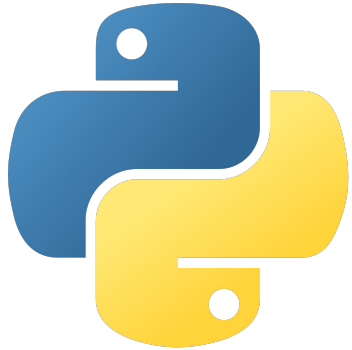
Introduction to Python: Week 2



Slides by Coding Leads
November 2023

Functions

What's a function?

- A function does something.
- It can take inputs, it can give back outputs - but doesn't need to do either.

Why use functions?

- To keep things tidy.
- To avoid repeating code (writing things in several different places).

Function examples

Going back to our hello world example.

```
def hello_world():
  print('hello world')
```

Then we ‘call’ the function later.

```
hello_world()
```

With inputs/outputs:

```
def print_x_times(my_phrase, x):
    print(my_phrase * x)
    return x + 1
```

We can specify the types of our inputs (e.g. `my_phrase: str`) to show people what we expect the inputs to be, but it doesn’t actually stop it from taking other input types.
e.g. `print_x_times(10, 10)`
prints the number 100.

Function questions.

- Write a function that takes a boolean as an input and returns the opposite.
- Write a function to calculate the area of a circle. Call your function with different values for the radius.
- Write a function to calculate acceleration. If your speed is 200 m/s, and after 10 seconds your speed was 500 m/s, what was the acceleration?

Let’s build a coin flipping game?

Coin flipping game.
Guess if it’s heads or tails.
Start with £5, lose £2 every time you are wrong, gain £1 every time you are right.

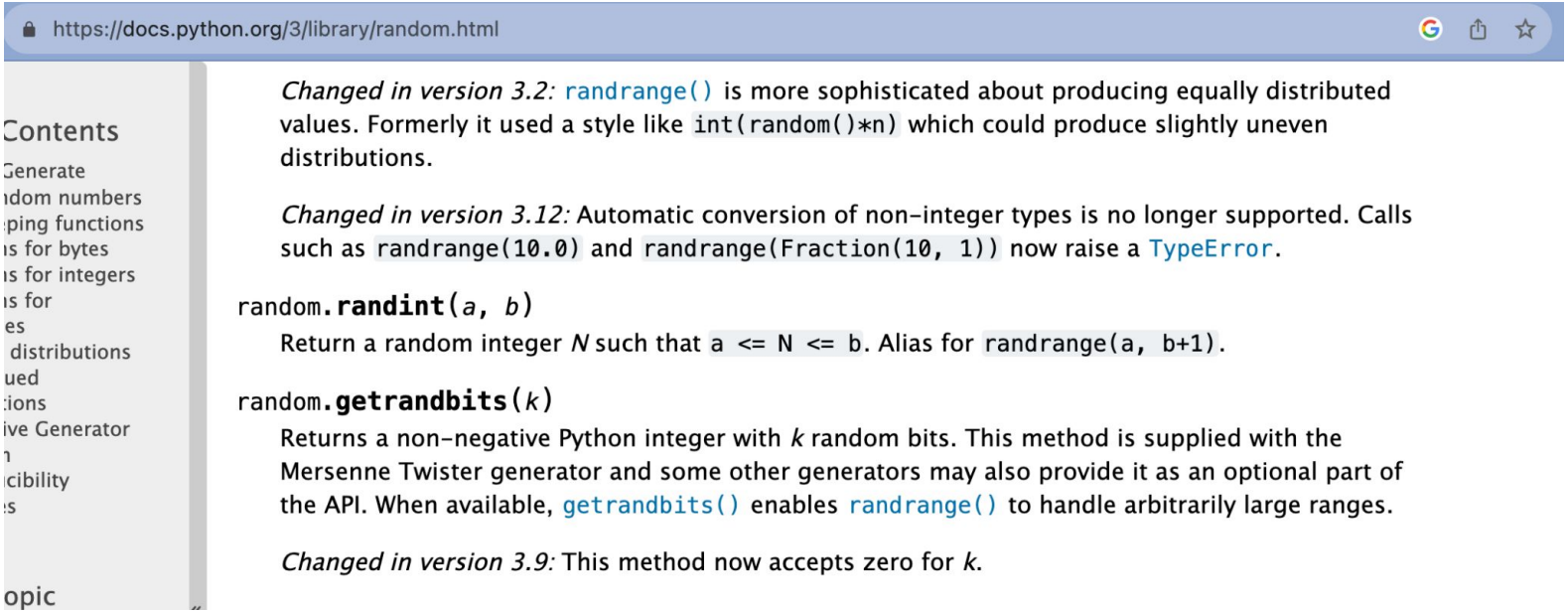
Write a function to get the guess from the user (hint: input)
Write a function to flip the coin (random)
If guess = coin toss -> increase winnings.
Write a function to increase winnings, which can also end the game when winnings reach 0.

```
input(“What’s your guess?”)
```

Note - you need double quotes for the string here because there’s an apostrophe.

Generate a random number

```
import random
random.randint(1,2)
```



From python docs

Arrays

- Data Type
- Holds multiple values, like a list.

Eg. coding_leads = [“Beeeeebek”, “Shady”, “Niki”]

There is a type in the 1st array element and we want to fix it. Here is what changing an array value looks like:

coding_leads[0] = “Beebek”

Arrays

If you need to store one of these array values in a variable you would do this:

lead1 = coding_leads[0] - What value would lead1 have?

We can use the len() method to return the length of an array.

How can we return the length of coding_leads? (Hint, it’s similar to print())

Arrays

The append() method lets you add an element to an array. Since the append() method is exclusive to arrays it has this syntax:
coding_leads.append(“Sophie”)

This adds Sophie to the coding_leads array.

- How would I print Sophie using the array?

The pop() method removes an array element using the location of the element:

coding_leads.pop(3) - What would this remove from the array?

Arrays

The remove() method removes an array element based on its value:

coding_leads.remove(“Shady”)

Both pop() and remove() push all elements down in the array.

For example if “Shady” is removed, “Niki” will then become the element in position coding_leads[1]

What is a “for loop” and why do you need it?

A tool that let us repeat a block of code multiple times.

Imagine: You want to print “hello” 3 times

```
print(“hello”)
print(“hello”)
print(“hello”)
```

For loop 3 times:

```
    print(“hello”)
```

For loop syntax in Python (Do **NOT** forget indentations!)

Guess the output:

```
for i in range(10):
    print(i)
```

```
for i in range(5,10):
    print(i)
```

```
for i in range(5,10, 2):
    print(i)
```

Why does this happen? (Python Documentation)

range(start, stop[, step])

Tasks:

Print all numbers from 30 to 60 (inclusive)

Print all odd numbers from 25 to 48 (inclusive)

Print all even numbers from 62 to 80 (exclusive) without using step argument

Answers

```
for i in range(30, 61):
    print(i)
```

```
for j in range(25, 49, 2):
    print(j)
```

```
for k in range(62, 80):
    if k % 2 == 0:
        print(k)
```

For loops with Arrays

```
letters = [“a”, “b”, “c”, “d”]


for i in range(len(letters)):
        print(letters[i])


for letter in letters:
        print(letter)
```

What is a “while loop” and why do you need it?

Used when you don't know how many times you'll loop.

Imagine: Filling a glass with water

```
for i in range(100):
    if glassFull() != True:
        pourWater()
    else:
        return
```

```
while !glassFull():
    pourWater()
```

Assumptions of this code?

Simple Counter Technique

```
count = 0


while count < 10:
        print(count)
        count += 1
```

Resources

https://docs.python.org/3/
https://www.w3schools.com/ (Tutorials and information for lots of languages, including python)
Freecodecamp.org

https://www.freecodecamp.org/news/python-array-tutorial-define-index-methods/ - For arrays

Our GitHub:
https://github.com/Shadyabu/EngAndTechSociety

Downloading. We recommend (all free):

IDEs - nice interfaces to write your code, run your code
https://code.visualstudio.com/
Supports lots of different languages

https://www.jetbrains.com/community/education/#students
Free for students.
They have lots of different apps for different languages.

Installing python and relevant packages / libraries (specific things e.g. for machine learning, plotting graphs, draw things)
https://brew.sh/
Installing python and packages on macOS

https://www.anaconda.com/
Also has Jupyter notebooks