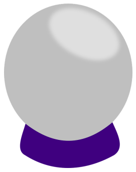
Function Fortune Teller

A Teknowledge Activity



Go to trinket.io → Sign in
Then go to **tinyurl.com/codingPython**
Continue with the rest of the lab!

Student Learning Goals:

- ***Learn about***: Functions -- why, what, and how

1.0 Your First Fortune!

Copy the following code.

```
import random

def randomAnimal():
    return random.choice(["dog", "snail"])

print(“Let me tell your fortune!”)
print(“tomorrow, you will see a”, randomAnimal(), “and a”, randomAnimal())
```

The **random.choice** function randomly returns one of the items in the list given to it. *Don’t worry about what a ‘list’ is just yet!*

Run your code several times.

- Why does it sometimes say the same animal twice in one fortune?
- Try adding some more animals to the fortune-teller’s vocabulary.
- Change the fortune to something new.

1.1 More Fortunes!

- Add a function called **randomColor()** which returns a random color. (Hint: It is very similar to how randomAnimal works)
- Change the fortune to something new that uses your animal and color functions.

2.0 How Many Functions?

Add the following function **before your fortune**.

```
def many(number, noun):
    if not (number == 1):
        noun += "s"
    return str(number) + " " + noun
```

and change your “tomorrow, you will see a…” fortune line to be:

```
numberOfPizzas = random.randint(1, 3)
print(“tomorrow, you will buy”, many(numberOfPizzas, “pizza”))
```

Run your code several times.

- What do you think **random.randint** does?
- What happens if you use the **many** function on a noun like “woman”? Add an if statement to handle this special case.

2.1 Using It All

Change your code to tell a fortune involving many of a random animal.
(Hint: use a variable and call the **randomAnimal** function from above)

3.0 This is Omenzing!

Functions are great because they can be combined and reused.
Add the following function.

```
def randomOmen():
    return "you see a " + randomColor() + " " + randomAnimal()
```

- Ask your mentor why the randomOmen function needs to be **after** the **randomColor** and **randomAnimal** functions.

Now you can have a complex fortune like this:

```
numberOfAnimals = random.randint(1, 3)
animal = randomAnimal()
animalGroup = many(numberOfAnimals, animal)
print("if", randomOmen() + ",", “look out for”, animalGroup)
```

- Before you run your code, walk through each line predicting what each could do!

4.0 Add New Features!

What fortunes can you create?
Here are some ideas to get you started:

- Add more randomly generated animals and colors.
- Add a function to generate random places, random people, random actions, or anything else. Use these to create new fortunes.
- Change **randomOmen()** to generate different styles of omens like this:

```
def randomOmen():
    n = random.randint(1, 3)
    if n == 1:
        return "you see a " + randomColor() + " " + randomAnimal()
    elif n == 2:
        return "SECOND OMEN STYLE HERE”
    else:
        return "THIRD OMEN STYLE HERE”
```

- Make one of the omen styles have a higher chance of being picked.
- Allow the user to pick whether he wants a good or bad omen. (Use input)
- Using a for loop, have multiple omens be told every time the code runs.
- Using a while loop, keep having omens told until you type a certain input.
- Add a function that tells you how ‘good’ an omen was based on criteria you create!

5.0 Saving Code

saveYourCode =
True