

Practice 1.1

The following questions provide practice with the concepts from part 1 of PH-204.

Functions and multiple returns

Write a function which returns both the first, second, third, and fourth power of a given number i.e.

$$f(x) = (x, x^2, x^3, x^4).$$

Calculate this for $x = 2$ and store the result in a, b, c, d.

Root finding

Use an appropriate method (`newton` or `brentq` perhaps) to find solutions to the following equations:

- $e^x = x^4$
- $x^4 - 3x^3 + 2x^2 - x + 1 = 0$
- $\sin(x) = \sqrt{x}$

Flow control

`break` and `continue` can be useful; read about them in Material 1 or in the Official Python3 Tutorial.

Write a function which returns a list of all odd integers from 0 to N inclusive with square less than 500.

See "flow control extras" in Material 1

A perfect number is equal to the sum of its positive divisors e.g. 6 divides into 1,2,3 and is therefore perfect.

Use a `for` loop to test each number from 0 to 10000 to see whether it is perfect. Exit the loop once you've found the first 4 perfect numbers. Wrap this up in a function if you like.

Lists and methods

A "method" is a function which is attached to and operates on an instance of a Python object, such as a list.

self reference

Make a list and use the `append` method to make the list contain itself. Try indexing it. Does anything break? Could this ever be a useful thing to do? Can you do this with a tuple?

sorting

Make a list of numbers. Sort it using the `sort` method.

What is different between the `sort` method and the in-built `sorted` function?

Make a list of people's names. Sort it by length using the optional argument `key` to method `sort` and the function `len` to measure the length of a string.

Planck spectrum

Thermal radiation from a blackbody is described by the Planck spectrum:

$$B(\nu, T) = \frac{2h\nu^3}{c^2} \frac{1}{e^{(h\nu)/(k_B T)} - 1}.$$

Physical constants are available in `scipy.constants`.

Write a function `B(nu, T)` which captures the above. Use as few brackets as possible.

Write a function `nu_max(T)` which returns the frequency at which spectral density is maximum, for a given temperature. **Hint:** Use `minimize` from `scipy.optimize`; minimizing $-B(\nu, T)$ maximizes $B(\nu, T)$.

Functions

Write `factorial` and `fibonacci` functions which do **not** use recursion.

Classic problems

sorting (medium difficulty)

Write a function which accepts a list and returns a sorted copy of that list. In practice, we'd use the in-built function `sorted`, but writing your own sort is instructive.

Algorithms include:

- insertion sort
- bubble sort
- quick sort (this one is usually implemented with recursion)

path finding (difficult)

Given a list of tuples specifying locations in space, find the shortest path which joins all points.

Distance between points is calculated by Pythagoras.

e.g. $[(2,4), (5,6), (-2,5), (-3,1)]$ specifies points at $(x = 2, y = 4)$, $(x = 5, y = 6)$ etc.

The shortest path in this example is $[4,3,1,2]$.

sudoku (extra difficult)

Consider the classic Sudoku puzzle.

Write a function which accepts a list of 27 integers (each of which is 0 to 9 with 0 representing an empty square) and returns the completed Sudoku. *This is a classic problem and you'll find plenty of code online, some of it brilliant and some just awful. Try to write your own.*