Randomness

Time

```
import time
print(time.time()) # try running it multiple times, last number will keep
on changing
```

Notice that it is probably printing the time in some weird format.

- This time is called Epoch time - number of seconds have passed since Jan 1, 1970, 00:00
- Advantage:
    - We can do simple Mathematics with it say add, subtract - will be difficult with string.
    - This time is time zone independent.

```
ticks  = time.time()
localtime = time.localtime(ticks)
print(localtime) # this will give time locatime object
print(time.asctime(localtime)) # more human readable format
```

Note that the local time will be affected by the time-zone you have selected on your computer.

Generate random numbers

Let's discuss different ways in which we can create random numbers. Whenever we are calling it, it should ideally give us different numbers.

```
import random
print(random.random()) # try runninng it multiple times
```

Although, the code to generate random number remains same, the thing that is changed in the environment is time. Random uses the current time to generate random numbers, even if you run it very quickly, since time.time() is very granular in nature. Though it can be a factor of various different factors for example, CPU cycle, some ID on your hardware besides the epoch time or current time.

Also, The input that goes in is called seed (fix the input).

```
random.seed(10)

print(random.random()) # try running multiple times, it will be give same
answer
print(random.random()) # if printed twice or more, it prints different
numbers
```

Although, it is printing four different numbers, the numbers remain same. By setting the seed, we have disabled the complete randomness - the input is fixed, the output is also fixed.

The question is why would you want to set the seed?
“”””Let's say you generate some random data, let's say, for your data science project, you may want to replicate the results when you are showing results to someone else.”””"

Generate random numbers in a specified range

We have noticed that the random.random generates numbers between [0,1).

```
r = random.random()
print(r * 100) # to generate random numbers in [0, 100)
print(90 + r*10) # generates number in [90, 100)
```