

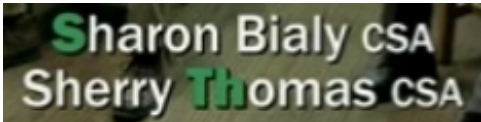
# Breaking bad

[Breaking Bad](#) is an American television drama series that tells the story of Walter White, a struggling high school chemistry teacher who is diagnosed with inoperable lung cancer. He turns to a life of crime, producing and selling methamphetamine with the aim of securing his family's financial future before he dies.



Logo of the American television series Breaking Bad.

If you've ever seen the series, you might have noticed how the opening and closing credits highlight some of the letters in the names of the cast with a green color. These letters refer to the chemical elements in the periodic table. The example below shows the letter **S** that corresponds to the chemical element sulfur and the letters **Th** that are used as the symbolic name of thorium.



The closing credits of the television series Breaking Bad highlight the symbolic names of the chemical elements in the periodic table using a green color.

The logo of the series also frames the first two letters of each word in the title within a green rectangle. **Br** is the symbolic name used for bromide and **Ba** is the symbolic name of barium.

## Assignment

1. Write a function `stressedWord` that takes exactly two arguments. The first argument is a string that contains a single word. The second argument is a list of strings. This list may represent a list of symbolic names of chemical elements, but any other strings are also allowed. The function must return a string containing the original word, where some of the letters have been stressed by surrounding them with square brackets. These letters correspond to the "leftmost and longest" occurrence of a symbol from the given list of symbols. With "leftmost and longest" we mean that the position of the opening square bracket ( [ ) should be as small as possible. If multiple symbols start from this position in the word, the longest one should be chosen. If none of the symbols from the given list of symbols occurs in the word, the original word without any square brackets must be returned by the function. Inside the string returned by the function there is at most one pair of square brackets. Comparisons between the given word and the symbols from the given list of symbols should be case insensitive, but the use of uppercase and lowercase letters in the word that was passed to the function should remain intact in the word that is returned by the function.

```
stressedWord(word, symbols)
```

2. Use the function `stressedWord` to write a function `stressedSentence`. The first string argument passed to this function represents a given sentence. The function must return a string containing the same sentence, in which each word highlights the "leftmost and longest"

occurrence of the letters of a symbol in between square brackets. By default, the function highlights the symbolic names from the chemical elements (in the example below, this list of symbolic names is assigned to the variable `chemicalSymbols`). If a list of strings is passed as a second argument to the function, these strings represent the symbols that must be highlighted with square brackets. You may assume that all words in the sentence are separated with a single space.

```
stressedSentence(sentence[, symbols])
```

## Example

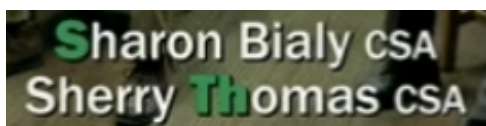
```
>>> chemicalSymbols = [  
... 'Ac', 'Ag', 'Al', 'Am', 'Ar', 'As', 'At', 'Au', 'B', 'Ba', 'Be', 'Bh',  
... 'Bi', 'Bk', 'Br', 'C', 'Ca', 'Cd', 'Ce', 'Cf', 'Cl', 'Cm', 'Cn', 'Co',  
... 'Cr', 'Cs', 'Cu', 'Db', 'Ds', 'Dy', 'Er', 'Es', 'Eu', 'F', 'Fe', 'Fl',  
... 'Fm', 'Fr', 'Ga', 'Gd', 'Ge', 'H', 'He', 'Hf', 'Hg', 'Ho', 'Hs', 'I',  
... 'In', 'Ir', 'K', 'Kr', 'La', 'Li', 'Lr', 'Lu', 'Lv', 'Md', 'Mg', 'Mn',  
... 'Mo', 'Mt', 'N', 'Na', 'Nb', 'Nd', 'Ne', 'Ni', 'No', 'Np', 'O', 'Os',  
... 'P', 'Pa', 'Pb', 'Pd', 'Pm', 'Po', 'Pr', 'Pt', 'Pu', 'Ra', 'Rb', 'Re',  
... 'Rf', 'Rg', 'Rh', 'Rn', 'Ru', 'S', 'Sb', 'Sc', 'Se', 'Sg', 'Si', 'Sm',  
... 'Sn', 'Sr', 'Ta', 'Tb', 'Tc', 'Te', 'Th', 'Ti', 'Tl', 'Tm', 'U', 'Uuo',  
... 'Uup', 'Uus', 'Uut', 'V', 'W', 'Xe', 'Y', 'Yb', 'Zn', 'Zr'  
... ]  
  
>>> stressedWord('Graham', chemicalSymbols)  
'G[ra]ham'  
>>> stressedWord('Python', ['Th', 'T', 'hon', 'on'])  
'Py[th]on'  
  
>>> stressedSentence("Monty Python's Life of Brian.")  
"[Mo]nty [P]ython's [Li]fe [o]f [Br]ian."  
>>> stressedSentence("Monty Python's Life of Brian.", ['Th', 'T', 'hon', 'on'])  
"M[on]ty Py[th]on's Life of Brian."
```

[Breaking Bad](#) is een Amerikaanse televisiereeks over een chemieleraar die drugsdealer wordt om zijn financiële problemen op te lossen. Maar van het een komt het ander en zo belandt hij helemaal in de criminaliteit.



Logo van de televisiereeks Breaking Bad.

Wie deze reeks gezien heeft, is het misschien ook opgevallen dat tijdens de begin- en eindgeneriek sommige letters in de namen van acteurs benadrukt worden aan de hand van een groene kleur. Deze letters verwijzen naar chemische elementen uit de tabel van Mendeljev. In het onderstaand voorbeeld komt de letter **S** overeen met het chemische element zwavel en staat **Th** symbool voor thorium.



De affiteling toont de symbolen van chemische elementen uit de periodieke tabel in het groen.

Ook in het logo van de reeks zijn de twee eerste letters van elk woord uit de titel omkaderd. **Br** staat voor broom en **Ba** verwijst naar barium.

## Opgave

1. Schrijf een functie `woordNadruk` waaraan exact twee argumenten moeten doorgegeven worden. Het eerste argument is string die één enkel woord voorstelt. Het tweede argument is een lijst van strings. Deze lijst kan een lijst met de symbolische namen van de chemische elementen zijn, maar kan ook andere strings bevatten. De functie moet als resultaat een string teruggeven, die het originele woord bevat waarin sommige letters tussen vierkante haakjes geplaatst werden om ze te benadrukken. Deze letters komen overeen met het "meest linkse en langste" voorkomen van een symbool uit de symbolenlijst. Met "meest linkse en langste" bedoelen we dat de positie waar het openend vierkant haakje (`()`) staat zo klein mogelijk moet zijn. Als vanaf deze positie in het woord meerdere symbolen starten, dan wordt deze met de langste lengte gekozen. Als er in het woord geen symbolen uit de symbolenlijst voorkomen dan wordt het woord zonder haakjes teruggegeven door de functie. Binnen de string die wordt teruggegeven staat er dus hoogstens één paar vierkante haakjes. De vergelijking tussen symbolen uit de symbolenlijst en letters uit het woord moet geen rekening houden met het verschil tussen hoofdletters en kleine letters, maar het gebruik van hoofdletters en kleine letters in het woord dat aan de functie werd doorgegeven, moet behouden blijven in het resultaat dat door de functie wordt teruggegeven.

```
woordNadruk(woord, symbolen)
```

2. Gebruik de functie `woordNadruk` om een functie `zinNadruk` te schrijven. Aan deze functie moet als eerste argument een string doorgegeven worden, die een zin voorstelt. De functie moet een string teruggeven die deze zin bevat, maar waarin voor elk woord het "meest linkse en langste" voorkomen van een symbool tussen vierkante haakjes staat. Standaard worden de symbolen uit het periodieke systeem gebruikt (in onderstaand voorbeeld wordt deze lijst van symbolen toegekend aan de variabele `chemischeSymbolen`). Indien er echter een lijst van strings als tweede argument aan de functie doorgegeven wordt, dan vormen deze strings de symbolen die tussen vierkante haakjes moeten geplaatst worden. Je mag veronderstellen dat de woorden in de zin telkens van elkaar gescheiden worden door één enkele spatie.

```
zinNadruk(zin[, symbolen])
```

## Voorbeeld

```
>>> chemischeSymbolen = [  
... 'Ac', 'Ag', 'Al', 'Am', 'Ar', 'As', 'At', 'Au', 'B', 'Ba', 'Be', 'Bh',  
... 'Bi', 'Bk', 'Br', 'C', 'Ca', 'Cd', 'Ce', 'Cf', 'Cl', 'Cm', 'Cn', 'Co',  
... 'Cr', 'Cs', 'Cu', 'Db', 'Ds', 'Dy', 'Er', 'Es', 'Eu', 'F', 'Fe', 'Fl',  
... 'Fm', 'Fr', 'Ga', 'Gd', 'Ge', 'H', 'He', 'Hf', 'Hg', 'Ho', 'Hs', 'I',  
... 'In', 'Ir', 'K', 'Kr', 'La', 'Li', 'Lr', 'Lu', 'Lv', 'Md', 'Mg', 'Mn',  
... 'Mo', 'Mt', 'N', 'Na', 'Nb', 'Nd', 'Ne', 'Ni', 'No', 'Np', 'O', 'Os',  
... 'P', 'Pa', 'Pb', 'Pd', 'Pm', 'Po', 'Pr', 'Pt', 'Pu', 'Ra', 'Rb', 'Re',  
... 'Rf', 'Rg', 'Rh', 'Rn', 'Ru', 'S', 'Sb', 'Sc', 'Se', 'Sg', 'Si', 'Sm',  
... 'Sn', 'Sr', 'Ta', 'Tb', 'Tc', 'Te', 'Th', 'Ti', 'Tl', 'Tm', 'U', 'Uuo',  
... 'Uup', 'Uus', 'Uut', 'V', 'W', 'Xe', 'Y', 'Yb', 'Zn', 'Zr'  
... ]
```

```
>>> woordNadruk('Graham', chemischeSymbolen)
```

```
'G[ra]ham'
```

```
>>> woordNadruk('Python', ['Th', 'T', 'hon', 'on'])
```

```
'Py[th]on'
```

```
>>> zinNadruk("Monty Python's Life of Brian.")
```

```
"[Mo]nty [P]ython's [Li]fe [o]f [Br]ian."
```

```
>>> zinNadruk("Monty Python's Life of Brian.", ['Th', 'T', 'hon', 'on'])
```

```
"M[on]ty Py[th]on's Life of Brian."
```