

```
<script>
(() => {
  'use strict';
  const form = document.querySelector('.needs-validation');
  form.addEventListener('submit', e => {
    if (!form.checkValidity()) {
      e.preventDefault();
      e.stopPropagation();
    } else {
      e.preventDefault();
      alert('Form submitted successfully!');
      form.reset();
    }
    form.classList.add('was-validated');
  });
})();
```

eita line by line bhujao eita tmi e dicho amke

Sure! Here's a breakdown of the script line by line:

javascript

Copy code

```
((() => {
  'use strict';
```

- This is an **Immediately Invoked Function Expression (IIFE)**, which runs as soon as it's defined. The `((() => { ... })())` is an arrow function that executes immediately.

Ask anything

Catching common coding mistakes.

javascript

Copy code

```
const form = document.querySelector('.needs-validation');
```

- This line uses `document.querySelector()` to select the first element in the HTML document with the class `.needs-validation` and assigns it to the constant variable `form`.
- This `form` variable now represents the form element on the page that you're trying to work with.

javascript

Copy code

```
form.addEventListener('submit', e => {
```

- This attaches an event listener to the `form` element. It listens for a `submit` event, which triggers when the form is submitted.
- When the form is submitted, the provided arrow function `e => { ... }` will run. The `e` is the event object.

javascript

Copy code

```
if (!form.checkValidity()) {
```

- This checks if the form is **valid** using the `checkValidity()` method, which checks whether all form inputs meet their validation requirements (like required fields, proper format, etc.).
- If the form is **not valid**, the code inside this block will run.

javascript

Copy code

```
  e.preventDefault();
  e.stopPropagation();
```

Ask anything

nothing does not happen.

- `e.stopPropagation()` stops the event from bubbling up the DOM. It prevents the event from being handled by any parent elements.

javascript

Copy code

```
  } else {
    e.preventDefault();
    alert('Form submitted successfully!');
    form.reset();
  }
```

- If the form is **valid** (`else` block), it does the following:
  - `e.preventDefault()` again to prevent the actual form submission (e.g., page refresh).
  - It shows an **alert** saying "Form submitted successfully!".
  - `form.reset()` resets all the fields in the form to their default values (usually empty or unselected).

javascript

Copy code

```
  form.classList.add('was-validated');
```

- This adds the class `was-validated` to the `form`.
- This is often used in frameworks like **Bootstrap** to apply styling to indicate that the form has been validated (such as showing red borders for invalid fields).

javascript

Copy code

```
});
})();
```

- This closes the event listener function and the IIFE, marking the end of the script.

Ask anything

- The script validates the form on submission.
- If the form is invalid, it prevents submission and stops the form from being processed.
- If the form is valid, it shows an alert, resets the form, and adds a class to style the validation feedback.

অবশ্যই নিচে স্ক্রিপ্টটি বাংলা ভাষায় ব্যাখ্যা করছি:

```
javascript Copy code

(() => {
  'use strict';
```

- এটি একটি Immediately Invoked Function Expression (IIFE)। এর মানে হলো, যখনই এই কোডটি লেখা হয়, তখনই এটি স্বয়ংক্রিয়ভাবে চালু হয়ে যায়।
- 'use strict'; দিয়ে strict mode চালু করা হয়, যা কোডে কিছু ভুল ধরতে সাহায্য করে এবং নিরাপদ কোড লেখার জন্য বাধ্য করে।

```
javascript Copy code

const form = document.querySelector('.needs-validation');
```

- এই লাইনটি document.querySelector() ব্যবহার করে HTML ডকুমেন্টে .needs-validation ক্লাসের সাথে প্রথম এলিমেন্টটি নির্বাচন করে এবং সেটি form নামে একটি কনস্ট্যান্ট ভেরিয়েবলে সংরক্ষণ করে।
- এই form ভেরিয়েবলটি এখন পেজের ঐ ফর্ম এলিমেন্টটিকে প্রতিনিধিত্ব করছে।

Ask anything

```
form.addEventListener('submit', e => {
```

- এখানে form -এ একটি submit ইভেন্টের জন্য event listener যোগ করা হয়েছে। অর্থাৎ, ফর্মটি যখন সাবমিট হবে, তখন এই ফাংশনটি কার্যকর হবে।
- e হলো ইভেন্ট অবজেক্ট, যেটি সাবমিশনের সাথে সম্পর্কিত তথ্য ধারণ করে।

```
javascript Copy code

  if (!form.checkValidity()) {
```

- এই লাইনটি চেক করে ফর্মটি valid (অথবা সঠিক) কিনা। checkValidity() ফর্মের ইনপুটগুলো যাচাই করে এবং দেখতে পারে যে সব ফিল্ড সঠিকভাবে পূর্ণ হয়েছে কিনা (যেমন: required ফিল্ড পূর্ণ হয়েছে, সঠিক ফর্ম্যাটে আছে ইত্যাদি)।
- যদি ফর্মটি অবৈধ হয়, তাহলে এই ব্লকের কোড কার্যকর হবে।

```
javascript Copy code

    e.preventDefault();
    e.stopPropagation();
```

- e.preventDefault() ফর্মের ডিফল্ট সাবমিশন আচরণ (যেমন পেজ রিলোড) বন্ধ করে দেয়।
- e.stopPropagation() ইভেন্টটি প্যারেন্ট এলিমেন্টগুলোর কাছে পৌঁছাতে বাধা দেয়, অর্থাৎ ইভেন্টটি আর উপরের দিকে চলবে না।

```
javascript Copy code

  } else {
    e.preventDefault();
    alert('Form submitted successfully!');
    form.reset();
  }
```

Ask anything

- আবার e.preventDefault() ব্যবহার করা হয়েছে, যাতে ফর্মটি সরাসরি সাবমিট না হয় (পেজ রিলোড না হয়)।
- তারপর একটি alert দেখানো হয় যা বলে "Form submitted successfully!"।
- form.reset() ফর্মটি পুনরায় রিসেট করে, অর্থাৎ সব ইনপুট ফিল্ড খালি হয়ে যায়।

```
javascript Copy code

  form.classList.add('was-validated');
```

- এই লাইনটি form -এ was-validated ক্লাস যোগ করে।
- এই ক্লাসটি সাধারণত Bootstrap মতো ফ্রেমওয়ার্কে ব্যবহার করা হয়, যেখানে ফর্মের ক্ষেত্রগুলোর ভ্যালিডেশন স্টাইল দেখানোর জন্য এটি ব্যবহার করা হয় (যেমন, অগ্রহণযোগ্য ইনপুট ফিল্ড রెড বর্ডার দেখানো হয়)।

```
javascript Copy code

  });
})();
```

- এখানে ইভেন্ট লিসেনারের ফাংশন এবং IIFE (Immediately Invoked Function Expression)-এর শেষ হয়ে যায়।

সারাংশ:

- স্ক্রিপ্টটি ফর্ম সাবমিশনকে যাচাই করে।
- যদি ফর্মটি অবৈধ হয়, তবে সাবমিট হওয়া বন্ধ করে এবং কোনো পরিবর্তন ছাড়াই ফর্মটি ফেরত পাঠায়।
- যদি ফর্মটি বৈধ হয়, তাহলে একটি সফল সাবমিশন মেসেজ দেখানো হয়, ফর্ম রিসেট করা হয়, এবং ফর্মের উপর ভ্যালিডেশন স্টাইল প্রয়োগ করা হয়।



Ask anything

📎 Attach 🔍 Search 📖 Study

🗣️ Voice