Guidelines for Programming with Functions

Some basic guidelines which every programmer must practice while writing a code with functions.

WE'LL COVER THE FOLLOWING

Creating Functions Wisely

Functions can include everything you can use in a regular program: variables, conditionals, loops, etc. Functions can call one another, giving the programmer an enormous amount of freedom for building programs. However, not everything deserves to be in its own function. It's better to write short and focused ones, in order to limit dependencies and improve program understanding.

Leveraging JavaScript Predefined Functions

We have already used several predefined JavaScript functions like `prompt()` and `alert()`. There are many others in the language specification. Get to know them instead of reinventing the wheel! Here's an example demonstrating two of the JavaScript mathematical functions.

```
console.log(Math.min(4.5, 5)); // 4.5
console.log(Math.min(19, 9));  // 9
console.log(Math.min(1, 1));   // 1
console.log(Math.random());    // A random number between 0 and 1
```

The function `Math.min()` returns the minimum number among its arguments. The function `Math.random()` generates a random number between 0 and 1. This course will introduce many other JavaScript functions.

Limiting Function Complexity

A function body must be kept simple, or otherwise split into several sub-functions. As a rule of thumb, 30 lines of code should be a max for non-specific cases.

Naming Functions and Parameters Well

Function naming is just as important as variable naming. You should choose names that express clearly the function purpose and follow a naming convention. Refer to the appendix for some useful advice.

> If you have difficulties coming up with a good name for a function, then maybe its purpose is not that clear and you should ask yourself if this function deserves to exist.