Random Numbers

In this lesson, we will learn about generating random numbers and how to choose random numbers from a set of numbers.

WE'LL COVER THE FOLLOWING ^

- Generating random numbers
- Not so random?
- Extracting random numbers

Generating random numbers

A random number generator lets you draw a random number from a specified distribution. Several random number generators are included in the random subpackage of numpy. We will import this package as rnd to save some typing.

import numpy.random as rnd

One of the examples for random number generators is the randint() function. The randint(low, high, size) function returns an array consisting of size integers, each is drawn randomly from low up to, but not including, high.

Use the help('numpy.random') command to learn about more random number generators in Python.

Suppose we flip a coin 10 times, assigning o to heads and 1 to tails. We will specify high as 1 + 1, which means it is 1 higher than the desired maximum value. So, this specific command generates an array of o and 1.

Let's see how we can implement this in our code.

```
import numpy as np
import numpy.random as rnd

print(rnd.randint(0, 1 + 1, 10))
```

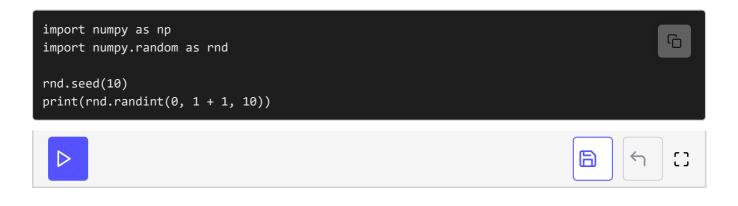
We will call randint() with the high argument set to 2, (1 higher than the maximum value) and the low argument set to 0. Run the code above again to see this.

Not so random?

Internally, the random number generator starts with what is called a **seed**. The seed is a number that is generated automatically (and supposedly at random) when you call the random number generator.

Since the value of the seed defines the sequence of random numbers that you get, some people may argue that the generated sequence is pseudo-random at best. You may not want to use the sequence for any serious cryptographic problems, but for our purposes, they are random enough.

In the code below, let's set seed equal to 10:



If we call the randint() function again, we get the same sequence of zeros and ones. Run the code above again to see this.

The ability to generate the exact same sequence is useful during code development. For example, by seeding the random number generator, you

can compare your output with the output of others trying to solve the same problem.

Extracting random numbers

The numpy.random module allows you to draw random numbers from a given
array using the choice() function. The input arguments of choice() are the
sample array and size of the randomly chosen values.

```
rnd.choice(array, size, replace)
```

The choice() function has an argument replace which specifies whether values are picked with replace or not. The value of replace is a boolean.



Run the code repeatedly and see the change in the output.

In the next lesson, we will learn about a common problem with random variables: flipping coins.