Simulating a Coin Flip

- Pick a random whole number between 0 and 1, inclusive
- Show coin flip in Python using `random.randint(0, 1)`

Pseudorandom vs. Random

- Programming languages have a `random` function that generates an *apparently* random number—but they are not truly random. They follow a deterministic algorithm.
- These numbers are referred to as *pseudorandom*, and they are generated by using an initial number called a *seed*, which has some irreversible mathematical operations applied to it.
- (If you've played Minecraft, this is essentially what the world seed does. If two players input the same seed, they get the same randomly generated world)
- We're not going to go into how these computers actually generate pseudorandom numbers, but it is important to know that although they seem random, they are actually quite predictable if you run the generator enough times.
- In some cases, such as in cryptography, we need *truly* random numbers. Quantum computers can do that, because quantum mechanics are fundamentally probabilistic.

Quantum Coin Flip

Simply apply a Hadamard gate to a single qubit and then run on the quantum computer (or simulator), passing `shots = 1` into the `execute()` method from Qiskit.

Quantum Random Number Generator

- Ask the user to input an upper bound for a random number
- Calculate the number of (qu)bits needed to represent that number

- Generate random values (coin flips!) for each qubit
- Convert the set of randomly generated qubits values to base 10
- Reject and re-generate if the result is over the upper bound provided by the user