Algorithms

A way too brief story of mathematics, algorithms and all the things I love

Algorithms and computation predate computers by a few thousand years and a few hundred years respectively.
The Egyptians used precise calculations to determine which crop field belonged to which farmer after the seasonal floods of the Nile, they also had a method to calculate fractional numbers, even achieving a pretty accurate approximation to π, that's $\frac{256}{81}$ or 3,1604938272 if you want to be exact - not bad for a 5000 year old culture. The Greeks used algorithms to track the movements of planets and to calculate the greatest common divisor, like Euclid's greatest common divisor calculation method.
An algorithm is therefore a process, strictly defined by a series of steps to achieve a certain goal, mathematics is the common language here, the tool that defines algorithms, which in turn are the basis for how computers work and the object of a whole field of mathematics - computation. A computer I might add for the sake of this historical note was a profession in the XX century, mostly done by women during the WWII effort. Much like today, a computer's job is to calculate the trajectory tables for ballistic missiles, a very necessary work at the time. Computation was at the basis of our victory over nazi Germany, by the code breaking genius of Alen Turing.
Compute however, even years after the war, remained as just another clerical work, similar to the work done by women in the old connection boards of old timey telephones.
As the importance of computers grew, the number of women in the field started to reduce and began to be taken over by a majority of men. Even though one of the pioneers in the construction of truly production-ready code (like the one that took us to space for example) has the groundbreaking brainpower of women behind it, like Margaret Hamilton. As the decades advanced, the figure of the programmer begun to take a more masculine turn. Programmers started to be seen as lonely, geeky-type people that lacked the social skills to interact with anything but a computer.

Moving forward to the XXI century, as things get more balanced, it's still admirable how far behind we are from mathematics. All the technological advancements of our time are but a discipline of mathematics, in a subset of computation. The computable algorithms - and that's a spec of sand in the vast sandbank of computation.

Context and functions

A computer program is a set of functions that depend on each other in a defined order, each requires it's own set of starting conditions.

The context of a program refers to *where you are*, whether it's inside the main function, assign a value to a variable, or you're using a variable that was defined somewhere else in your code. *Where you are* tells you if a certain variable is usable, or if a certain function can be called. There are two different contexts in any given program, the context of a function and the context of the program.

Take the following piece of code:

```javascript

```

```
// Example 1
var world = "world";

function printer(world) {
  console.log("Hello " + world);
}

function anotherPrinter(world) {
  var world = "dog";

  console.log("Hello " + world);
}

function yetAnotherPrinter(text) {
  var world = "cat";

  console.log("Hello " + world);
}
```

```

This example shows a program that has a global context. It has a variable `world` that can be used by any function inside the example - but there's a catch - both functions, `anotherPrinter()` and `yetAnotherPrinter()` have a variable `world` created.
Isn't this wrong? - Yes and no.
The `anotherPrinter()` function gets the global variable `world` as an argument that has the value `world`, which is then overridden by the value `dog` and at last prints `Hello dog`. The
```

variable `world` is overridden in the context of that function.
The `yetAnotherPrinter()` is a different case, it has a local variable called `world` that doesn't override any variable name - the argument variable it receives is called `text`. No problem there.

The bottom two functions of example are wrong, not because the program doesn't run but because they are creating a mess. We get 3 different variables called `world`, in different contexts and with different values. This is a very simple example and it's quite easy to see what every result will be. Given the complexity of actual programs, it could create real problems as you could expect a certain value for `world` that just isn't correct.