



## **SESSION 09:**

# **VARIABLES, DATA TYPES, OPERATORS AND WINDOW-DOM METHOD**

Module 1: Web Application UI Design

Version: 1.0



- 1. Tổng quan Javascript**
- 2. Variable – Biến**
- 3. Data types – Các kiểu dữ liệu**
- 4. Các phương thức Window, DOM**
- 5. Operators – Các toán tử**

# TỔNG QUAN JAVASCRIPT

- Javascript là ngôn ngữ lập trình được nhà phát triển sử dụng để tạo trang web tương tác
- Là ngôn ngữ lập trình kịch bản phía client và server
- Công nghệ cốt lõi của World Wide Web
- Là ngôn ngữ thông dịch
- 3 cách nhúng Javascript vào HTML

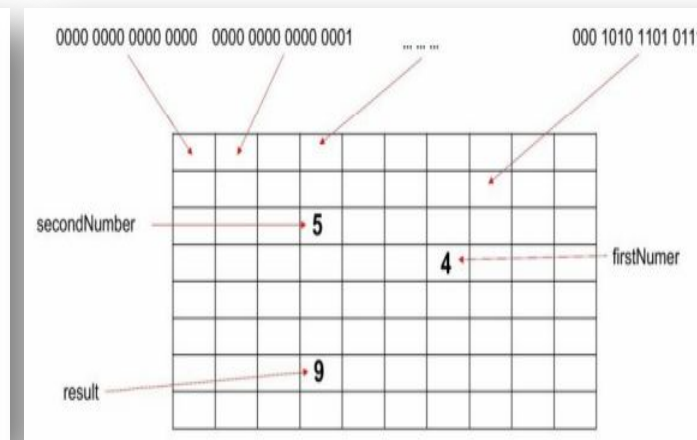
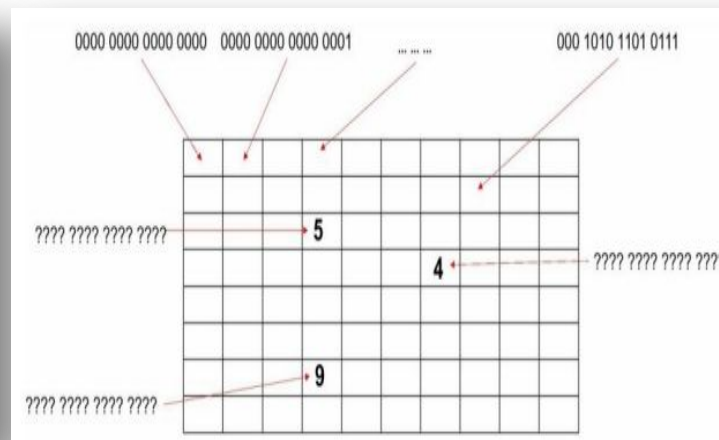
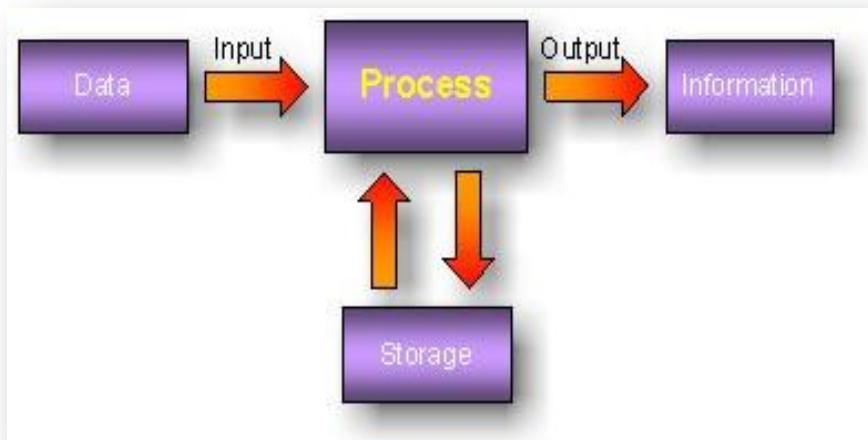
- `<button onclick="alert('Hello World')">Click here</button>`

- ```
<script>
let rikkeiAcademy = "Hệ thống đào tạo CNTT chất lượng Nhật Bản";
</script>
```

- `<script src="demo.js"></script>` ể script



# BIẾN - 1



- **Variable – Biến**

- Biến là một tên gọi được gán cho một vùng nhớ chứa dữ liệu
- Dữ liệu được lưu trữ trong vùng nhớ của biến được gọi là giá trị (value)
- Có thể truy nhập, gán hay thay đổi giá trị của biến
- Khi gán một giá trị mới thì giá trị cũ sẽ bị ghi đè lên
- Cần phải khai báo biến trước khi sử dụng

# BIẾN - 2

## • Cú pháp khai báo biến

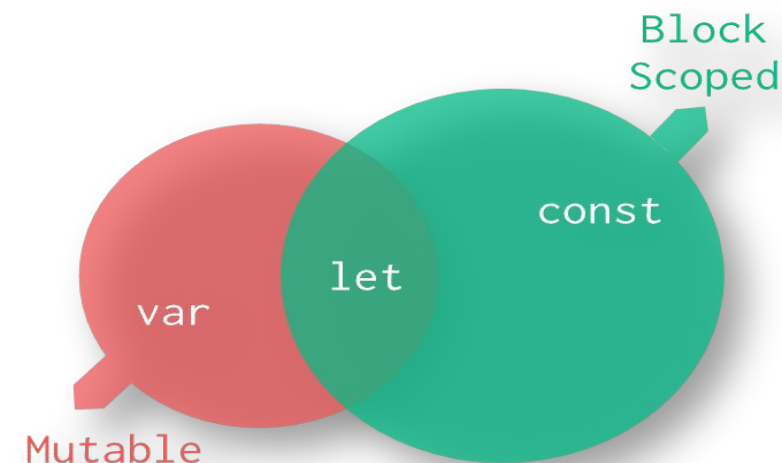
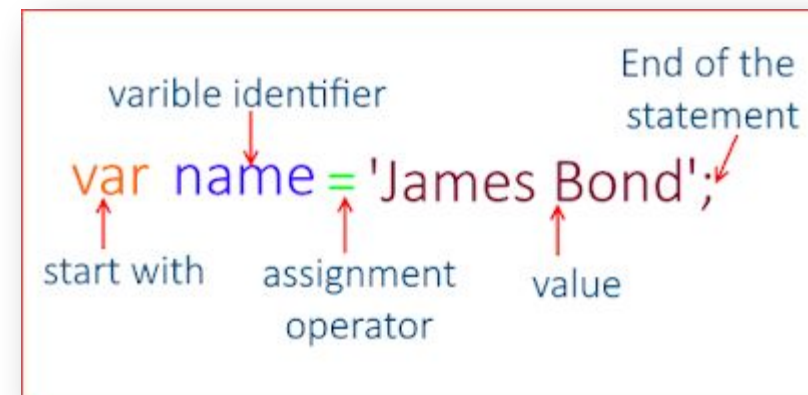
**var** | **let** | **const** [**variable\_name**] = **value**;

## • Quy tắc đặt tên biến

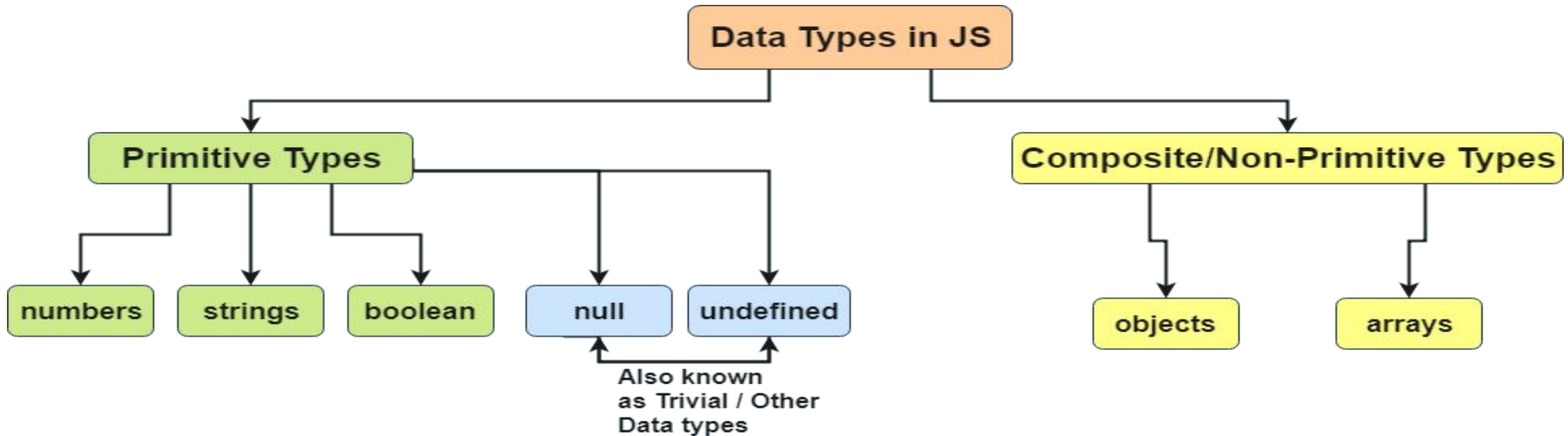
- Tên biến phải bắt đầu bằng một ký tự alphabet (a-zA-z\_), \$ và dấu gạch dưới \_
- Theo sau ký tự đầu có thể là các ký tự chữ, số ...
- Không trùng với các từ khóa của javascript
- Không chứa các ký tự đặc biệt
- Tên biến rõ ràng và có ý nghĩa
- Nên đặt theo quy tắc Camel Case

## • Từ khóa khai báo biến

- **var**:
  - khai báo biến toàn cục hoặc cục bộ, có thể khai báo lại và gán lại giá trị
  - Hoisting là khái niệm chỉ mọi khai báo biến với từ khóa var sẽ được chuyển lên trên cùng
- **let**: khai báo trong khối lệnh, có thể gán lại nhưng không thể khai báo lại
- **const**: khai báo trong khối lệnh, không thể gán lại và khai báo lại



# CÁC KIỂU DỮ LIỆU - 1



- Kiểu dữ liệu là một cách phân loại dữ liệu cho trình biên dịch hoặc thông dịch hiểu các lập trình viên muốn sử dụng dữ liệu
- Kiểu dữ liệu mô tả loại dữ liệu sẽ được lưu trong biến
- Khi khai báo biến và gán giá trị cho biến một giá trị đồng nghĩa xác định kiểu dữ liệu của biến đó
- Sử dụng `typeof` để kiểm tra kiểu dữ liệu của biến

```
<script>
  let numberOne = 10;
  typeof numberOne; //number
</script>
```

# CÁC KIỂU DỮ LIỆU - 2

- **Number – Kiểu dữ liệu số**

- Number trong JavaScript dùng cho cả số nguyên (integer) và số thực (float), và luôn là số có dấu (signed)
- Ví dụ:
  - Số nguyên: 10, -6
  - Số thực: 15.8, -4,6
  - NaN (Not a Number), Infinity(số vô cực)

- **String – Kiểu dữ liệu chuỗi**

- Chuỗi bao gồm các ký tự liên tiếp nhau
- Chuỗi bao gồm các ký tự liên tiếp nhau
- Ví dụ: “Rikkei Academy”

- **Boolean**

- Kiểu dữ liệu boolean chỉ có hai giá trị là true và false

- **Undefined**

- Không xác định
- Khai báo một biến nhưng chưa gán giá trị cho nó

- **Null**

- Giá trị rỗng hoặc giá trị không tồn tại
- Có thể được sử dụng để gán cho một biến như là một đại diện không có giá trị

# CÁC PHƯƠNG THỨC WINDOW, DOM

- **Window**

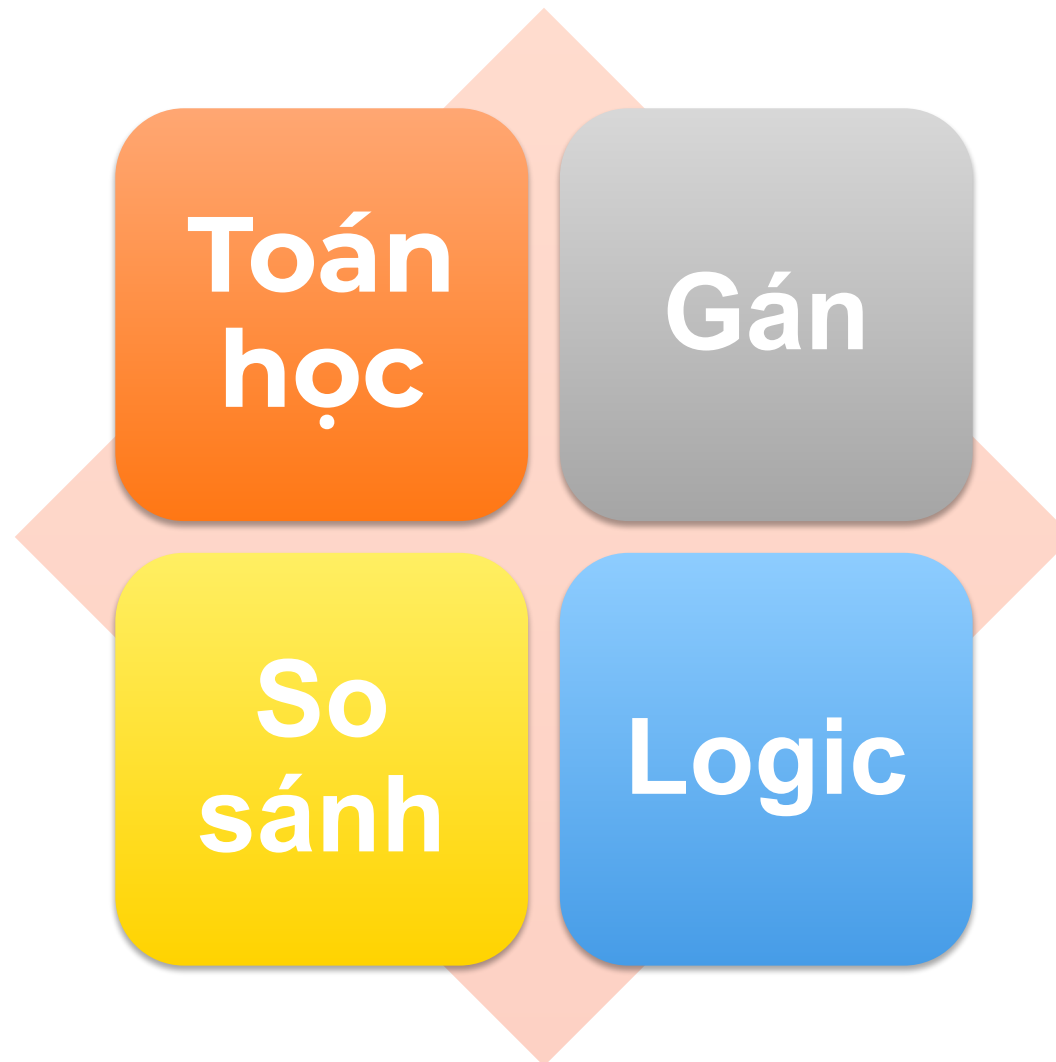
- **alert("message")**
  - Hiển thị hộp thông báo có nội dung message và nút OK
  - Sử dụng khi muốn thông tin đến người dùng
- **confirm("message")**
  - Hiển thị hộp thông báo có nội dung message và nút OK, Cancel
  - Trả về **true** nếu chọn OK, ngược lại là **false**
- **prompt("text","defaultText")**
  - Hiển thị hộp thông báo có nội dung text và cho người dùng nhập dữ liệu
  - Trả về giá trị người dùng nhập nếu chọn OK, ngược lại null
- **Console.log("message")**
  - Ghi một thông báo tới console

- **DOM**

- **document.write(exp1, exp2,..., expN):** In ra thông tin các đối số exp1,...,expN



# CÁC TOÁN TỬ - 1



# CÁC TOÁN TỬ - TOÁN HỌC - 2

| Toán tử | Mô tả            | Ví dụ     | Kết quả |
|---------|------------------|-----------|---------|
| +       | Cộng             | $10 + 5$  | 15      |
| -       | Trừ              | $10 - 5$  | 5       |
| *       | Nhân             | $10 * 5$  | 50      |
| /       | Chia             | $10 / 5$  | 2       |
| %       | Chia lấy phần dư | $10 \% 5$ | 0       |
| ++      | Tăng 1 giá trị   | $10 ++$   | 11      |
| --      | Giảm 1 giá trị   | $10 --$   | 9       |

# CÁC TOÁN TỬ - GÁN - 3

| Toán tử | Ví dụ   | Tương đương |
|---------|---------|-------------|
| =       | $x=y$   | $x=y$       |
| +=      | $x+=y$  | $x=x+y$     |
| -=      | $x-=y$  | $x=x-y$     |
| *=      | $x*=y$  | $x=x*y$     |
| /=      | $x/=y$  | $x=x/y$     |
| %=      | $x\%=y$ | $x=x\%y$    |

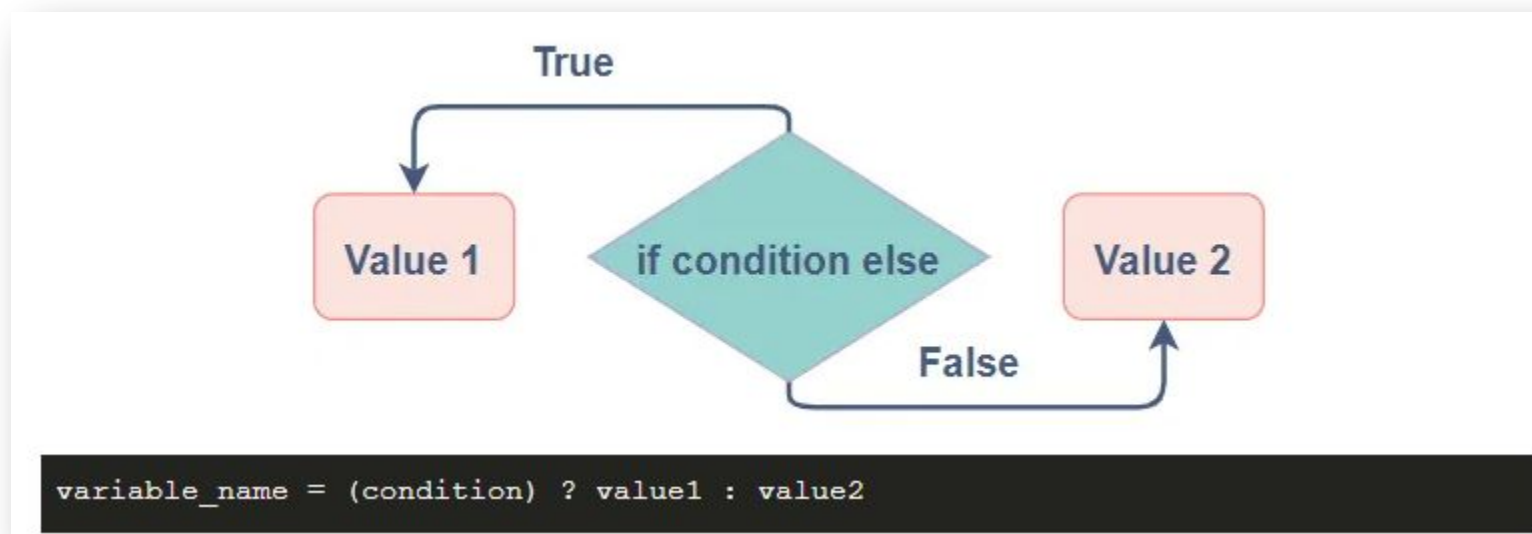
# CÁC TOÁN TỬ - SO SÁNH - 4

| Toán tử      | Mô tả                                          | Ví dụ     | Kết quả |
|--------------|------------------------------------------------|-----------|---------|
| <b>==</b>    | So sánh bằng (so sánh giá trị)                 | 10=="10"  | True    |
| <b>===</b>   | So sánh bằng (so sánh giá trị và kiểu dữ liệu) | 10==="10" | False   |
| <b>!=</b>    | So sánh khác                                   | 10!=10    | False   |
| <b>!==</b>   | Không cùng kiểu dữ liệu hoặc bằng giá trị      | 10!==10   | False   |
| <b>&gt;</b>  | Lớn hơn                                        | 10>10     | False   |
| <b>&lt;</b>  | Nhỏ hơn                                        | 8<10      | True    |
| <b>&gt;=</b> | Lớn hơn hoặc bằng                              | 10>=10    | True    |
| <b>&lt;=</b> | Nhỏ hơn hoặc bằng                              | 10<=10    | True    |

# CÁC TOÁN TỬ - LOGIC - 5

| Toán tử           | Mô tả          | Ví dụ         | Kết quả |
|-------------------|----------------|---------------|---------|
| <b>&amp;&amp;</b> | Logic và       | true && false | false   |
| <b>  </b>         | Logic hoặc     | true    false | true    |
| <b>!</b>          | Logic phủ định | !true         | false   |

- **Toán tử 3 ngôi**



# CÁC TOÁN TỬ - ĐỘ ƯU TIÊN - 6

- Trong một biểu thức có nhiều phép toán thì chúng sẽ lần lượt được đánh giá dựa vào độ ưu tiên
- Có thể sử dụng dấu ngoặc “()” để thay đổi độ ưu tiên của các toán tử
- Các toán tử có cùng độ ưu tiên thì sẽ thực hiện từ trái sang phải

| Operators            | Precedence                             |
|----------------------|----------------------------------------|
| postfix              | expr++ expr--                          |
| unary                | ++expr --expr +expr -expr ~ !          |
| multiplicative       | * / %                                  |
| additive             | + -                                    |
| shift                | << >> >>>                              |
| relational           | < > <= >= instanceof                   |
| equality             | == !=                                  |
| bitwise AND          | &                                      |
| bitwise exclusive OR | ^                                      |
| bitwise inclusive OR |                                        |
| logical AND          | &&                                     |
| logical OR           |                                        |
| ternary              | ? :                                    |
| assignment           | = += -= *= /= %= &= ^=  = <<= >>= >>>= |



# KẾT THÚC

HỌC VIỆN ĐÀO TẠO LẬP TRÌNH CHẤT LƯỢNG NHẬT BẢN