Introduction to the Document Object Model

How a Browser Renders a Web Page

- **HTML is “served” to the browser**
  - Typically, the response from an HTTP request, triggered by the URL bar, link or other programatic navigation
  - Sometimes, simply opening an ‘.html’ file on your computer

How a Browser Renders a Webpage

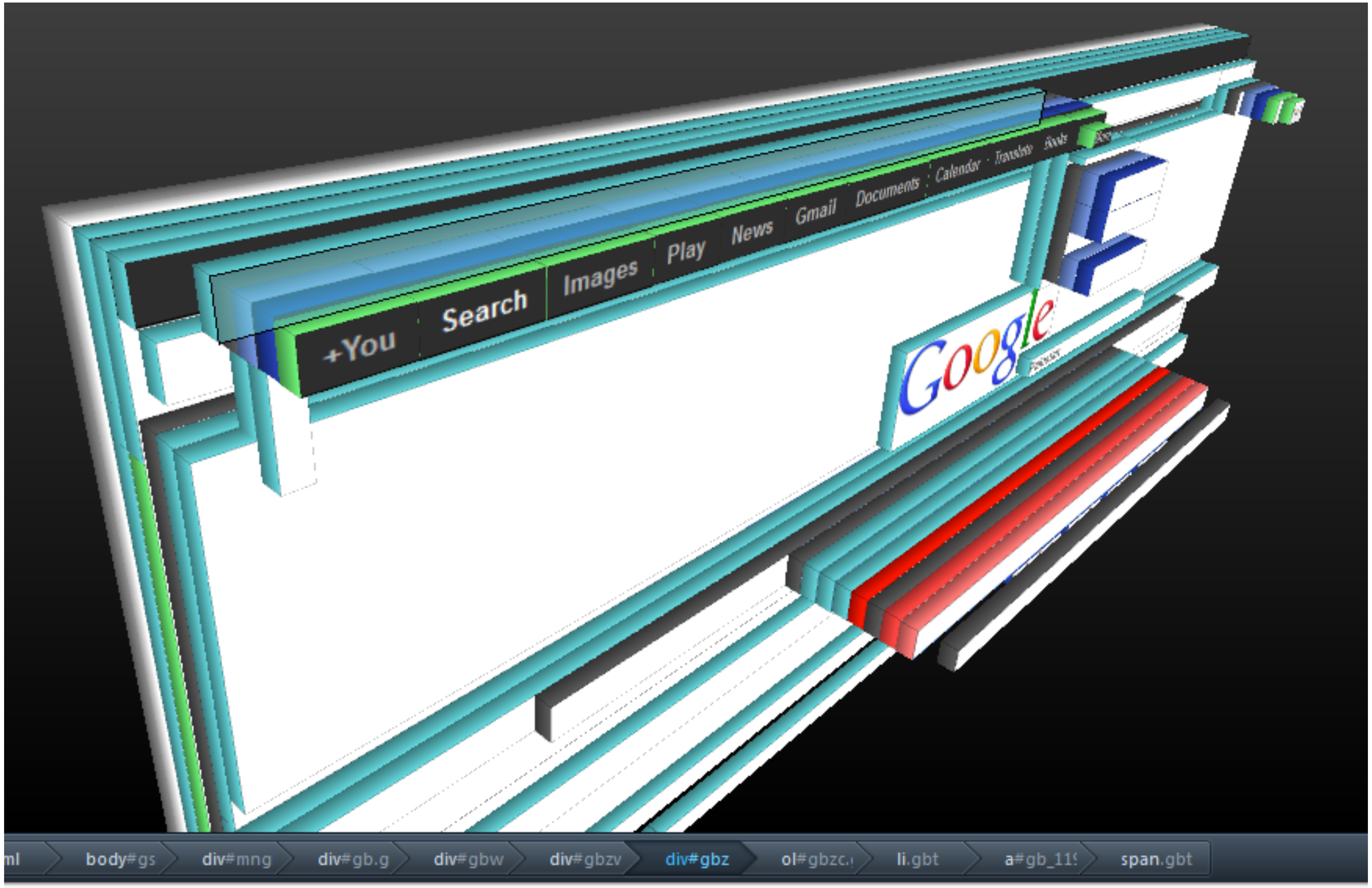
- **HTML is “served” to the browser**
  - Typically, the response from an HTTP request, triggered by the URL bar, link or other programatic navigation
  - Sometimes, simply opening an ‘.html’ file on your computer
- **The browser deserializes HTML (text) into an information structure (connected objects)**

How a Browser Renders a Webpage

- HTML is “served” to the browser
  - Typically, the response from an HTTP request, triggered by the URL bar, link or other programatic navigation
  - Sometimes, simply opening an ‘.html’ file on your computer
- The browser deserializes HTML (text) into an information structure (connected objects)
- Using these connected objects, which have dynamic properties, the browser “paints” a visualization

How a Browser Renders a Webpage

- **HTML is “served” to the browser**
  - Typically, the response from an HTTP request, triggered by the URL bar, link or other programatic navigation
  - Sometimes, simply opening an ‘.html’ file on your computer
- **The browser deserializes HTML (text) into an information structure (connected objects)**
- **Using these connected objects, which have dynamic properties, the browser “paints” a visualization**
- **The structure of connected objects is what is known as the Document Object Model**


+You
Search
Images
Play
News
Gmail
Documents
Calendar
Translate
Books
Google
ml
body#gs
div#mng
div#gb.g
div#gbw
div#gbzv
div#gbz
ol#gbzc.
li.gbt
a#gb_11
span.gbt

Why study the DOM?

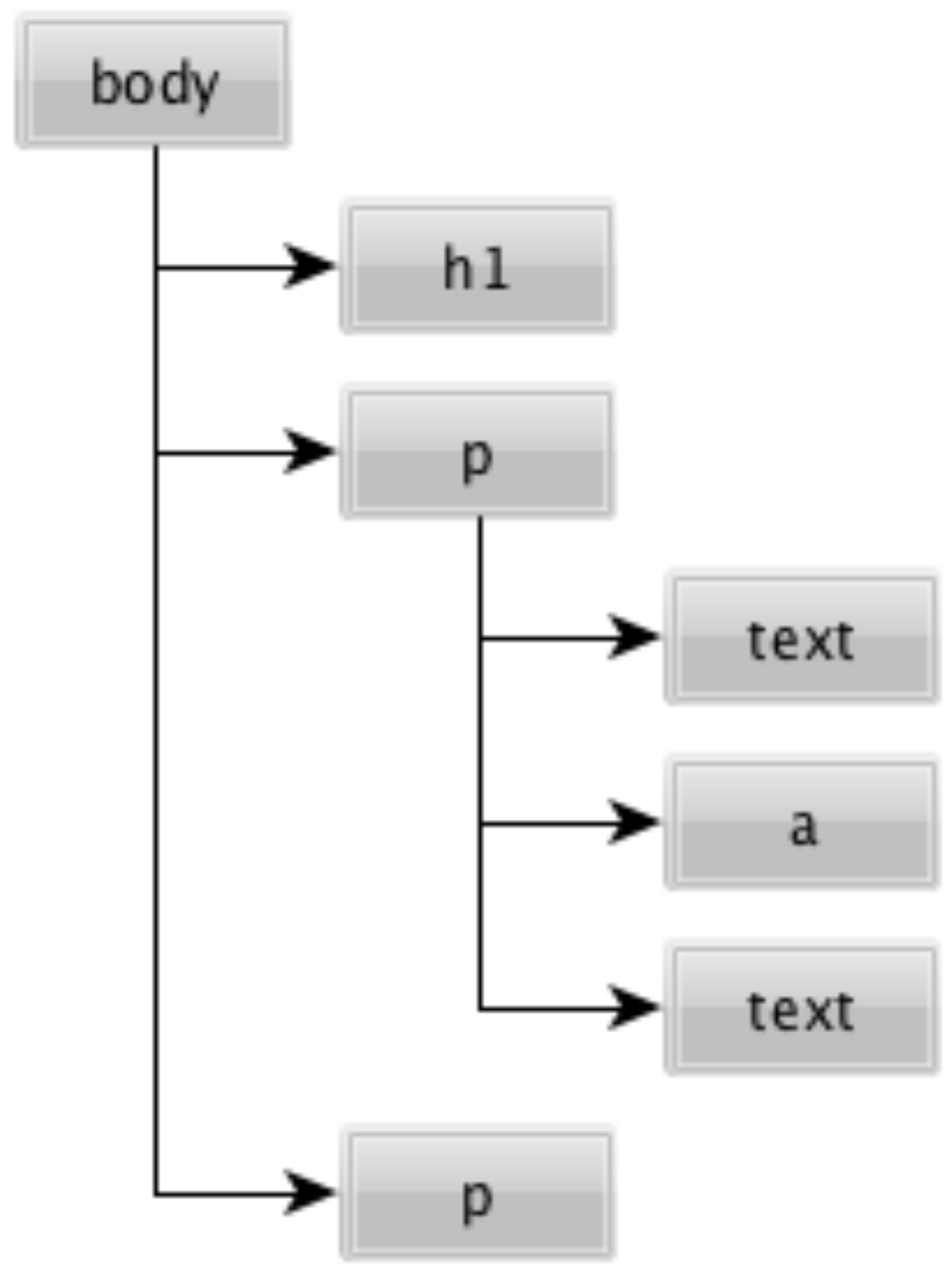
- **The Document Object Model is:**
  - The most powerful publishing platform ever created
  - What allows web pages to render, respond to user events and change
  - **Connects JavaScript to HTML**

The *document* Object

- Global reference to the HTML document
- Provides methods for:
  - Navigating the DOM
  - Manipulating the DOM
- The *document* object is the important connection between the DOM and JavaScript code

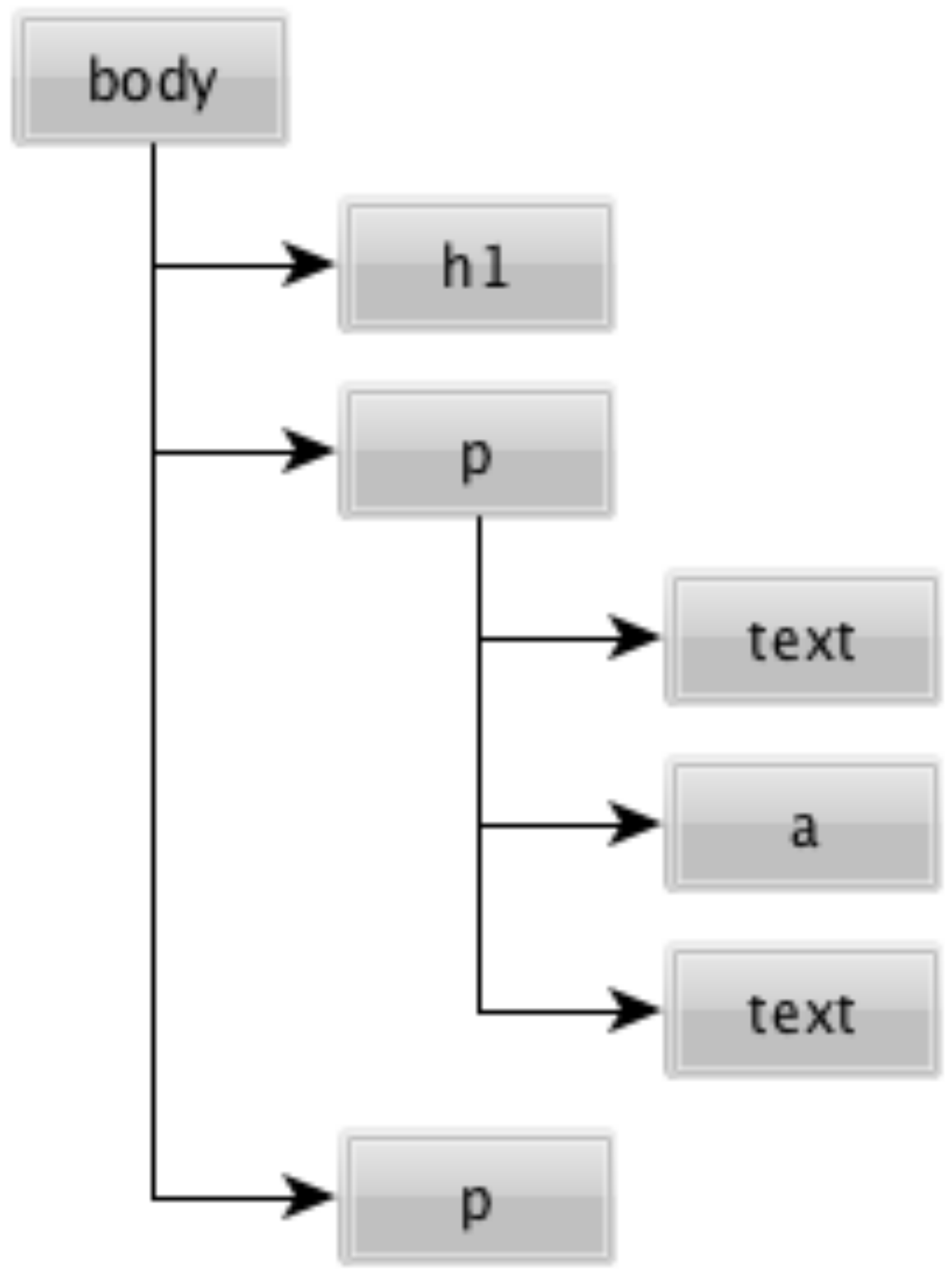
The DOM is a Tree

- Trees are an ubiquitous data structure
- The main idea here: There is a Node that branches into other Nodes (its children Nodes)
  - Each Node can have 0 to many children Nodes
  - Nodes can have 0 or 1 parent
  - Nodes can have 0 to many Sibling Nodes



The DOM is a Tree

```
<body>

   <h1>Hello</h1>

   <p>

      Check out my

      <a href="/page">Page!</a>

      It's the best page out there
   </p>



   <p>Come back soon!</p>
</body>
```



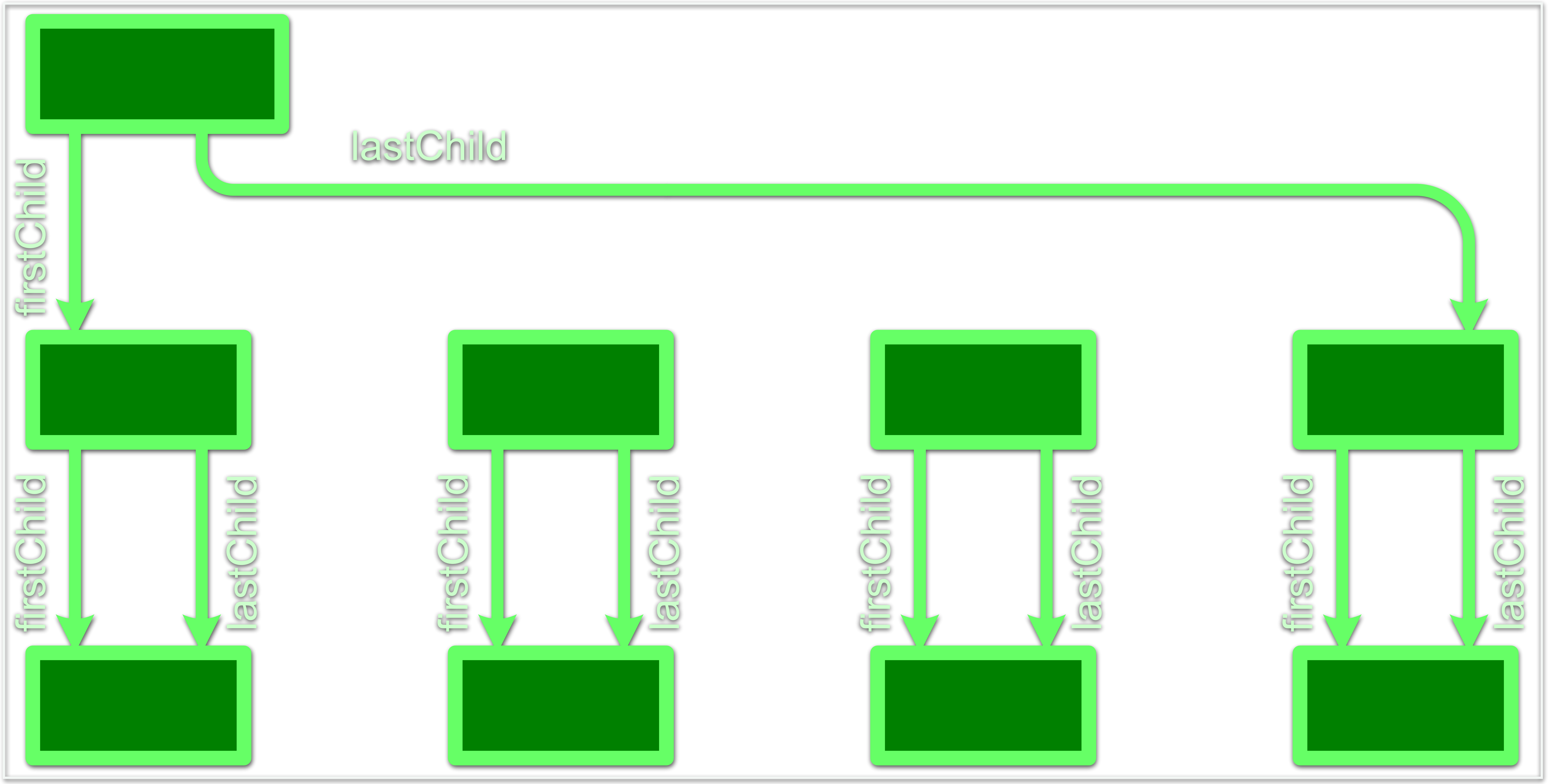
Indentation Is Important!

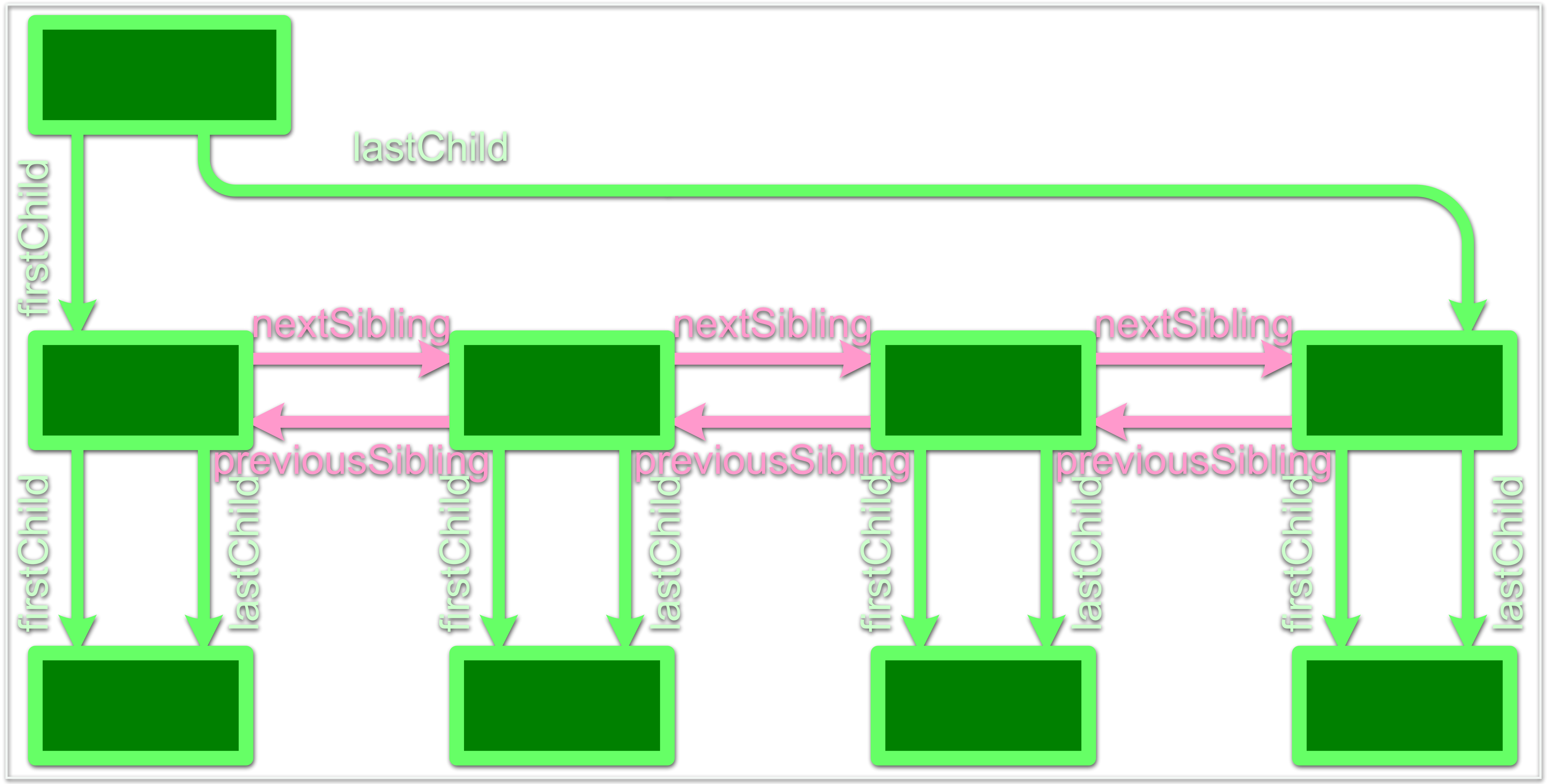
- No indentation makes it hard to see the tree structure:

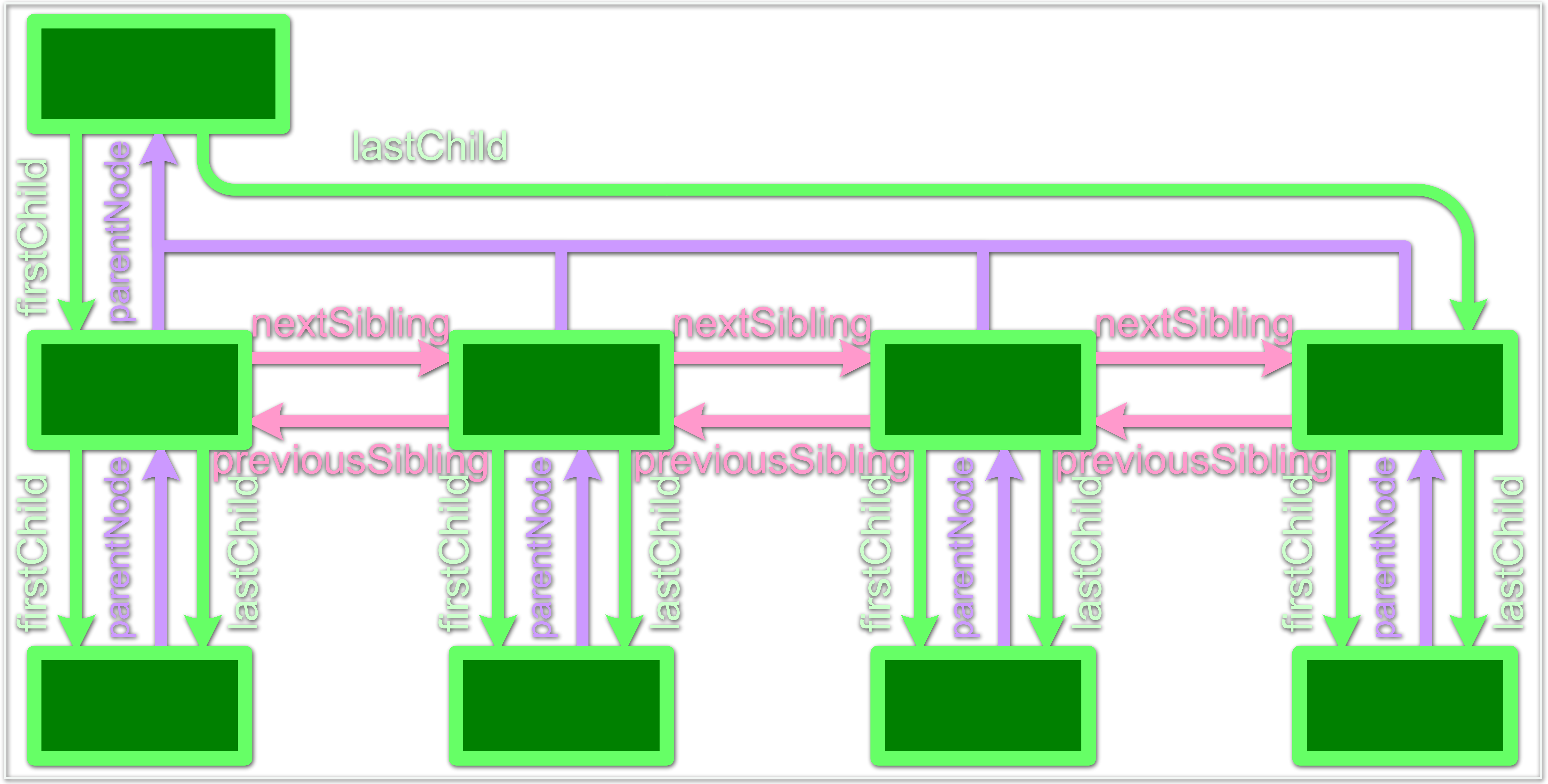
```
<body>
<h1>Hello</h1>
<p>
Check out my
<a href="/page">Page!</a>
It's the best page out there
</p>
<p>Come back soon!</p>
</body>
```

Tree Structures are easy to navigate

- At any point in the DOM you are at a Node
- No matter where you go, you're still at a Node
  - Child
  - Parent
  - Sibling
  - All return Nodes
- All Nodes share similar DOM navigation methods


lastChild
firstChild
firstChild
lastChild
firstChild
lastChild
firstChild
lastChild
firstChild
lastChild

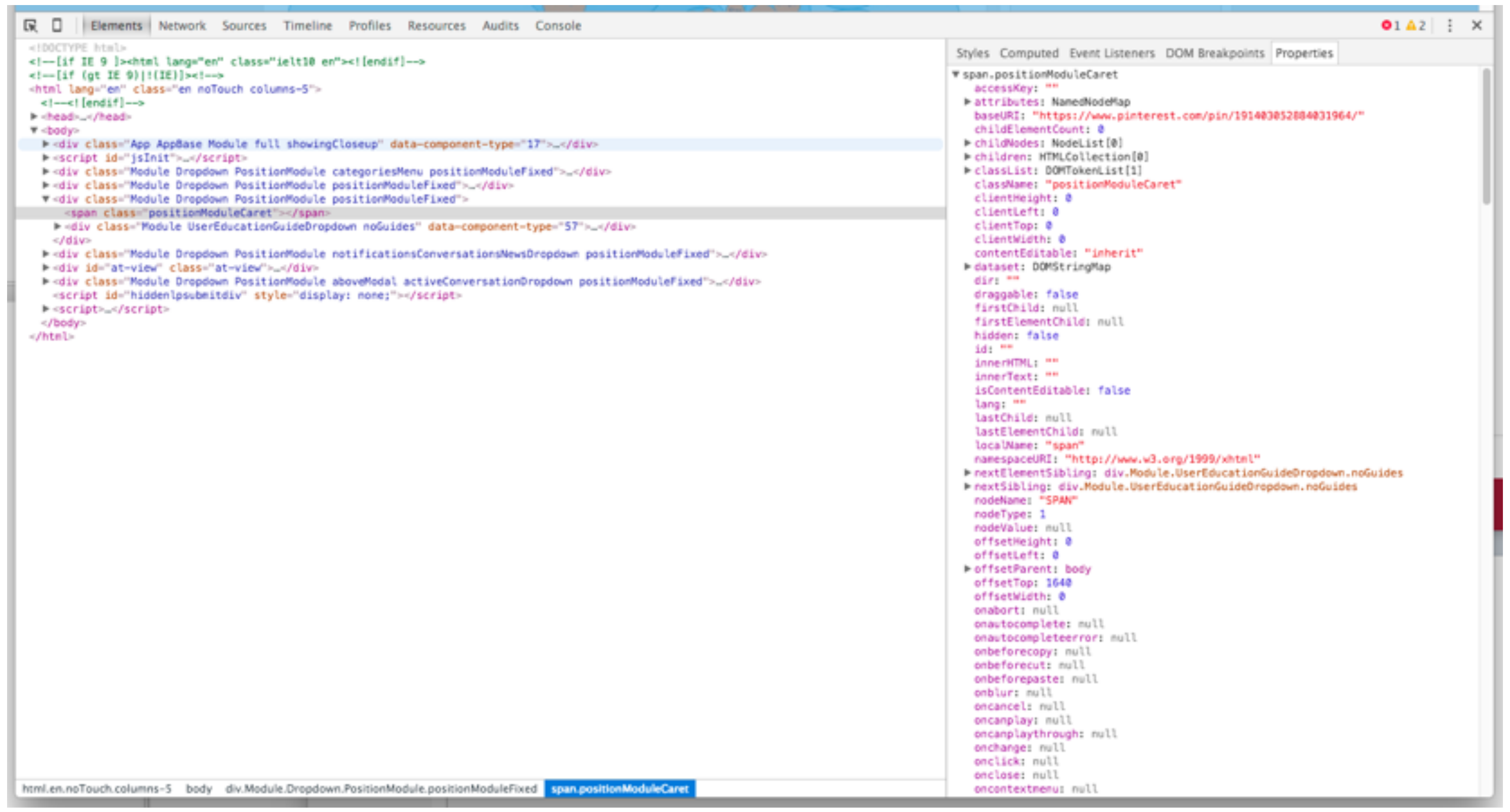

lastChild
firstChild
nextSibling
nextSibling
nextSibling
previousSibling
previousSibling
previousSibling
firstChild
lastChild
firstChild
lastChild
firstChild
lastChild
firstChild
lastChild


lastChild
firstChild
parentNode
nextSibling
nextSibling
nextSibling
previousSibling
previousSibling
previousSibling
firstChild
parentNode
lastChild
firstChild
parentNode
lastChild
firstChild
parentNode
lastChild
firstChild
parentNode
lastChild

Nodes have lots of Attributes

- Nodes are JavaScript Objects
- Nodes have Attributes that are JavaScript properties
- Attributes define how the Node looks and responds to User activity

one node



```
<!DOCTYPE html>
<!--[if IE 9 ]><html lang="en" class="ielt10 en"><![endif]-->
<!--[if (gt IE 9)|!(IE)]><!-->
<html lang="en" class="en noTouch columns-5">
  <!--<![endif]-->
▶ <head>…</head>
▼ <body>
  ▶ <div class="App AppBase Module full showingCloseup" data-component-type="17">…</div>
  ▶ <script id="jsInit">…</script>
  ▶ <div class="Module Dropdown PositionModule categoriesMenu positionModuleFixed">…</div>
  ▶ <div class="Module Dropdown PositionModule positionModuleFixed">…</div>
  ▼ <div class="Module Dropdown PositionModule positionModuleFixed">
      <span class="positionModuleCaret"></span>
    ▶ <div class="Module UserEducationGuideDropdown noGuides" data-component-type="57">…</div>
    </div>
  ▶ <div class="Module Dropdown PositionModule notificationsConversationsNewsDropdown positionModuleFixed">…</div>
  ▶ <div id="at-view" class="at-view">…</div>
  ▶ <div class="Module Dropdown PositionModule aboveModal activeConversationDropdown positionModuleFixed">…</div>
    <script id="hiddenlpsubmitdiv" style="display: none;"></script>
  ▶ <script>…</script>
  </body>
</html>
```

```
▼ span.positionModuleCaret
    accessKey: ""
  ▶ attributes: NamedNodeMap
    baseURI: "https://www.pinterest.com/pin/191403052884031964/"
    childElementCount: 0
  ▶ childNodes: NodeList[0]
  ▶ children: HTMLCollection[0]
  ▶ classList: DOMTokenList[1]
    className: "positionModuleCaret"
    clientHeight: 0
    clientLeft: 0
    clientTop: 0
    clientWidth: 0
    contentEditable: "inherit"
  ▶ dataset: DOMStringMap
    dir: ""
    draggable: false
    firstChild: null
    firstElementChild: null
    hidden: false
    id: ""
    innerHTML: ""
    innerText: ""
    isContentEditable: false
    lang: ""
    lastChild: null
    lastElementChild: null
    localName: "span"
    namespaceURI: "http://www.w3.org/1999/xhtml"
  ▶ nextElementSibling: div.Module.UserEducationGuideDropdown.noGuides
  ▶ nextSibling: div.Module.UserEducationGuideDropdown.noGuides
    nodeName: "SPAN"
    nodeType: 1
    nodeValue: null
    offsetHeight: 0
    offsetLeft: 0
  ▶ offsetParent: body
    offsetTop: 1640
    offsetWidth: 0
    onabort: null
    onautocomplete: null
    onautocompleteerror: null
    onbeforecopy: null
    onbeforecut: null
    onbeforepaste: null
    onblur: null
    oncancel: null
    oncanplay: null
    oncanplaythrough: null
    onchange: null
    onclick: null
    onclose: null
    oncontextmenu: null
```

html.en.noTouch.columns-5   body   div.Module.Dropdown.PositionModule.positionModuleFixed   span.positionModuleCaret

hundreds of properties!

Navigating the DOM

- Searching the DOM
  - getElementById (find nodes with a certain ID attribute)
    - `document.getElementById(“will”);`
  - getElementsByClassName (find nodes with a certain CLASS ATTRIBUTE)
    - `document.getElementByClassName(“will”);`
  - getElementsByTagName (find nodes with a certain HTML tag)
    - `document.getElementByTagName(“div”);`

Traversing the DOM

- Access children
  - `element.children, element.lastChild, element.firstChild`
- Access siblings
  - `element.nextElementSibling, element.previousElementSibling`
- Access parent
  - `element.parentElement`

```
document.querySelector(“p.news”);
```