**LAB 10 – GIẢI THÍCH**

**Câu 3. Giải thích:**

* **Cho biết ý nghĩa của toán tử “==” và “===” trong JavaScript.**
  + Giải thích:
    - == : toán tử so sánh lỏng lẻo, JS sẻ chuyển đổi dữ liệu của hai toán tử về cùng một kiểu trước khi so sánh.
    - === : toán tử so sánh chặt chẽ, JS sẽ so sánh giá trị và kiểu dữ liệu mà không thực hiện ép kiểu.
* **Giải thích câu lệnh:**

const form = document.forms['Calculator'];

const a = parseFloat(form['numberA'].value);

const b = parseFloat(form['numberB'].value);

* + Giải thích:
    - Câu lệnh, const form = document.forms['Calculator']; :
      * Trong đó, document.forms là một thuộc tính đối tượng document trong DOM, chứa tất cả phần tử <form> có trong trang HTML.
      * Khi truy cập document.forms['Calculator'], bạn đang lấy form có thuộc tính name là "Calculator".
      * Kết quả, biến biến form sẽ đại diện cho phần tử <form name="Calculator"> trong HTML, cho phép truy cập và thao tác các trường nhập dữ liệu bên trong form.
    - Câu lệnh, const a = parseFloat(form['numberA'].value); và const b = parseFloat(form['numberB'].value);
      * Trong đó, form['numberA']/ form['numberB'] truy cập vào phần tử input có trong form và có thuốc tính name là “numberA/numberB”
      * .value lấy giá trị hiện tại mà người dùng nhập vào ô input.
      * Hàm parseFloat() được sử dụng để chuyển chuỗi đó thành giá trị số thực. Nếu chuỗi không chứa một số hợp lệ, kết quả là NaN.
      * Biến a/b được gán giá trị số sau khi chuyển đổi.
* **Giải thích câu lệnh: if (isNaN(a) || isNaN(b))**
  + Giải thích:
    - Hàm isNaN() trong JS trả về hàm true nếu tham số không phải là một số, ngược lại trả về false.
    - Câu lệnh if (isNaN(a) || isNaN(b)): nếu a không phải là số hoặc b không phải là số, thì hiển thị thông báo ‘Vui lòng nhập một số hợp lệ!’. Và dừng thực thi thông qua hàm return.

**Câu 5: Giải thích**

* **Cho biết ý nghĩa của toán tử “!=” và “!==” trong JavaScript.**
  + Giải thích:
    - “!=” so sánh không bằng lỏng lẻo. Khi so sánh hai giá trị thì thực hiện ép kiểu giống nhau nếu cần thiết.
    - “!==” so sánh không bằng chặt chẽ. So sánh mà không cần ép kiểm, nó kiểm tra giá trị và kiểu dữ liệu, kết quả trả về true nếu giá trị hoặc kiểu dữ liệu của chúng giống nhau.
* **Cho biết ý nghĩa của onsubmit="return validateForm()".**
  + Giải thích:
    - Thuộc tính onsubmit được sử dụng để chỉ định một hàm JS sẽ chạy trước khi form được gửi đi.
    - onsubmit="return validateForm()":
      * Khi người dùng nhấn nút submit, hàm validateForm() sẽ được gọi
      * Nếu hàm trả về true, quá trình gửi form sẽ được tiếp diễn. Ngược lại nếu trả về false, việc gửi form sẽ bị hủy.
* **Cho biết ý nghĩa của minlength="8" maxlength="20" pattern="(?=.\*\d)(?=.\*[a z])(?=.\*[A-Z]).{8,}" title="Mật khẩu phải chứa ít nhất 1 chữ cái viết hoa, 1 chữ cái viết thường và 1 số.".**
  + Ý nghĩa:
    - minlength="8" và maxlength="20": yêu cầu người dùng nhập vào ít nhất 8 kí tự/giới hạn số kí tự nhập vào không vượt quá 20 kí tự. Nếu nhập vào nhỏ hơn 8 và nhiều hơn 20 kí tự, trình duyệt sẽ thông báo lỗi.
    - pattern="(?=.\*\d)(?=.\*[a-z])(?=.\*[A-Z]).{8,}": Thuộc tính dùng để khai báo một biểu thức mà giá trị nhập vào phải phù hợp. Chi tiết:
      * (**?=.\*\**d): Yêu cầu trong chuỗi phải có ít nhất 1 chữ số (\d là ký hiệu cho chữ số).
      * (?=.\*[a-z]): Yêu cầu có ít nhất 1 ký tự thường (từ a đến z).
      * (?=.\*[A-Z]): Yêu cầu có ít nhất 1 ký tự in hoa (từ A đến Z).
      * .{8,}: Sau đó chuỗi phải có ít nhất 8 ký tự (dấu chấm . đại diện cho bất kỳ ký tự nào, và {8,} có nghĩa là tối thiểu 8 ký tự).
    - Title= “…”: Thuộc tính title cung cấp thông tin hướng dẫn cho người dùng. Khi nhập giá trị không khớp, trình duyệt sẽ hiển thị nội dung của title như một thông báo lỗi.
* **Regex là gì? Cho một vài ví dụ.**
  + Định nghĩa:
    - **Regex – Biểu thức chính quy:** Là một tập hợp các kí tự được định nghĩa theo một cấu trúc mẫu, dùng để tìm kiếm, so sánh hoặc thay thế các chuỗi kí tự. Regex được sử dụng rộng rãi để kiểm tra định dạng dữ liệu, chẳng hạn như email, số điện thoại, mật khẩu..v.v
  + Ví dụ:
    - Kiểm tra email hợp lệ: ^[a-zA-Z0-9.\_%+-][+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$](mailto:+@[a-zA-Z0-9.-]+\.%5ba-zA-Z%5d%7b2,%7d$)
    - Kiểm tra số điện thoại gồm 10 chữ số: ^\d{10}$
    - Mật khẩu chứa tối thiểu 8 ký tự, có ít nhất 1 chữ số và 1 chữ cái: (?=.\*\d)(?=.\*[a-zA-Z]).{8,}
* **Cho biết ý nghĩa của birthdayInput.setCustomValidity("Bạn phải trên 18 tuổi!").**
  + Giải thích:
    - setCustomValidity là một phương thức của đối tượng input trong DOM, được sử dụng để thiết lập một thông báo lỗi tùy chỉnh cho trường nhập dữ liệu nếu nhập vào không hợp lệ.
    - Trong code trên, nếu người dùng chưa đủ 18 tuổi thì sẽ xuất hiện thông báo lỗi với nội dung “Bạn phải trên 18 tuổi!”. Nếu trường hợp không có lỗi, trình duyệt sẽ ngăn gửi form và thông báo lỗi cho người dùng.
    - Nếu đã hợp lệ, form sẽ được gửi đi.