Table of Contents

Introduction

This document will introduce the concept and importance of modules, and documentation in Python.

Videos

- Modules and Documentation (0:00 - 11:48)
    - Covers the official Python documentation, random module documentation, and how to use and understand documentation

1 Modules

A module is a file that contains function definitions, variables and other types of statements. This is done to keep related code grouped together and easily accessible to provide functionality specific to certain tasks. It is also used to break down large programs into more manageable sizes, which help organize the code.

To use functions and other things from a module, we'll first need to 'import' it into our code. Let's take a look at the Python random module for example:

> The Python random module contains many functions that can allow us to obtain a random number from a given range, obtain a random item from a given list, etc. More information on the Python random module can be found For this example, let's pick a random item from a list, using the function `random.choice()`.

```
# making the random module available to us
import random
my_list = ['apple', 'pear', 'cherry', 'watermelon']

# picking a random item from the list
random_item = random.choice(my_list)

# printing the random item
print(random_item)
```

There are many other modules available such as the string or math modules among many others. More info on modules can be found in the module section of the official Python documentation, and for a full list of Python modules, refer to the Python module index. Understanding when to use them can make coding easier and more efficient.

2 Documentation

Documentation is an important part of coding. It is mainly used to describe how the code works and how it can be used, but documentations can also be used to help keep track of what needs to be done, or what a block of code does. Even if a program or a piece of code is amazing, if it cannot be understood by others easily, it will likely not be used.

Documentation is organized in a manual like format with mostly text and little code. The Python Standard Library has documentation for concepts, different built-in functions, and modules of Python (eg. random and time).

Let's take a look at the Python random module for example:

The documentation begins by explaining what this module does, how it works, and additional functionalities that it can provide. It then goes on to explain each function found within this module. For this example, we'll take a look at the `random.choice()` function, which can be found under the functions for sequences section.

> `random.`**`choice`**(*`seq`*)
>
> Return a random element from the non-empty sequence *seq*. If *seq* is empty, raises `IndexError`.

The first line shows the syntax for the function call, including what arguments are needed for the function to function properly. The next lines describe what the function does, if it returns anything, and all of the listed arguments, if any. In this case, the description tells us that this function takes in one argument (referred to as *seq* here), which has to be a sequence (for example an object of type list) that contains at least one item, and from this sequence, one item will be selected and returned. It also states that if the given sequence is empty, there will be an error.

`>>> import random`
`>>> my_list = ['cat', 'dog', 'bird']`
`>>> random.choice(my_list)`
`'cat'`
	To access functions from a module, we must first import it. Then we can call the choice function, passing in an argument that is a sequence (in this case, a list of str objects), which will choose a random object from the list and return it. Here, the str object `'cat'` is randomly chosen.