

Building an IT System: The Foolish Dictionary


Now that we have learnt a wide range of the principles involved in “building IT systems”, in this exercise you and your colleagues will build a non-trivial IT system as a team. This will be done in a way similar to the corresponding lecture demonstration, but taking a few short-cuts to make the process shorter.

Requirement

Accompanying these instructions is a copy of an old book, *The Foolish Dictionary*, written by the pseudonymous author Gideon Wurdz in 1904. This book is in the public domain and can be found in many versions on the Internet, such as the online version shown below.

The
FOOLISH
DICTIONARY
online edition

A chip of the old block -- A daughter of the Tenderloin.




One man's meat is another man's finish -- Canned Beef in Cuba.


ANGAROO A hard drinker from Australia, especially fond of hops, and generally carrying a load.

KATYDID A gossiping grasshopper who is always meddling in Katy's affairs.

KEEPSAKE Something given us by someone we've forgotten.

KERNELS or COLONELS Articles often found in cores (or corps) and frequently surrounded by shells.

KEROSENE

An alleged provider of heat and light. From Lat. *carus*, meaning expensive and *seneo*, to be weak; expensive but weak. For further explanation, consult Standard Oil Company.

KEYHOLE

A frequent test for sobriety.

KID Either a boxing-glove or a first-born. In either case, hard to handle until well tanned.

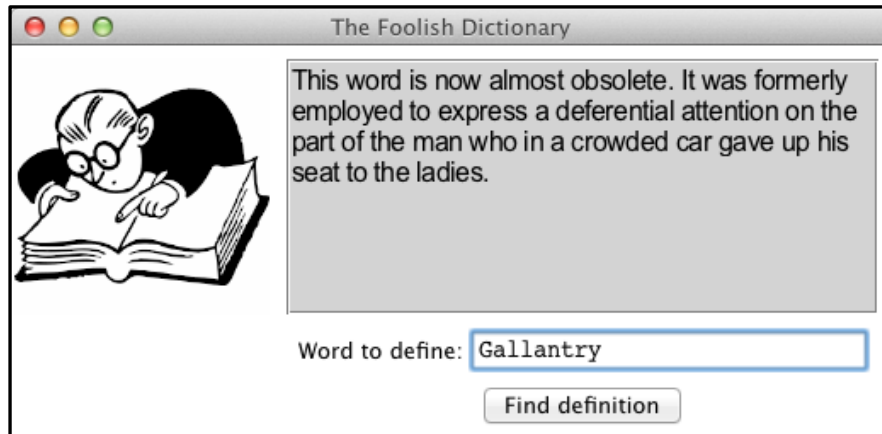
KILTS A Scotchman's apology for indecent exposure.

The book is a collection of *supposedly* humorous definitions for various words. In the intervening century, however, much of the humour seems to have dissipated! There's little in this book to “LOL” about for a 21st century reader! Many of the terms and references to contemporary society are lost on a modern audience.

Nevertheless, your task is to develop an “app” which allows the user to easily find definitions for various words appearing in this dictionary. To do so, you will work with your workshop

colleagues as a team, separately preparing each part of the necessary program code and putting the result together at the end.

The basic idea is to produce a program with a Graphical User Interface something like the following. Feel free to change its “look-and-feel”.



There is a text entry field for the user to enter a word, a button to push to make the corresponding definition appear, and a text area to display the definition from the dictionary (if one can be found). For visual interest we have also added an icon to the GUI in this particular example.

Step 0 - A Crude Prototype

We have provided a plain text (sans illustrations) copy of this dictionary in the file `TheFoolishDictionary_GideonWurdz_1904.txt`. Open this file in a text editor and study its contents. As a very simple version of the “app” we require, you can just use the search/find feature built into your text editor to find definitions. Do this to find the entry for the word “GALLANTRY” as shown above.

Obviously this approach works, but it’s not very convenient. Your team’s job is to build a nicer application for this purpose. This will be done in two parts, the back-end function that searches through the dictionary and produces the definition, and the front-end user interface. Divide your workshop group into two teams, one to work on each part.

Step 1a - The Back-End Search Function

This team’s goal is to develop the back-end function which finds words in the dictionary and displays their definition. See the “Step 1a” Python template file for more detail.

As you will have seen when studying the dictionary’s plain text source, each of the word definitions is formatted in a similar way, as shown overleaf.

```
=CHAMPAGNE= The stuff that makes the world go round.
*      *      *      *      *

=CHAIR= Four-legged aid to the injured.

=CHARITY= Forehanded aid to the indigent.
*      *      *      *      *

=CHAUFFEUR= A man who is smart enough to operate an automobile,
but clever enough not to own one.
*      *      *      *      *

=CHRISTIAN= A member of any orthodox church.
*      *      *      *      *
```

Note that the word being defined appears in capital letters with equals signs (=) on either side, followed by either a space or a newline character. This point marks the beginning of the definition. The definition ends when either an asterisk (*) or another equals sign is encountered. Your back-end function should take advantage of this pattern to produce the function which finds and prints the definition.

At this stage there is no front-end Graphical User Interface for your function, so it should just accept the word of interest as a parameter and print the results to Python's shell window. Later this code will be properly integrated into the GUI.

The back-end function also needs to be robust to potential errors. In particular, it should not crash if the dictionary file is missing, but should produce a meaningful error message in this case. Similarly, if the requested word does not appear in the dictionary, a message to this effect should be returned.

Step 1b - Graphical User Interface Prototype

While Team A is developing the back-end function, Team B should be developing the user interface, with a "stub" representing the incomplete back end. See the "Step 1b" Python template file for more detail and use the example on the previous page as a guide.

Step 2 - Integration and Fine Tuning

When both teams have completed their separate work they should get back together and integrate their results to produce the final "app". See the "Step 2" Python template file for more detail. The teams may need to modify the code they've written to make them fit together properly. This is also the opportunity to make final improvements to the complete IT system before delivering it to the satisfied customer!