når me ynskjer å samanlikne tekst utan å ta omsyn til storleiken på bokstavane:

  ```
  >>> answer = "JA"

  >>> answer == "ja" # JA og ja er ikkje like
  False
  ```

```
>>> answer = "JA"

>>> answer.lower() == "ja" # konverter JA til ja for testen
True
```

Du må hugse på at desse funksjonane **ikkje** endrar på variabelen. Difor må du lagre resultatet i ein ny variabel om du vil handsame endringa di:

```
>>> s = "tekst"

>>> s.upper() # Me endrar ikkje på variabelen!
'TEKST'

>>> s # Framleis små bokstavar
'tekst'

>>> s = s.upper() # No endrar me på variabelen

>>> s # Denne gongen er det store bokstavar
'TEKST'
```

Skriv ditt eige program

Lag eit program som skriv ut filmtitlar med store bokstavar fyrst i kvart ord.

Programmet skal sjå slik ut:

```
>>>
Skriv inn ein filmtittel: alice in wonderland
Alice In Wonderland
```

Dette må du gjere:

- [ ] Be om at brukaren skriv inn ein filmtittel.
- [ ] Lagre filmtittelen i ein variabel.
- [ ] Manipuler strengen slik at resultatet blir som beskrive over.
- [ ] Skriv ut den nye strengen.

Teljing av tekst

Ved hjelp av `s.count()` kan me finne ut om ein streng inneheldt ein bestemt tekst og kor mange gonger den finst i strengen. Til dømes inneheldt strengen `"Hei verd!"` teksten `"verd"` ein gong.

Tenk deg at du ynskjer å finne ut kor mange komma som er i `"A, B, C, D, E, F, G, H, I, J, K, L"`. Det er enkelt å telje for hand, men ikkje like moro som å la datamaskina gjere det:

```
>>> s = "A, B, C, D, E, F, G, H, I, J, K, L"

>>> s.count(",")
11
```

Me kan òg telje tekst som er lengre, til dømes `"Per"`:

```
>>> s = "Per, Ada, Kim, Per, Kim, Per"

>>> s.count("Per")
3
```

Skriv ditt eige program

Lag eit program som tel kor mange ord det er i det brukaren skriv inn. Antal ord kan reknast ut ved å telje talet på mellomrom og så leggje til 1. Forstår du kvifor me må leggje til 1?

Slik skal programmet sjå ut:

```
>>>
Skriv inn ein streng: Hei på deg
Du skreiv inn 3 ord.
```

Dette må du gjere:

- [ ] Be brukaren om tekst.
- [ ] Lagre teksten til ein variabel.
- [ ] Rekne ut kor mange ord som er i teksten.

- [ ] Skriv ut kor mange ord teksten inneheldt.

  Hint: Hugs å konvertere frå tal til tekst med `str()` -funksjonen.

Erstatte tekst

Me kan bruke `s.replace()` for å byte ut tekst i ein streng med ein annan tekst. Kva om me vil byte ut alle komma med semikolon?

```
>>> s = "A, B, C, D, E, F, G, H, I, J, K, L"

>>> s.replace(",", ";")
'A; B; C; D; E; F; G; H; I; J; K; L'
```

Her får `s.replace()` to argument - fyrst teksten me skal erstatte i strengen `s` , og så teksten me skal erstatte med. Me kan òg bruke `s.replace()` for å fjerne tekst. Me kan til dømes fjerne alle mellomrom:

```
>>> s = "1 2 3 4 5"

>>> s.replace(" ", "")
'12345'
```

Skriv ditt eige program

Nokre operativsystem og program oppfører seg rart viss ein lagar filnamn med mellomrom i. Du vil difor lage eit program som byttar ut alle mellomrom med ein understrek. I tillegg skal du syte for at det berre blir brukt små bokstavar i filnamnet. Det skal fungere som i programmet under:

```
>>>
Skriv inn eit filnamn: Mi HemMelege FIL.txt
mi_hemmelege_fil.txt
```

Dette må du gjere:

- [ ] Spør brukaren om eit filnamn.

- [ ] Endre filnamnet slik som beskrive over.

- [ ] Skrive ut det nye filnamnet.

Lisens: CC BY-SA 4.0 (http://creativecommons.org/licenses/by-sa/4.0/deed)