

## Lecture 3: Intended learning outcomes

By the end of this lecture, you will be able to:

- refactor Python code using for loops;
- improve the performance of code;
- write instructions on how to produce Pascal's triangle.

### 3 More program writing

#### 3.1 Refactoring

Last time we were trying to find the factors of all numbers up to 50,000 using trial division and ended up with the following code.

```

1 | n = 1
2 | while n <= 50000:
3 |     print("n =", n)
4 |     i = 1
5 |     while i <= n:
6 |         if n % i == 0:
7 |             print(i)
8 |             i = i + 1
9 |     n = n + 1

```

We have subsequently learnt about the **for** loop, which will achieve the same as our **while** loops in this case. We can *refactor* the code, that is rewrite it to make it more readable, but without changing the way it works.

#### 3.2 Improving the performance

The following code is now more readable and easier to understand. This tidying up of the code is like rewriting rough work.

```

1 | for n in range(1, 50001):
2 |     print("n =", n)
3 |
4 |     for i in range(1, n+1):
5 |         if n % i == 0:
6 |             print(i)

```

This code runs very slowly for big numbers. Can we do anything to speed it up and *improve the performance*? Go back and think about the method we were using.

#### 3.3 Timing a program

There are several ways that you can time how long a program takes to run. One way is to import and use the `time` module, as follows.

```

1 | import time
2 |
3 | start = time.time()
4 | # Insert your program here
5 | end = time.time()
6 |
7 | print("This program took", end - start, "seconds")

```

(You will learn more about Python modules in Week 8.)

### 3.4 Task: Producing Pascal's triangle I

Next week's homework is to write a program to draw Pascal's triangle. We will now look at what steps are needed to do that, but we won't produce the Python code.

- Task.** (a) Write out the first five lines of Pascal's triangle.  
(b) What is the 8th line of Pascal's triangle?  
(c) What is the  $n$ th line of Pascal's triangle?

### 3.5 Task: Producing Pascal's triangle II

**Task.** Produce detailed written instructions on how to draw the first  $n$  lines of Pascal's triangle, for  $n \in \mathbb{N}$ .

Things you should assume about the reader:

- they know how to calculate  $\binom{n}{i}$ ;
- they will follow your instructions literally (they won't try to guess what you *really* mean);
- they have not seen Pascal's triangle before;
- when writing they start at the left of the page;
- they can move down to the beginning of the next line but never up;
- on each line they can write something, or leave gaps of a given number of spaces, or move to the next column;
- they are writing on paper that extends infinitely to the right.

[The reader will behave like a computer.]

### 3.6 Task: Producing Pascal's triangle III

After a short break...

**Task.** Take a fresh piece of paper and *precisely* follow the instructions you have been given, using  $n = 6$ . You should do no more and no less than is written in the instructions.