

CSCI 127: Joy and Beauty of Data

Lecture 8: Modules

Reese Pearsall

Snowmester 2020

<https://reese.github.io/classes/127/main.html>

Announcements

Lab 5 due Tuesday 12/15 @ 11:59 PM (files)

Program 2 due Wednesday 12/16 @ 11:59 PM



Today

Modules (Math and Random), and Strings

Modules

Modules are an external resource (python code) that provide a series of functionality for us to use.
We can import them to easily do things without needing to code it ourselves

For example, python can't generate random numbers on its own, so we imported the **random** module to give us the ability to generate random numbers

There are thousands of modules out there (some require a more tedious download process)

As programmers, we often rely on external modules and libraries

List of Python modules we can easily import

Global Module Index - <https://docs.python.org/3/py-modindex.html>

Math Module

The math module gives us access to plethora of mathematical functions

<https://docs.python.org/3/library/math.html#module-math>

Random Module

The random module gives us access to **(pseudo)** random number generation

<https://docs.python.org/3/library/random.html#module-random>

What is a pseudo random number generator?

A **deterministic** mathematical formula for generating sequences of numbers

Not truly random— How are we supposed to tell our computer to pick a random number?

A **seed** is given to the formula as a starting point (this seed changes each time you run a program)

What is a pseudo random number generator?

A **deterministic** mathematical formula for generating sequences of numbers

Not truly random— How are we supposed to tell our computer to pick a random number?

A **seed** is given to the formula as a starting point (this seed changes each time you run a program)

42



PRNG

Seed is passed into the
formula to generate the first
number (this is usually based
your computer's clock)



PRNG

First number is generated



31



Previous value gets
passed into the
formula to generate
next number

PRNG

31

PRNG

Second number is generated




31 17



PRNG

31 17 42



PRNG

31 17 42 56 74 11 12 75 66 32 54 61 22 16 39 64 98 50 49 41 42 36 28 45 7 47 76 97 69
37 2 73 34 60 92 59 18 82

PRNG

31 17 42 56 74 11 12 75 66 32 54 61 22 16 39 64 98 50 49 41 42 36 28 45 7 47 76 97 69
37 2 73 34 60 92 59 18 82

[...]

PRNG

31 17 42 56 74 11 12 75 66 32 54 61 22 16 39 64 98 50 49 41 42 36 28 45 7 47 76 97 69
37 2 73 34 60 92 59 18 82

[...]

... 93 38 86 6 40 20 5 88 4 68 70 95 84 35 9 83 77 19 17 62 48 15 33 43 46 63 31 17 42 56
74 11 12 75 66 32 54 61 22 16 39 64 98 50 49 41 42 36 28 45 7 47 76 97 69 37 2 73 34 60
92 59 18 82

PRNG

Notice anything?

31 17 42 56 74 11 12 75 66 32 54 61 22 16 39 64 98 50 49 41 42 36 28 45 7 47 76 97 69
37 2 73 34 60 92 59 18 82

[...]

... 93 38 86 6 40 20 5 88 4 68 70 95 84 35 9 83 77 19 17 62 48 15 33 43 46 63 31 17 42 56
74 11 12 75 66 32 54 61 22 16 39 64 98 50 49 41 42 36 28 45 7 47 76 97 69 37 2 73 34 60
92 59 18 82

PRNG

Our sequence will eventually start over and repeat!

31 17 42 56 74 11 12 75 66 32 54 61 22 16 39 64 98 50 49 41 42 36 28 45 7 47 76 97 69
37 2 73 34 60 92 59 18 82

[...]

... 93 38 86 6 40 20 5 88 4 68 70 95 84 35 9 83 77 19 17 62 48 15 33 43 46 63 **31 17 42 56**
74 11 12 75 66 32 54 61 22 16 39 64 98 50 49 41 42 36 28 45 7 47 76 97 69 37 2 73 34 60
92 59 18 82

PRNG

How long until the random number sequence starts to repeat?

31 17 42 56 74 11 12 75 66 32 54 61 22 16 39 64 98 50 49 41 42 36 28 45 7 47 76 97 69
37 2 73 34 60 92 59 18 82

[...]

}
????

... 93 38 86 6 40 20 5 88 4 68 70 95 84 35 9 83 77 19 17 62 48 15 33 43 46 63 **31 17 42 56**
74 11 12 75 66 32 54 61 22 16 39 64 98 50 49 41 42 36 28 45 7 47 76 97 69 37 2 73 34 60
92 59 18 82

PRNG

This is typically referred to as the period. The period for Python PRNG's is $2^{19937}-1$

31 17 42 56 74 11 12 75 66 32 54 61 22 16 39 64 98 50 49 41 42 36 28 45 7 47 76 97 69
37 2 73 34 60 92 59 18 82

[...]

} ????

... 93 38 86 6 40 20 5 88 4 68 70 95 84 35 9 83 77 19 17 62 48 15 33 43 46 63 31 17 42 56
74 11 12 75 66 32 54 61 22 16 39 64 98 50 49 41 42 36 28 45 7 47 76 97 69 37 2 73 34 60
92 59 18 82

PRNG



This is typically referred to as the period. The period for Python PRNG's is $2^{19937}-1$

31 17 42 56 74 11 12 75 66 32 54 61 22 16 39 64 98 50 49 41 42 36 28 45 7 47 76 97 69
37 2 73 34 60 92 59 18 82

[...]

} ????

... 93 38 86 6 40 20 5 88 4 68 70 95 84 35 9 83 77 19 17 62 48 15 33 43 46 63 31 17 42 56
74 11 12 75 66 32 54 61 22 16 39 64 98 50 49 41 42 36 28 45 7 47 76 97 69 37 2 73 34 60
92 59 18 82

So, what does this mean???

If you know how the underlying PRNG works for some application, and you know what you **seed** you are giving it:

You can determine what random numbers will be generated

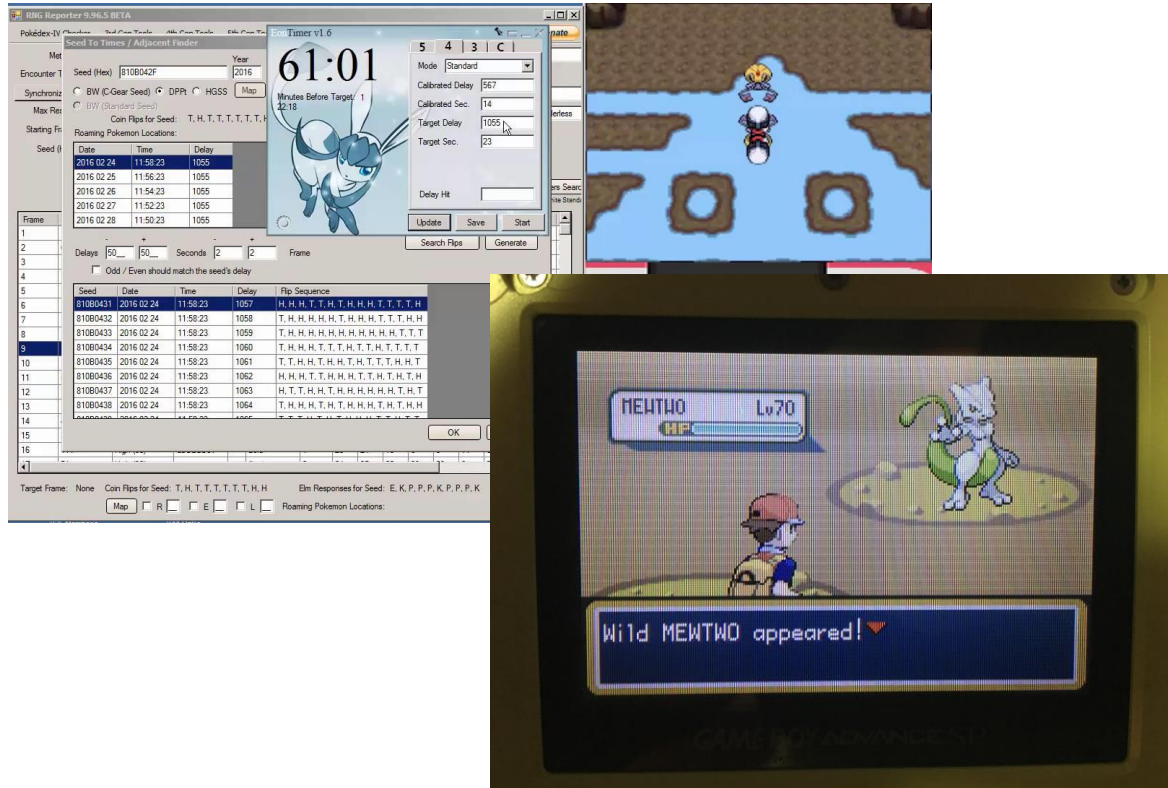
This is something to think about--- We rely on random number from computers all the time

FYI -- It's really difficult to successfully crack PRNGs like this. There are many protections in place to prevent humans from cheating the system

But let's look at times when PRNGs have been exploited.....

Video Games

Video games **heavily** rely on random numbers. The gameplay and what players experience is often determined by RNG



Some players have figured out how certain game's PRNGs work, and they determined how the seed is determined

Which means if they have the correct setup, they can **control** the outcome of the game

Gambling Machines

Hacking Slot Machines by Reverse-Engineering the Random Number Generators

Interesting story:

The venture is built on Alex's talent for reverse engineering the algorithms — known as pseudorandom number generators, or PRNGs — that govern how slot machine games behave. Armed with this knowledge, he can predict when certain games are likeliest to spit out moneyinsight that he shares with a legion of field agents who do the organization's grunt work.

These agents roam casinos from Poland to Macau to Peru in search of slots whose PRNGs have been deciphered by Alex. They use phones to record video of a vulnerable machine in action, then transmit the footage to an office in St. Petersburg. There, Alex and his assistants analyze the video to determine when the games' odds will briefly tilt against the house. They then send timing data to a custom app on an agent's phone; this data causes the phones to vibrate a split second before the agent should press the "Spin" button. By using these cues to beat slots in multiple casinos, a four-person team can earn more than \$250,000 a week.