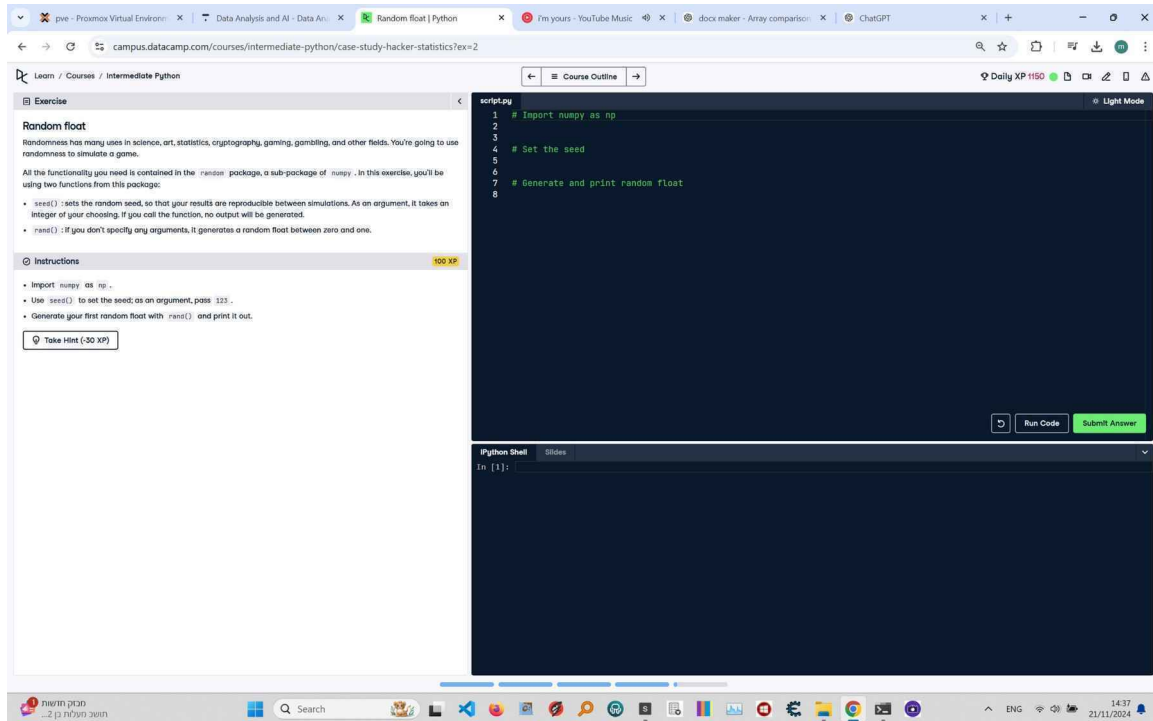


Random Float in Python



****Question:****

Randomness has many uses in science, art, statistics, cryptography, gaming, gambling, and other fields. You're going to use randomness to simulate a game.

All the functionality you need is contained in the `random` package, a sub-package of `numpy`. In this exercise, you'll be using two functions from this package:

- `seed()`: sets the random seed so that your results are reproducible between simulations. As an argument, it takes an integer of your choosing; if you call the function, no output will be generated.
- `rand()`: if you don't specify any arguments, it generates a random float between zero and one.

****Instructions:****

1. Import `numpy` as `np`.
2. Use `seed()` to set the seed; as an argument, pass `123`.
3. Generate your first random float with `rand()` and print it out.

****Answer:****

Here is the Python code that solves the problem:

```
# Import numpy
import numpy as np

# Set the random seed
np.random.seed(123)

# Generate and print random float
random_float = np.random.rand()
print(random_float)
```

****Explanation:****

1. ****Importing numpy****: The `numpy` library is imported as `np`, allowing you to use its functions with the alias `np`.
2. ****Setting the random seed****: The `np.random.seed()` function ensures reproducibility of the random number generation. Passing `123` as the argument initializes the random number generator to a fixed state.
3. ****Generating a random float****: The `np.random.rand()` function generates a random float uniformly distributed between 0 and 1.
4. ****Printing the result****: The generated random float is stored in the variable `random_float` and printed to the console.