# [Functional Programming](https://www.youtube.com/watch?v=LnX3B9oaKzw&list=WL&index=12&t=16s&ab_channel=Computerphile)

Essentially programming where you **don’t have side effects** and the **inputs are not modified** by the function. The benefits of functional programming is that it avoids introducing bugs.

Another way of saying it: Functional languages try to isolate the transformations of data from the definition of the data.

# Pure Functions

Producing a program where your functions:

1. Avoids side effects (eg. performing a function won’t modify a global variable)
2. Given an input, you will always get the same output. It does not modify the inputs. (eg. If the parameters/inputs are x and y, then z will always be returned)

Example of a pure function, note that after the function is executed, the inputs are not modified.

**Function addElementToArray (element, array)**

**Return […array].push(element)**

# Easy Way to Convert Digits to Binary in Your Head

If you want to know 13 (which is a digit) in binary, just add up the hexadecimals (second row of image). Notice that 8 + 4 + 1 = 13.

