Una consulta por favor, es posible en javascript ocultar el innerHTML de un input?

No, **no es posible ocultar el innerHTML de un <input>** porque los elementos input no tienen innerHTML en el sentido tradicional de los elementos HTML que contienen contenido entre sus etiquetas de apertura y cierre (como un div, p, o span).

El valor de un <input> se almacena en su atributo value. Los input son elementos vacíos (void elements), lo que significa que no pueden tener hijos ni contenido interno.

Si lo que quieres es que el valor del input no sea visible para el usuario, tienes varias opciones:

1. **Cambiar el type del input a hidden:** Esta es la forma más común y semánticamente correcta de ocultar un input. El valor seguirá siendo enviado con el formulario, pero el usuario no lo verá ni podrá interactuar con él.

HTML

<input type="hidden" id="miInputOculto" name="miDato" value="valor\_secreto">

Si el input ya es type="date" y quieres mantener la funcionalidad del calendario (como entendí en nuestra conversación anterior), entonces type="hidden" no te servirá porque ocultaría el calendario.

1. **Ocultar el elemento input completo con CSS:** Si quieres que el input mantenga su type="date" (para que el calendario funcione cuando hagas clic en él o en otro elemento que lo active), pero que el input en sí no muestre su valor ni esté visible, puedes usar CSS:

HTML

<input type="date" id="expected\_closing\_date\_picker" name="expected\_closing\_date" style="display: none;">

<span id="expected\_closing\_date\_display"></span>

Y en JavaScript, simplemente te aseguras de que el <span> muestre la fecha, y que al hacer clic en el <span> se abra el calendario del input oculto (esto requiere un poco más de lógica JS).

**Modificación al JavaScript que ya teníamos (para activar el input oculto al hacer clic en el <span>):**

Si ocultas el input con display: none;, para que el usuario pueda abrir el calendario haciendo clic en el <span> de visualización, necesitarías algo así:

JavaScript

// ... tu código existente ...

const expectedClosingDatePicker = document.getElementById('expected\_closing\_date\_picker');

const expectedClosingDateDisplay = document.getElementById('expected\_closing\_date\_display');

// Ocultar el input original

if (expectedClosingDatePicker) {

expectedClosingDatePicker.style.display = 'none';

}

// Asegúrate de que el span sea clickeable o tenga un cursor indicativo

if (expectedClosingDateDisplay) {

expectedClosingDateDisplay.style.cursor = 'pointer'; // O un estilo de botón

// Cuando se hace clic en el SPAN, activamos el input de fecha oculto

expectedClosingDateDisplay.addEventListener('click', function() {

if (expectedClosingDatePicker) {

expectedClosingDatePicker.showPicker(); // Método para mostrar el calendario (no soportado en todos los navegadores)

// O, alternativamente, puedes forzar el foco y el navegador podría abrirlo

// expectedClosingDatePicker.focus();

// expectedClosingDatePicker.click(); // Simular un click para abrir el picker

}

});

}

// ... la función formatDateForDisplay y el listener 'change' del input ...

if (expectedClosingDatePicker && expectedClosingDateDisplay) {

expectedClosingDatePicker.addEventListener('change', function() {

const selectedDate = this.value; // Esto será en formato YYYY-MM-DD

expectedClosingDateDisplay.textContent = formatDateForDisplay(selectedDate);

});

}

// Inicializa el span si ya hay un valor

if (expectedClosingDatePicker.value && expectedClosingDateDisplay.textContent.trim() === '') {

expectedClosingDateDisplay.textContent = formatDateForDisplay(expectedClosingDatePicker.value);

}

**Consideraciones para showPicker():** El método showPicker() es el más directo para abrir el calendario de un input type="date", pero **no es compatible con todos los navegadores** (es una API relativamente nueva). Si la compatibilidad con navegadores antiguos es crucial, la combinación de focus() y click() puede ser una alternativa, aunque su comportamiento varía más entre navegadores.

En resumen, la respuesta directa a tu pregunta es: **no, no puedes ocultar el innerHTML de un input porque no tiene uno.** Lo que puedes hacer es ocultar el elemento input completo con CSS (display: none; o visibility: hidden;) y manejar la interacción a través de otro elemento visible.

Voy a plantear la pregunta de otra forma ¿hay alguna forma de hacer que el texto del input type=date no sea visible para el usuario?

Sí, entiendo perfectamente la intención. Quieres la funcionalidad del calendario nativo (type="date") pero sin que el usuario vea el texto en formato YYYY-MM-DD dentro del propio input después de seleccionar una fecha.

Dado que un input no tiene innerHTML y su contenido visual es directamente su value, y ese value es gestionado por el navegador para el type="date", **no hay una forma directa y estándar de decirle al navegador que "oculte solo el texto" dentro de un input type="date" mientras mantiene la funcionalidad del calendario visible**. El navegador está diseñado para mostrar el valor seleccionado en el campo.

Las soluciones que te permiten lograr el efecto deseado (calendario nativo + dd/mm/yyyy visible para el usuario + YYYY-MM-DD enviado al servidor) siempre implican **ocultar o estilizar fuertemente el input type="date" original** y usar un elemento **separado** (como un <span> o <div>) para mostrar el formato dd/mm/yyyy.

Aquí te reitero y detallo las mejores estrategias para lograr esto, y cómo aplicarlas a tu create.blade.php:

### Estrategia 1: Ocultar el input type="date" y activar el calendario al hacer clic en el <span> (Recomendada si quieres que el input sea *completamente* invisible)

Esta es la solución más cercana a lo que pides. El input type="date" estará oculto con CSS, y tu <span> personalizado será el elemento interactivo para el usuario.

**Modificación en resources/views/salesperson/leads/create.blade.php:**

**1. HTML (Dentro de tu div.mb-4 para la fecha):**

HTML

<div class="mb-4">

<label for="expected\_closing\_date\_display" class="block text-gray-700 text-sm font-bold mb-2">Data de Fechamento Esperada (Opcional):</label>

<div class="flex items-center">

<input type="date"

id="expected\_closing\_date\_picker"

name="expected\_closing\_date"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline @error('expected\_closing\_date') border-red-500 @enderror"

value="{{ old('expected\_closing\_date') }}"

style="display: none;"> {{-- Alternativa a style="display: none;": <input type="date" ... hidden> --}}

<span id="expected\_closing\_date\_display"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline ml-2 text-gray-700 font-medium cursor-pointer"

tabindex="0"> {{-- Añadimos tabindex para hacerlo enfocable con teclado --}}

@if(old('expected\_closing\_date'))

{{ \Carbon\Carbon::parse(old('expected\_closing\_date'))->format('d/m/Y') }}

@endif

@if(!old('expected\_closing\_date'))

Selecione uma data

@endif

</span>

</div>

@error('expected\_closing\_date')

<p class="text-red-500 text-xs italic">{{ $message }}</p>

@enderror

</div>

**2. JavaScript (@push('scripts')):**

JavaScript

@push('scripts')

<script>

document.addEventListener('DOMContentLoaded', function() {

// ... (Tu código existente para el checkbox de la compañía y búsqueda de productos) ...

// =============================================================

// CÓDIGO PARA LA FECHA dd/mm/yyyy Y OCULTAR EL INPUT ORIGINAL

// =============================================================

const expectedClosingDatePicker = document.getElementById('expected\_closing\_date\_picker');

const expectedClosingDateDisplay = document.getElementById('expected\_closing\_date\_display');

// Función para formatear YYYY-MM-DD a DD/MM/YYYY para la visualización

function formatDateForDisplay(dateString) {

if (!dateString) return '';

const date = new Date(dateString + 'T00:00:00'); // Añadir 'T00:00:00' para evitar problemas de zona horaria

if (isNaN(date.getTime())) { // Validar si la fecha es válida

return dateString; // Si no es una fecha válida, devuelve la cadena original

}

const day = String(date.getDate()).padStart(2, '0');

const month = String(date.getMonth() + 1).padStart(2, '0'); // getMonth() es base 0

const year = date.getFullYear();

return `${day}/${month}/${year}`;

}

// Listener para el input type="date"

// Cuando el usuario selecciona una fecha con el calendario (invisible), actualizamos el <span>

if (expectedClosingDatePicker && expectedClosingDateDisplay) {

expectedClosingDatePicker.addEventListener('change', function() {

const selectedDate = this.value; // El valor del input type="date" siempre es YYYY-MM-DD

expectedClosingDateDisplay.textContent = formatDateForDisplay(selectedDate);

});

// Al hacer clic en el SPAN, abrimos el calendario del input oculto

expectedClosingDateDisplay.addEventListener('click', function() {

// Utiliza showPicker() si quieres la forma más directa (navegadores modernos)

if (expectedClosingDatePicker.showPicker) {

expectedClosingDatePicker.showPicker();

} else {

// Fallback para navegadores antiguos: enfocar y simular clic

expectedClosingDatePicker.focus();

expectedClosingDatePicker.click();

}

});

// Para manejar la fecha inicial si viene de old()

if (expectedClosingDatePicker.value) {

expectedClosingDateDisplay.textContent = formatDateForDisplay(expectedClosingDatePicker.value);

} else {

expectedClosingDateDisplay.textContent = 'Selecione uma data'; // Placeholder si no hay fecha

}

}

});

</script>

@endpush

**Explicación de los cambios clave:**

* **HTML del input:** Ahora tiene style="display: none;" (o puedes usar el atributo hidden directamente <input type="date" ... hidden>). Esto lo hace completamente invisible, pero sigue siendo funcional y su valor se enviará con el formulario.
* **HTML del <span>:**
  + Le he añadido las mismas clases de Tailwind CSS que tu input para que tenga el mismo aspecto visual (borde, padding, etc.), haciendo que parezca un campo de entrada.
  + Se le agrega cursor-pointer para indicar que es clickeable.
  + Se le añade tabindex="0" para que sea accesible con el teclado (puedas navegar a él con Tab).
  + Ahora incluye un if/else para mostrar "Selecione uma data" si no hay un old() valor, lo cual es mejor para la experiencia del usuario.
* **JavaScript:**
  + El input.style.display = 'none'; se asegura de que el campo original esté oculto al cargar la página.
  + El eventListener en expectedClosingDateDisplay (el <span>) ahora usa expectedClosingDatePicker.showPicker() (si está disponible) o un fallback de focus() y click() para abrir el calendario nativo del input oculto cuando el usuario hace clic en el <span>.
  + La inicialización del <span> al cargar la página ahora también maneja el caso donde no hay un valor old(), mostrando el placeholder.

Esta es la forma más limpia y robusta de lograr el efecto que deseas, dándole al usuario la interfaz que espera (dd/mm/yyyy) y manteniendo la funcionalidad nativa de selección de fecha y el envío correcto al backend (YYYY-MM-DD).

Y que tal darle al texto color=white?

Show thinking

¡Excelente idea! Utilizar el color del texto para "ocultarlo" es una forma ingeniosa de mantener el input type="date" visible (con sus bordes, padding y el ícono de calendario nativo del navegador) mientras el texto del YYYY-MM-DD se vuelve invisible contra un fondo blanco.

Esta aproximación es muy buena si quieres que el input *parezca* un campo donde se puede interactuar, pero que el texto de la fecha formateada esté en el <span> de al lado.