Tree view with a scroll pane in HTML+JavaScript+CSS

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1">

<style>

ul, #myUL {

list-style-type: none;

}

#myUL {

margin: 0;

padding: 0;

}

.caret {

cursor: pointer;

-webkit-user-select: none; /\* Safari 3.1+ \*/

-moz-user-select: none; /\* Firefox 2+ \*/

-ms-user-select: none; /\* IE 10+ \*/

user-select: none;

}

.caret::before {

content: "\25B6";

color: black;

display: inline-block;

margin-right: 6px;

}

.caret-down::before {

-ms-transform: rotate(90deg); /\* IE 9 \*/

-webkit-transform: rotate(90deg); /\* Safari \*/'

transform: rotate(90deg);

}

.nested {

display: none;

}

.active {

display: block;

}

/\* width \*/

::-webkit-scrollbar {

width: 20px;

}

/\* Track \*/

::-webkit-scrollbar-track {

box-shadow: inset 0 0 5px grey;

border-radius: 10px;

}

/\* Handle \*/

::-webkit-scrollbar-thumb {

background: red;

border-radius: 10px;

}

/\* Handle on hover \*/

::-webkit-scrollbar-thumb:hover {

background: #b30000;

}

</style>

</head>

<body>

<h2>Custom Scrollbar Example</h2>

<h2>Tree View</h2>

<p>A tree view represents a hierarchical view of information, where each item can have a number of subitems.</p>

<p>Click on the arrow(s) to open or close the tree branches.</p>

<ul id="myUL">

<li><span class="caret">Beverages</span>

<ul class="nested">

<li>Coffee</li>

<li><span class="caret">Tea</span>

<ul class="nested">

<li>Black Tea</li>

<li><span class="caret">Green Tea</span>

<ul class="nested">

<li>Sencha</li>

<li>Gyokuro</li>

<li>Matcha</li>

<li>Pi Lo Chun</li>

</ul>

<li>White Tea</li>

</li>

</ul>

<li>Water</li>

</li>

</ul>

</li>

</ul>

<script>

var toggler = document.getElementsByClassName("caret");

var i;

for (i = 0; i < toggler.length; i++) {

toggler[i].addEventListener("click", function() {

this.parentElement.querySelector(".nested").classList.toggle("active");

this.classList.toggle("caret-down");

});

}

</script>

</body>

</html>

Another example @   
<https://stackoverflow.com/questions/5636375/how-to-create-a-collapsing-tree-table-in-html-css-js>

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1">

<style>

body {

font-family: Arial;

}

ul.tree li {

list-style-type: none;

position: relative;

}

ul.tree li ul {

display: none;

}

ul.tree li.open > ul {

display: block;

}

ul.tree li a {

color: black;

text-decoration: none;

}

ul.tree li a:before {

height: 1em;

padding:0 .1em;

font-size: .8em;

display: block;

position: absolute;

left: -1.3em;

top: .2em;

}

ul.tree li > a:not(:last-child):before {

content: '+';

}

ul.tree li.open > a:not(:last-child):before {

content: '-';

}</style>

</head>

<body>

<h2>Custom Scrollbar Example</h2>

<h2>Tree View</h2>

<p>A tree view represents a hierarchical view of information, where each item can have a number of subitems.</p>

<p>Click on the arrow(s) to open or close the tree branches.</p>

<ul class="tree">

<li><a href="#">Part 1</a>

<ul>

<li><a href="#">Item A</a>

<ul>

<li><a href="#">Sub-item 1</a></li>

<li><a href="#">Sub-item 2</a></li>

<li><a href="#">Sub-item 3</a></li>

</ul>

</li>

<li><a href="#">Item B</a>

<ul>

<li><a href="#">Sub-item 1</a></li>

<li><a href="#">Sub-item 2</a></li>

<li><a href="#">Sub-item 3</a></li>

</ul>

</li>

<li><a href="#">Item C</a>

<ul>

<li><a href="#">Sub-item 1</a></li>

<li><a href="#">Sub-item 2</a></li>

<li><a href="#">Sub-item 3</a></li>

</ul>

</li>

<li><a href="#">Item D</a>

<ul>

<li><a href="#">Sub-item 1</a></li>

<li><a href="#">Sub-item 2</a></li>

<li><a href="#">Sub-item 3</a></li>

</ul>

</li>

<li><a href="#">Item E</a>

<ul>

<li><a href="#">Sub-item 1</a></li>

<li><a href="#">Sub-item 2</a></li>

<li><a href="#">Sub-item 3</a></li>

</ul>

</li>

</ul>

</li>

<li><a href="#">Part 2</a>

<ul>

<li><a href="#">Item A</a>

<ul>

<li><a href="#">Sub-item 1</a></li>

<li><a href="#">Sub-item 2</a></li>

<li><a href="#">Sub-item 3</a></li>

</ul>

</li>

<li><a href="#">Item B</a>

<ul>

<li><a href="#">Sub-item 1</a></li>

<li><a href="#">Sub-item 2</a></li>

<li><a href="#">Sub-item 3</a></li>

</ul>

</li>

<li><a href="#">Item C</a>

<ul>

<li><a href="#">Sub-item 1</a></li>

<li><a href="#">Sub-item 2</a></li>

<li><a href="#">Sub-item 3</a></li>

</ul>

</li>

<li><a href="#">Item D</a>

<ul>

<li><a href="#">Sub-item 1</a></li>

<li><a href="#">Sub-item 2</a></li>

<li><a href="#">Sub-item 3</a></li>

</ul>

</li>

<li><a href="#">Item E</a>

<ul>

<li><a href="#">Sub-item 1</a></li>

<li><a href="#">Sub-item 2</a></li>

<li><a href="#">Sub-item 3</a></li>

</ul>

</li>

</ul>

</li>

<li><a href="#">Part 3</a>

<ul>

<li><a href="#">Item A</a>

<ul>

<li><a href="#">Sub-item 1</a></li>

<li><a href="#">Sub-item 2</a></li>

<li><a href="#">Sub-item 3</a></li>

</ul>

</li>

<li><a href="#">Item B</a>

<ul>

<li><a href="#">Sub-item 1</a></li>

<li><a href="#">Sub-item 2</a></li>

<li><a href="#">Sub-item 3</a></li>

</ul>

</li>

<li><a href="#">Item C</a>

<ul>

<li><a href="#">Sub-item 1</a></li>

<li><a href="#">Sub-item 2</a></li>

<li><a href="#">Sub-item 3</a></li>

</ul>

</li>

<li><a href="#">Item D</a>

<ul>

<li><a href="#">Sub-item 1</a></li>

<li><a href="#">Sub-item 2</a></li>

<li><a href="#">Sub-item 3</a></li>

</ul>

</li>

<li><a href="#">Item E</a>

<ul>

<li><a href="#">Sub-item 1</a></li>

<li><a href="#">Sub-item 2</a></li>

<li><a href="#">Sub-item 3</a></li>

</ul>

</li>

</ul>

</li>

</ul>

<script>

var tree = document.querySelectorAll('ul.tree a:not(:last-child)');

for(var i = 0; i < tree.length; i++){

tree[i].addEventListener('click', function(e) {

var parent = e.target.parentElement;

var classList = parent.classList;

if(classList.contains("open")) {

classList.remove('open');

var opensubs = parent.querySelectorAll(':scope .open');

for(var i = 0; i < opensubs.length; i++){

opensubs[i].classList.remove('open');

}

} else {

classList.add('open');

}

e.preventDefault();

});

}

</script>

</body>

</html>

Using JSON to represent a tree:  
<https://stackoverflow.com/questions/37150668/json-schema-for-tree-structure>

{

"Id": 1,

"Label": "A",

"Child": [

{

"Id": 2,

"Label": "B",

"Child": [

{

"Id": 5,

"Label": "E"

}, {

"Id": 6,

"Label": "E"

}, {

"Id": 7,

"Label": "E"

}

]

}, {

"Id": 3,

"Label": "C"

}, {

"Id": 4,

"Label": "D",

"Child": [

{

"Id": 8,

"Label": "H"

}, {

"Id": 9,

"Label": "I"

}

]

}

]

}

Schema for this data with the following requirements: Each node has

* an id (an integer, required),
* a label (a string, optional), and
* an array of child nodes (optional).

{

"$schema": "http://json-schema.org/draft-04/schema#",

"$ref": "#/definitions/node",

"definitions": {

"node": {

"properties": {

"Id": {

"type": "integer"

},

"Label": {

"type": "string"

},

"Children": {

"type": "array",

"items": {

"$ref": "#/definitions/node"

}

}

},

"required": [

"Id"

]

}

}

}

JSON for the tea data