Javascript

Sean Keenan

Educational Technology Specialist

Summer 2024 DSSF

What is JS?

- Javascript was created as a scripting language to add functionality to browsers.

How is Javascript different from HTML/CSS?

- HTML/CSS build up the **structure and style** of your code
- Javascript builds up the **functionality** of your code
- Together these three make up the basics of “Front-end” development

Scripting Language?

Javascript allows you to type up a list of instructions for the computer to carry out

You can...

- Make calculations
- Read and write files
- Send data over a network
- Access the functionality of other programs

Syntax?

...A little more complicated then with HTML/CSS

JS is one of the most used coding languages out there because of it's versatility - we will be learning a small corner of (vanilla) JS.

What we're learning is a subsection of JS as a starting point based on your HTML/CSS experience

Javascript

JS is one of the most used coding languages out there because of it's versatility - we will be learning a small corner of (vanilla) JS.

What we're learning is subsection of JS a starting point based on your HTML/CSS experience

Variables

What is a variable?

a fundamental building block of JS, a named container for some value. You can conceptualize it as a box with a label, which you will fill with a section of your code.

Variables

What is a variable?

a fundamental building block of JS, a named container for some value. You can conceptualize it as a box with a label, which you will fill with a section of your code.

Variables are like variables in math that are x and y

Variable names can be made up of numbers, letters, and $ or _ but cannot contain spaces or start with a number.

Always end a line with ;

Variables

You store a value in a variable with the assignment operator (=)

EXAMPLE:

VariableExample

=

30;

Variables

You declare a variable by putting a keyword in front of it (let or const; var is outdated)

const cannot be reassigned but **let** can

```
const VariableExample = 30;
VariableExample = 20;
```

❌

```
let VariableExample = 30;
VariableExample = 20;
```

✅

Variables

After value is assigned to a variable using the assignment operator , you can assign the value of that variable to another variable using the assignment operator

```
Const myVar;

myVar = 5;

Const myNum;

myNum = myVar;
```

Challenge Q:
What will the value of MyNum + 5 be?

Strings

Strings are characters inside single or double quotations

Example:

```
Let myFirstName = “Sean”;
Let myLastName=”Keenan”;
```

Strings are a type of Data Type, like in previous examples using Numbers. Boolean is another Data Type which defines a variable either as true or false

Functions

A way to write reusable code

Write the code in one place, use it elsewhere as many times as you like

Defining a function

Define (create) a function by using a function declaration

Example:

```
function sayHello ( ) {
…Some code here…
}
```

Functions

```
function sayHello ( ) {
  …Some code here…
            }
```

The parentheses ( ) provides the parameter list

The parameter list ( ) is used to pass values into the function, if needed

If no outside values are needed in the function declaration, it stays empty.

Functions

Defining a function does not call it

You will need more then the Function Declaration to get the code to run

EXAMPLE FUNCTION CALL

```
sayHello( );
```

So to wrap up what this looks like:

Function Declaration

```
function sayHello ( ) {
…. some code…

 }
```

Function Call

```
sayHello( );
```

Functions

Example function

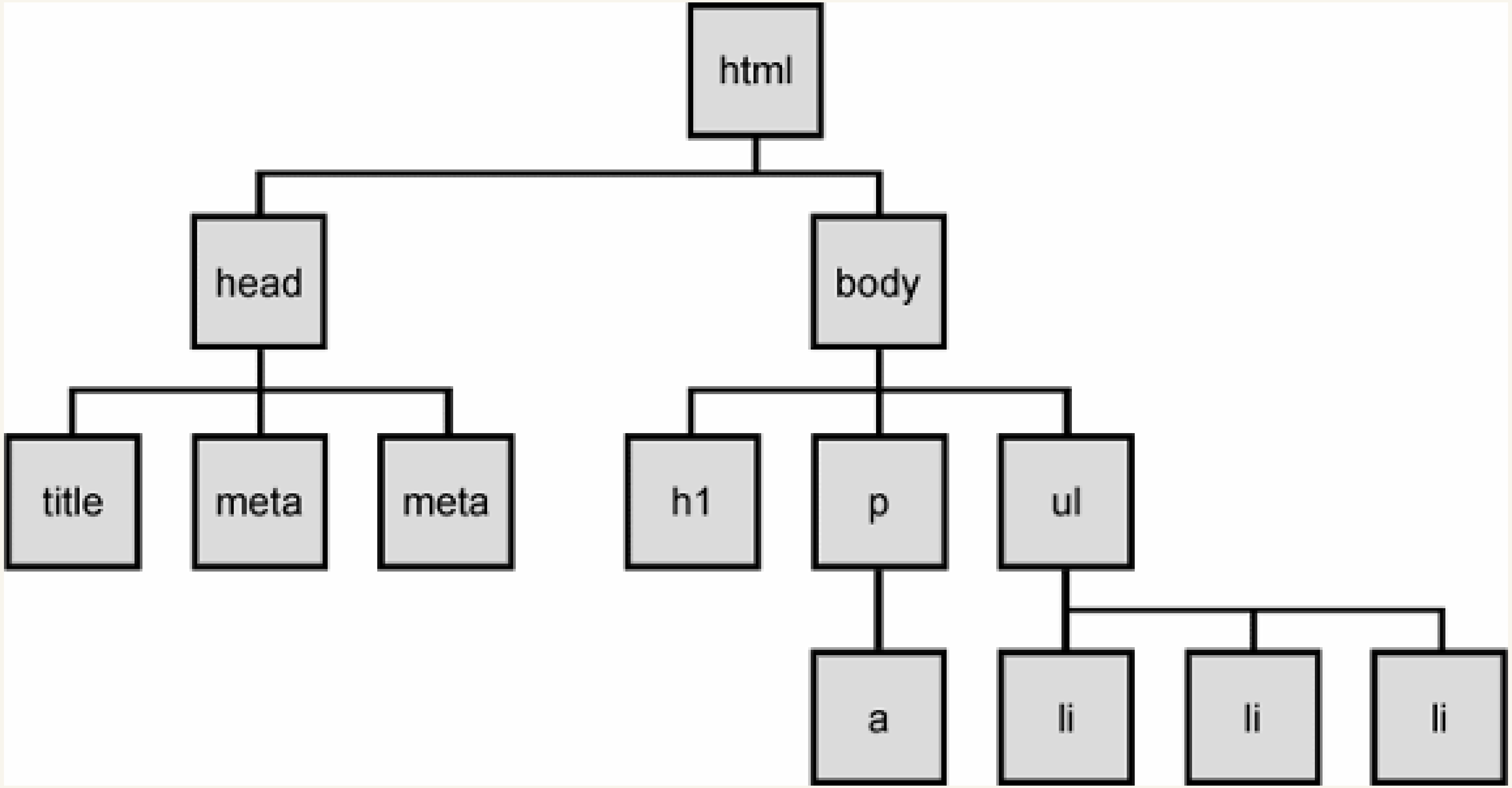
```
Function getArea (width, length) {
Return width * length;
}

getArea(3,4);
```

The return statement ends function execution and specifies a value to be returned to the function caller.

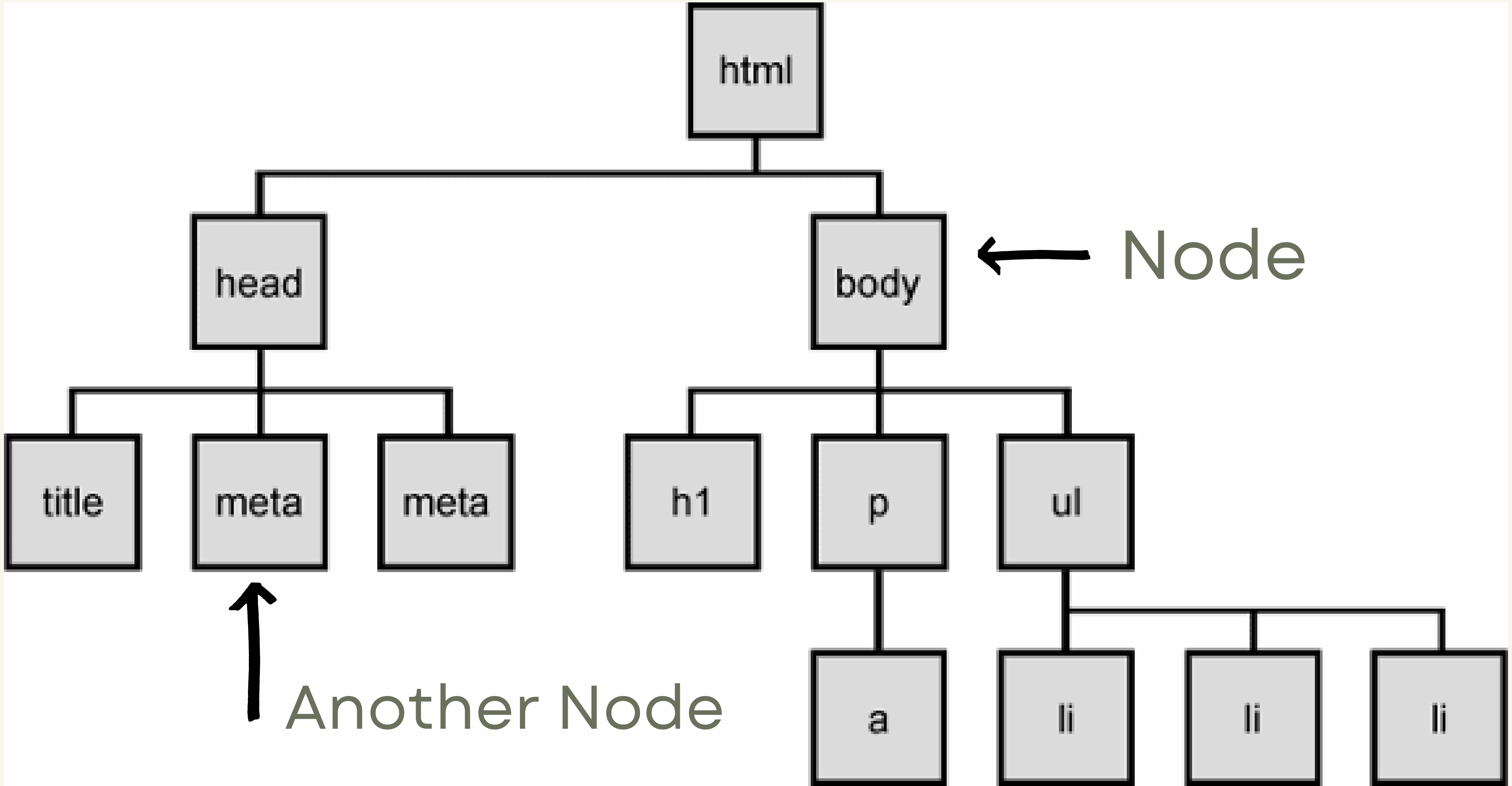
DOM Manipulation

Document Object Model



Nodes

Each of the points on the tree graph are called nodes

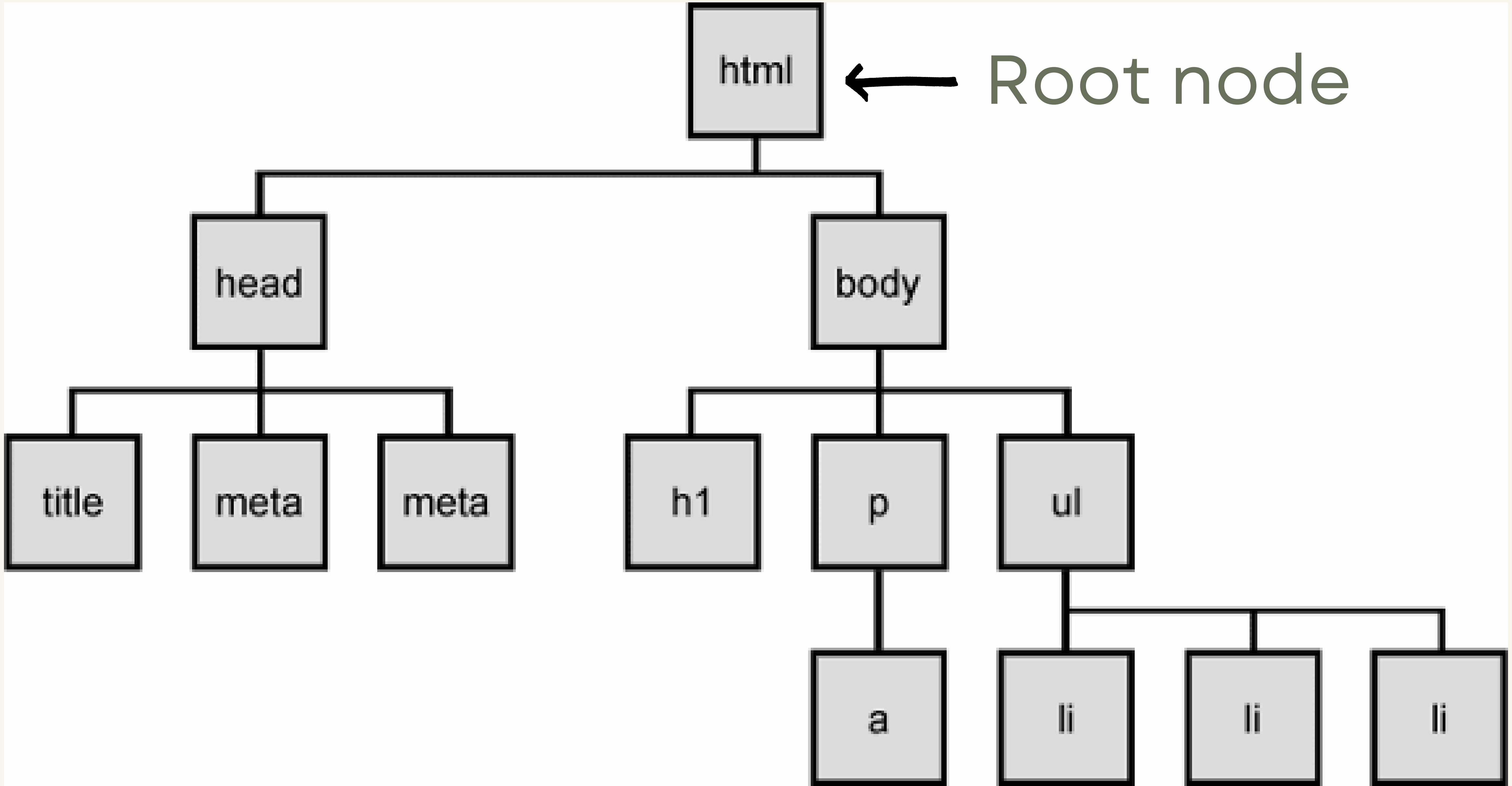


Node

Another Node

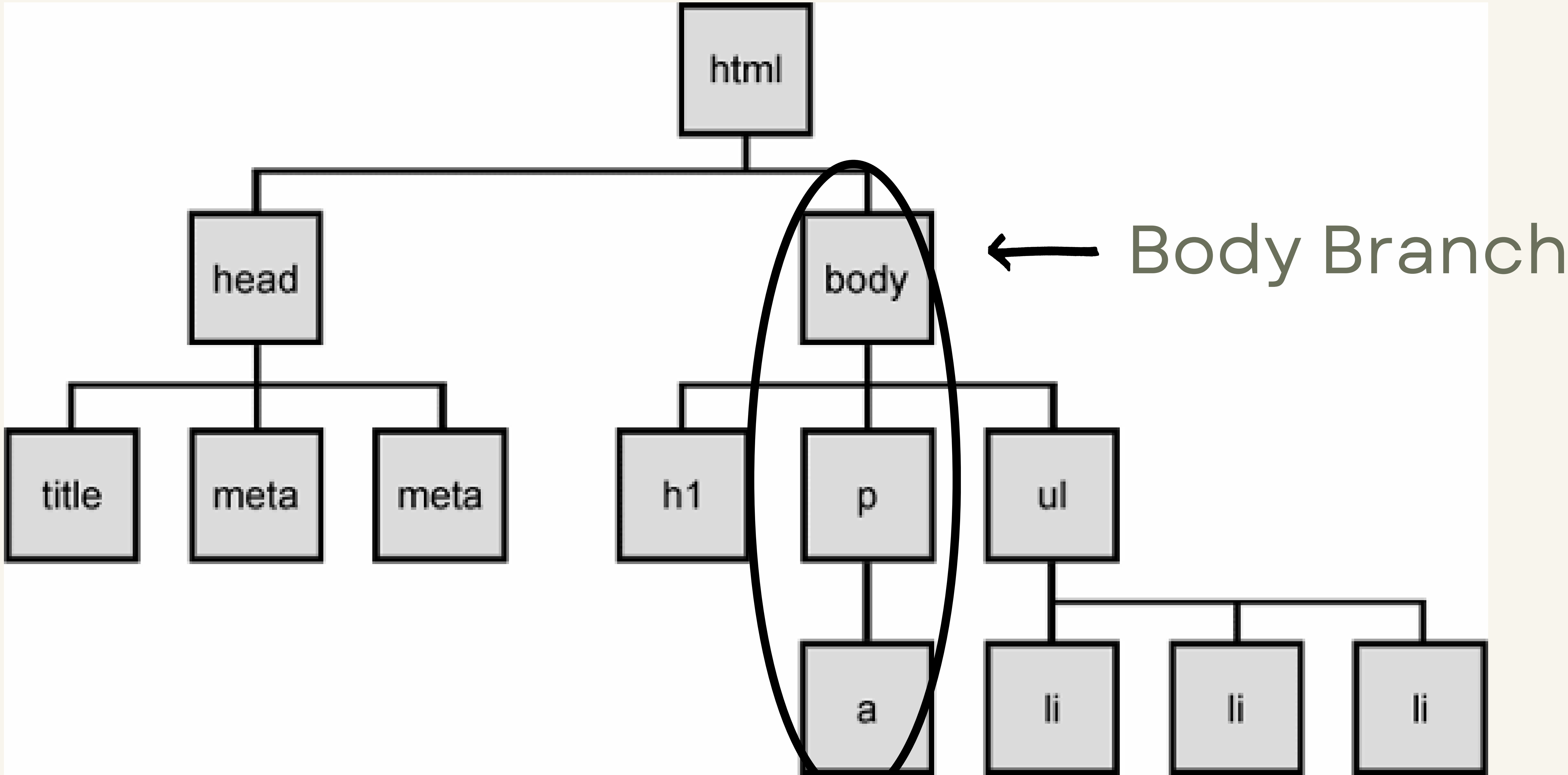
Nodes

The start of the the tree graph is known as the root node



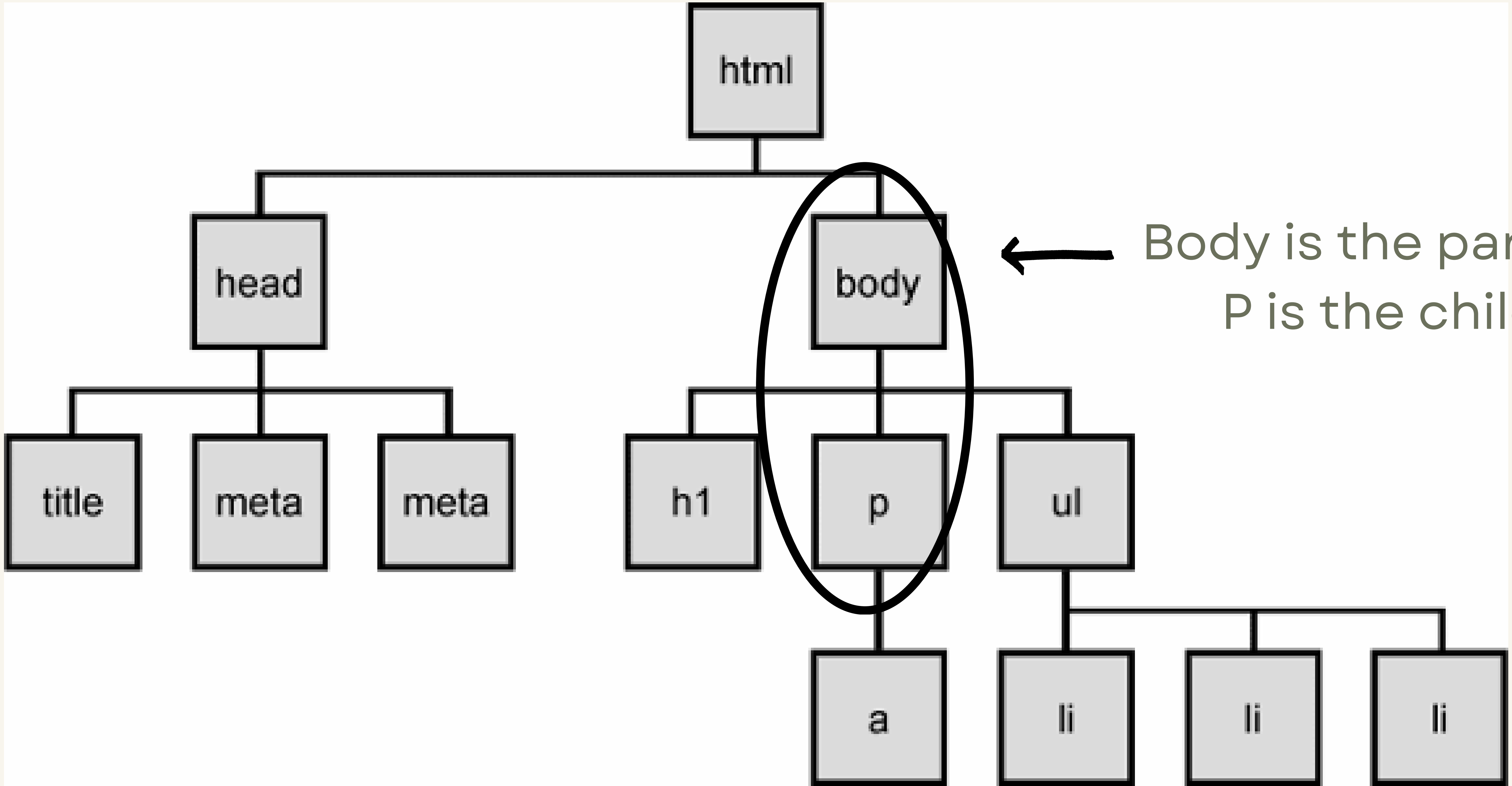
Nodes

We can refer to sections of the tree graph as branches



Parent, Child, and Sibling Nodes

If a node in encompassing another node, it is the "parent node"
The node that is being encompassed is the "child node"

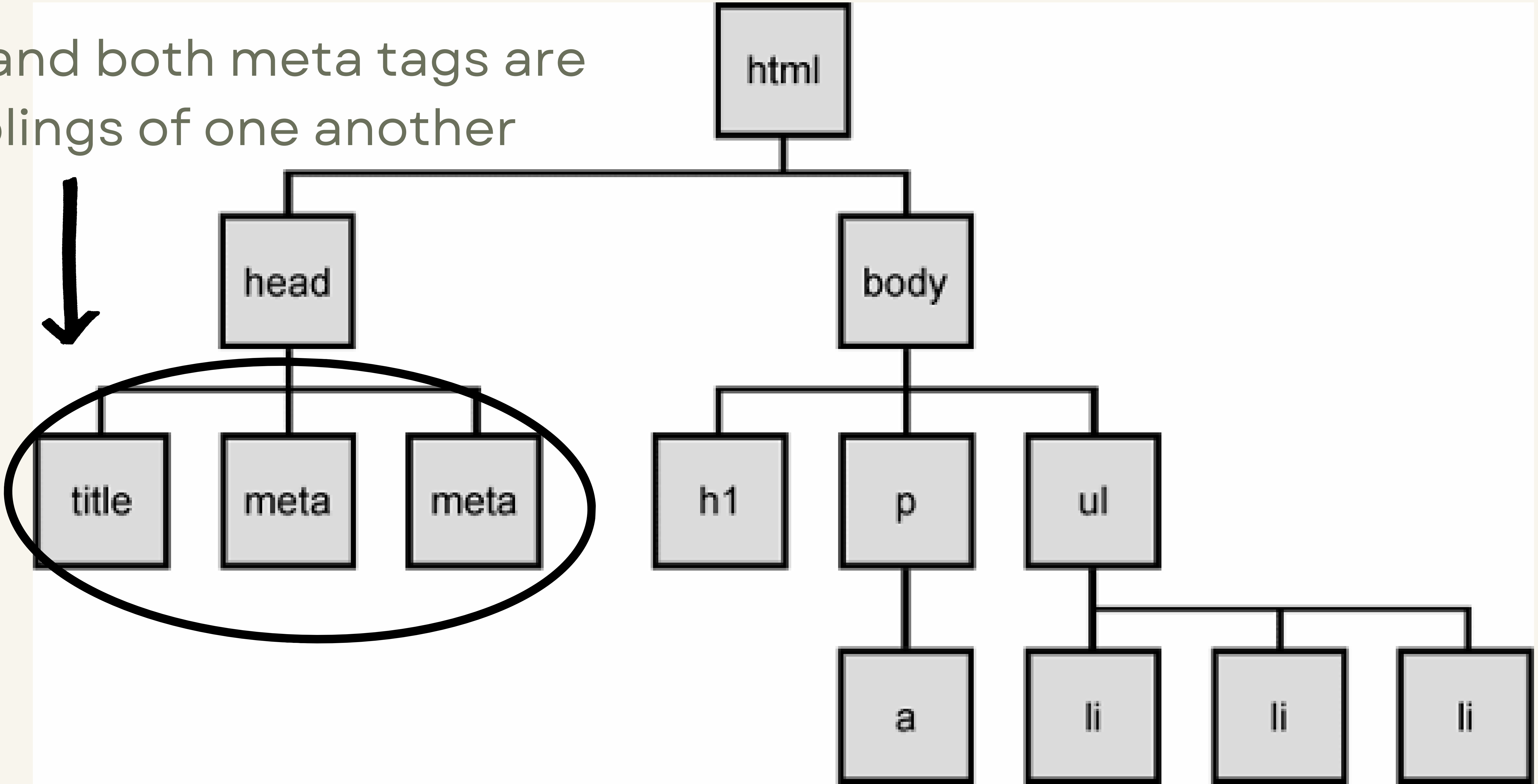


Body is the parent,
P is the child

Parent, Child, and Sibling Nodes

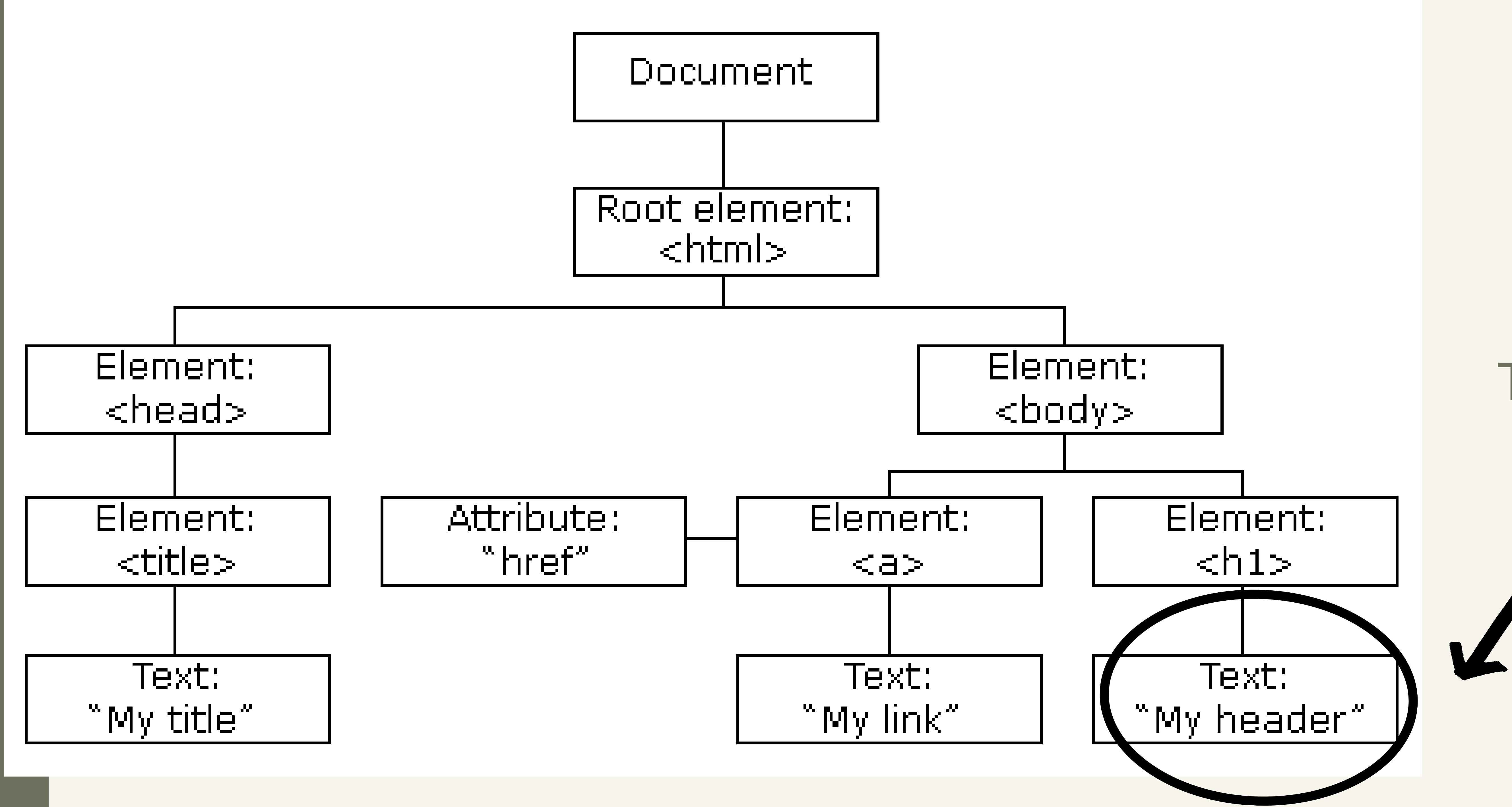
All the "children" being encompassed within a "parent" are referred to as "sibling nodes"

Title and both meta tags are siblings of one another



Text Nodes

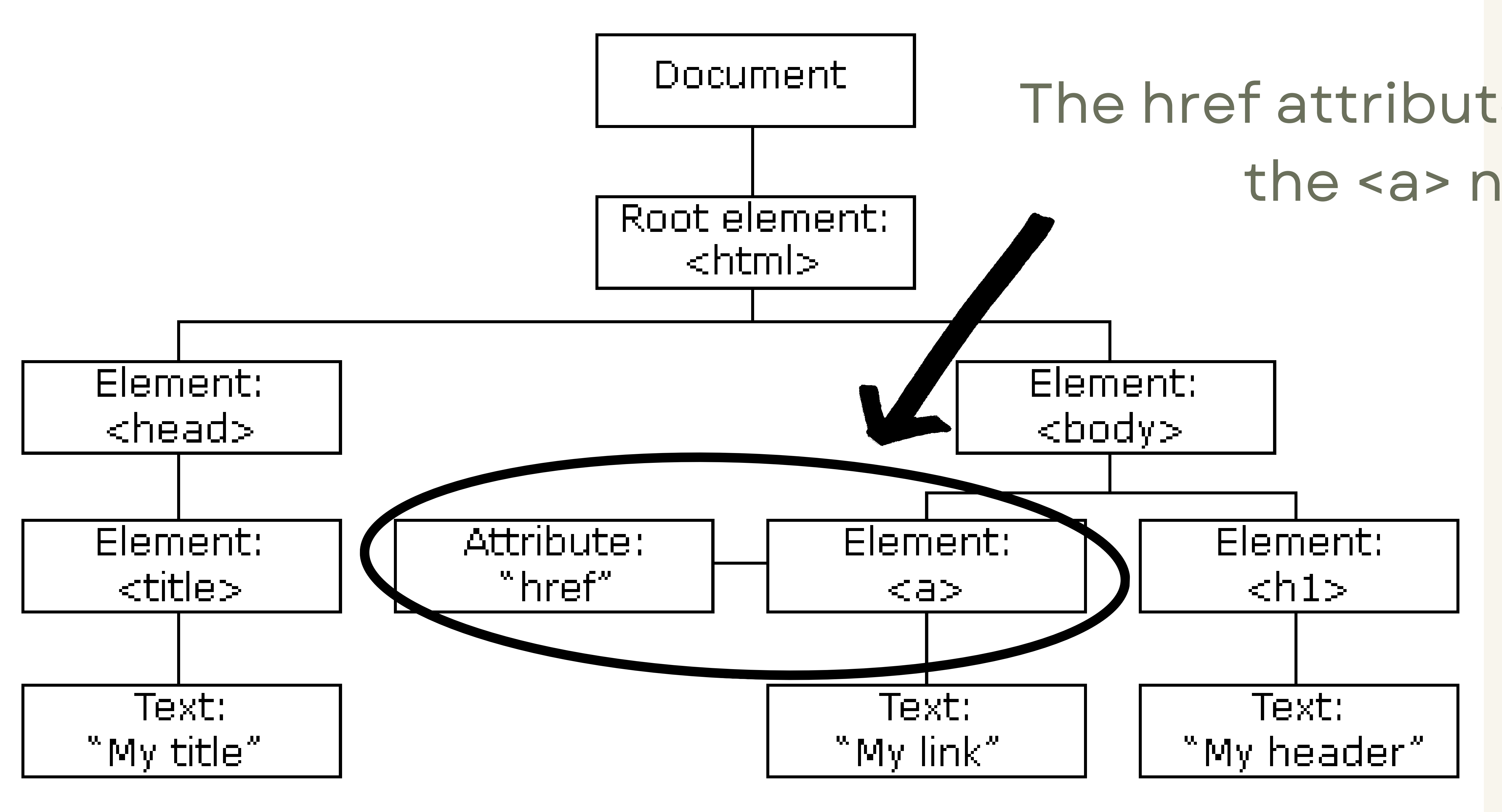
The text within a <a> or <h1> or <title> and other text elements are referred to as “text nodes”. Text nodes don’t have children of their own, neither do void nodes like <br>



Text node for h1 element

Attribute Nodes

Attributes are apart of their respective Element Node
They are not given their own node

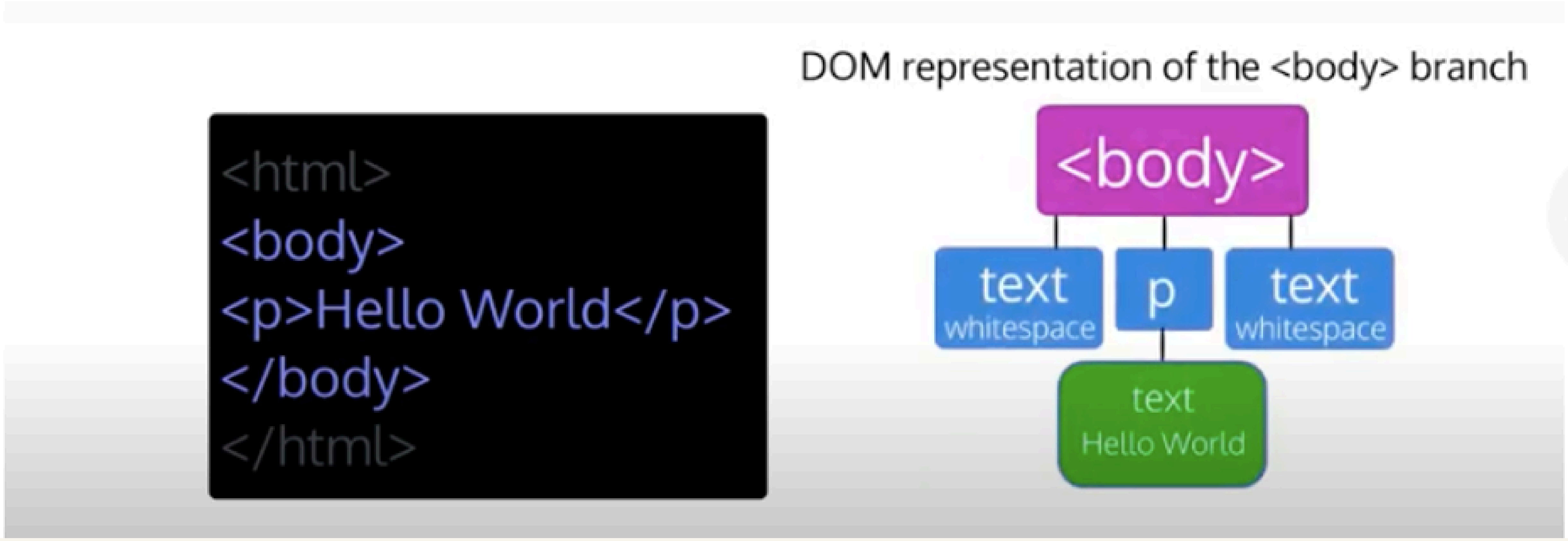


The href attribute is apart of the <a> node

White Space Nodes

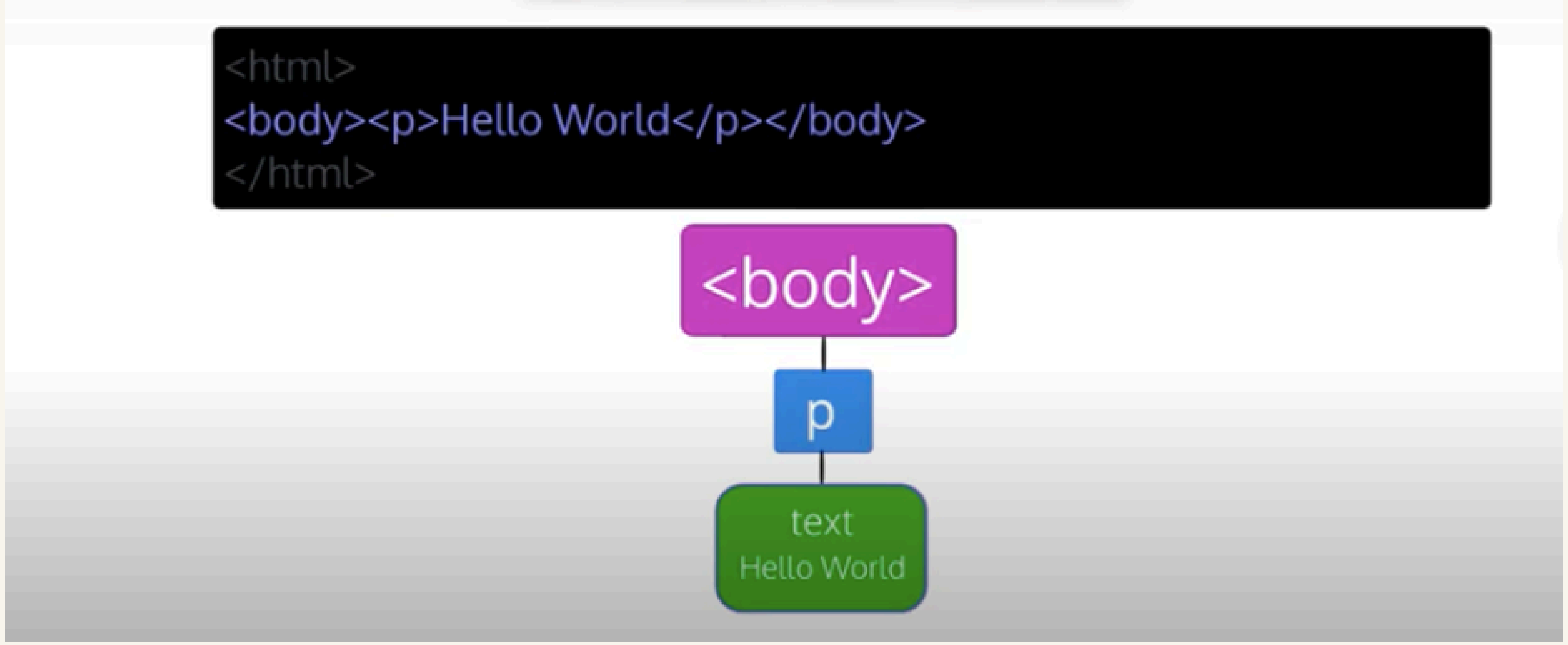
The white space you can see in the source code that you can't see in the browser

This includes line breaks, tabs, and any spaces that are not a part of text content.



DOM representation of the <body> branch

```
<html>
<body>
<p>Hello World</p>
</body>
</html>
```



```
<html>
<body><p>Hello World</p></body>
</html>
```

How do we use JS to access the DOM? Dom Transversal

Find our way into the DOM via the body node

How do we reference the body node(and therefor all the children under it?)

```
let node=document.body;
```

The body is a good entry point because there is only ever one Body in a document

This is a variable name - you can call this anything

refers to the HTML document running the code

How do we move around the DOM?

.parentNode - gives the parent of a node

.firstChild - first child of a node

.lastChild - last child of a node

.nextSibling - very next sibling of a node

.previousSibling - immediate previous sibling of a node

How do we move around the DOM?

.childNodes[...] - generates a list containing all child nodes of a parent node. Use bracket notation with index to access each child in the list. **Starts with 0**

Ex:
.childNodes[0] is the first child of the parent node
.childNodes[1] is the second child of the parent node
.childNodes[2] is the third child of the parent node

Challenge Q - which of these would equal .firstChild?

Methods of manipulating the DOM

DOM Transversal:
Jumping between parents, children, and siblings

Example:
document.body.childNodes[3].style.color = "red";

getElementsByTagName:
Chooses the element based on the tag you're using

Example:
document.getElementsByTagName("p")[0].style.color = "orange";

Methods of manipulating the DOM

getElementsById:
Selects an element by ID

Example:
document.getElementById("textToChange").style.color = "blue";

querySelector:
Selects an element based on it's CSS selector, meaning you can select an element by ID, Class, and other selectors
https://www.w3schools.com/cssref/css_selectors.php

Example:
document.querySelector("#textToChange").style.color = "green"