

Generators

Consider the following list of numbers

Please see `synth\start.ipynb` for practical use generating an sound wave

```
In [ ]: a = range(5)
        list(a)
```

```
Out[ ]: [0, 1, 2, 3, 4]
```

A list is good, but what about an infinite list? With a generator you can do that. Yield will return a value from the function, but unlike return the state of the function is mainted for the next time it is called. This is also lazy execution, it will only generate the new value once needed. It will only occupy enough memory for 1 number. So its very effician in time and memory.

```
In [ ]: def infinite_sequence():
        num = 0
        while True:
            yield num
            num += 1

        gen = infinite_sequence()
        print(next(gen))
        print(next(gen))
        print(next(gen))

        #if you run the example below get ready to stop it with the interput button!
        # for i in infinite_sequence():
        # print(i, " ")
```

```
0
1
2
```