Getting Started

Preline JavaScript

This page explains how Preline JavaScript works, its methodology, and provides some examples.

Heads up!

For detailed documentation on plugins, please visit the plugins page.

Learn more >

Helper for dynamic added components

Preline JavaScript plugins include `autoInit` static method, it's useful when you need to reinitialize all elements on the page.

```
window.HSStaticMethods.autoInit();
```

This method can also be used with certain collections of initialized elements.

```
window.HSStaticMethods.autoInit(['carousel', 'dropdown']);
```

AJAX example:

```
<select id="dynamic-select-options" data-hs-select='{
  "placeholder": "Select option...",
  ...
}' class="hidden --prevent-on-load-init"></select>
<button type="button" id="load-options">Load Options</button>

<script>
```

On this page

```
  document.getElementById("load-options").addEventListener("click", function() {
    loadOptions();
  });

  function loadOptions() {
    const xhr = new XMLHttpRequest();

    xhr.onreadystatechange = function() {
      if (this.readyState == 4) {
        if (this.status == 200) {
          const options = JSON.parse(this.responseText);
          populateOptions(options);
        } else {
          console.error("Failed to load options:", this.status, this.statusText);
        }
      }
    };

    xhr.open("GET", "https://some-api.com/options", true);
    xhr.send();
  }

  function populateOptions(options) {
    const selectElement = document.getElementById("dynamic-select-options");
    selectElement.innerHTML = "";

    options.forEach(function(option) {
      const optionElement = document.createElement("option");
      optionElement.value = option.value;
      optionElement.textContent = option.text;

      selectElement.appendChild(optionElement);
    });

    window.HSStaticMethods.autoInit(['select']);
  }
</script>
```

Usage of static methods inside TypeScript (TS) files

To use static methods inside TS files, it is necessary to declare the interface inside the files where [illegible]ed, this will prevent possible warnings and errors.

```
...

import { IStaticMethods } from "preline/preline";
declare global {
  interface Window {
    HSStaticMethods: IStaticMethods;
  }
}

...

window.HSStaticMethods.autoInit();
```

Prevent auto initialize

To prevent an element from auto-initializing, you can add the `--prevent-on-load-init` CSS class.

This can be useful if you want to initialize elements using an event other than load.

```
<select data-hs-select='{
  "placeholder": "Select option...",
  ...
}' class="hidden --prevent-on-load-init">
  <option value="">Choose</option>
  ...
</select>

<script>
  document.addEventListener('DOMContentLoaded', () => {
    document.querySelectorAll('[data-hs-select].--prevent-on-load-init').forEach((el) =
  });
</script>
```