**!!!:** Bài này đã bỏqua phần truthy và falsy.

**Toán tử số học:**

**Lũy thừa:** a\*\*b -> ab

**Toán tử +:** chỉ được xem là cộng giữa 2 số với nhau khi cả 2 đều là số, còn lại sẽ được xem là cộng chuỗi.

**Các giá trị được xem là điều kiện sai:**

+ 0

+ false

+ ‘’ “”

+ undefined

+ NaN

+ null

Nhấn mạnh lại là chỉ có 6 giá trị ở trên, các giá trị như mảng rỗng, object rỗng đều là true.

**Kiểu dữ liệu nguyên thủy:** primitive data

+ Number

+ String

+ Boolean

+ Undefined -> Khai báo ra 1 biến nhưng không gán giá trị cho nó

+ Null // var name = null;

+ Symbol // var id = Symbol(‘nội dung’)

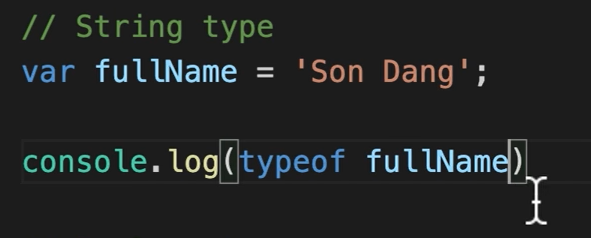
Đối với kiểu dữ liệu nguyên thủy khi gán lại giá trị cho biến thì ô nhớ mới được tạo ra để lưu giá trị mới, chứ không ghi đè lên ô nhớ của giá trị cũ.

**Dữ liệu phức tạp:** complex data

+ Function

+ Object

**Kiểm tra kiểu dữ liệu:** typeof



Null có kiểu null, nhưng khi typeof null thì sẽ ra object

Kết quả trả về của typeof có kiểu string.

**Truthy vs Falsy được trình bày ở đây không chi tiết, xem lại bài viết trên F8 sẽ rõ hơn.**

Document.all là một ngoại lệ chính thức duy nhất theo đặc tả ECMA (phiên bản 5). Đặc tả này mô tả toàn bộ các object khi chuyển sang boolean sẽ là true. Tuy nhiên, Document.all là một ngoại lệ.

Cụ thể như sau:

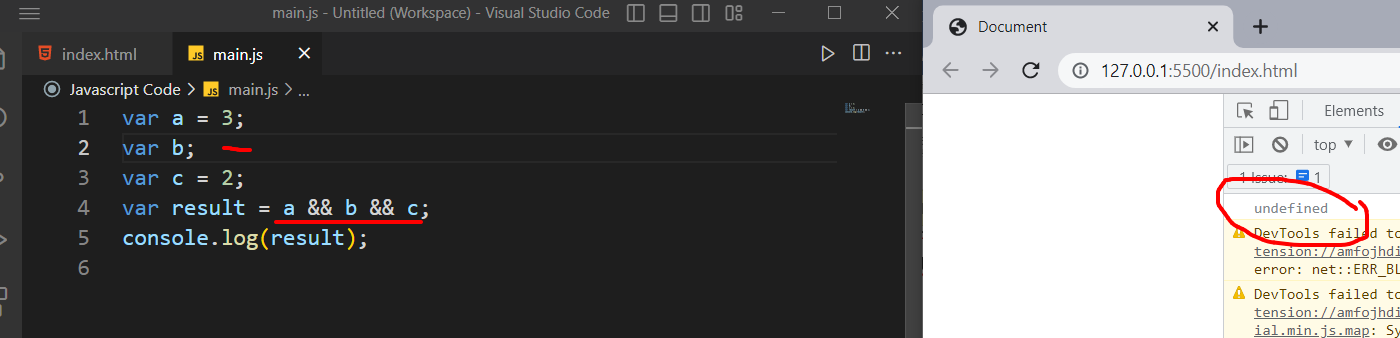
document.all chuyển sang boolean sẽ là false

document.all khi là toán hạng của toán tử so sánh == hoặc != sẽ là undefined

Khi typeof document.all sẽ trả về "undefined"

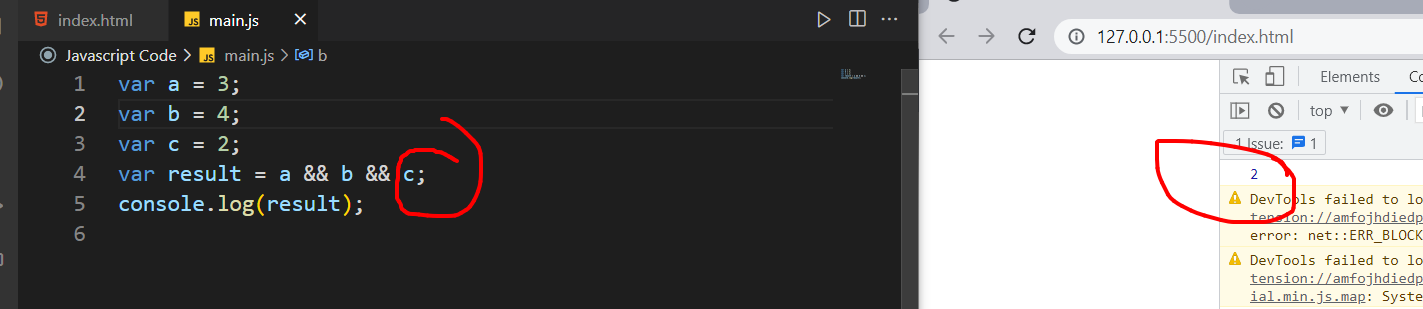
**Toán tử logic: not, and, or**

**And:**



Toán tử && hoạt động theo kiểu đi từ trái sang phải, nếu không phải là 6 giá trị false thì sẽ đi tiếp, nếu đến cuối cùng mà vẫn không nhận được false nào thì sẽ nhận giá trị của biến cuối cùng.

Trong ảnh phía trên do b là undefined nên biến result nhận được là undefined



Trong ảnh này do đi đến cuối phép so sánh mà không nhận được false nào cả thì sẽ lấy giá trị của biến c.

**Or:** cách hoạt động tương tự phép and.