**I. HÀM LÀ GÌ? (FUNCTION)**

**1. Giới thiệu**

* Trước khi đến với **Function**, bạn hãy viết 1 chương trình:
  + In ra màn hình đoạn text sau: ***"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua."***
  + In ra màn hình câu **"Hello"**.
  + Tiếp tục in ra màn hình đoạn text ***"Lorem ipsum..."*** một lần nữa.
  + In ra màn hình câu **"How are you today?"**.
  + Tiếp tục in ra màn hình đoạn text ***"Lorem ipsum..."*** một lần nữa.
  + In ra màn hình câu **"Have a good day!"**.
* Sau khi làm xong, bạn sẽ nhận thấy có 1 vấn đề trong code của bạn, đó là cùng một đoạn code in ra màn hình đoạn text ***"Lorem ipsum..."***, bị lặp lại đến 3 lần.

→ Liệu có cách nào chỉ cần viết đoạn code in ra màn hình đoạn text ***"Lorem ipsum..."*** một lần, nhưng lại có thể sử dụng lại được nhiều lần không? Đây là lúc cần tới **Function**.

**2. Khái niệm**

* **Function** là một phương pháp lập trình nhằm giảm tải công sức của người lập trình: **Viết code 1 lần, sử dụng mãi mãi**
* Không chỉ vậy, khi sử dụng **function**, code cũng trở nên sáng sủa và dễ đọc, dễ hiểu hơn.
* Ý tưởng ở đây sẽ là thực hiện viết đoạn code bị lặp lại 1 lần, sau đó gói lại vào một cái hộp (**function**), đặt tên cho cái hộp. Mỗi khi cần dùng tới đoạn code chỉ cần lấy cái hộp đó mở ra dùng, không cần phải viết lại code 1 lần nữa.

**3. Cú pháp**

function tenFunction() {

// Your code

}

* **function**: **keyword** bắt buộc cần phải có khi khai báo một **Function**.
* **tenFunction**: tên của function, quy tắc đặt tên giống với quy tắc đặt tên biến.
* Bên trong sẽ là đoạn code mà bạn muốn sử dụng mỗi khi cần.
* Cú pháp bên trên chỉ là bạn khai báo ra một **function** có tên là **tenFunction**, chứ chưa hề sử dụng nó. Cũng giống như việc bạn biết trên thế giới này có một người tên là **Quân**, nhưng nếu bạn không gọi **Hey, Quân** thì người đó sẽ không biết bạn có nhu cầu cần nhờ đến **Quân**.

→ Vậy cũng giống như gọi tên người, bạn chỉ cần gọi tên **function** đã được khai báo để sử dụng nó. Đằng sau tên có kèm theo cặp ngoặc tròn. Cú pháp:

tenFunction();

* Khi đó, phần code bên trong **function tenFunction** mới được thực hiện.

**4. Ví dụ**

* Để giải quyết bài toán ở đầu chương, bạn khai báo một **function** như sau:
* function renderLorem() {
* console.log("Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.");

}

* Ở ví dụ trên, một **function** được khai báo ra với tên **renderLorem**, có nhiệm vụ thực hiện in ra màn hình đoạn text ***"Lorem ipsum..."*** mỗi khi được gọi.
* Sau khi đã có **function renderLorem**, thực hiện gọi ở những vị trí tương ứng như yêu cầu của đề bài:
* renderLorem();
* console.log("Hello");
* renderLorem();
* console.log("How are you today?");
* renderLorem();

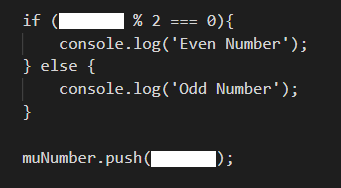
console.log("Have a good day!");

***Bài tập***: Viết một function có nhiệm vụ in ra màn hình đoạn text **"Hello World!"** 3 lần.

**II. THAM SỐ CỦA HÀM (FUNCTION PARAMETERS)**

**1. Giới thiệu**

* Bạn hãy viết 1 chương trình:
  + Tạo ra một array myNumber không có phần tử nào bên trong.
  + Cho người dùng nhập vào một số bất kỳ, sau đó kiểm tra xem số đó là số chẵn hay số lẻ, thông báo kết quả ra màn hình. Sau khi thông báo, thêm số mà người dùng đã nhập vào một array tên là myNumber được tạo ra từ trước.
  + Tiếp tục cho người dùng nhập một số khác → kiểm tra số đó là chẵn hay lẻ → thông báo kết quả ra màn hình → thêm vào myNumber.
  + Lặp lại quá trình trên 1 lần nữa (Nhập → Kiểm tra → Thông báo → Thêm vào array).
* Nếu không sử dụng function, bạn sẽ gặp phải vấn đề giống như bài toán ở đầu chương, một đoạn code bị lặp lại nhiều lần.
* Nếu sử dụng function, thì function được tạo ra sẽ không biết phải nhét gì vào những ô màu trắng như hình dưới:

[](https://github.com/edtechkidsvn/c4ejs-student-book/blob/master/images/function/function_img_1.png)

→ Cần phải cung cấp cho **function** một **biến** để nó có thể hoàn thành được nhiệm vụ của mình. **Biến** cung cấp cho **function**, được gọi là **tham số (parameter)**. Cú pháp:

function tenFunction(parameter) {

// Use parameter to complete your code

}

* **parameter** của một function sẽ được đặt trong cặp ngoặc tròn, bên cạnh tên function.

**2. Ví dụ, cách sử dụng**

* Đi ngay vào giải quyết bài toán kiểm tra chẵn lẻ bên trên sẽ giúp bạn hiểu rõ nhất về cách sử dụng **parameter**.
* Khai báo một **function** với tên **checkNumber**, có nhiệm vụ kiểm tra 1 số là chẵn hay lẻ, thông báo cho người dùng và thêm vào array myNumber:
* function checkNumber(numberparameter) {
* if (numberparameter % 2 === 0){
* console.log('Even Number');
* } else {
* console.log('Odd Number');
* }
* myNumber.push(numberparameter);

}

* Việc tiếp theo bạn cần làm không dừng lại ở việc chỉ cần gọi **checkNumber()** nữa, mà bạn sẽ cần phải cho **function checkNumber** biết nó sẽ phải kiểm tra số nào. Cú pháp:
* let myNumber = [];
* let number1 = prompt("Input your number: ");
* checkNumber(number1);
* let number2 = prompt("Input your number: ");
* checkNumber(number2);
* let number3 = prompt("Input your number: ");

checkNumber(number3);

* Đoạn code trên được giải thích thông qua câu nói vui sau:  
  "Hey **checkNumber**, mày hãy nhận lấy số **number1** mà người dùng vừa nhập vào, **coi nó là numberparameter** của mày, sau đó thực hiện công việc bên trong nhé."

***Lưu ý:*** Một **function** có thể khai báo một hoặc nhiều parameter, ngăn cách nhau bởi dấu phẩy (,). Khai báo ra bao nhiêu parameter, sử dụng bấy nhiêu.

***Bài tập:*** Viết một function có chức năng tính tổng 2 số bất kỳ do người dùng nhập vào và in kết quả ra màn hình.