

Algorithms - a simple introduction in Python: Part One

"The notion of an algorithm is basic to all of computer programming, so we should begin with a careful analysis of this concept."

Donald Knuth *The Art of Computer Programming*

An algorithm is simply a way of doing something. When you follow a recipe or use instructions to build flat-pack furniture, you are using an algorithm.

When talking about computer programs, we need algorithms that will do something for us efficiently. By this we mainly mean that they must:

- give the correct output
- be fast
- use as little memory as possible

Let's start! Here is a program that will square a number.

```
# square.py - simple squaring program

def square(x):
    """Takes a number and returns its square."""
    return x*x

# main
print(square(int(input("Number to square: "))))
```

TASK: When you have typed this in and checked that it works, write a similar program that gives the cube of a number.

Here's a program that will raise a number to the power of any whole number.

```
# power.py - demonstrates raising numbers to integer powers
# by successive squaring.

def power(x, y):
    """Takes a real number x and an integer y and returns x^y."""
    z = 1
    while y != 0:
        if y % 2 == 0:
            x *= x
            y //= 2
        else:
            z *= x
            y -= 1
    return z

# main
print(" *** exponentiation by successive squaring ***")
print("This program will raise x to the power of y")
print("as long as y is a whole number!")
a = float(input("x: "))
b = int(input("y: "))
print("Result: ", power(a, b))
```

The program works because: $x^{2y} = (x^y)^2$ and $x^y = x \cdot x^{y-1}$.
You might want to check this yourself and then confirm that the program gives the correct output.