

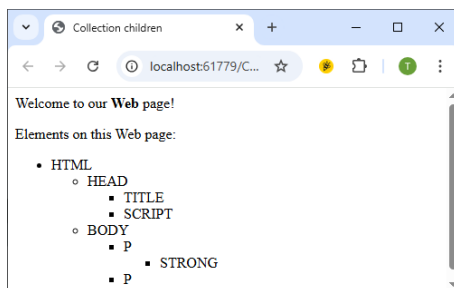
7.3 Collection children

```
1 <!-- Children.html -->
2 <html>
3   <head>
4     <title>Collection children</title>
5     <script>
6       elements = "<ul>"
7       function child(object)           // Rekursiv //
8       {                               // funktion //
9         elements += "<li>" + object.tagName + "<ul>"
10        for (var i=0; i < object.children.length; i++)
11        {
12          if (object.children[i].children.length)
13            child(object.children[i]) // Anropar //
14          else                               // sig själv //
15            elements += "<li>" +
16              object.children[i].tagName + "</li>"
17        }
18        elements += "</ul>"
19      }
20    </script>
21  </head>
22  <body onload = "child(document.all[0])
23    myDisplay.outerHTML += elements"&>
24    <p>Welcome to our <strong>Web</strong> page!</p>
25    <p id = "myDisplay">Elements on this Web page:</p>
26  </body>
27 </html>
```

Element och barnelement

Collection **children** innehåller för ett specifikt element elementets barnelement. T.ex. har elementet **html** två barnelement: **head** och **body**. Scriptet ovan använder collection **children** för att gå igenom alla element samt barnelement, i den ordning de förekommer i dokumentet.

Det som skiljer scriptet **Children** från scriptet **All** (sid 134) är att funktionen **child()** som är definierad på rad 7, anropar



sig själv på rad **13**. Detta är ett programmeringstekniskt koncept som kallas för *rekursion* och är ett alternativ till loopar.

Funktionen `child()`

Funktionen `child()` använder rekursion för att samla alla element och deras barn som finns i scriptet. Den börjar med elementet `html` på rad **2** med koden `document.all[0]` på rad **22** och går igenom alla barnelement av `html`-elementet. Om den träffar på ett element som har sina egna barn (rad **12**) anropar den rekursivt funktionen `child()` genom att skicka som parameter det nya elementets objekt som funktionen loopar. När denna loop slutar, går den anropade funktionen till nästa varv. Attributet `tagName` används för att samla namnen på de taggar som hittats när man loopade genom dokumentet. De placeras i strängen `elements`. Scriptet använder `ul`- och `li`-taggarna för att skriva ut dem i en korrekt hierarkisk struktur.

När funktionen `child()`:s ursprungliga anrop är avslutat (rad **22**), byter rad **23** `outerHTML`-attributet till `p`-elementet `myDisplay` till strängen `elements`. `outerHTML`-attributet liknar `innerHTML`-attributet som introducerades tidigare. Skillnaden är att `outerHTML` även inkluderar de omgivande start- och sluttaggarna samt innehållet, i det här fallet tagarna `<p id = "myDisplay">` och `</p>`.