

### :host

The `:host` pseudo-class matches the shadow tree's host element. Given the following code, the selector `:host` matches the `<x-foo>` element. Compound selectors are also supported, such as `:host(.foo)` matches the `<x-foo>` element as well.

```
<x-foo class="foo">
  <"shadow tree">
    <style>
      :host { /* matches x-foo */ }
    </style>
    <div class="foo">...</div>
  </>
</x-foo>
```

Note that using the `.foo` selector inside the shadow tree will only match the `div.foo` element and not the host `x-foo.foo` element.

### ::shadow

The `::shadow` pseudo-element matches the shadow tree's top-level elements. See the following code for examples.

```
<style>
x-foo::shadow {
  /* matches div and span#root */
}
x-foo::shadow > span {
  /* matches only span#root */
}
x-foo::shadow span {
  /* matches all spans */
}
</style>
<x-foo class="foo">
  <"shadow tree">
    <div>
      <span id="child">...</span>
    </div>
    <span id="root">...</span>
  </>
</x-foo>
```

### :host-context()

The `:host-context()` pseudo-class matches a parent element outside the shadow tree. Given the following code, the selector `:host-context()` matches the `<div>` element with the `green-theme` class. This can be used to coordinate styles for a group of web components, for example when theming.

```
<div class="green-theme">
  <x-foo>
  <"shadow tree">
    <style>
      :host-context(.green-theme) {
        color: #060;
      }
    </style>
    <div class="foo">...</div>
  </>
</x-foo>
</div>
```

