Lesson Eleven

Math 6080 (for the Masters Teaching Program), Summer 2020

The Sieve of Eratosthenes. We start with a list of the integers from 0 to 999. (You can prompt the user to make this any list from 0 to n.)

```
Sieve = []
for i in range(1000):
    Sieve = Sieve + [i]
```

Our goal is to put zeroes in this list wherever there is a non-prime.

```
Sieve[1] = 0
```

The first non-zero element we find is 2, which we use to turn all multiples of 2 (other than 2 itself) into zeroes. The next non-zero element after that is 3, which we use to turn all multiples of 3 (other than 3 itself) into zeroes. The next non-zero element after that is 5, etc.

```
d = 2
while d**2 <= n:
    if d == 0: continue
    else:
        i = 2
        while d∗i< n:
            Sieve[d*i] = 0
            i = i + 1
    d = d + 1
print(Sieve)
```

Try it out!

Exercise. Convert this, by use of the .pop command, into a list of the primes less than 1000 (removing all the zeroes). This is, I claim, a very efficient way to conjure up lists of primes. We will use the Sieve with the zeroes intact in our first extended project in Lesson Twelve.