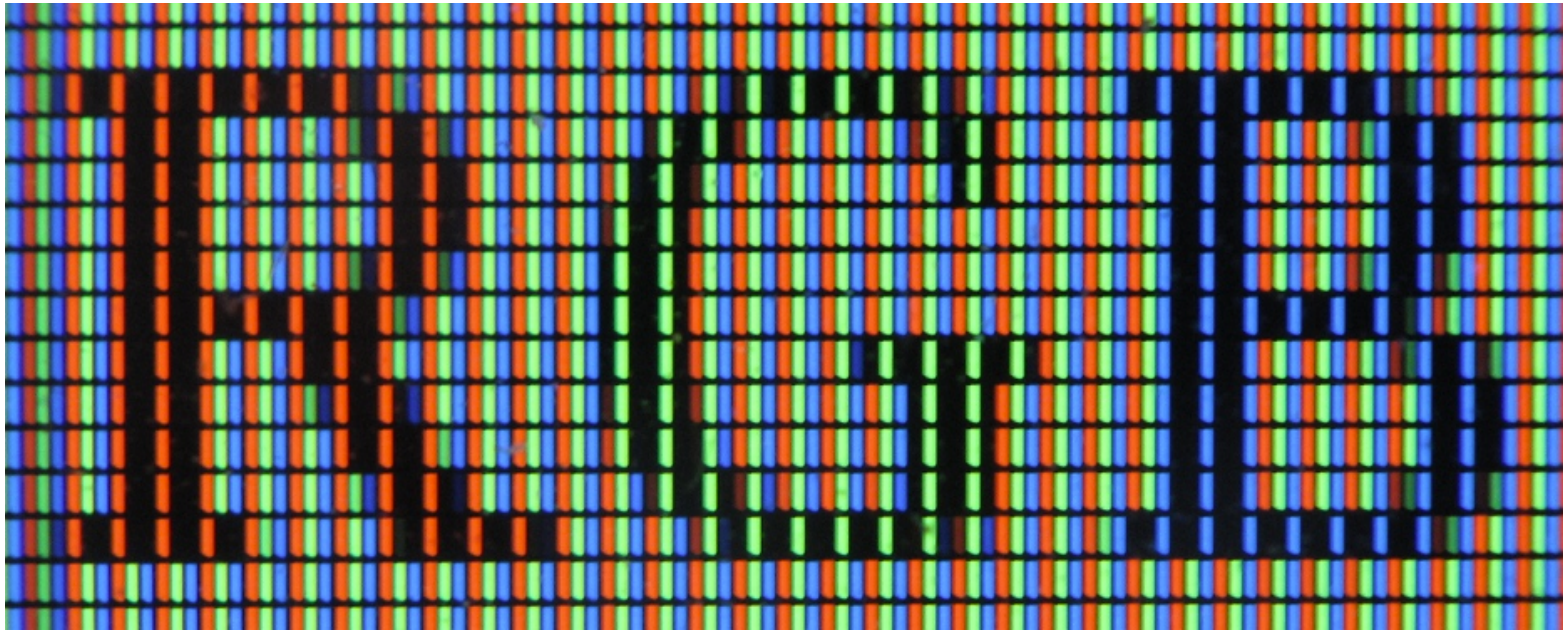
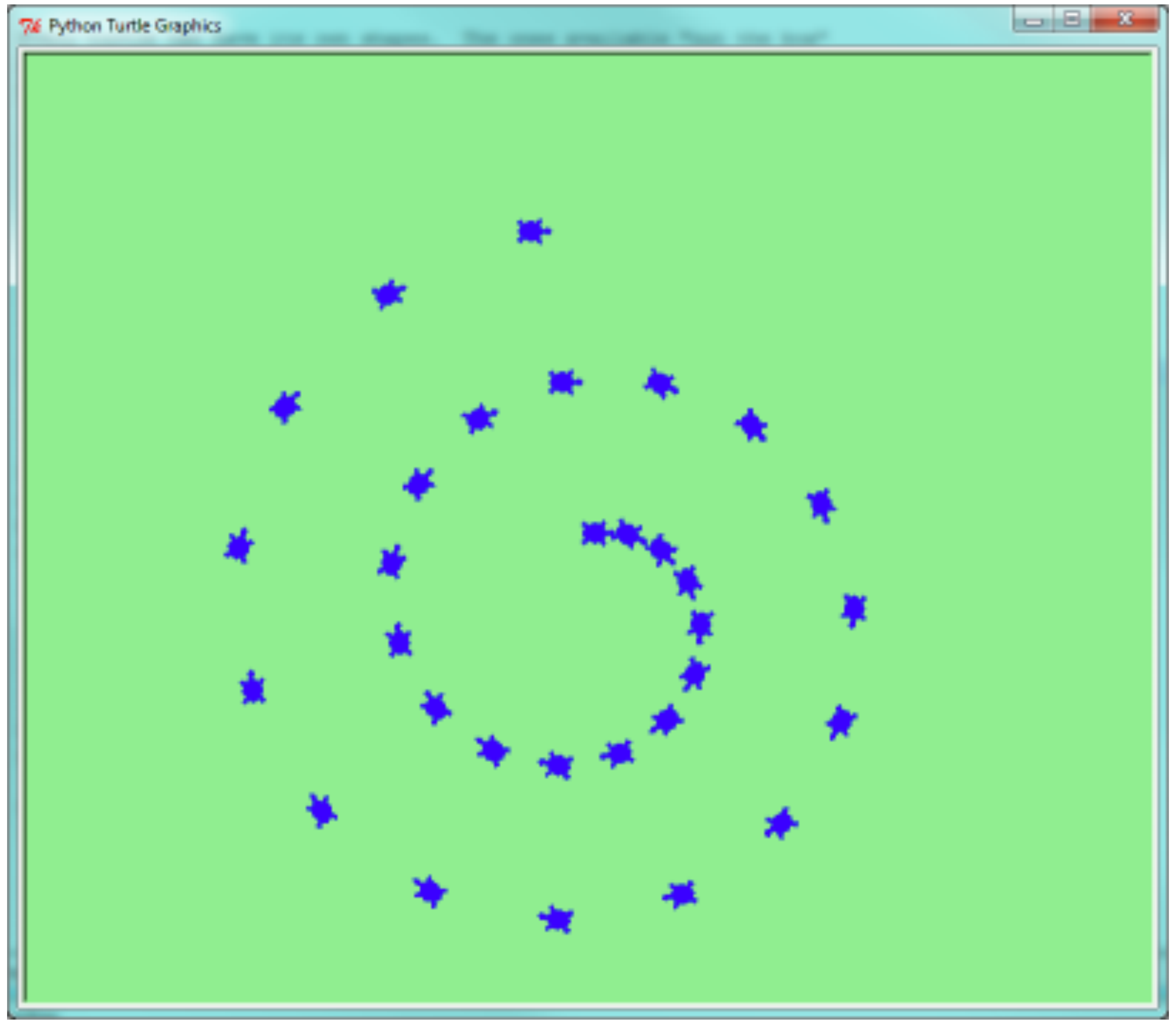
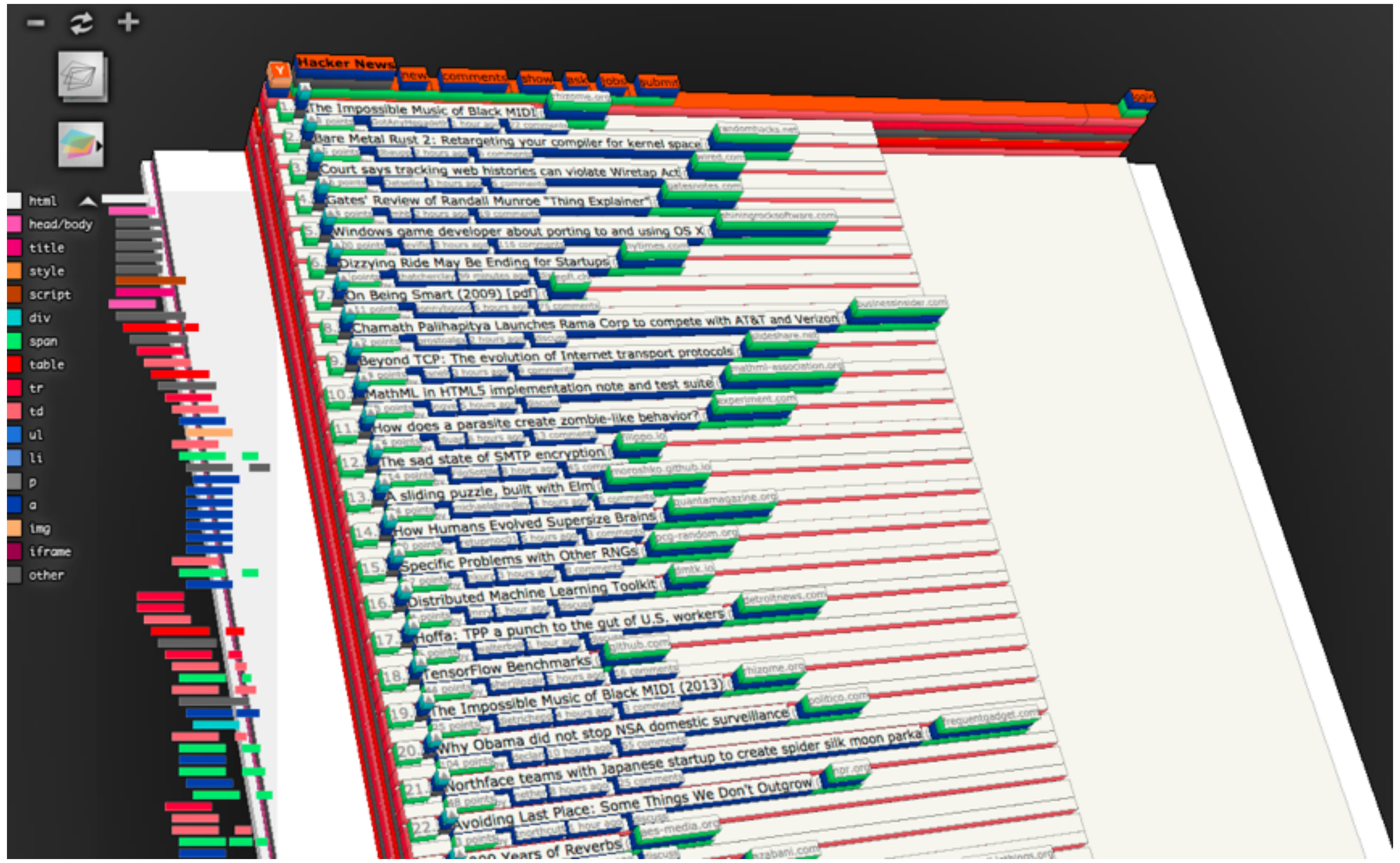
Introduction to the Document Object Model




Python Turtle Graphics

```
<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>
```



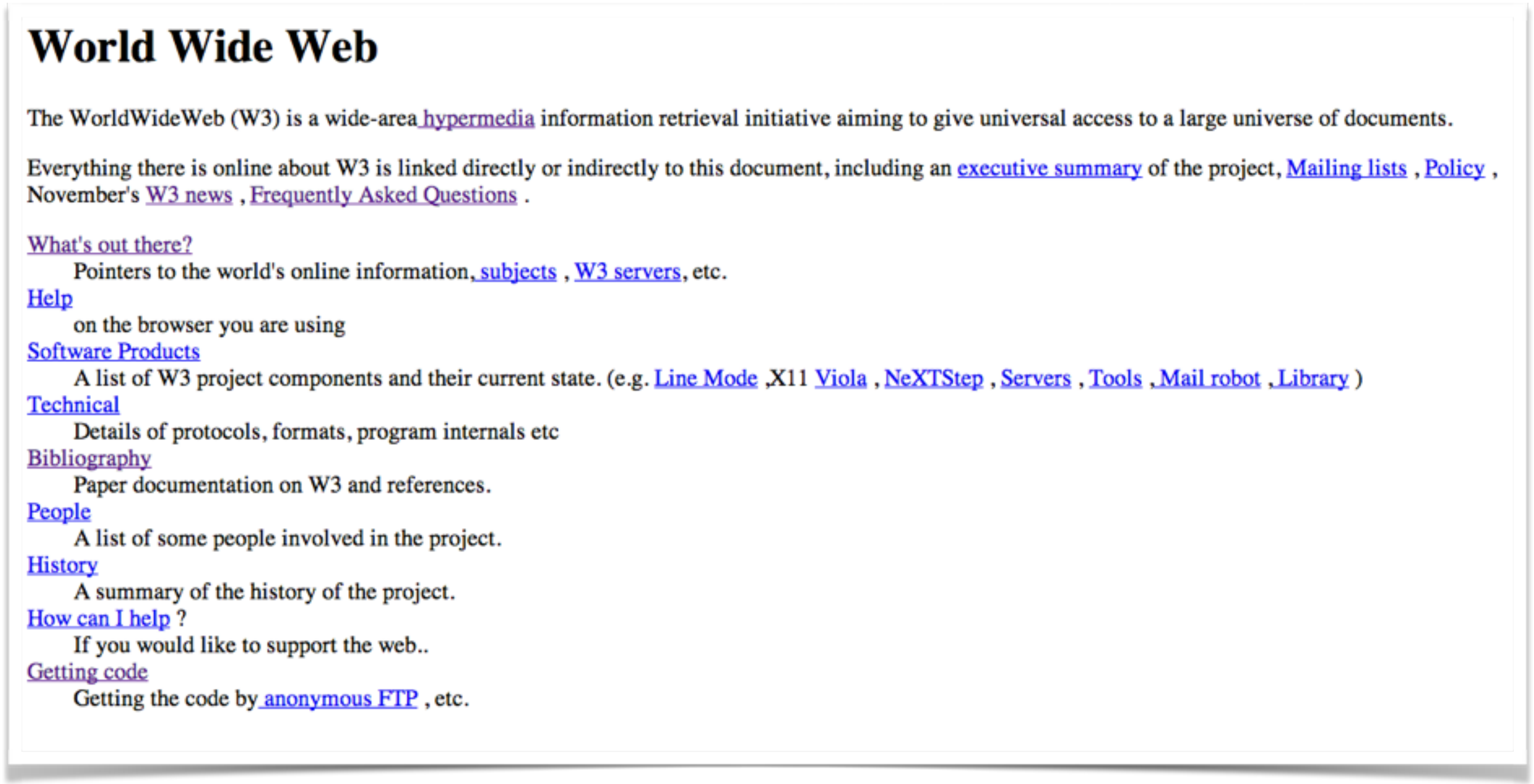
Hacker News
new
comments
show
ask
jobs
submit
The Impossible Music of Black MIDI
Bare Metal Rust 2: Retargeting your compiler for kernel space
Court says tracking web histories can violate Wiretap Act
Gates' Review of Randall Munroe "Thing Explainer"
Windows game developer about porting to and using OS X
Dizzying Ride May Be Ending for Startups
On Being Smart (2009) [pdf]
Chamath Palihapitiya Launches Rama Corp to compete with AT&T and Verizon
Beyond TCP: The evolution of Internet transport protocols
MathML in HTML5 implementation note and test suite
How does a parasite create zombie-like behavior?
The sad state of SMTP encryption
A sliding puzzle, built with Elm
How Humans Evolved Supersize Brains
Specific Problems with Other RNGs
Distributed Machine Learning Toolkit
Hoffa: TPP a punch to the gut of U.S. workers
TensorFlow Benchmarks
The Impossible Music of Black MIDI (2013)
Why Obama did not stop NSA domestic surveillance
Northface teams with Japanese startup to create spider silk moon parka
Avoiding Last Place: Some Things We Don't Outgrow
html
head/body
title
style
script
div
span
table
tr
td
ul
li
p
a
img
iframe
other

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 History of the DOM

- The original World Wide Web was a simple idea: document retrieval through hyperlinks between documents
- No concepts of:
  - User Interactivity
  - Sessions
  - Presentation (no CSS!)



World Wide Web

The WorldWideWeb (W3) is a wide-area hypermedia information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an executive summary of the project, Mailing lists , Policy , November's W3 news , Frequently Asked Questions .

What's out there?
    Pointers to the world's online information, subjects , W3 servers, etc.
Help
    on the browser you are using
Software Products
    A list of W3 project components and their current state. (e.g. Line Mode ,X11 Viola , NeXTStep , Servers , Tools , Mail robot , Library)
Technical
    Details of protocols, formats, program internals etc
Bibliography
    Paper documentation on W3 and references.
People
    A list of some people involved in the project.
History
    A summary of the history of the project.
How can I help ?
    If you would like to support the web..
Getting code
    Getting the code by anonymous FTP , etc.

A technically correct definition of the DOM

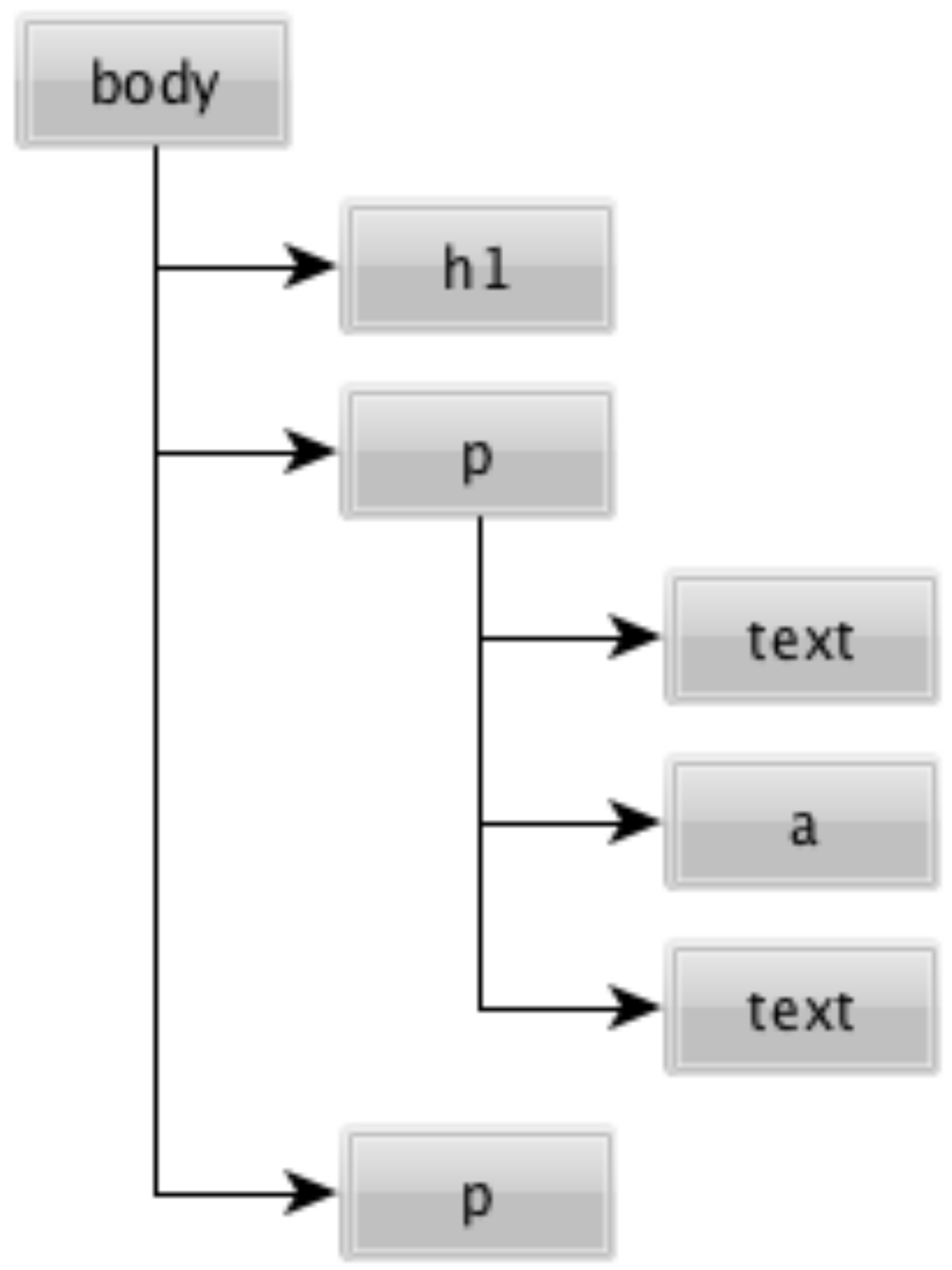
The Document Object Model (DOM) is a **cross-platform** and **language-independent** convention for representing and interacting with objects in **HTML, XHTML, and XML documents**.

The **nodes** of every **document** are organized in a **tree structure**, called the **DOM tree**. **Objects** in the DOM tree may be addressed and manipulated by using methods on the objects. The public interface of a DOM is specified in its application programming interface (API).

?

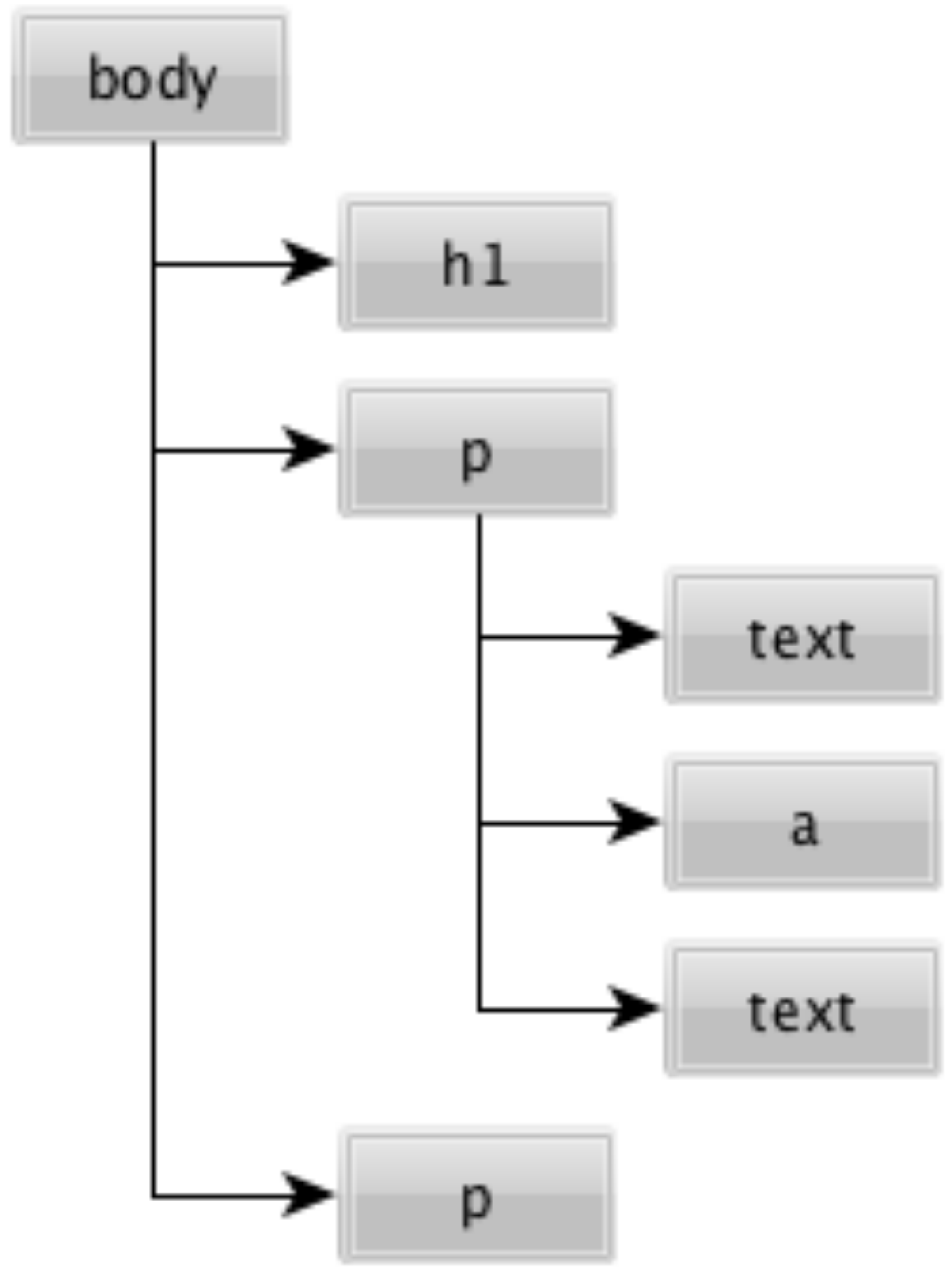
The DOM is a Tree

- Trees are a data structure from computer science
- 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>
```



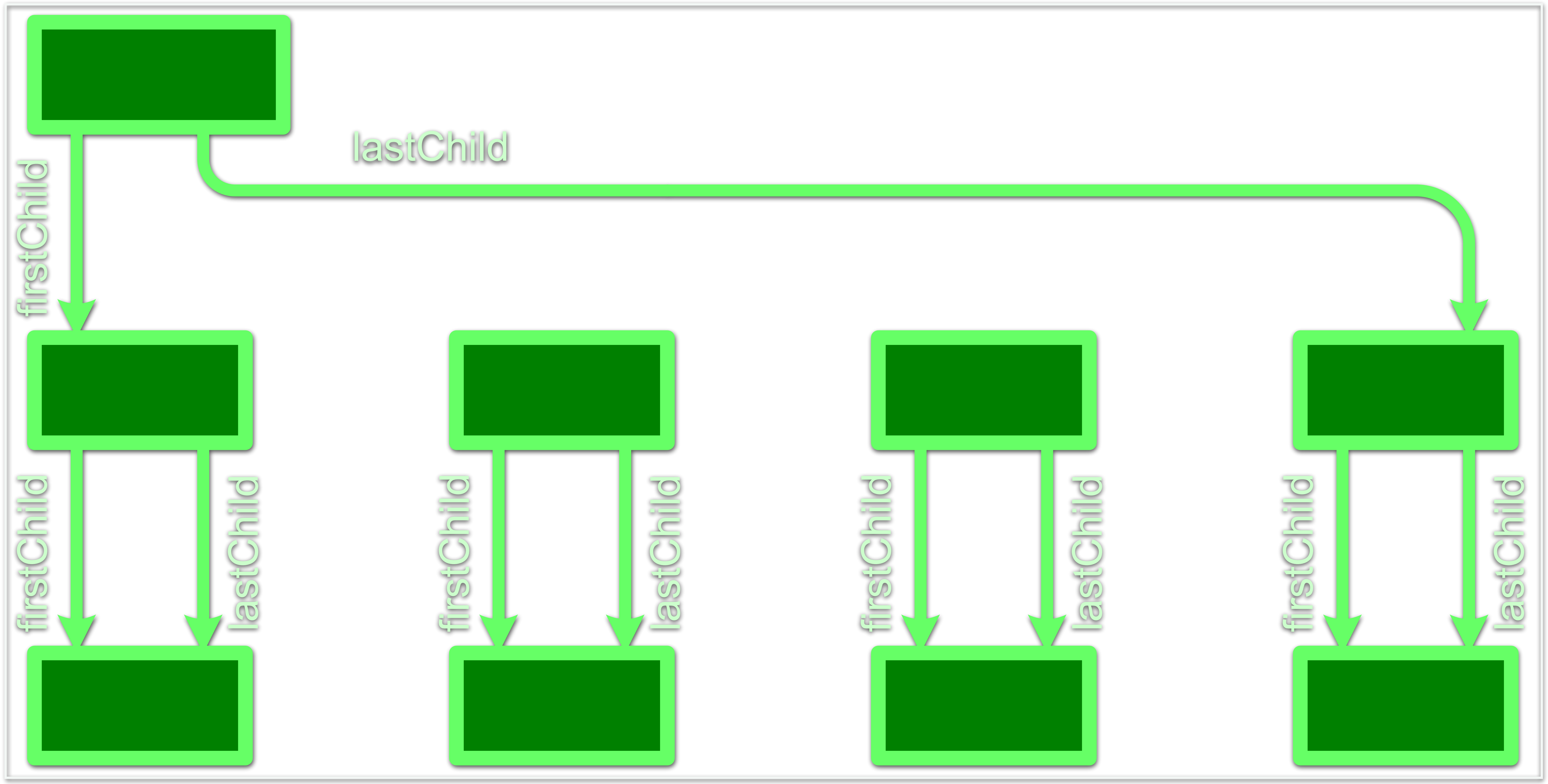
Indentation Is Important!

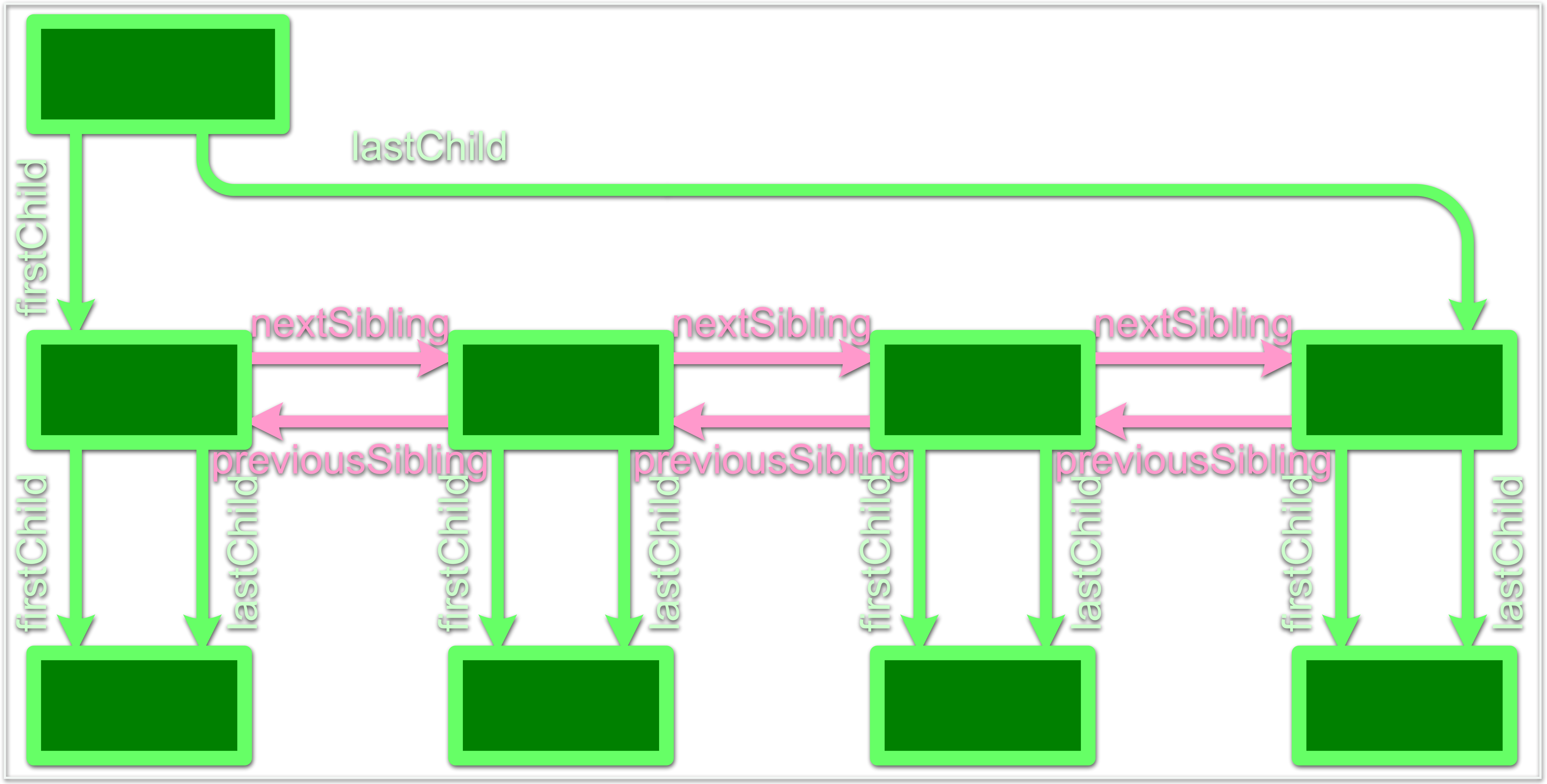
- No indentation makes it hard to see the tree structure:

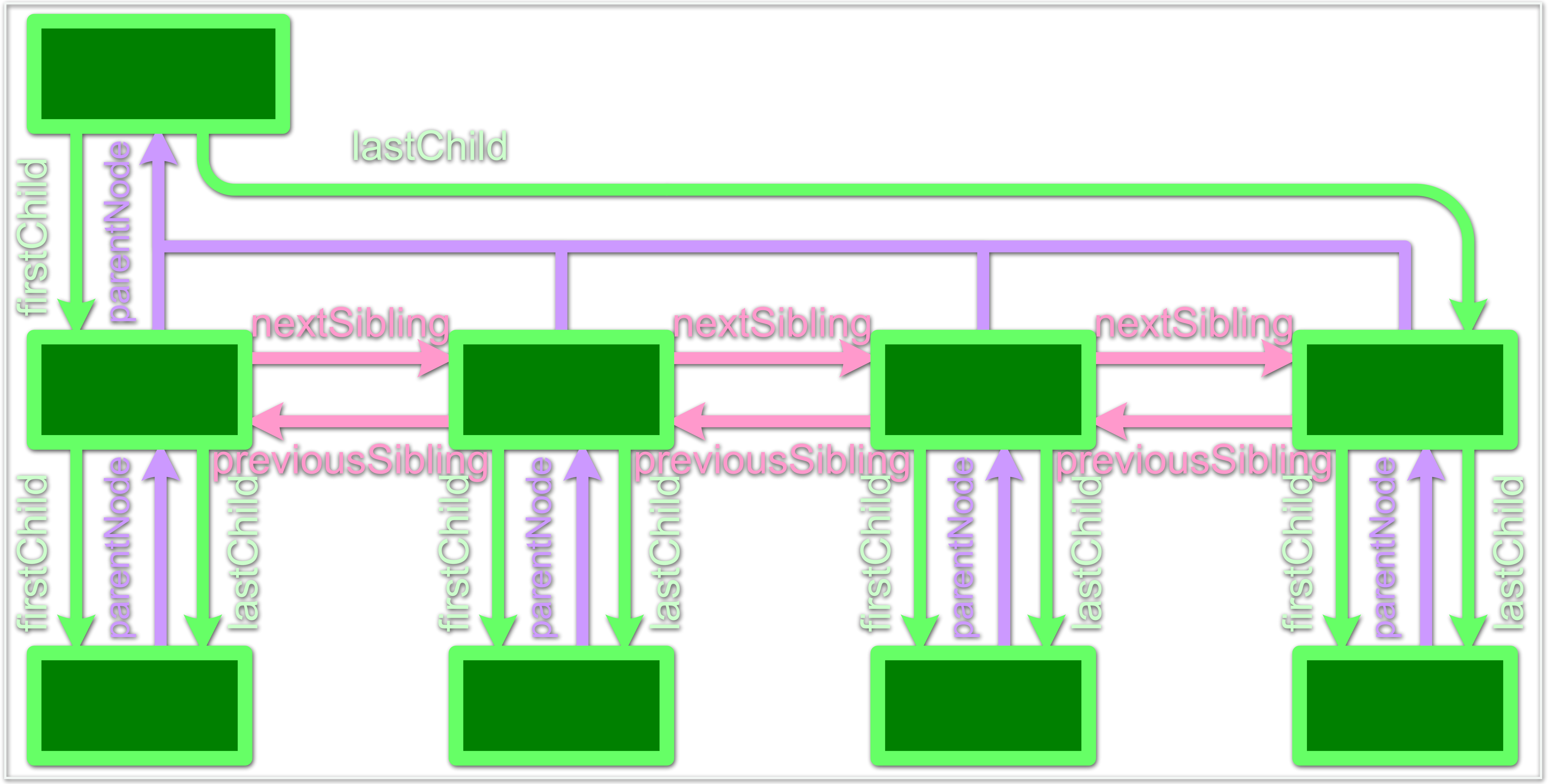
```
<body>
<h1>Hello</h1>
<p>
Check out my
<a href="/page">Page!</a>
It's the best page out there
</p>
<p>Come back soon!</p>
</body>
```

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

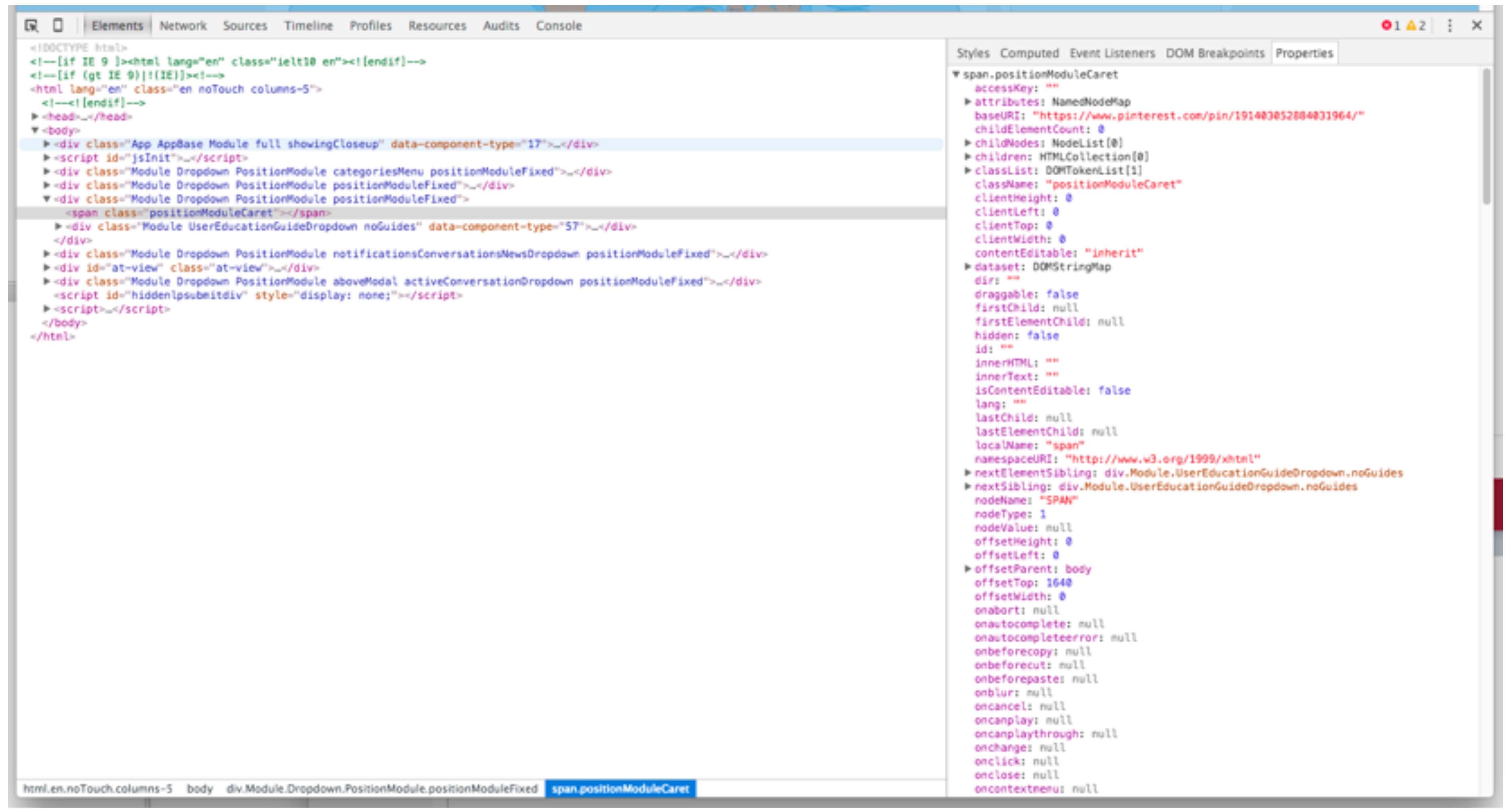

lastChild
firstChild
parentNode
nextSibling
nextSibling
nextSibling
previousSibling
previousSibling
previousSibling
firstChild
parentNode
lastChild
firstChild
parentNode
lastChild
firstChild
parentNode
lastChild
firstChild
parentNode
lastChild

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

hundreds of properties!



```
<!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>
```

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

```
▼ 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
```

JavaScript 💚 DOM

- **We have this Object Model of an HTML document, how do we manipulate it?**
  - <script> elements!
  - We can put <script> elements of JavaScript into our DOM that can interact with the DOM
- **How do you reference the DOM inside JavaScript?**
  - The *document* object

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**

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”);`
  - querySelector, querySelectorAll (search using CSS selectors)
    - `document.querySelector(“#will .will:first-child”);`

Traversing the DOM

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