### I. BIẾN LÀ GÌ? (VARIABLES)

**1. Ví dụ thực tế**

* Đức là một nhân viên phục vụ bàn trong một nhà hàng tên là Pectopah, mức lương cố định của Đức là 100k/giờ.
* Ngoài mức lương cố định trên, mỗi giờ Đức có thể được nhận thêm tiền boa (**tips**) từ khách hàng.
* Số tiền mà Đức nhận được mỗi giờ có thể được biểu diễn bởi biểu thức sau:

100+tips

* Số tiền boa (**tips**) mỗi giờ có thể ít, có thể nhiều hoặc không có tùy thuộc vào nhiều yếu tố như: giờ đó đông hay vắng khách, thái độ của khách đang vui hay buồn,...Nói tóm lại, **tips có thể thay đổi (vary)**.
* Vậy tổng số tiền mà Đức nhận được mỗi giờ phụ thuộc vào số tiền tips mà Đức nhận được trong giờ đó.

Trong ví dụ trên, **tips** được gọi là **biến** (biến trong biến đổi), vì nó không có giá trị cố định; giá trị của nó thay đổi theo những yếu tố khác. Trái lại, số tiền 100k được gọi là hằng số vì giá trị của nó không thay đổi.

Một số ví dụ khác về **biến** trong thực tế: thái độ của người yêu bạn ngày hôm nay (phụ thuộc vào độ quan tâm của bạn cho người ta trong vài ngày vừa rồi), hay số lượng hoa đào nở Tết này (phụ thuộc vào thời tiết ấm hay lạnh)

**2. Khái niệm**

* Trong máy tính, biến là một vùng không gian trống trong RAM, dùng để lưu trữ giá trị (value). Nói cách khác, trong lập trình có thể coi biến như một cái hộp, bên trong nó chứa một value nào đó.
* Giá trị (value) nằm trong hộp không cố định mà có thể thay đổi tuỳ theo những yếu tố khác hoặc theo sự sắp đặt của bạn. Hôm nay trong hộp chứa giá trị a, ngày mai có thể chứa giá trị b, cũng giống như tiền tips của Đức lúc 8h-9h là 200k, nhưng lúc 9h-10 lại chỉ có 10k vậy.
* Tại một thời điểm, một biến chỉ có thể lưu trữ được một giá trị (value).
* Trong lập trình, biến là một công cụ cực kỳ quan trọng vì có những chức năng không thay thế được như sau
  + Chứa giá trị (value) có thể dùng đi dùng lại được, ở ví dụ mục 1, tips ngoài việc tính lương của Đức còn có thể được dùng để xem liệu khách hàng có hài lòng với dịch vụ của hàng không (lấy tips so sánh với 5, nếu lớn hơn được tính là hài lòng chẳng hạn)
  + Chứa giá trị có khả năng thay đổi trong tương lai, ví dụ tiền tips của Đức có thể sẽ còn 80% vào cuối ngày vì nhà hàng có quy định nộp lại 20% tips. Một ví dụ khác có thể kể đến như nhiệt độ từ đài khí tượng lúc đầu có thể là 25 độ C nhưng đến trưa có thể sẽ được cập nhật thành 30 độ.

**3. Cú pháp**

* Một biến cần phải được khai báo trước khi sử dụng.
* Cú pháp khai báo biến trong javascript như sau:

let firstVariable;

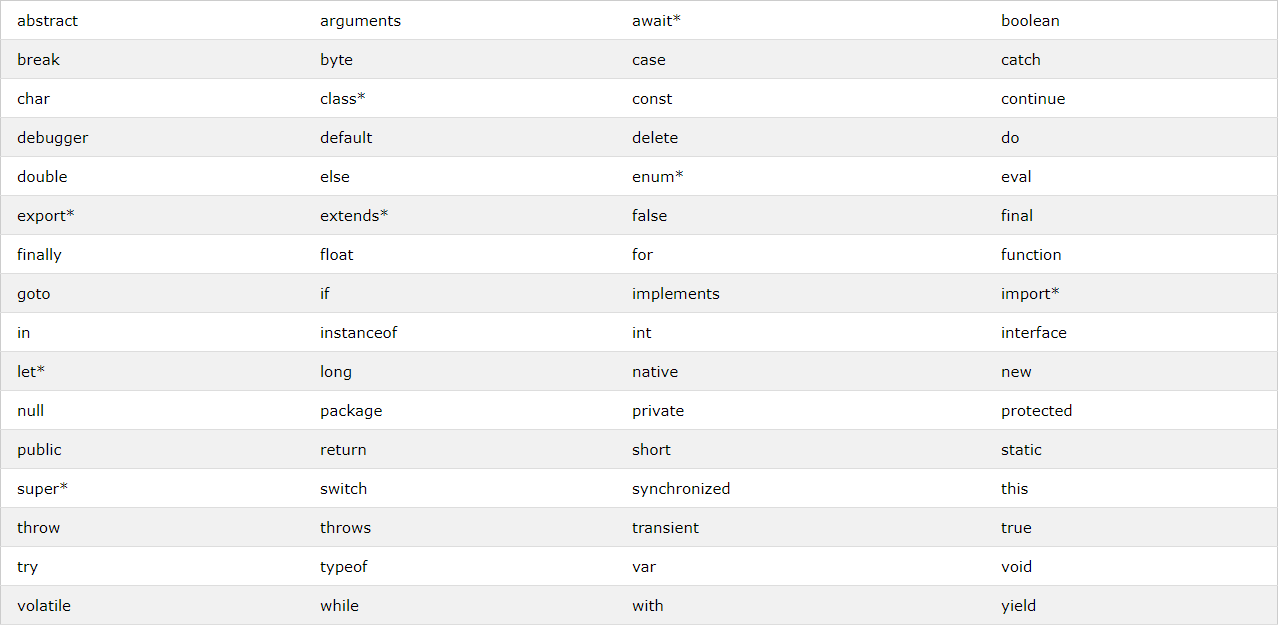
* Ở câu lệnh trên:
  + "firstVariable" là tên biến, có thể đặt tùy ý, lưu ý là cần phải tuân thủ theo các quy tắc đặt tên và nên đặt tên biến có ý nghĩa. Các quy tắc đặt tên biến sẽ được trình bày rõ hơn ở mục 4
  + "let" là từ khóa (keyword) của javascript, sử dụng để khai báo một biến. Có 3 cách để khai báo một biến trong javascript, tương ứng với 3 keywords **var**, **let** và **const**. Chúng được gọi là **declarations of variables**.
  + Trong các **declarations of variables**, chỉ có **let** là được sử dụng phổ biến hơn cả, số lượng các biến được khai báo bởi **let** chiếm tới 95% trong một dự án. Chính vì vậy trong chương này sẽ chỉ tập trung vào **let**. Ý nghĩa và cách sử dụng **let** sẽ được đề cập cụ thể ở phần II.
* Một vài ví dụ về cách khai báo biến:
* let myMovie;
* const bestSchool;

var dateOfBirth;

**4. Quy tắc đặt tên**

Một đoạn code viết ra với những tên biến mang ý nghĩa rõ ràng, sẽ khiến cho việc đọc code và sửa lỗi trở nên dễ dàng hơn. Ngoài việc cần phải mang ý nghĩa rõ ràng, tên biến trong lập trình sẽ phải tuân thủ một số quy tắc đặt tên như sau:

* Có thể bao gồm cả chữ và số.
* **Không được** bắt đầu bằng số.
* **Không được** chứa những ký tự đặc biệt (khoảng trắng cũng tính là ký tự đặc biệt), ngoại trừ "\_" và "$".
* **Không được** trùng với các từ khóa (keywords) của javascript.
* Các keyword có trong javascript:

[](https://github.com/edtechkidsvn/c4ejs-student-book/blob/master/images/variable/keywords.png)  
(Nguồn: [*https://www.w3schools.com*](https://www.w3schools.com/))

* Tên của biến là duy nhất, có thể coi tên của biến là định danh (indentifiers).
* Ví dụ:
  + Tên biến hợp lệ: dateOfBirth, temp99, $credit, \_name,...
  + Tên biến không hợp lệ: 22CThanhCong, @gmail.com, break,...

***Bài tập***

* + Sau khi chạy những đoạn code dưới đây, x bằng bao nhiêu?
    - Code 1
    - let x = 5;

let x = 10;

* + - * Code 2

let x = 5;

x = 10;

* + Những cách khai báo biến nào sau đây hợp lệ:
  + let break;
  + var Break;
  + let year-of-birth;
  + const MYFAVORITE;
  + const tech\_kids;

const 4ever;

### II. CHI TIẾT KEYWORD LET, PHẠM VI HOẠT ĐỘNG CỦA BIẾN (TYPE SYNTAX, VARIABLE SCOPE)

**1. Phạm vi hoạt động của biến (Variable Scope)**  
Trước khi đến với ý nghĩa và cách sử dụng keyword **let** để khai báo biến, chúng ta hãy đi tìm hiểu khái niệm **Scope**.

Liên hệ một chút với ví dụ ở đầu chương:

* Khi còn đang ở trong nhà hàng Pectopah, tất cả khách hàng đều có quyền yêu cầu Đức phục vụ họ, vì lúc này Đức đang đóng vai trò là nhân viên phục vụ bàn, Đức có trách nhiệm phải hoàn thành nhiệm vụ của một người phục vụ.
* Hết giờ làm việc, Đức bước ra khỏi nhà hàng, không có bất cứ một ai có quyền yêu cầu Đức phục vụ món ăn nữa, vì lúc này đã là ngoài giờ làm việc, Đức không đóng vai trò là nhân viên phục vụ bàn. (Giả sử Đức không được phép ra khỏi nhà hàng trong giờ làm việc)

Trong ví dụ trên, có thể hiểu nhà hàng Pectopah là **Scope** hay **phạm vi hoạt động** của Đức - trong vai trò nhân viên phục vụ. Ra khỏi **Scope**, Đức không còn hoạt động như một nhân viên phục vụ bàn nữa.

Khi đã hình dung ra được **Scope**, chúng ta có thể dễ dàng nắm được ý nghĩa và cách sử dụng của keywords **let** như sau:

* Keyword **let** khai báo một biến chỉ có thể truy cập được trong **Scope** bao quanh nó.

Ví dụ

{

let x = 5;

};

console.log(x); //x is not defined

Trong ví dụ trên, chương trình sẽ báo lỗi "x is not defined", vì biến x được khai báo bởi **let** nằm bên trong cặp {} (**Scope**), trong khi câu lệnh console.log(x) lại nằm bên ngoài, vì vậy nên không thể in được x ra màn hình.

***Bài tập***  
Giá trị của x, và y là bao nhiêu sau khi chạy đoạn code dưới đây?

{

let x = 30;

{

let y = 20;

}

console.log(x);

console.log(y);

}

### III. CÁC KHÁI NIỆM KHÁC

**1. Declaring variables và Evaluating variables**

* **Declaring variables**: Việc lập trình viên lần đầu tiên viết ra một biến để dùng về sau như let x = 20 ở trên được gọi là **Khai báo biến**, tiếng anh là **Declaring variables**.

Hiểu một cách đơn giản thì đây là chúng ta đang báo cho JavaScript nhớ rằng có 1 biến, tên là x, có giá trị là 20, từ giờ trở đi, khi gọi đến x, ví dụ console.log(x) hay x = 34 thì chính là x mà ta đã dặn JavaScript ghi nhớ từ trước

``` javascript

const theBestSchool = "TechKids";

let mySport = "Football";

let myPhoneNumber = "0123456789";

```

* **Evaluating variables**: Cách khai báo biến ở mà có sẵn giá trị như ở trên không phải là cách duy nhất để sinh ra một biến, trong thực tế sẽ có những lúc lập trình viên, tại thời điểm khai báo biến, không biết được giá trị đó là bao nhiêu, trong những trường hợp như vậy giá trị của biến có thể được để trống như dưới.
* const theBestSchool;
* let mySport;

let myPhoneNumber;

Những biến được khai báo nhưng không gán giá trị như trên sẽ mặc định được JavaScript để giá trị là undefined