Name: Date:

Lesson 6: My capital dictionary

Dictionary of me

Last week we were introduced to the dictionary data type, a very important structure which holds a collection of data by key and their associated values just like a dictionary holds words + their definitions. Values can be a wide variety of data types including string, integer, bool, list or even other dictionaries. In this example, one of the values is a string, the other a list:

```
>>> d={'animal':'monkey','vegetable':['carrot','potato']}
>>> d.keys()
['animal','vegetable']
>>> d.values()
['monkey',['carrot','potato']]
```

We can mix up data types to create a "dictionary of me" which describes some key facts about yourself. Your forename and surname could be string values in this dictionary, your age an integer, your gender a boolean, your favourite sports a list. You could also have a friends key that could be a list of dictionaries of other descriptions of others.

TRY IT OUT #1: Try creating a Python dictionary describing yourself. It can be built of any combination of other Python data types you like. List the keys and values.

Here's a simple example of a dictionary of me as outlined above. We can likewise use a for loop to iterate the dictionary built around a dictionary `.items()` method:

```
>>> john={'forename':'John', 'surname':'Smith', 'age':10,
'sports':['Rugby','Cricket'], 'boy':True, 'friends':[]}
>>> for k,v in d.items():
...         print(k,v)
('forename','John')
('surname','Smith')
('age',10)
('sports',['Rugby','Cricket'])
('boy',True)
('friends',[])
```

TRY IT OUT #2: Try iterating over the dictionary you already developed with a `for` loop.

raw_input

It's worth knowing at this point how to pick up user input using the `raw_input` built-in function. `raw_input` blocks at the prompt waiting for the user to type their input and hit ENTER. It returns the input string the user entered:

```
>>> s=raw_input("hello, print something>")
hello, print something>something
>>> s
'something'
```

random

Python has a special module called `random` which is used to support random number generation. The `randint()` method can be called on random to generate an integer in the supplied range. In this example we are generating random numbers in the range 0 to 10 and then 100 random numbers in the range 0 to 1.

```
>>> import random
>>> random.randint(0,10)
7
>>> random.randint(0,10)
3
>>> for i in range(0,100):
...   print(random.randint(0,1)
```

TRY IT OUT #4: Try generating random numbers in a range. Count how many of each number in the range are output. Do you notice anything interesting?

You should find that you have a close to equal count of all the numbers in a range. So, if you generate 100 random numbers in the range 0 to 1 as we are above, approximately 50 will be 0 and 50 will be 1.

Random number generation is very important for a variety of applications including lottery number and generating really secure codes that cannot easily be guessed. Such codes can be used in more complicated cipher systems.

Constructing a capital quiz

We can use the combination of `raw_input()`, `random` and a dictionary to build a quiz! Let's start with our dictionary which will have country names as its keys and corresponding capitals as its values.

TRY IT OUT #3: Try building a dictionary of capital cities – it should have a `len` of at least 10.

```
>>> capitals={'UK':'London', 'France':'Paris', 'Germany':'Bonn',
'Zimbabwe':'Harare', 'Ireland':'Dublin', 'Morocco':'Rabat'}
```

Now we can look at how we might pick a random country. To do this we need to retrieve our list of keys, choose a random integer i that lies between 0 and `len(dictionary)` and then retrieve the i'th value from the list. This snippet shows you how to do that:

```
i=random.randint(0,len(d)-1)
country=d.keys()[i]
capital=d[country]
```

We are picking a random number in the range required and then using that as the index into the keys array to retrieve our country. That country can in turn be used as the key into our countries dictionary to retrieve the corresponding capital.

Bringing it all together ….

At this point, you have the main elements you need to build an interactive capital quiz! That's what this week's Code Quest will focus on. As with other recent lessons, the full solution will be presented on the last page for you to look at.

SECRET CODE QUEST!

Write a Python program that uses a dictionary and `raw_input()` with `random` to create and interactive country capital quiz. It should ask the user to enter their name to start with and then guide them through typing in their responses to randomly selected countries in the dictionary. You will need to construct a country-capitals dictionary. You could use the one you constructed earlier in Try It Out #3. Or you might want to explore online resources online options. For instance, you could cut-paste the data here and then adapt your program to use it:

https://gist.githubusercontent.com/pamelafox/986163/raw/3d48bc4c21b8932332b97f329f91e5bc2853c322/countryinfo.py

Can you think how to make it work so that it ignores letter cases? In other words so that "Paris" and "paris" are equivalent as the capital of France?

```
import random

d={'UK':'London',
      'USA':'Washington',
      'Morocco':'Rabat',
      'Zimbabwe':'Harare',
      'India':'New Delhi',
      'Egypt':'Cairo',
      'Italy':'Rome',
      'Spain':'Madrid',
      'China':'Beijing',
      'France':'Paris'}

scoring=True
score=0
tries=0
name=raw_input("Please enter your name\n> ")
print("------------------------------------------------------------")
print("|                                                          |")
print("          Welcome to the capital quiz, %s!             " % name)
print("          We're going to give you a country            ")
print("          and ask you to type in its capital           ")
print("          You can press 'x' to exit at any time        ")
print("|                                                          |")
print("------------------------------------------------------------")
s=raw_input("Press any key to continue, %s " % name)
print("------------------------------------------------------------")
while scoring:
    i=random.randint(0,len(d)-1)
    country=d.keys()[i]
    capital=d.get(country)
    guess=raw_input("What is the capital of '%s'> " % country)
    if guess=='x' or tries>=10:
        scoring=False
        print("Exiting after %d tries...." % tries)
    else:
        if guess==capital:
            print("☺ the capital of '%s' is '%s'" % (country,guess))
            score+=1
        else:
            print("☹ The capital of '%s' is '%s'" % (country,capital))
        tries+=1
        print("------------------------------------------------------")
print("============================================================")
percentage=0
if tries:
    percentage=(float(score)*100)/tries
print("------------------------------------------------------------")
print("|                                                          |")
print("   Goodbye %s!  You scored %d/%d (%s%%) on capitals today!" %
(name,score,tries,round(percentage,1)))
print("|                                                          |")
print("------------------------------------------------------------")
```