

# Selftest

## Not so anonymous functions

An anonymous function does not have a name, but we can give one to it. For example, we can say:

```
square = lambda x: x*x
```

to define a function square instead of the more traditional (and preferable)

```
def square2(x):  
    return x*x
```

Use this to define functions cube and bisquare.

## Rewriting with keyword arguments

The following code calculates the integral  $\int_a^b f(x)dx$  as the average value of  $f$  in the interval

from  $a$  to  $b$  estimated by taking 100 random points in  $[a, b]$  multiplied by the length  $(b - a)$  of the interval. The programmer choose to give descriptive names to the variables. Change the function so that it forces the user to use named arguments.

```
def integral(function, lower, upper):  
    suma = 0  
    for _ in range(100):  
        suma += function(random.uniform(lower, upper))  
    return suma/100 * (upper-lower)
```

Introduce (after the first line) an explanation what the function does using triple quotation marks `"""`. Then see what `help(integral)` does.