

Buzz-phrases

A **buzzword** is a word that becomes very popular for a period of time. It may be a technical term and may have little meaning, being simply used to impress others. Buzzwords often originate in jargon, acronyms or neologisms. Business speech is particularly vulnerable to buzzwords. Examples of overworked business buzzwords include *synergy*, *vertical*, *dynamic*, *cyber* and *strategy*. If the meme contains multiple words, it is called a **buzz-phrase**. A common buzz-phrase is *think outside the box*.



In 1968, *Newsweek* published an article [How to Win at Wordsmanship](#) by Philip Broughton. It describes a sure-fire method for converting frustration into fulfillment (jargonwise). Euphemistically called the *Systematic Buzz Phrase Projector*, Broughton's system employs a lexicon of 30 carefully chosen buzzwords.

0. integrated	0. management	0. options
1. total	1. organizational	1. flexibility
2. systematized	2. monitored	2. capability
3. parallel	3. reciprocal	3. mobility
4. functional	4. digital	4. programming
5. responsive	5. logistical	5. concept
6. optional	6. transitional	6. time-phase
7. synchronized	7. incremental	7. projection
8. compatible	8. third-generation	8. hardware
9. balanced	9. policy	9. contingency

The procedure is simple. Think of any three digit number, then select the corresponding buzzword from each column. For instance, number 257 produces *systematized logistical projection*, a buzz-phrase that can be dropped into virtually any technical conversation or report to give it that ring of decisive, knowledgeable authority. "No one will have the remotest idea of what you are talking about," says Broughton, "but the important thing is that they're not about to admit it."

Many buzz-phrase generators have been developed since the original publication of the article, including a [corporate-speak generator](#) (*holistically embrace customer-directed imperatives*) and a [Shakespearean insult generator](#) (*Thou impertinent urchin-faced miscreant!*). You can Google for "buzz-phrase generator" to find lots of them on the web, but it's most fun to build your own.

Assignment

In this assignment we represent a **word list** as a sequence (a list or a tuple) of strings. Your task:

- Write a function `buzzphrase1` that takes a sequence (a list or a tuple) as its only argument. This sequence must contain one or more word lists. The function must randomly pick a word from each word list, and concatenate these words together (in the order of the given word lists) into a single string. Successive words should be separated by a single space. The function must return this string of words as its result.
- Write a function `buzzphrase2` that takes one or more word lists as its arguments. The function must randomly pick a word from each word list, and concatenate these words together (in the order the word lists are passed to the function) into a single string. Successive words should be separated by a single space. The function must return this string of words as its result.

Example

```
>>> buzz1 = ('integrated', 'total', 'systematized', 'parallel', 'functional', 'responsive', 'optional', 'synchronized', 'compatible', 'balanced')
>>> buzz2 = ('management', 'organizational', 'monitored', 'reciprocal', 'digital', 'logistical', 'transitional', 'incremental', 'third-generation', 'policy')
>>> buzz3 = ('options', 'flexibility', 'capability', 'mobility', 'programming', 'concept', 'time-phase', 'projection', 'hardware', 'contingency')
>>> buzzphrase1((buzz1, buzz2, buzz3))
'compatible monitored capability'
>>> buzzphrase2(buzz1, buzz2, buzz3)
'optional logistical time-phase'

>>> shakespeare1 = ['Thou']
>>> shakespeare2 = ['artless', 'bawdy', 'beslubbering', 'bootless', 'churlish', 'cockered', 'clouted', 'craven', 'currish', 'dankish', 'dissembling', 'droning', 'errant', 'fawning', 'fobbing', 'froward', 'frothy', 'gl']
>>> shakespeare3 = ['base-court', 'bat-fowling', 'beef-witted', 'beetle-headed', 'boil-brained', 'clapper-clawed', 'clay-brained', 'common-kissing', 'crook-pated', 'dismal-dreaming', 'dizzy-eyed', 'doghe']
>>> shakespeare4 = ['apple-john', 'baggage', 'barnacle', 'bladder', 'boar-pig', 'bugbear', 'bum-bailey', 'canker-blossom', 'clack-dish', 'clotpole', 'coxcomb', 'codpiece', 'death-token', 'dewberry', 'flap-dra']
>>> buzzphrase1((shakespeare1, shakespeare2, shakespeare3, shakespeare4))
'Thou wayward doghearted joltthead'
>>> buzzphrase2(shakespeare1, shakespeare2, shakespeare3, shakespeare4)
'Thou fobbing half-faced flap-dragon'
```

Een **zoemwoord** (Engels: *buzzword*) is een modewoord dat geen duidelijke betekenis heeft. Een voorbeeld is het woord *manager*: zonder verdere omschrijving is het immers niet duidelijk wat dit beroep precies inhoudt. Als de term uit meerdere woorden bestaat, dan spreken we van een **zoemzin**. Een voorbeeld hiervan is *think outside the box*.



In 1968 publiceerde *Newsweek* een humoristische artikel [How to Win at Wordsmanship](#) van de hand van Philip Broughton. Hierin beschreef hij een waterdichte methode om frustratie om te zetten in (taalkundige) daadkracht. Broughton's methode werd eufemistisch de *Systematic Buzz Phrase Projector* gedoopt, en gebruikt een lexicon van 30 zorgvuldig uitgekozen zoemwoorden.

0. integrated	0. management	0. options
1. total	1. organizational	1. flexibility
2. systematized	2. monitored	2. capability
3. parallel	3. reciprocal	3. mobility
4. functional	4. digital	4. programming
5. responsive	5. logistical	5. concept
6. optional	6. transitional	6. time-phase
7. synchronized	7. incremental	7. projection
8. compatible	8. third-generation	8. hardware
9. balanced	9. policy	9. contingency

De procedure is heel eenvoudig: denk aan een getal dat bestaat uit drie cijfers en selecteer het corresponderende zoemwoord uit elke kolom. Zo produceert het getal 257 de zoemzin *systematized logistical projection* die je in elke technische discussie of elk technisch rapport kunt laten vallen om gewichtig over te komen. Broughton zei hierover: "Geen kat zal weten waarover je het hebt, maar het belangrijkste is dat ze dat nooit zullen toegeven."

Sinds dit artikel werden er tal van generatoren voor zoemzinnen ontwikkeld, onder andere voor [ambitieuze managers die carrière willen maken in het bedrijfsleven](#) (*holistically embrace customer-directed imperatives*) of [om te kunnen vloeken zoals Shakespeare](#) (*Thou impertinent urchin-faced miscreant!*). Als je Google gebruikt voor "buzz-phrase generator" dan vind je heel wat voorbeelden op het Internet, maar het leukste is natuurlijk om er zelf een in elkaar te knutselen.

Opgave

In deze opgave stellen we een **woordenlijst** voor als een sequentie (een lijst of een tuple) van strings. Gevraagd wordt:

- Schrijf een functie `zoemzin1` waaraan een sequentie (een lijst of een tuple) als argument moet doorgegeven worden. Deze sequentie moet één of meerdere woordenlijsten bevatten. De functie moet willekeurig uit elke woordenlijst een woord kiezen, en deze woorden samenvoegen tot één enkele string (in de volgorde van de gegeven woordenlijsten). Hierbij moeten de woorden telkens van elkaar gescheiden worden door één enkele spatie. De functie moet deze string van woorden als resultaat teruggeven.
- Schrijf een functie `zoemzin2` waaraan één of meerdere woordenlijsten als afzonderlijke argumenten moeten doorgegeven worden. De functie moet willekeurig uit elke woordenlijst een woord kiezen, en deze woorden samenvoegen tot één enkele string (in de volgorde waarin de woordenlijsten aan de functie worden doorgegeven). Hierbij moeten de woorden telkens van elkaar gescheiden worden door één enkele spatie. De functie moet deze string van woorden als resultaat teruggeven.

Voorbeeld

```
>>> buzz1 = ('integrated', 'total', 'systematized', 'parallel', 'functional', 'responsive', 'optional', 'synchronized', 'compatible', 'balanced')
>>> buzz2 = ('management', 'organizational', 'monitored', 'reciprocal', 'digital', 'logistical', 'transitional', 'incremental', 'third-generation', 'policy')
>>> buzz3 = ('options', 'flexibility', 'capability', 'mobility', 'programming', 'concept', 'time-phase', 'projection', 'hardware', 'contingency')
>>> zoemzin1(buzz1, buzz2, buzz3)
'compatible monitored capability'
>>> zoemzin2(buzz1, buzz2, buzz3)
'optional logistical time-phase'

>>> shakespeare1 = ['Thou']
>>> shakespeare2 = ['artless', 'bawdy', 'beslubbering', 'bootless', 'churlish', 'cockered', 'clouted', 'craven', 'currish', 'dankish', 'dissembling', 'droning', 'errant', 'fawning', 'fobbing', 'froward', 'frothy', 'gl']
>>> shakespeare3 = ['base-court', 'bat-fowling', 'beef-witted', 'beetle-headed', 'boil-brained', 'clapper-clawed', 'clay-brained', 'common-kissing', 'crook-pated', 'dismal-dreaming', 'dizzy-eyed', 'doghe']
>>> shakespeare4 = ['apple-john', 'baggage', 'barnacle', 'bladder', 'boar-pig', 'bugbear', 'bum-bailey', 'canker-blossom', 'clack-dish', 'clotpole', 'coxcomb', 'codpiece', 'death-token', 'dewberry', 'flap-dr']
>>> zoemzin1([shakespeare1, shakespeare2, shakespeare3, shakespeare4])
'Thou wayward dogheated jolthead'
>>> zoemzin2(shakespeare1, shakespeare2, shakespeare3, shakespeare4)
'Thou fobbing half-faced flap-dragon'
```