**Puoi spiegarmi perchè bisogna mettere 'return false' dentro la funzione?**

Ottima domanda! Il return false è fondamentale per controllare il comportamento del form HTML. Ti spiego cosa succede:

**Comportamento predefinito del form**

Quando un form HTML viene inviato (submit), il browser ha un **comportamento predefinito**:

1. Raccoglie tutti i dati del form
2. Li invia all'URL specificato nell'attributo action
3. **Ricarica la pagina** o naviga verso una nuova pagina

Nel tuo caso:

<form action="#" method="post" onsubmit="return saveData()">

Senza return false, quando premi "SEND":

1. La funzione saveData() viene eseguita
2. Il calcolo viene fatto e il risultato mostrato
3. **Ma subito dopo** il form viene inviato e la pagina si ricarica
4. Risultato: vedi il numero per una frazione di secondo, poi la pagina si ricarica e tutto scompare!

**Come funziona return false**

Il return false nella funzione **impedisce** l'invio del form:

function saveData(){

x = document.getElementById("1st").value;

y = document.getElementById("2nd").value;

let division = x / y;

document.getElementById("res").innerHTML = division;

return false; // ← Questo dice al browser: "NON inviare il form!"

}

Quando la funzione restituisce false, il browser capisce che non deve procedere con l'invio del form.

**Prova pratica**

Se rimuovi il return false e provi il codice, vedrai che:

* Il risultato appare per un attimo
* Poi la pagina si ricarica immediatamente
* E il risultato scompare

**Alternative moderne**

Oggi si usa spesso anche questo approccio più esplicito:

function saveData(event){

event.preventDefault(); // Impedisce il comportamento predefinito

x = document.getElementById("1st").value;

y = document.getElementById("2nd").value;

let division = x / y;

document.getElementById("res").innerHTML = division;

}

Entrambi i metodi fanno la stessa cosa: impediscono al form di essere inviato, permettendo al JavaScript di gestire tutto lato client senza ricaricare la pagina.