The principles of DRY

Ever find yourself writing the same lines of code over, and over, and over again? Well.. **DRY** your eyes, we've all been there**. DRY** is something to keep at the forefront of your mind when writing code, rather than a set of instructions, so take a moment to reflect and even go back to your old projects - could things be simpler?

What is DRY?

Aside from the Atacama Desert, and my humour writing this.. **DRY** stands for *"Don't Repeat Yourself"* and is one of the many software development principles you should keep in mind, especially in an object-orientated world. **DRY**, in principle, is the reduction of the amount of duplicated code we write which in turn lowers said code count, makes refactoring easier, and enables simpler maintenance.

The principle itself comes thanks to Andy Hunt and Dave Thomas in their book "The Pragmatic Programmer" (A strongly recommended read for those wishing to expand their book collection). In their words: *"Every piece of knowledge must have a single, unambiguous, authoritative representation within a system"*.

In simpler terms we should write code that truly works for us; it should be thoughtful, distinctive, and reusable.

Let's take a closer look:

Let's presume we have an array containing fruits and we need to log each output on an individual line.

```
let myFruityArray = ["apple", "banana", "pear"];

console.log(myFruityArray[0]);
console.log(myFruityArray[1]);
console.log(myFruityArray[2]);
```

Imagine how repetitive this would be if we had, lets say, 50 items? Even simple tasks have a non-repetitive, and more elegant, solution.

```
let myFruityArray = ["apple", "banana", "pear"];

for (let i = 0; i < myFruityArray.length; i++){
    console.log(myFruityArray[i]);
}
```

What about multi-dimensional arrays?

```
let myFruityArray = [
    ["apple", "banana", "pear"],
    ["grape", "strawberry", "kiwi"],
    ["orange", "blueberry", "raspberry"]
];

for(let i = 0; i < myFruityArray.length; i++) {
   for(let j = 0; j < myFruityArray[i].length; j++) {
      console.log(myFruityArray[i][j]);
   }
}
```

What if we need to do this more than once? Make it a function! Now that’s **DRY**.

```
// array declaration
let myFruityArray = [
    ["apple", "banana", "pear"],
    ["grape", "strawberry", "kiwi"],
    ["orange", "blueberry", "raspberry"]
];

// function declaration
function myFunction(myArray){
    for(let i = 0; i < myArray.length; i++) {
        for(let j = 0; j < myArray[i].length; j++) {

            console.log(myArray[i][j]);
        }
    }
}

// function call
myFunction(myFruityArray);
```

In conclusion

We’ve all been there, we’ve all looked back at old projects with an unrelenting urge to scream “WHY” to the heavens.. And eventually we’ve all picked up someone elses project which has fundamentally changed our views of them forever (well, until the next coffee).

Code is beautiful, and should be beautifully written. If you find yourself repeating the same thing over, and over, and over again.. Get a little more **DRY**.

- **Write code or logic once**
    - Writing, or copy and pasting, the same code/logic over and over again is a big no-no. Make use of functions and smarter logic.

- **Use thoughtful naming conventions and create clear linkages**
    - Descriptive variables/functions/methods etc make their contents clear and easier to reuse in the future
    - Make use of code comments, they don’t always have to be exhaustively extensive

- **Divide code into small reusable chunks which can be called upon when needed**
    - This way changes can be made to code in one place
    - Less is more: save yourself time and effort

Your fingers, your brain, and your fellow developers will thank you.