level

JavaScript DOM

Fact #1

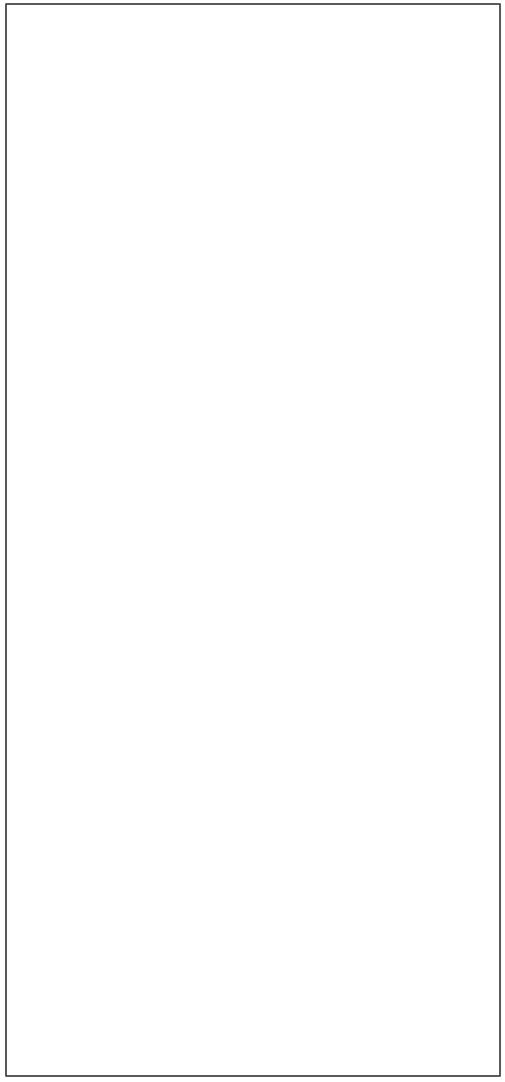
“Firstly you will hate me. Then you will depend on me. Finally You will really love me.”

-- JavaScript

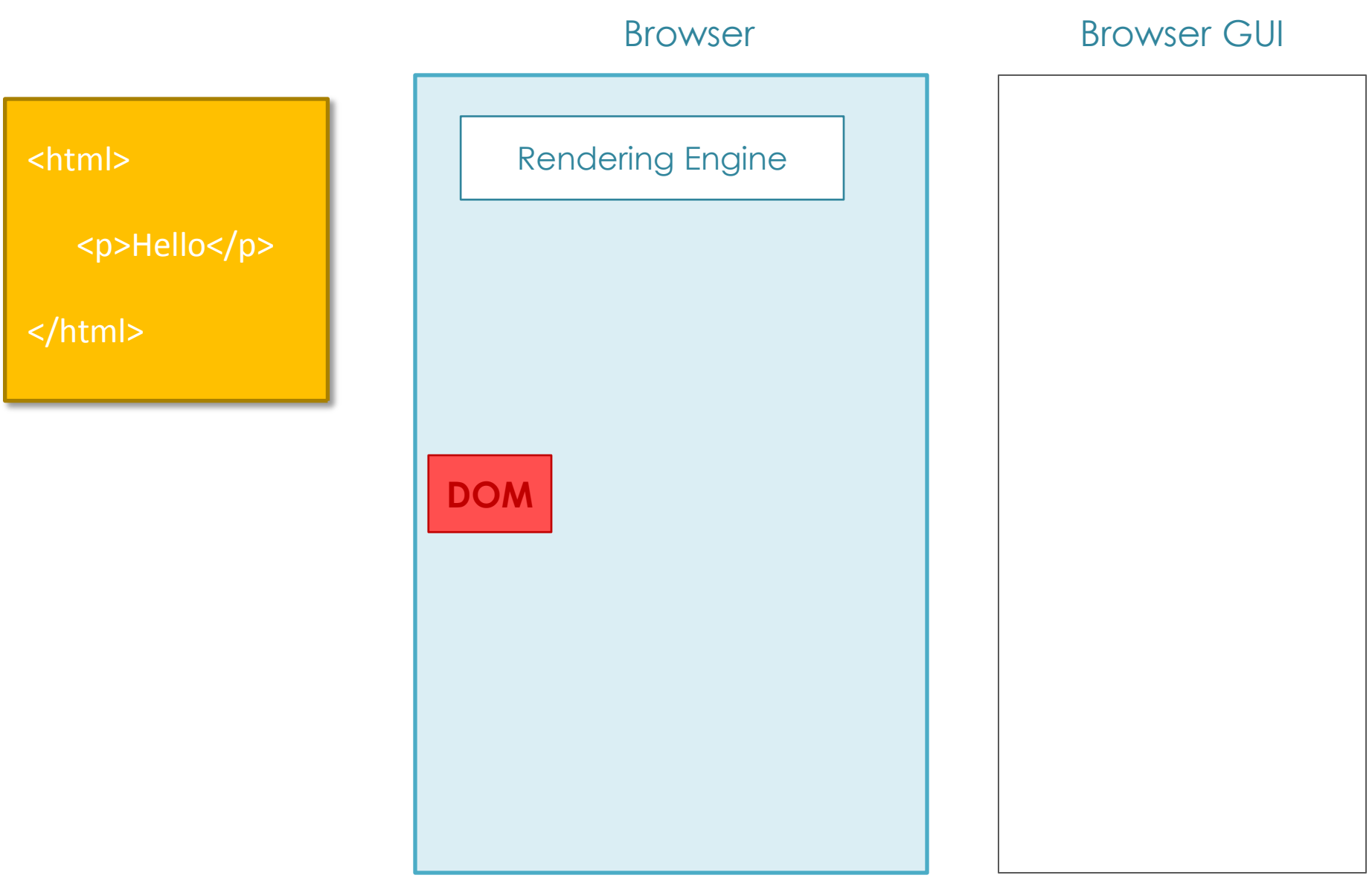
DOM

```
<html>

    <p>Hello</p>

</html>
```

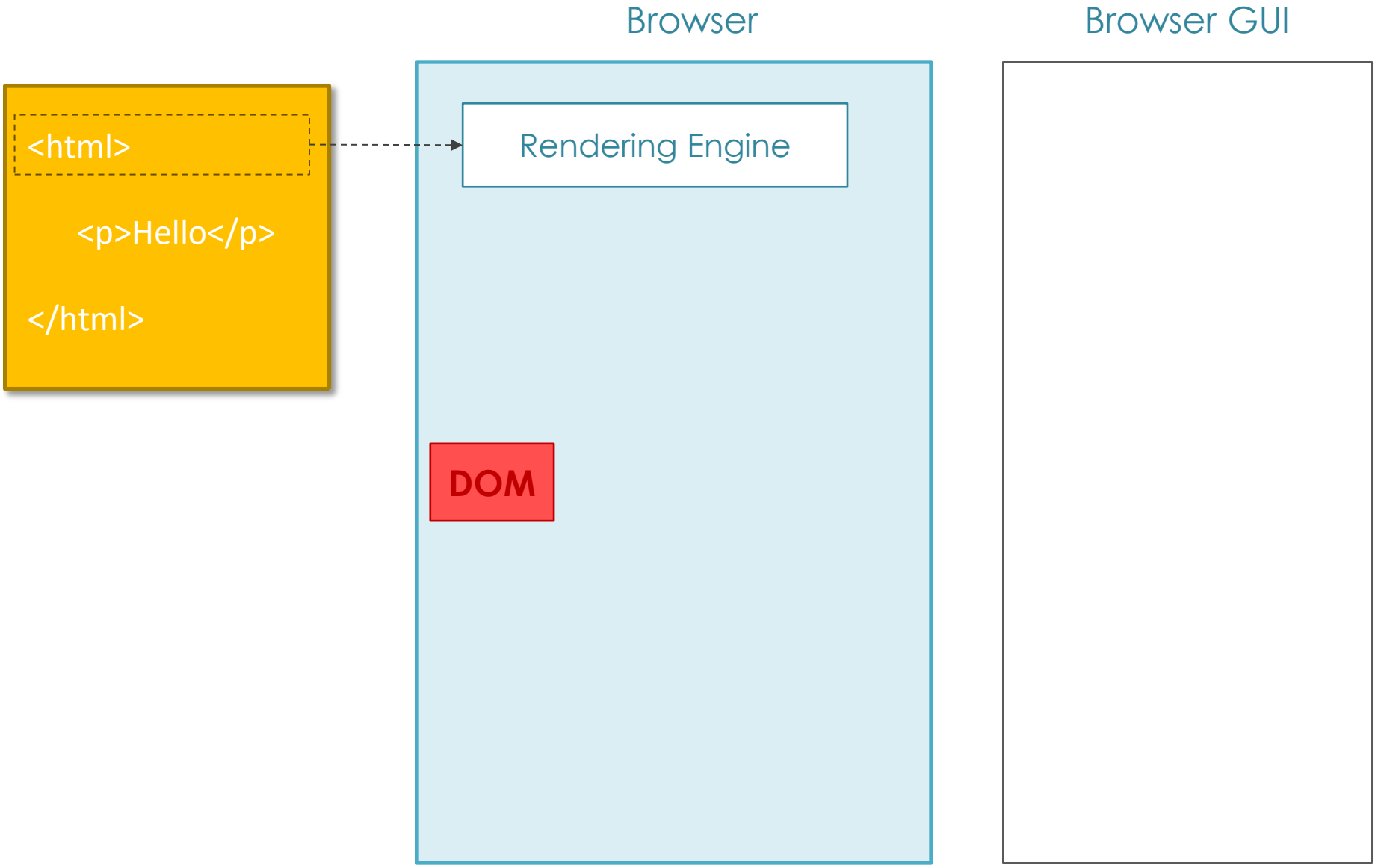
Browser GUI



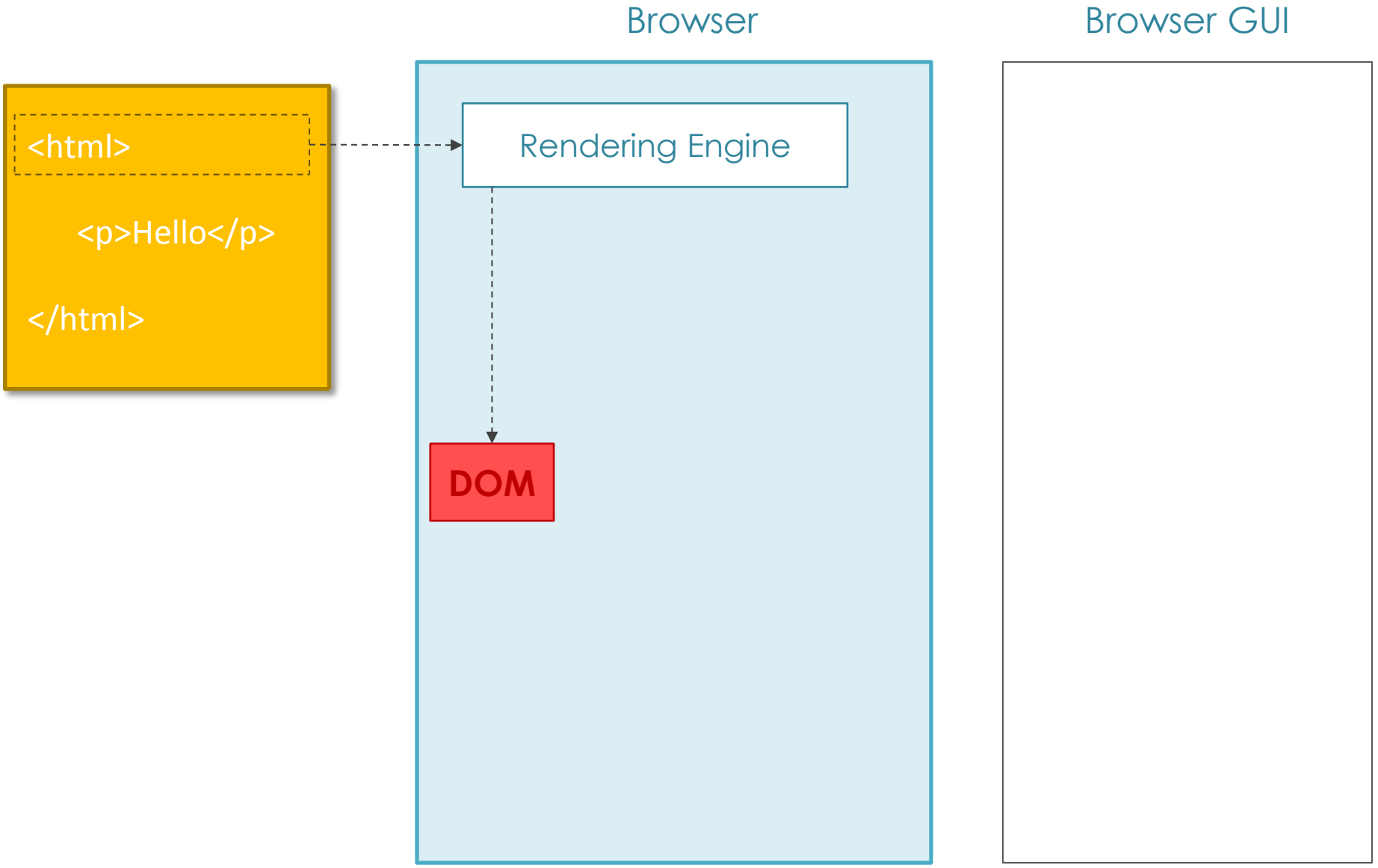
DOM



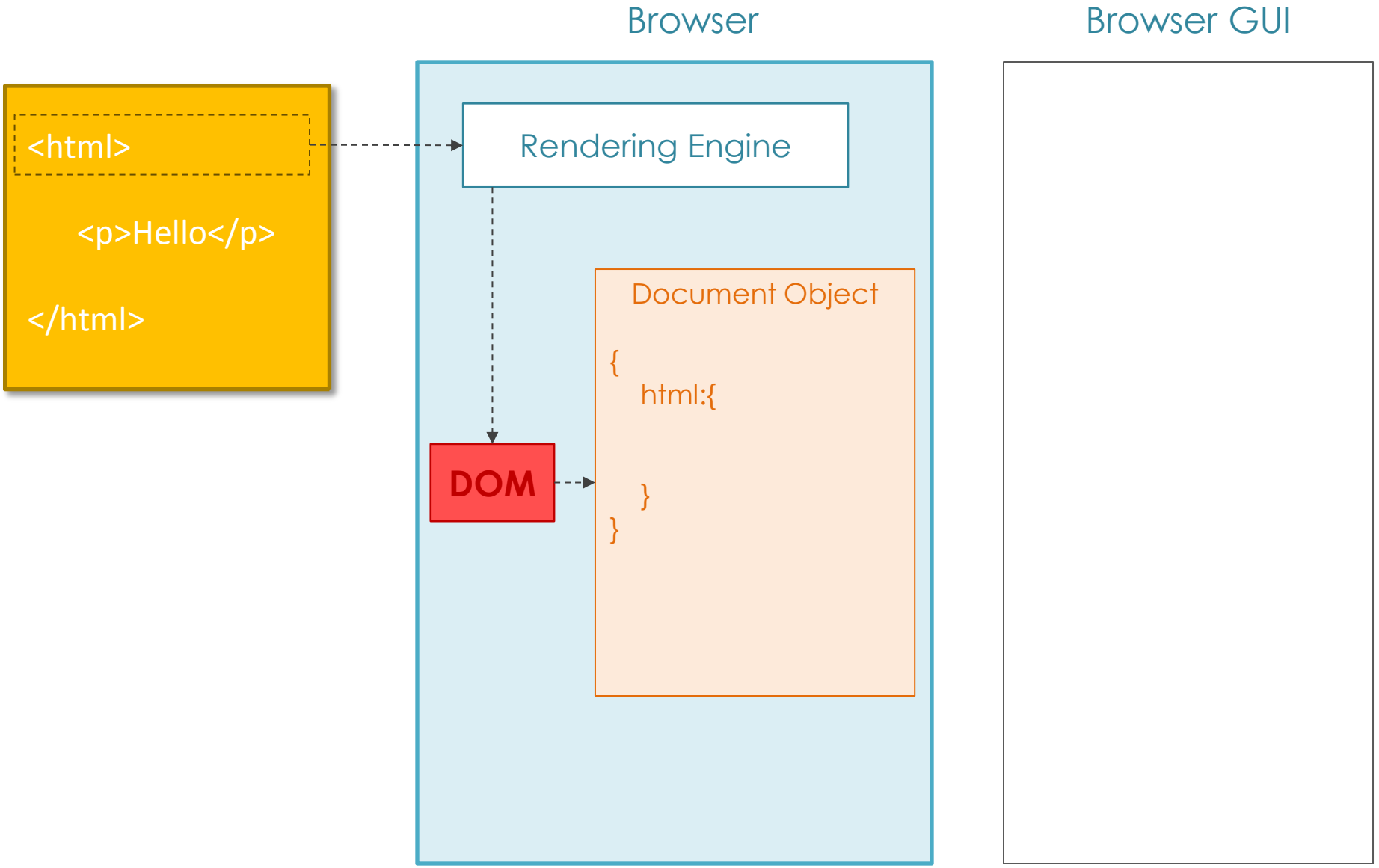
DOM



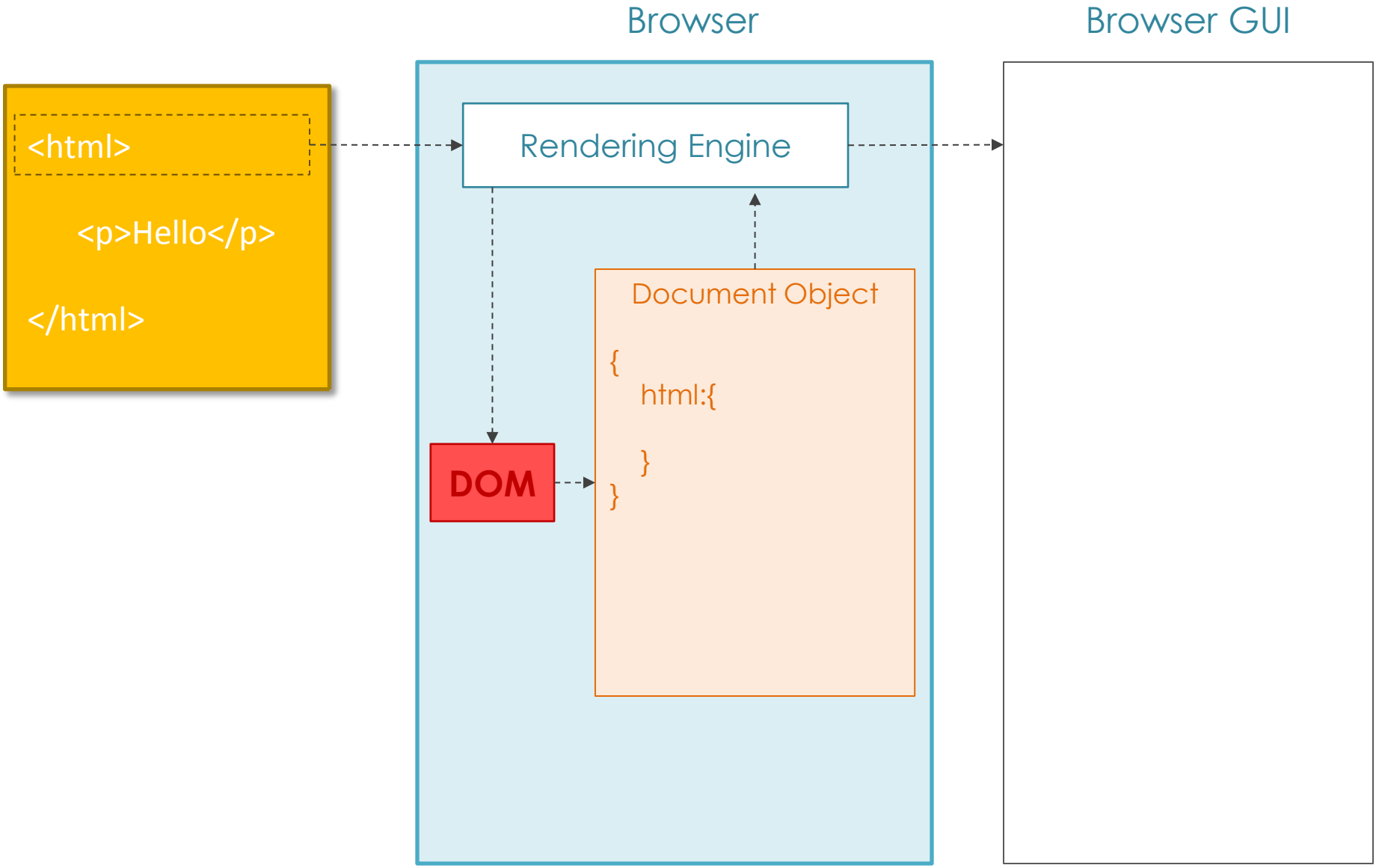
DOM



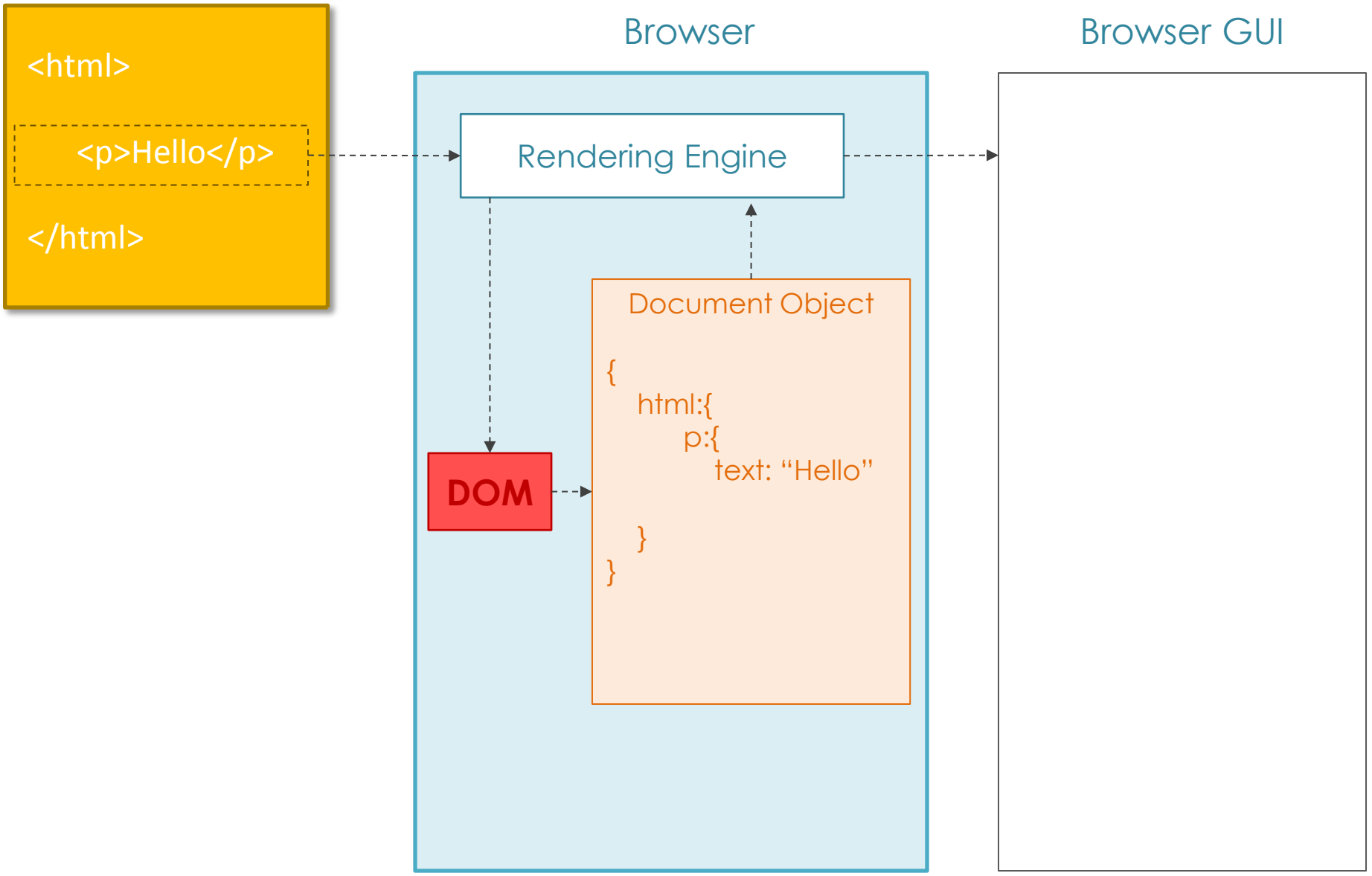
DOM


Browser
Browser GUI
<html>
Hello
</html>
Rendering Engine
Document Object
{
html:{
}
}
DOM

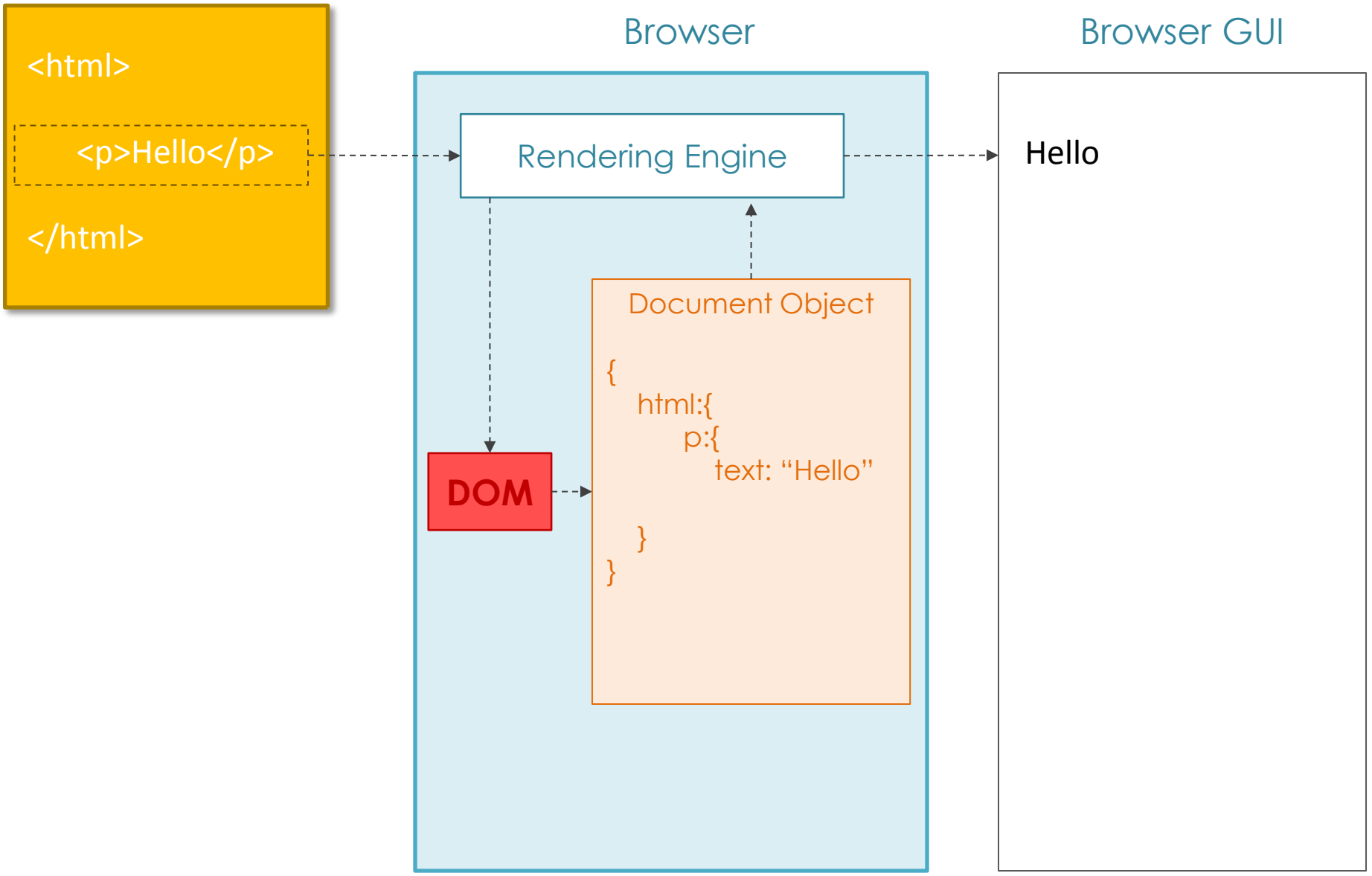
DOM


Browser
Browser GUI
<html>
Hello
</html>
Rendering Engine
Document Object
{
html:{
}
}
DOM

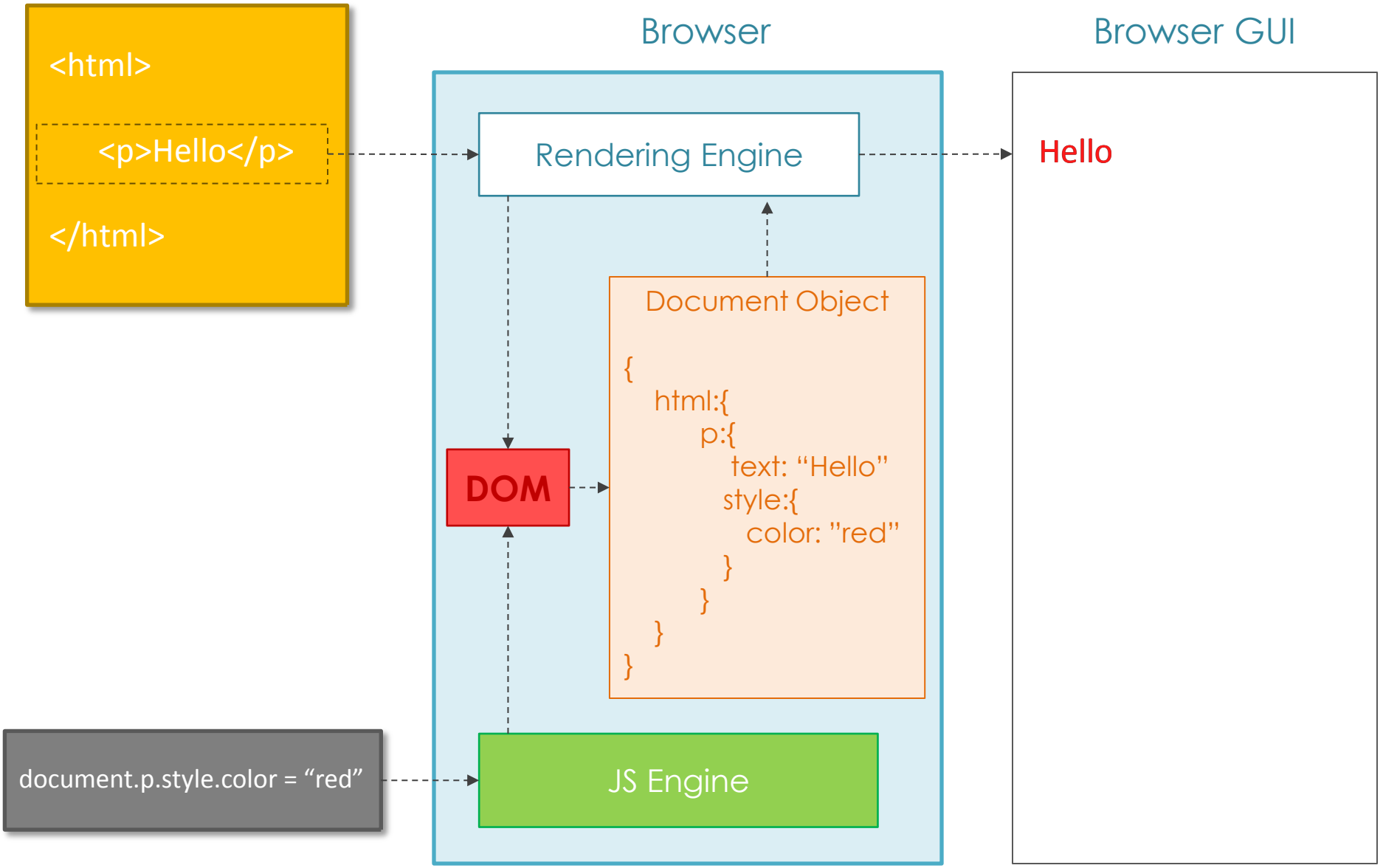
DOM


<html>
Hello
</html>
Browser
Rendering Engine
DOM
Document Object
{
html:{
p:{
text: "Hello"
}
}
}
Browser GUI

DOM


<html>
Hello
</html>
Browser
Rendering Engine
DOM
Document Object
{
html:{
p:{
text: "Hello"
}
}
}
Browser GUI
Hello

DOM


<html>
Hello
</html>
Browser
Rendering Engine
Document Object
{
html:{
p:{
text: "Hello"
style:{
color: "red"
}
}
}
}
DOM
JS Engine
document.p.style.color = "red"
Browser GUI
Hello

HTML DOM

The **HTML DOM** is a standard **object** model and **programming interface** for HTML.

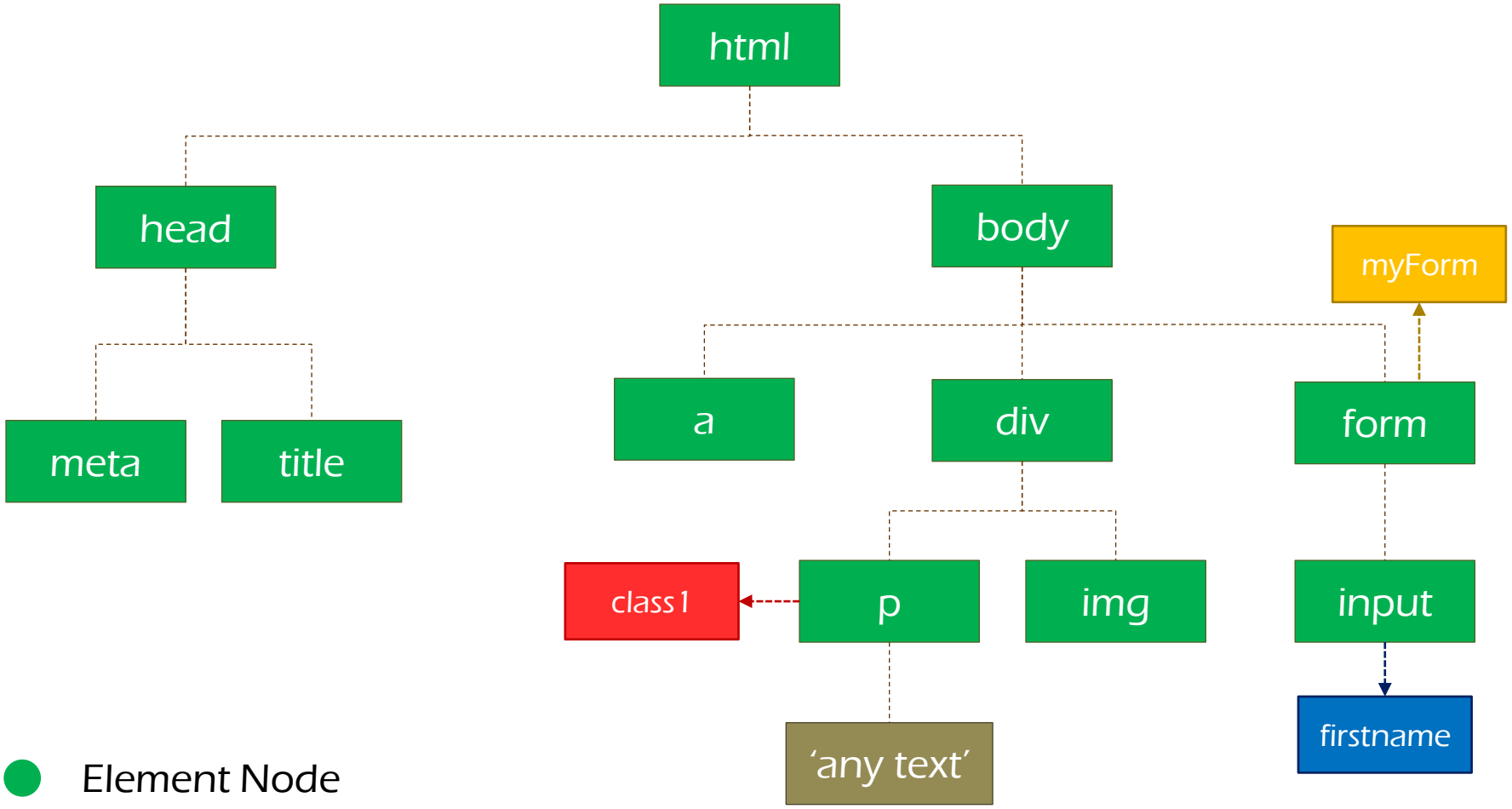
It defines:

- The HTML elements as objects
- The properties of all HTML elements
- The methods to access all HTML elements
- The events for all HTML elements

In other words:

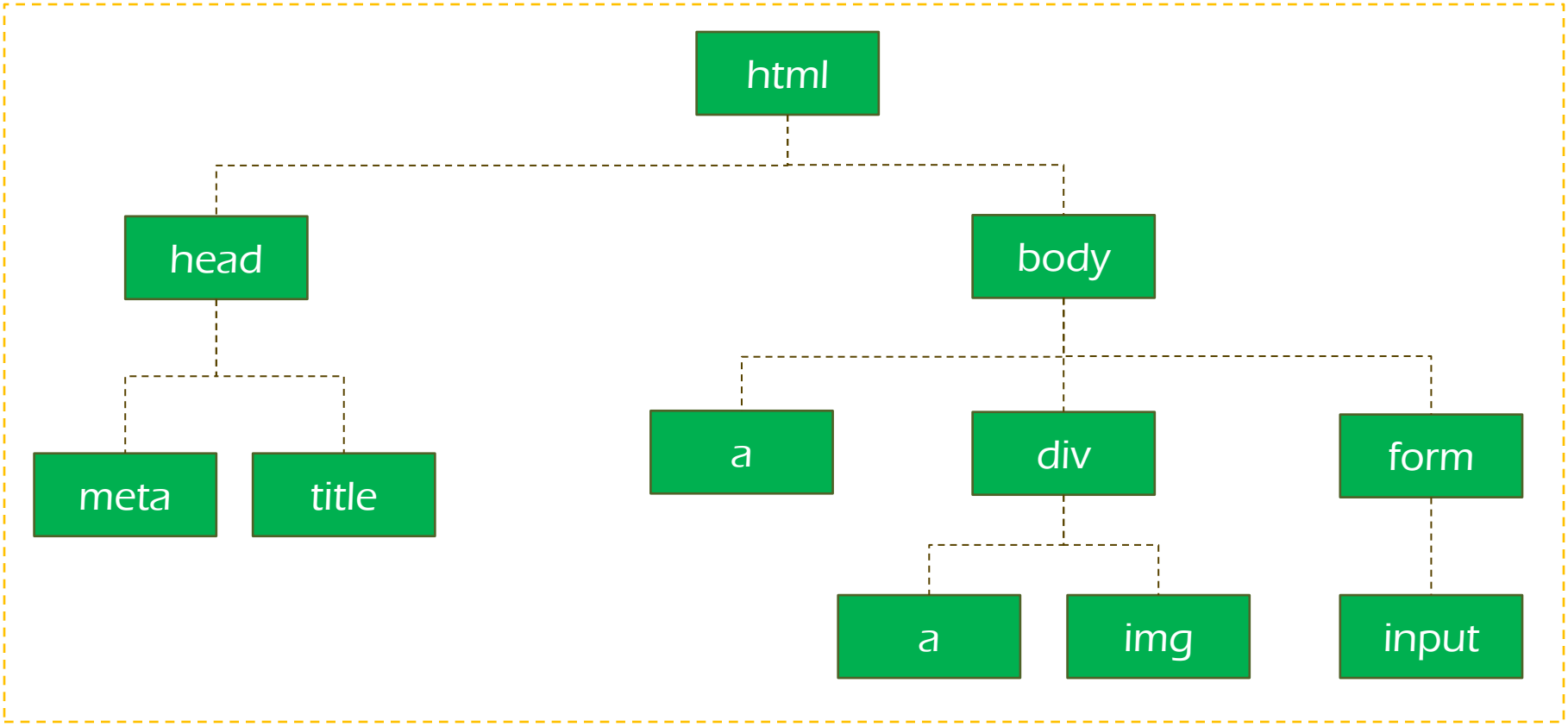
The **HTML DOM** is a standard for how to get, change, add, or delete HTML elements.

DOM TREE


html
head
body
meta
title
a
div
form
myForm
class1
p
img
input
'any text'
firstname
Element Node
Text Node
Attribute Node

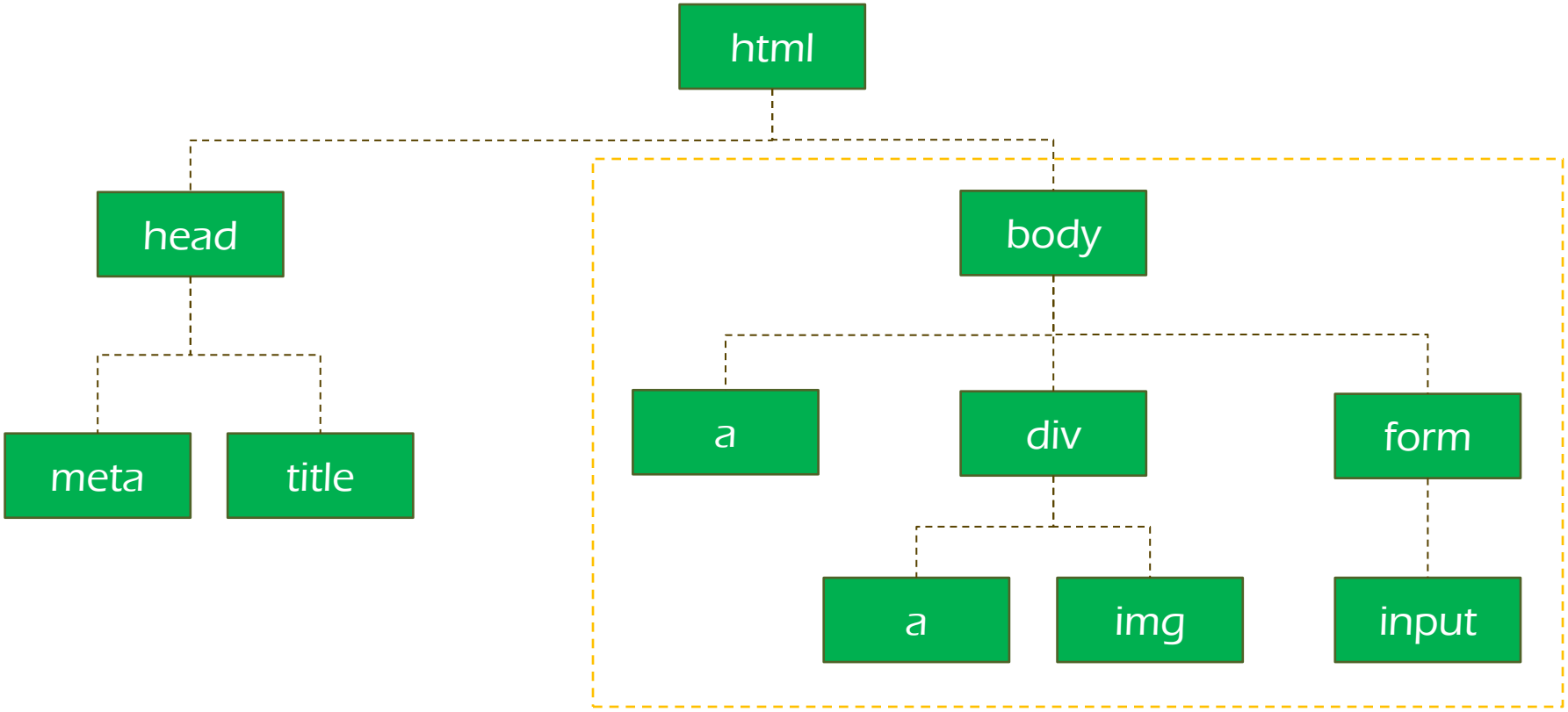
DOCUMENT OBJECT

document. documentElement



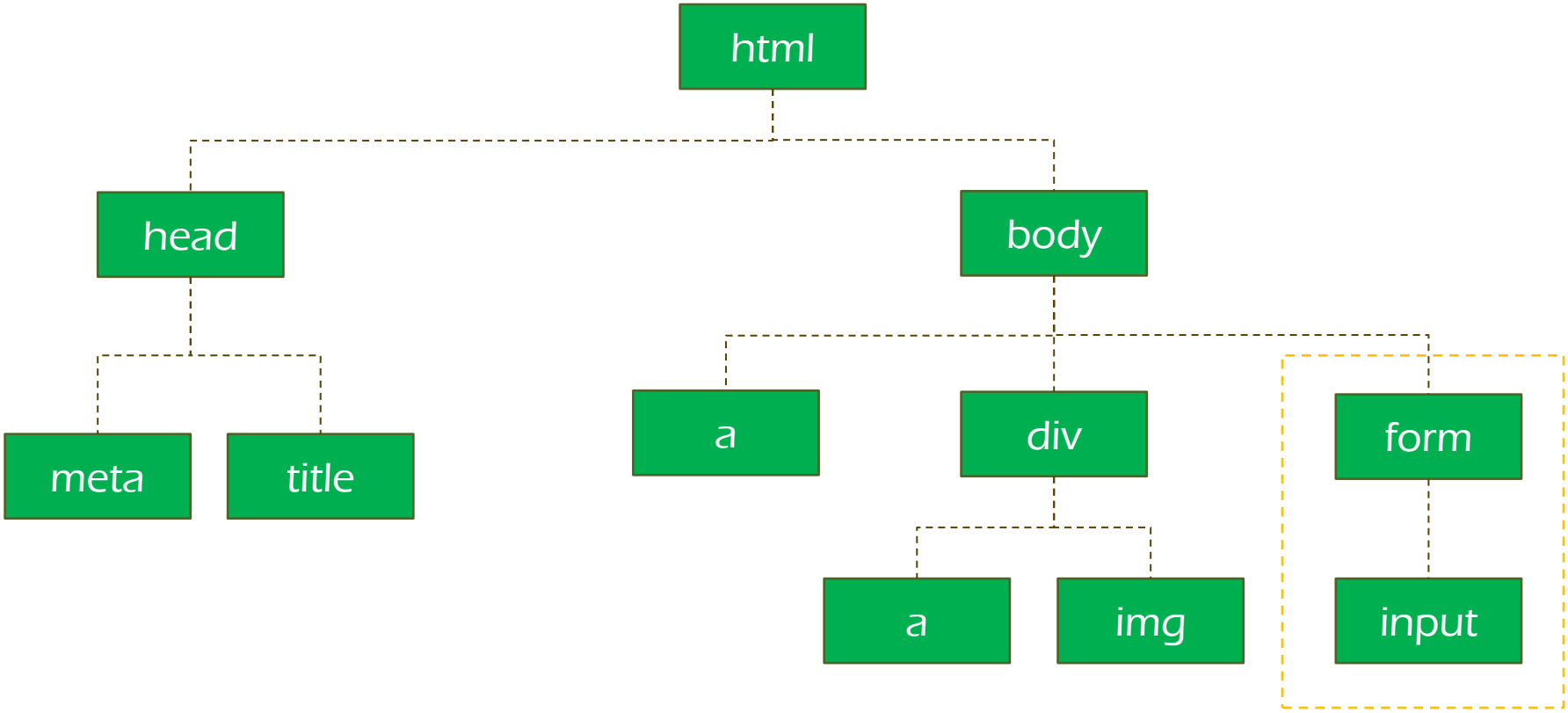
DOCUMENT OBJECT

document. body



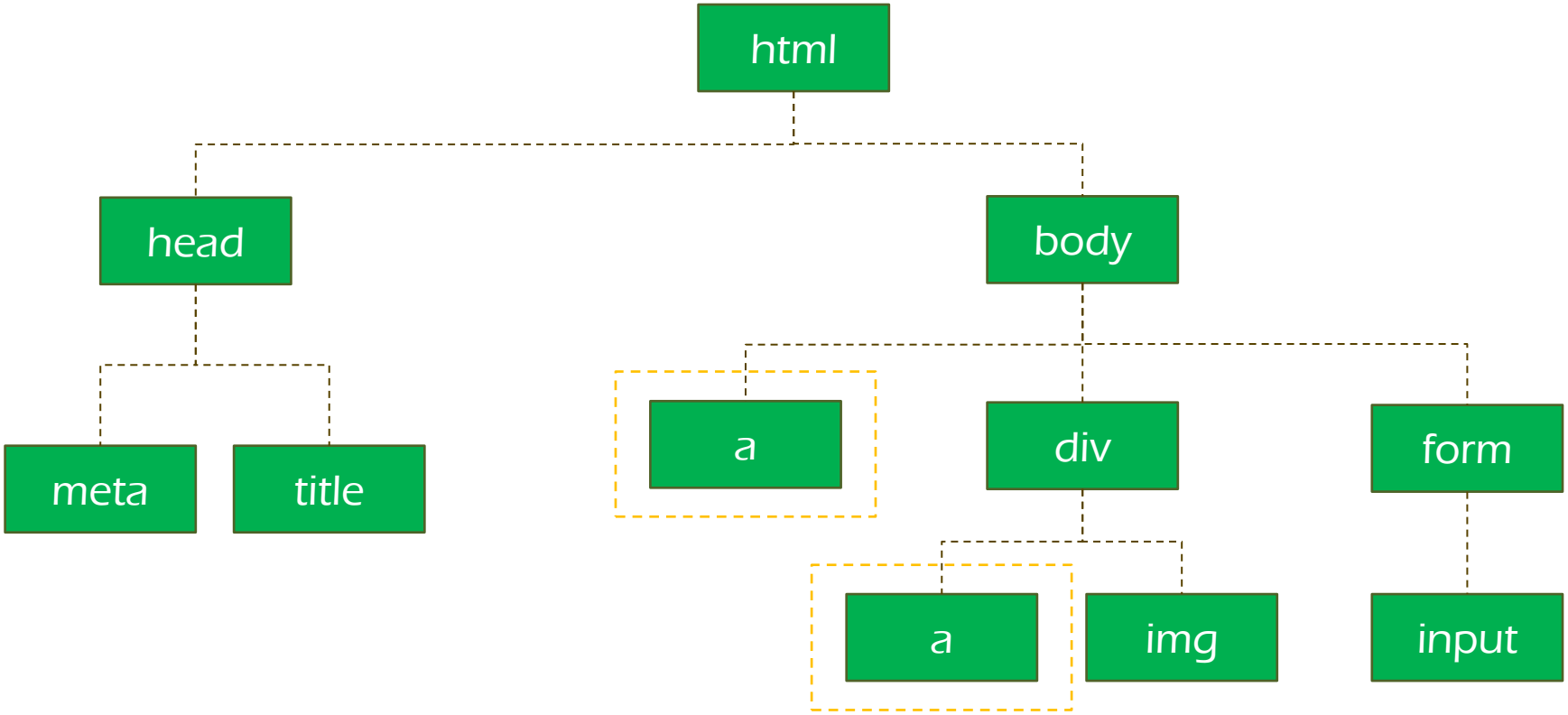
DOCUMENT OBJECT

document. forms



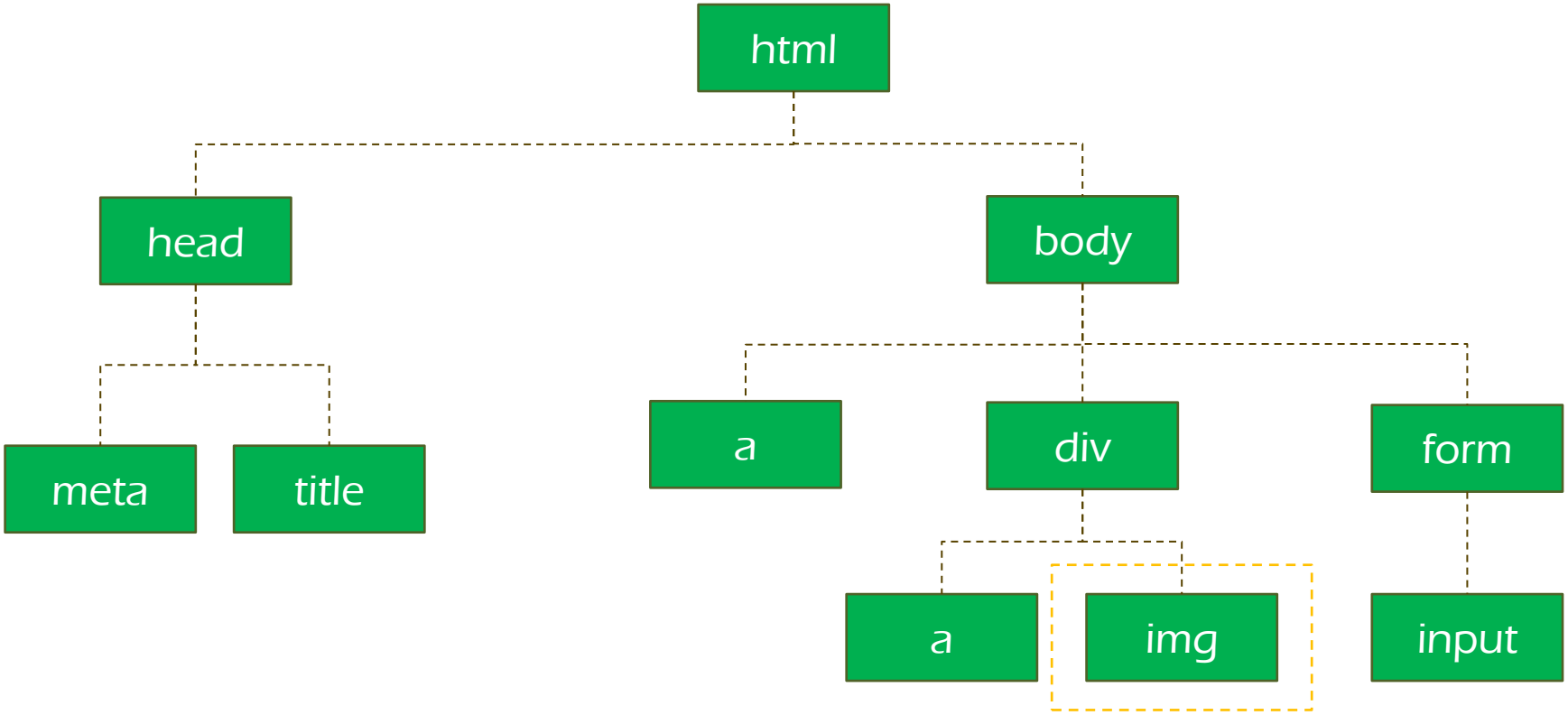
DOCUMENT OBJECT

document. links



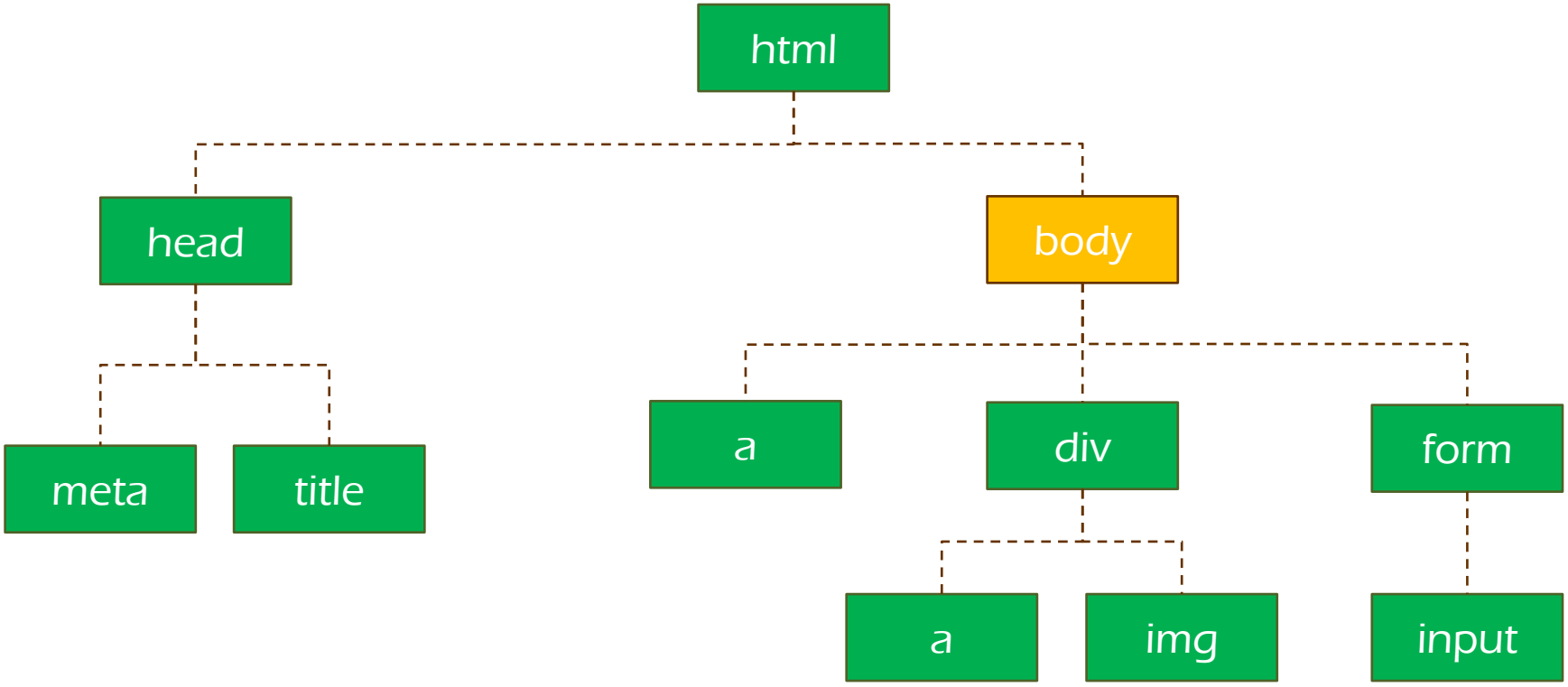
DOCUMENT OBJECT

document. images



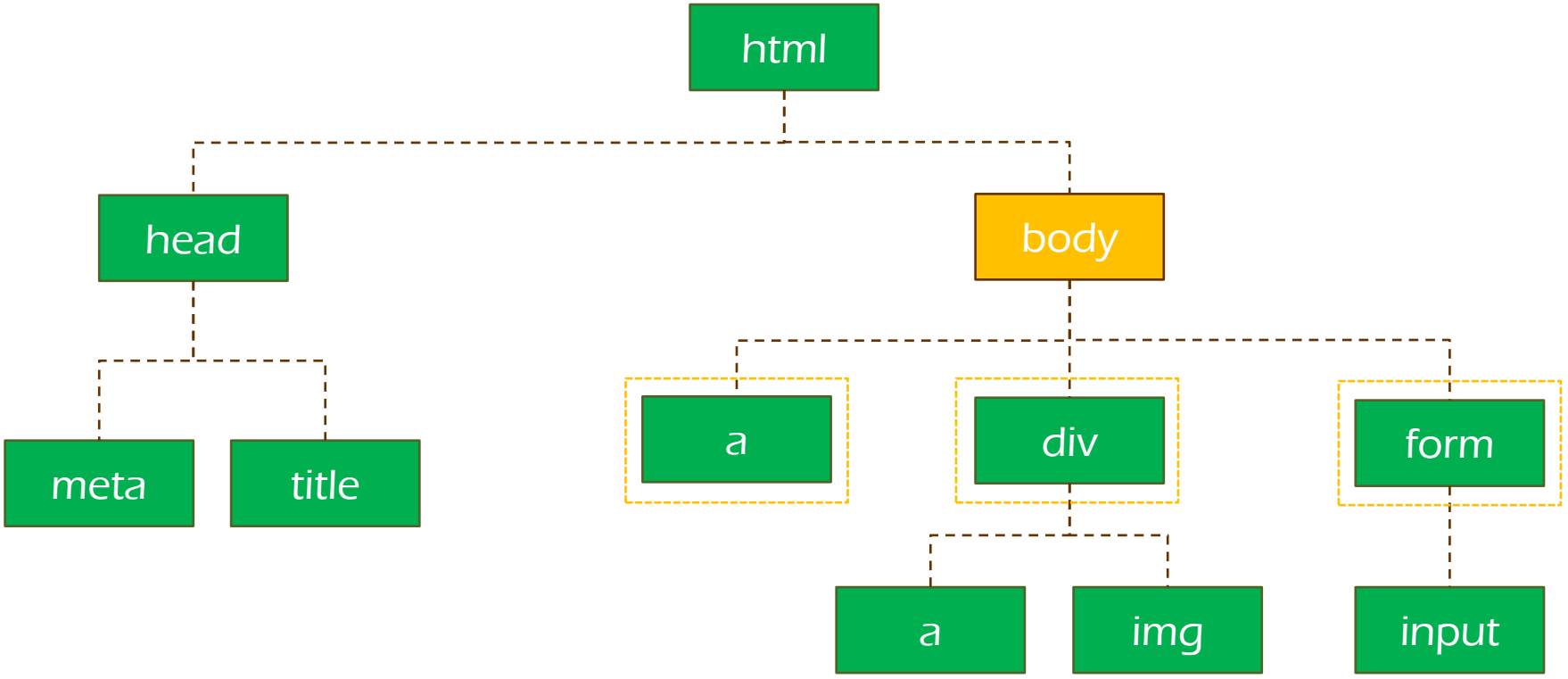
DOM NAVIGATION

document.body.



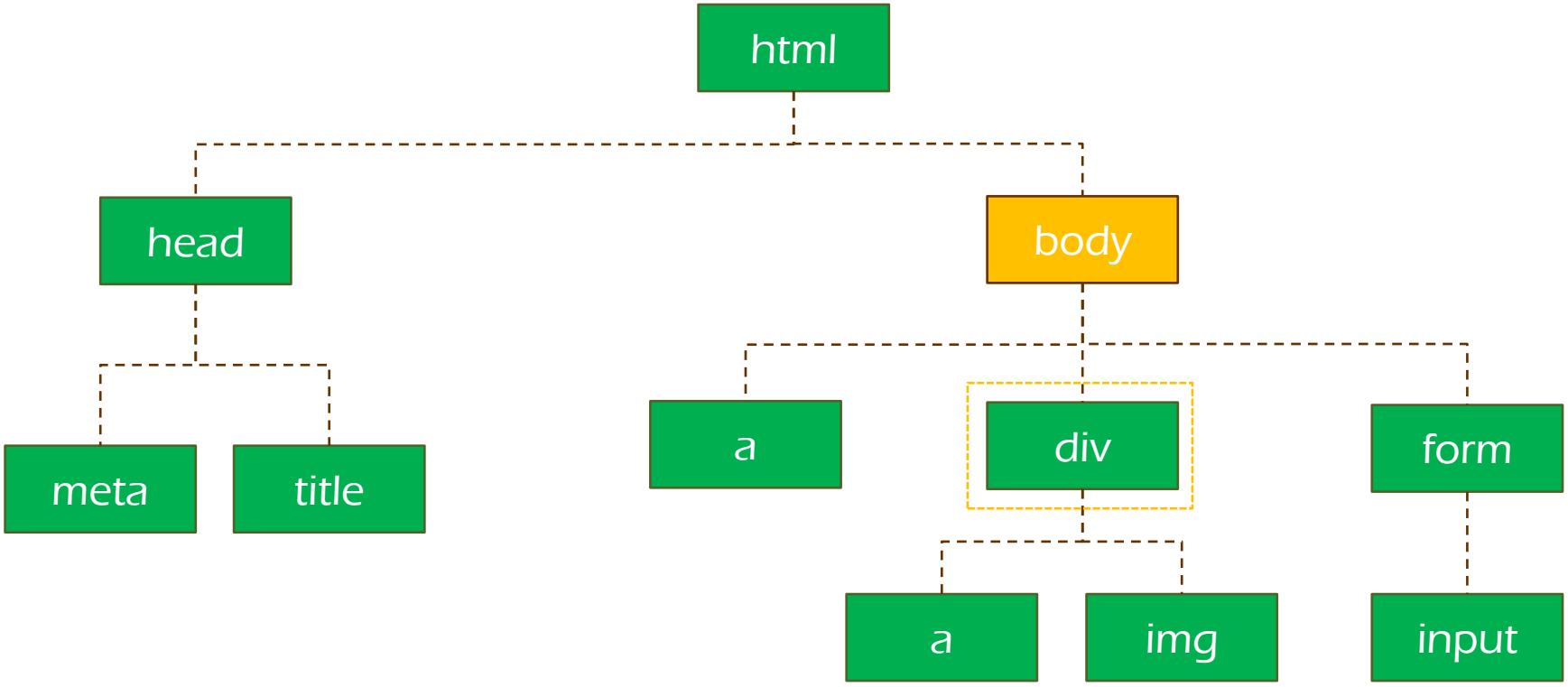
DOM NAVIGATION

document.body. children



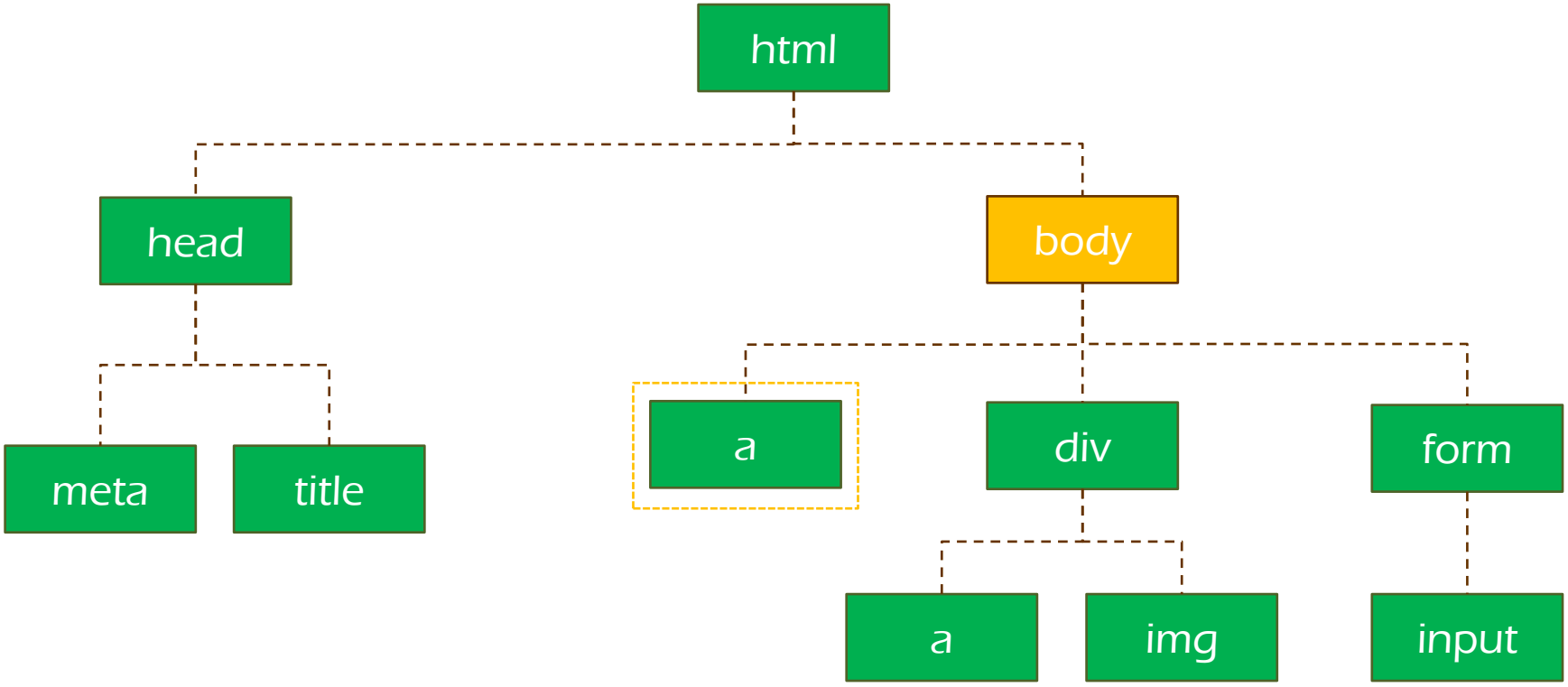
DOM NAVIGATION

document.body. children[1]

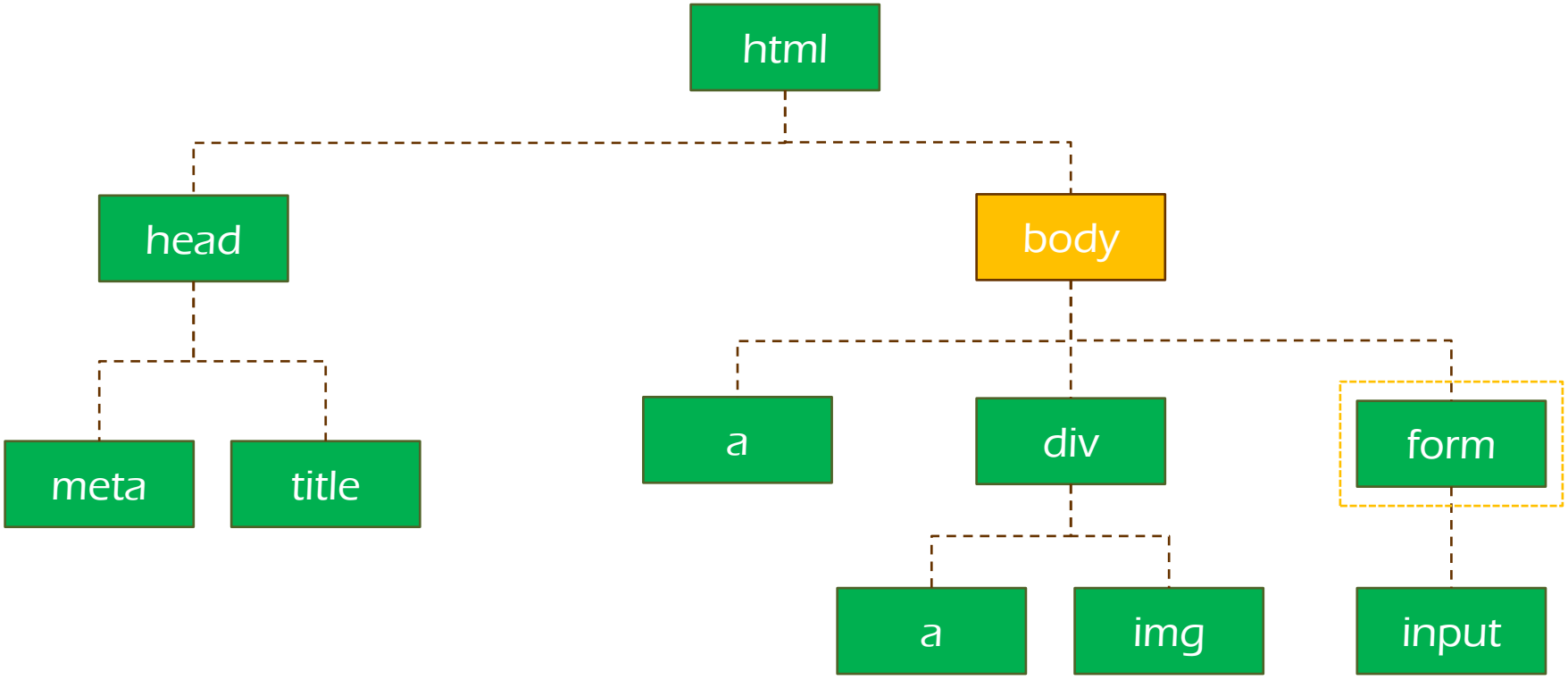


DOM NAVIGATION

document.body. firstElementChild

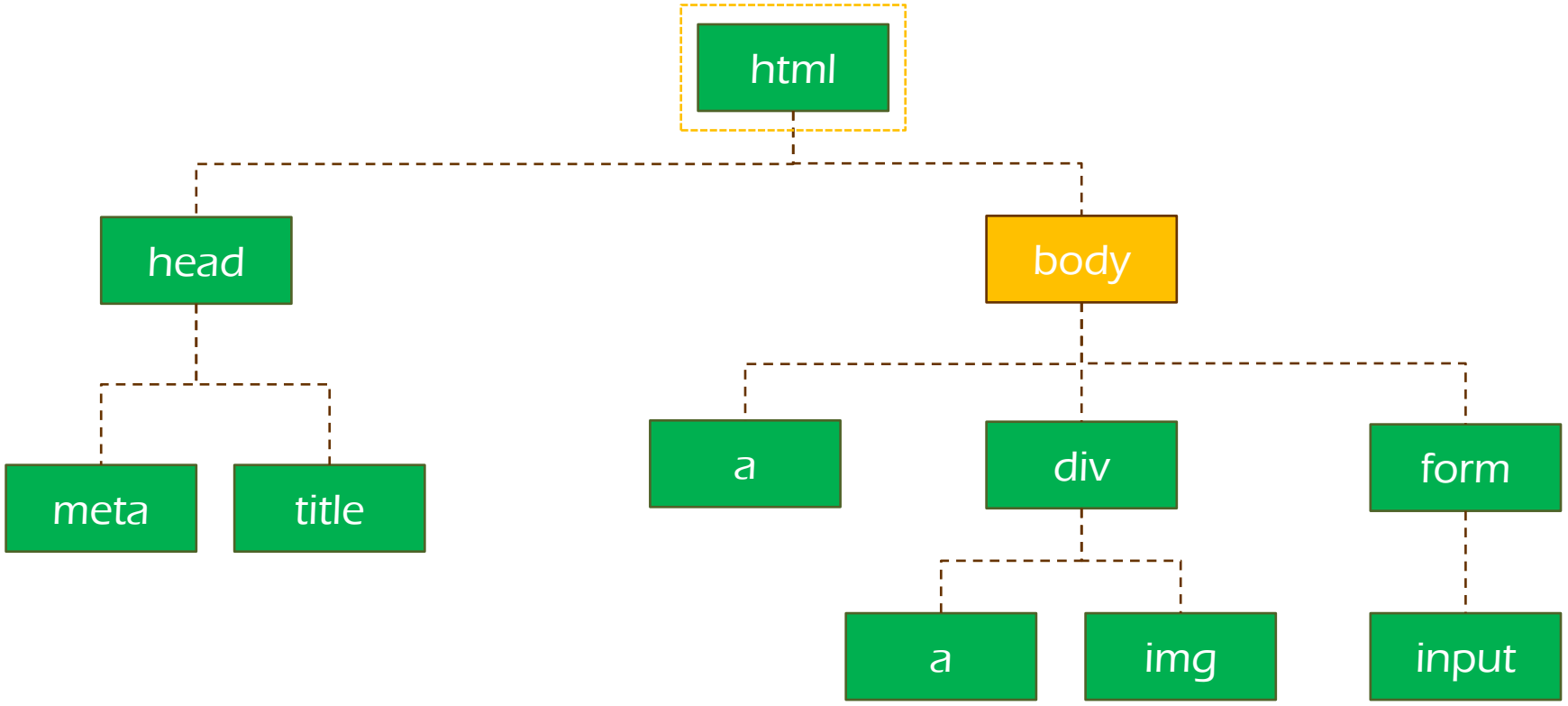


document.body. lastElementChild



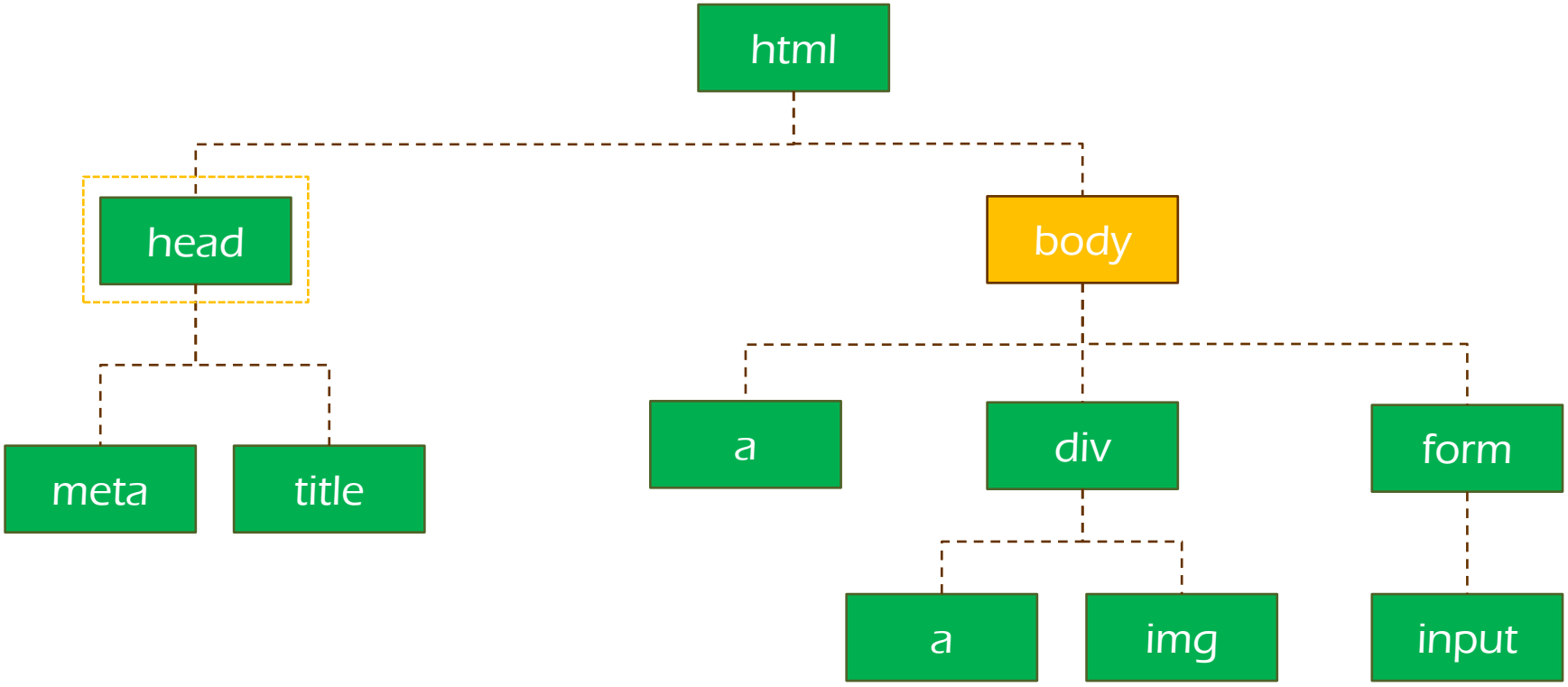
DOM NAVIGATION

document.body. parentElement



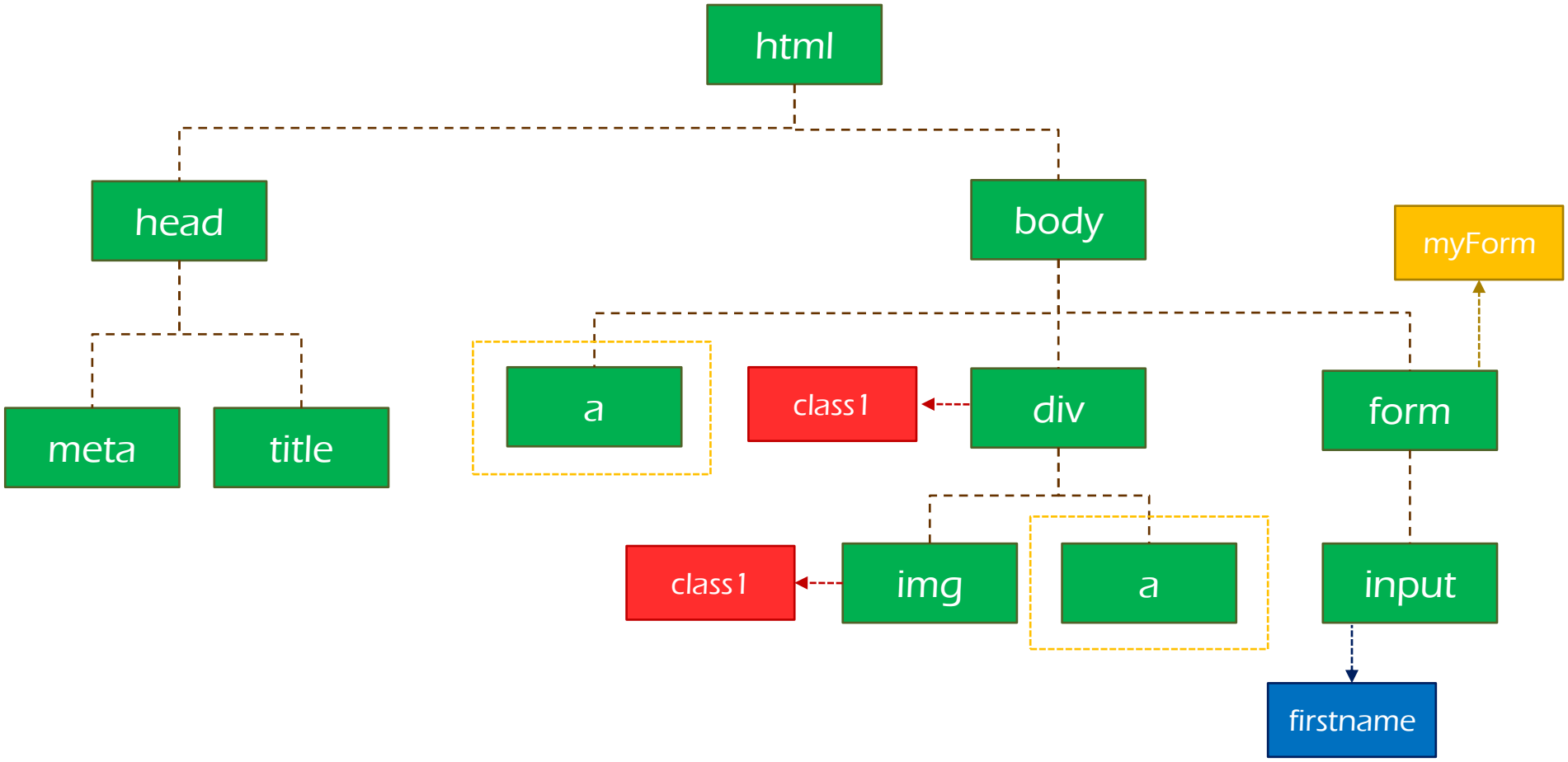
DOM NAVIGATION

document.body. previousElementSibling



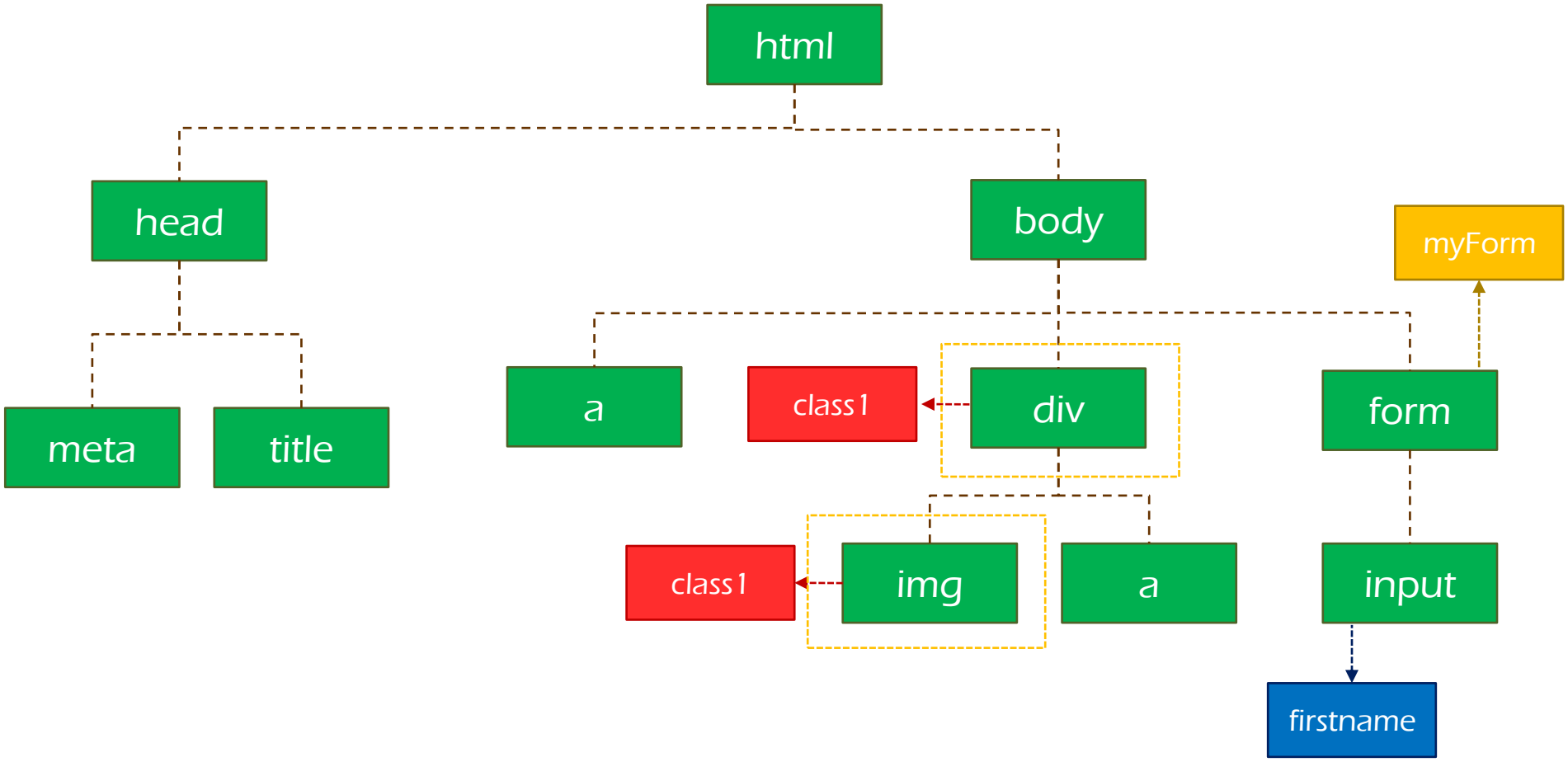
FIND ELEMENTS IN DOM

```
document.getElementsByTagName('a');
```



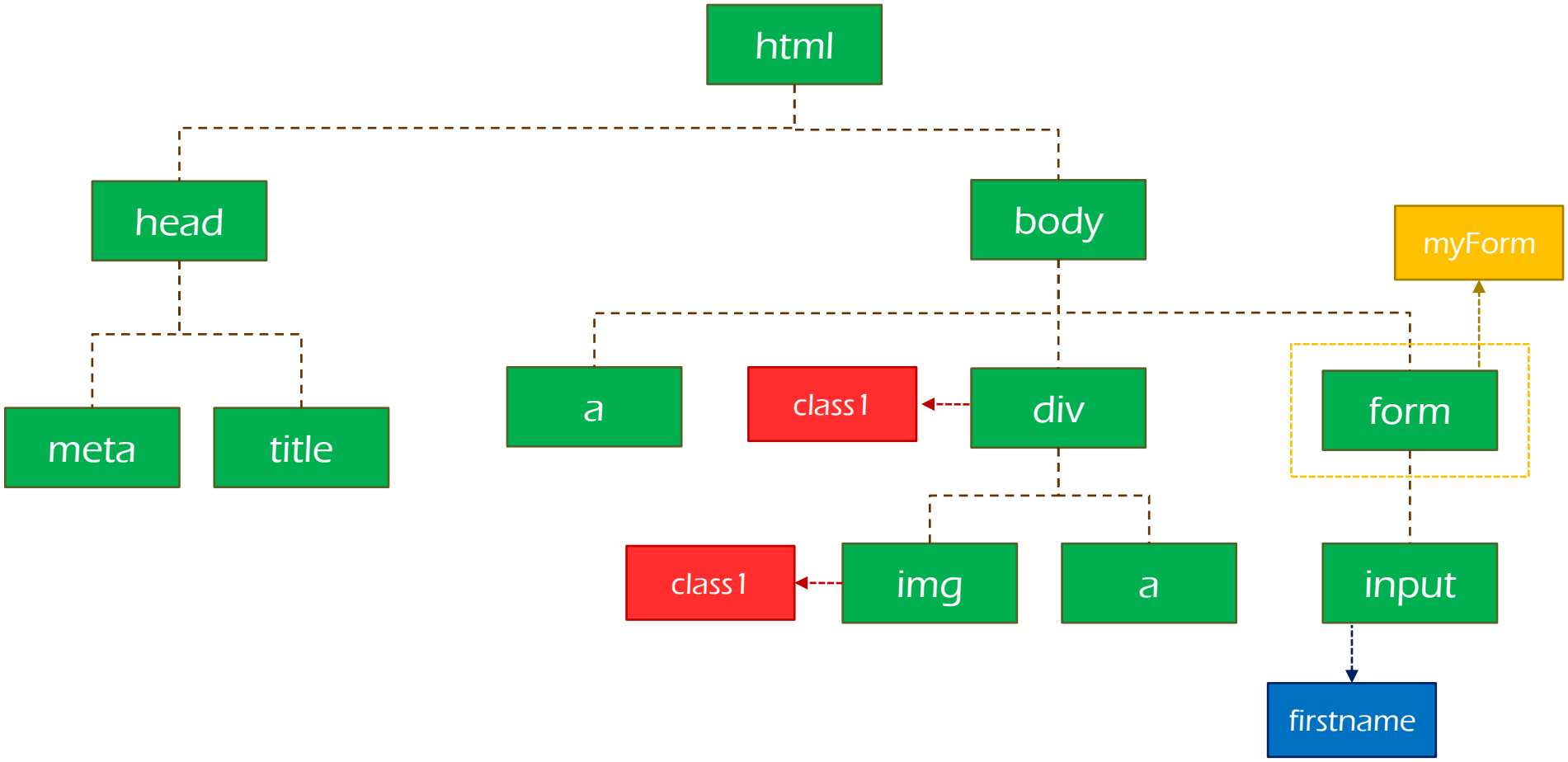
FIND ELEMENTS IN DOM

```
document.getElementsByClassName('class1');
```



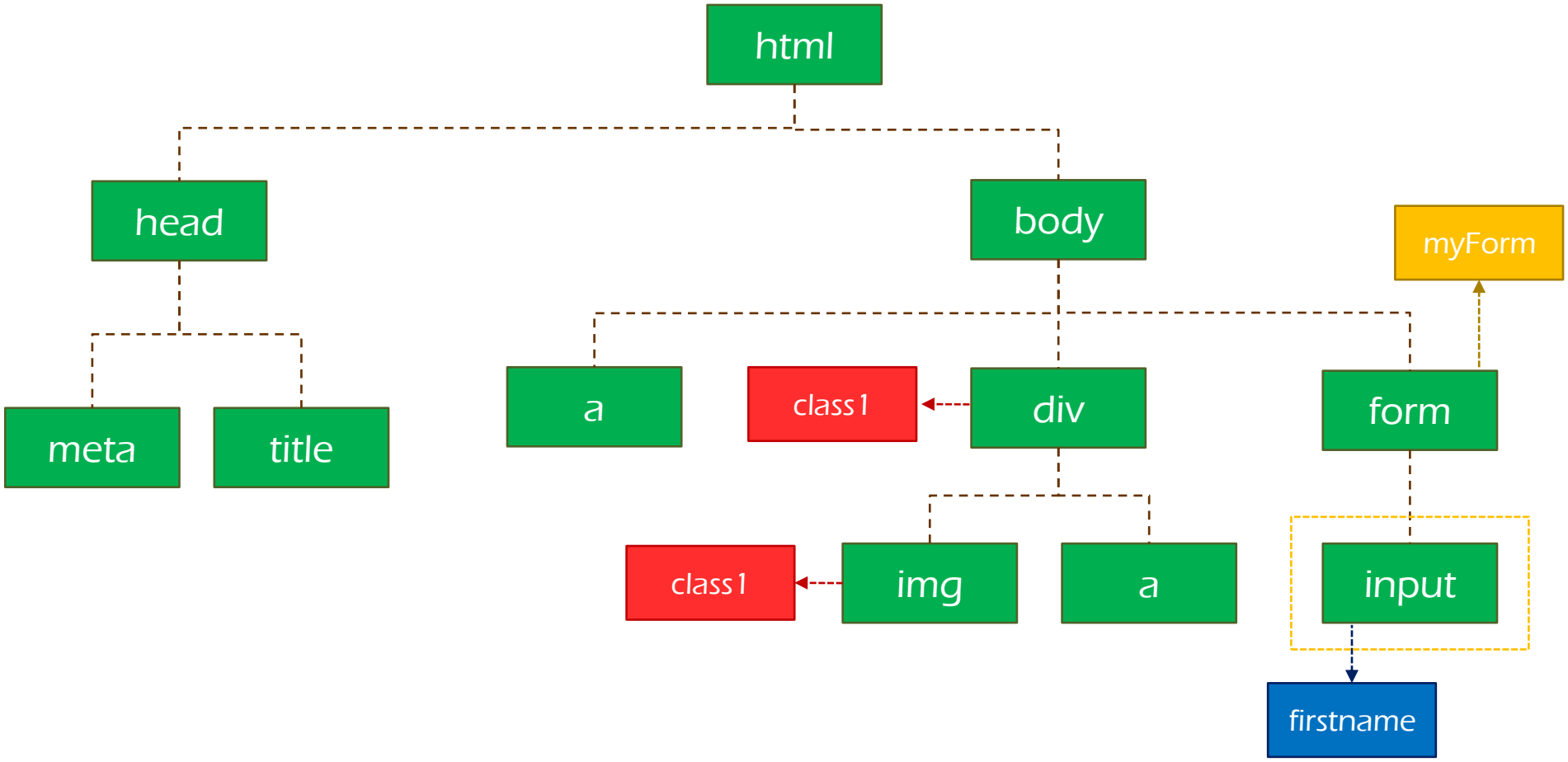
FIND ELEMENTS IN DOM

```
document.getElementById('myForm');
```



FIND ELEMENTS IN DOM

```
document.getElementsByName('firstname');
```



1 Creating The Element:

```
var paragraph = document.createElement("p");
```

p

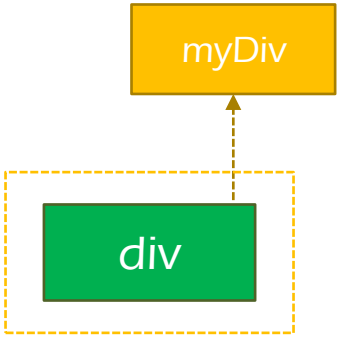
ADDING ELEMENTS | Append Child

1 Creating The Element:

```
var paragraph = document.createElement("p");
```

2 Adding this Element:

```
var myDiv = document.getElementById('myDiv');
```

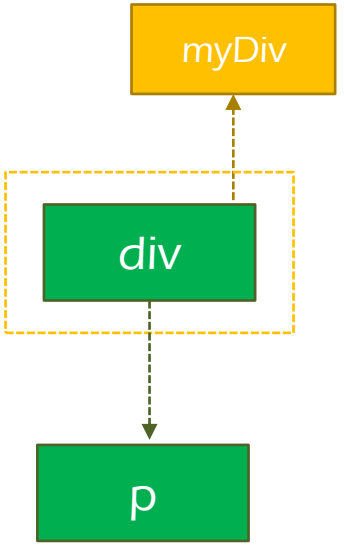


1 Creating The Element:

```
var paragraph = document.createElement("p");
```

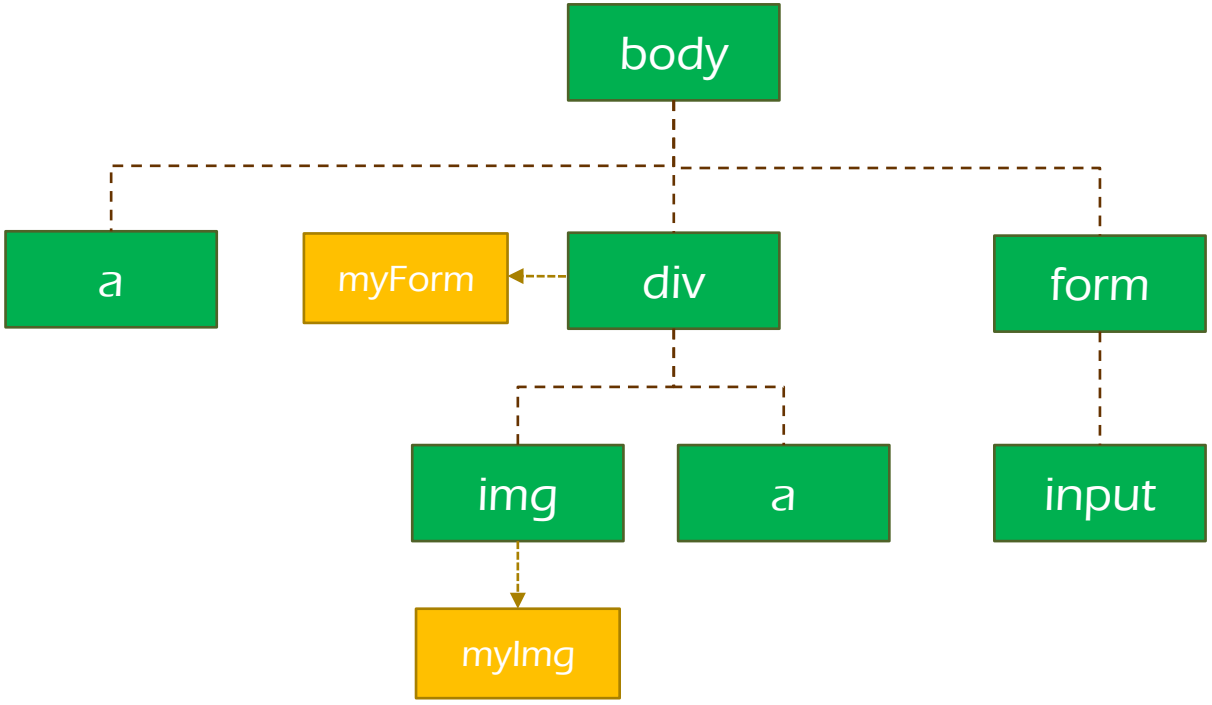
2 Adding this Element:

```
var myDiv = document.getElementById('myDiv');

myDiv.appendChild(paragraph);
```



ADDING ELEMENTS | Insert Before

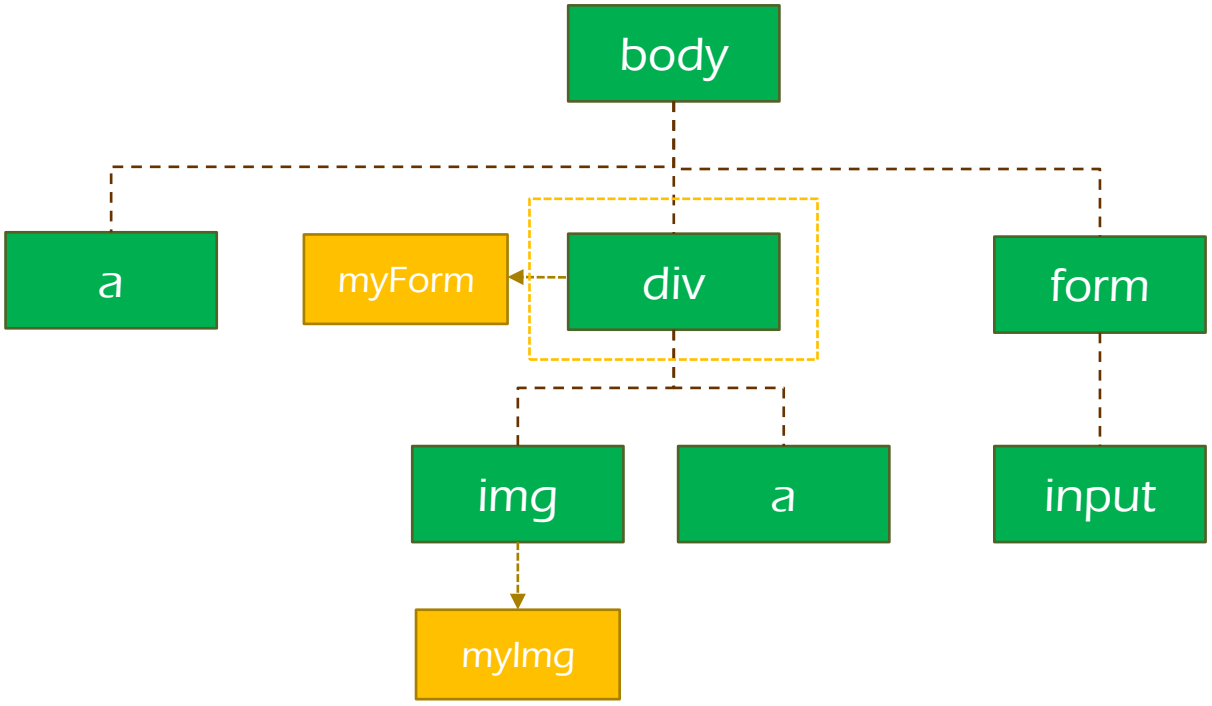
```
var paragraph = document.createElement("p");
```



ADDING ELEMENTS | Insert Before

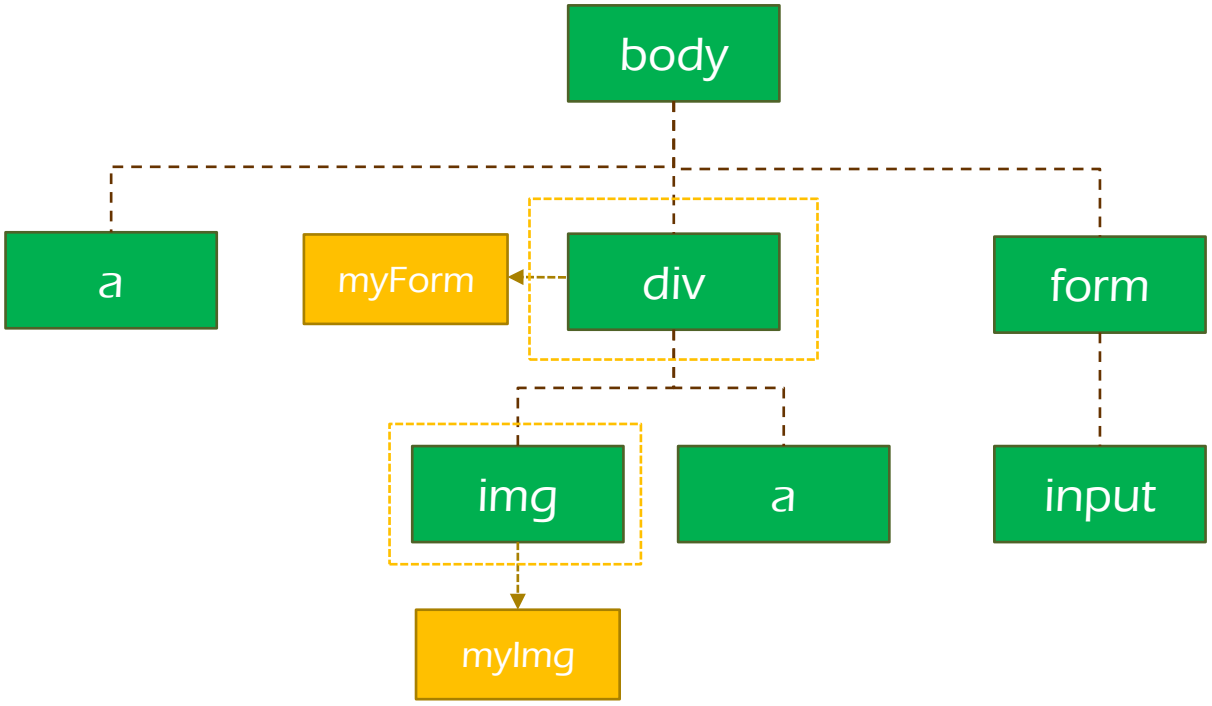
```
var paragraph = document.createElement("p");
var parent = document.getElementById('myDiv');
```

p



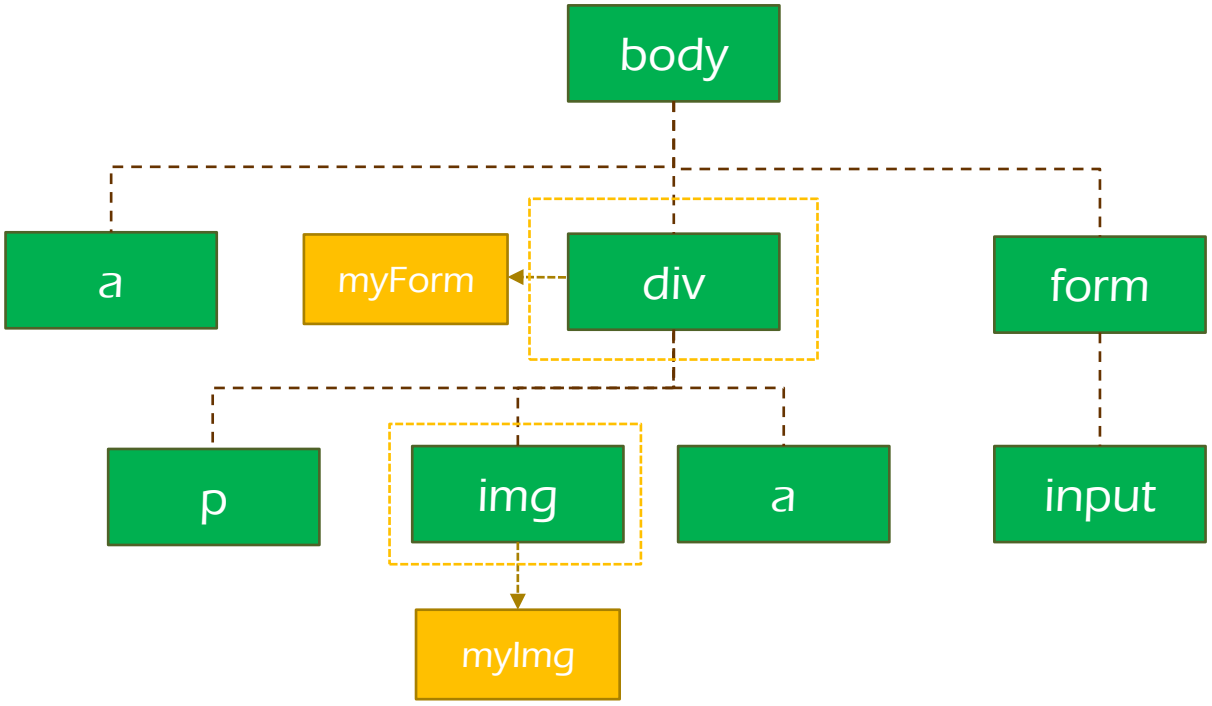
```
var paragraph = document.createElement("p");
var parent = document.getElementById('myDiv');
var child = document.getElementById('myImg');
```

p



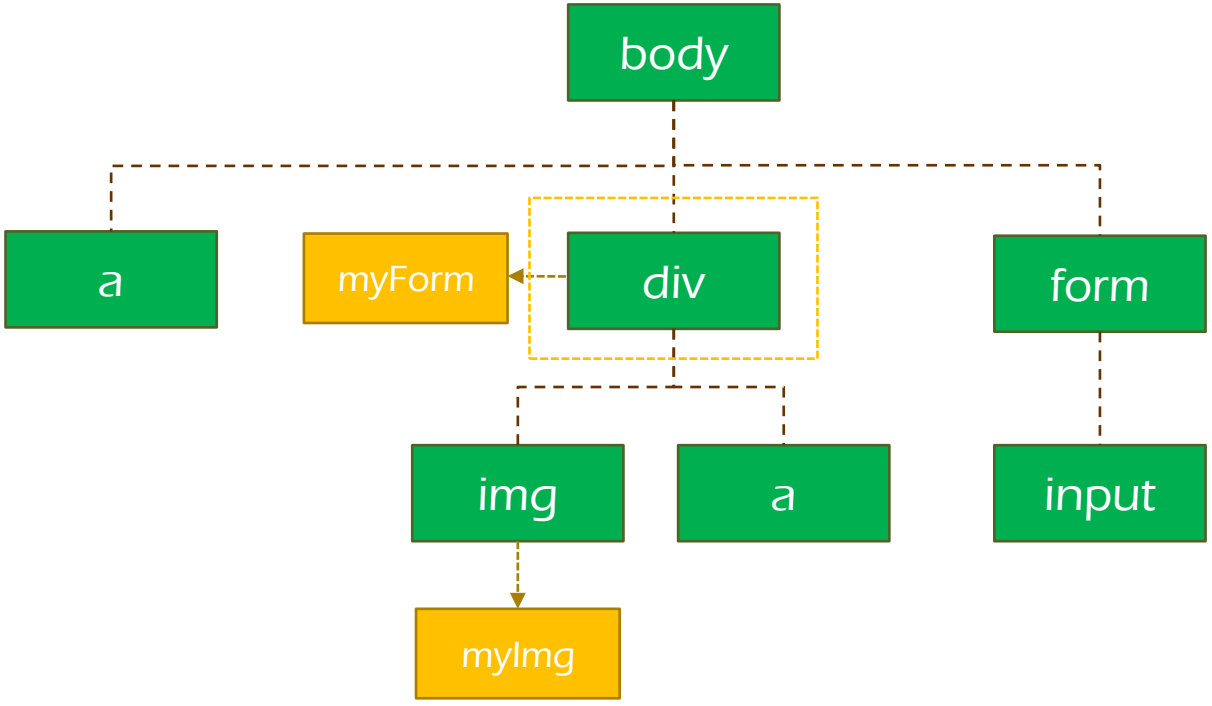
JS

```
var paragraph = document.createElement("p");
var parent = document.getElementById('myDiv');
var child = document.getElementById('myImg');
parent.insertBefore(paragraph, child);
```



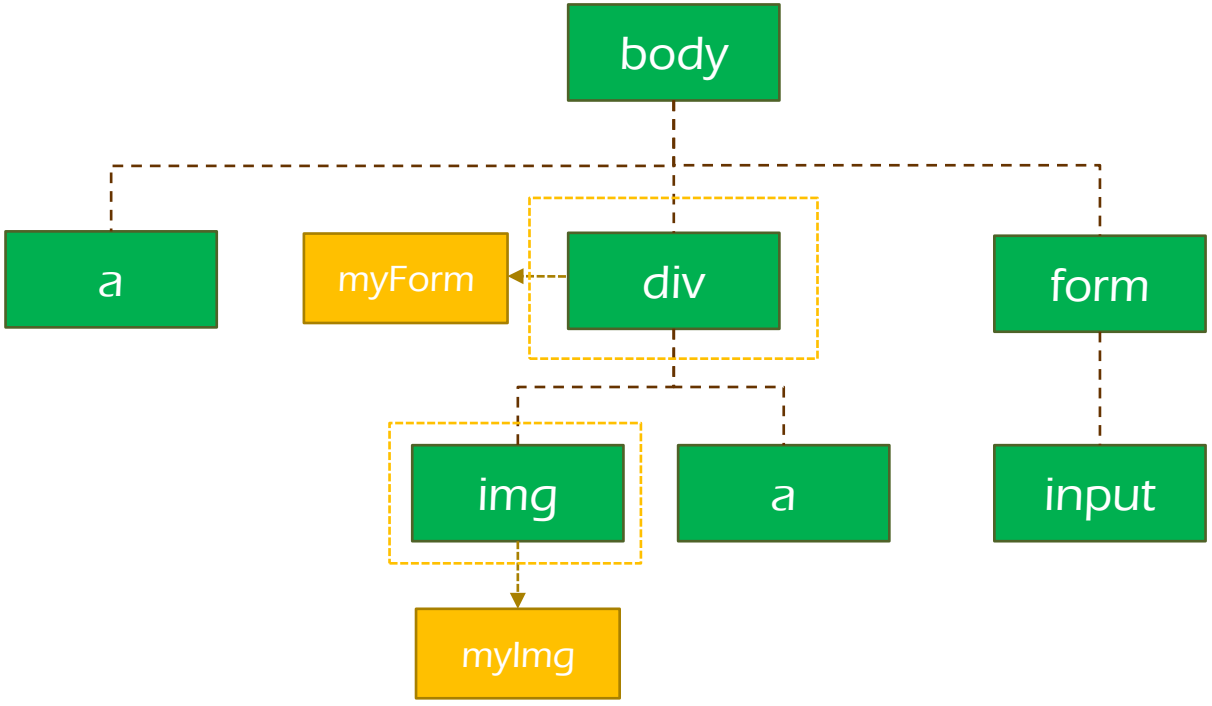
REMOVING ELEMENTS

```
var parent = document.getElementById('myDiv');
```



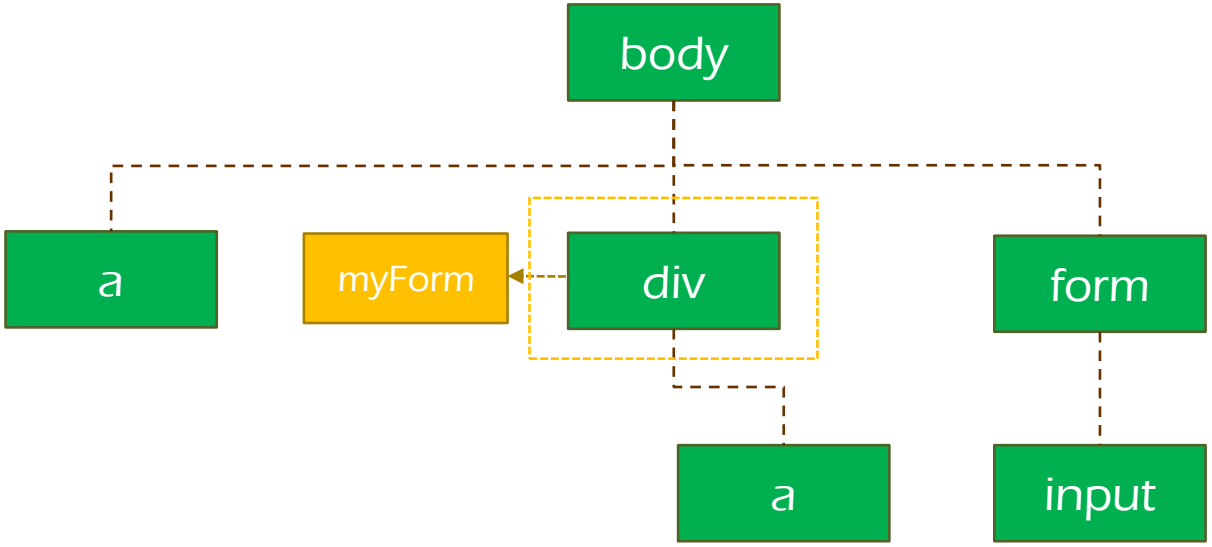
REMOVING ELEMENTS

```
var parent = document.getElementById('myDiv');

var child = document.getElementById('myImg');
```



REMOVING ELEMENTS

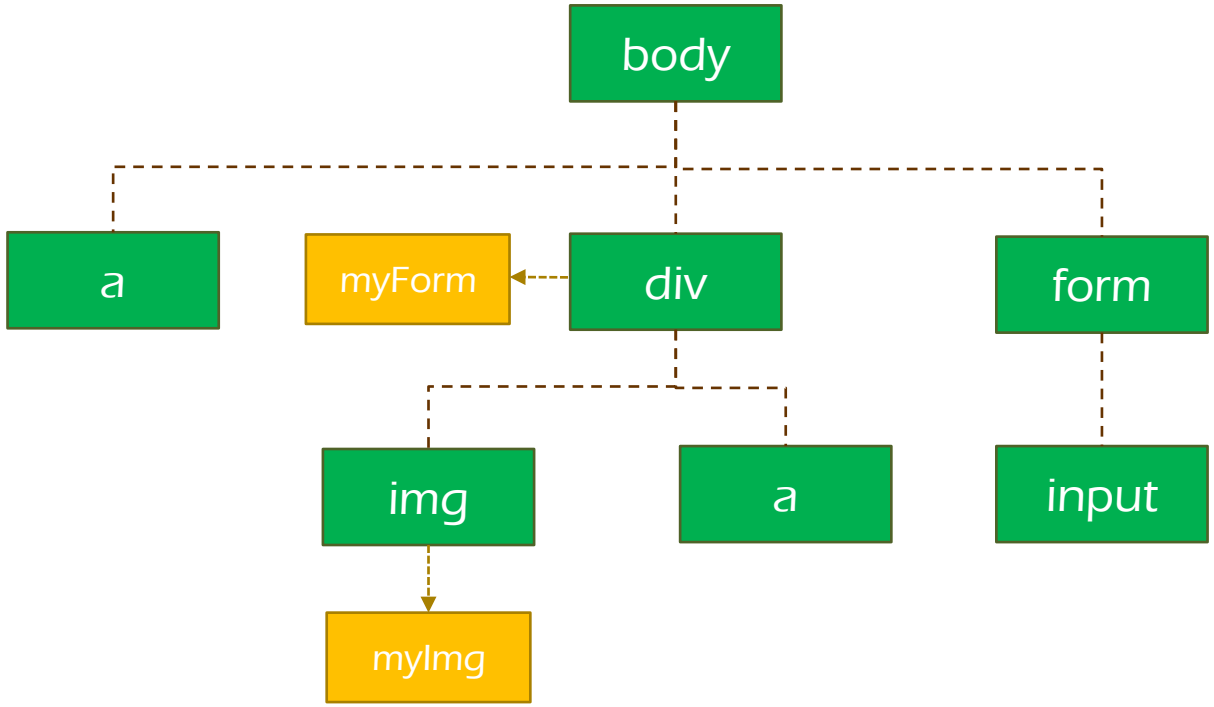
```
var parent = document.getElementById('myDiv');

var child = document.getElementById('myImg');

parent.removeChild(child);
```



REPLACING ELEMENTS

```
var paragraph = document.createElement("p");
```

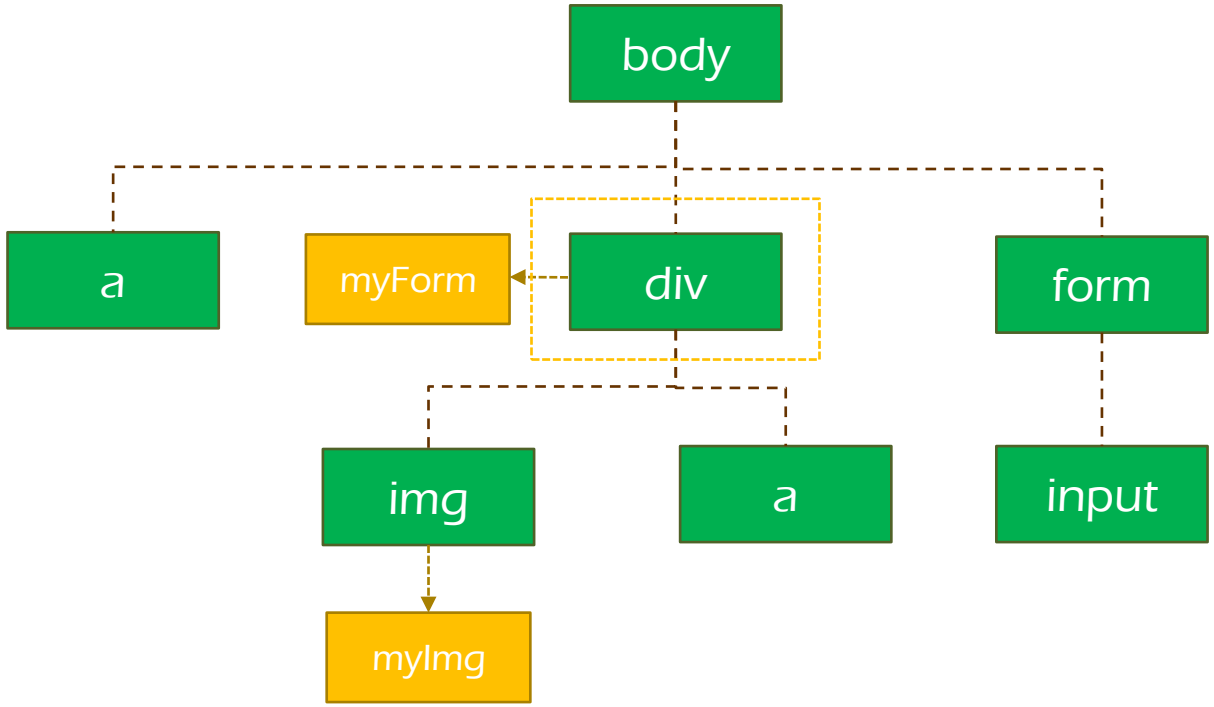
p



REPLACING ELEMENTS

```
var paragraph = document.createElement("p");

var parent = document.getElementById('myDiv');
```

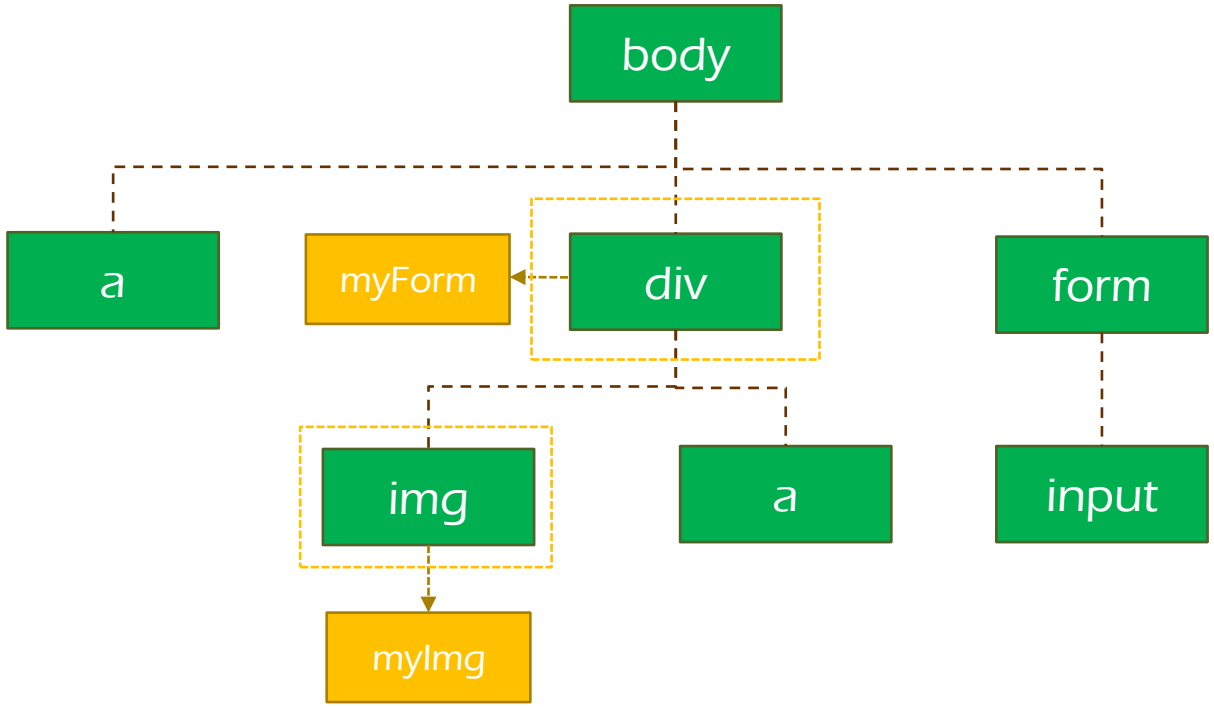
p



REPLACING ELEMENTS

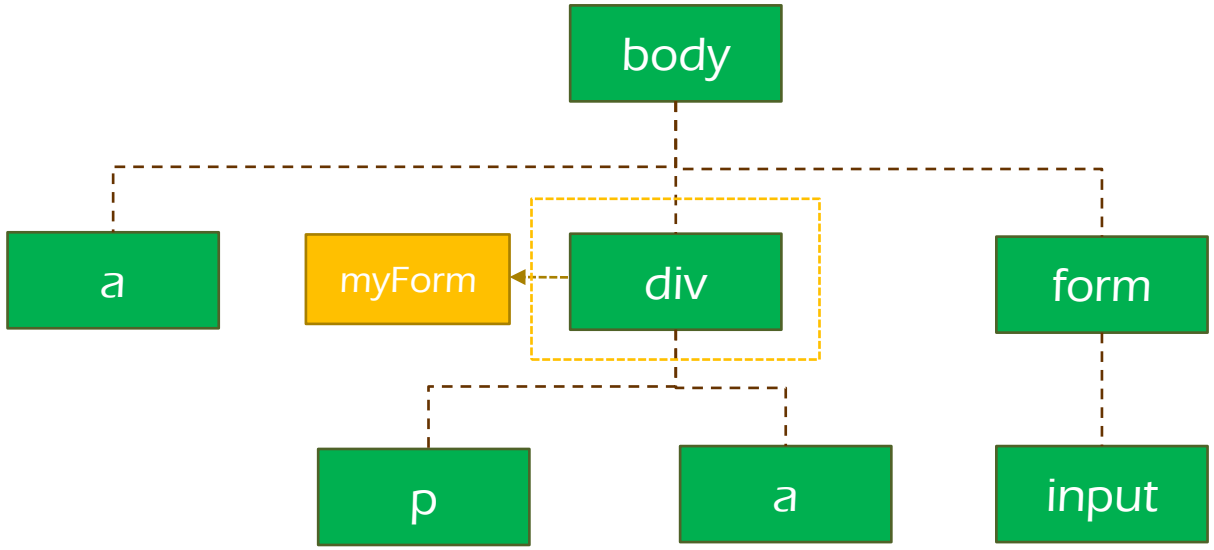
```
var paragraph = document.createElement("p");
var parent = document.getElementById('myDiv');
var child = document.getElementById('myImg');
```

p



REPLACING ELEMENTS

```
var paragraph = document.createElement("p");
var parent = document.getElementById('myDiv');
var child = document.getElementById('myImg');
parent.replaceChild(paragraph, child);
```



innerHTML

Getting the HTML inside an Element

```
var html = document.getElementById('myDiv').innerHTML;

console.log(html);
```

Setting the HTML inside an Element

```
document.getElementById('myDiv').innerHTML ='<new HTML Content/>'
```

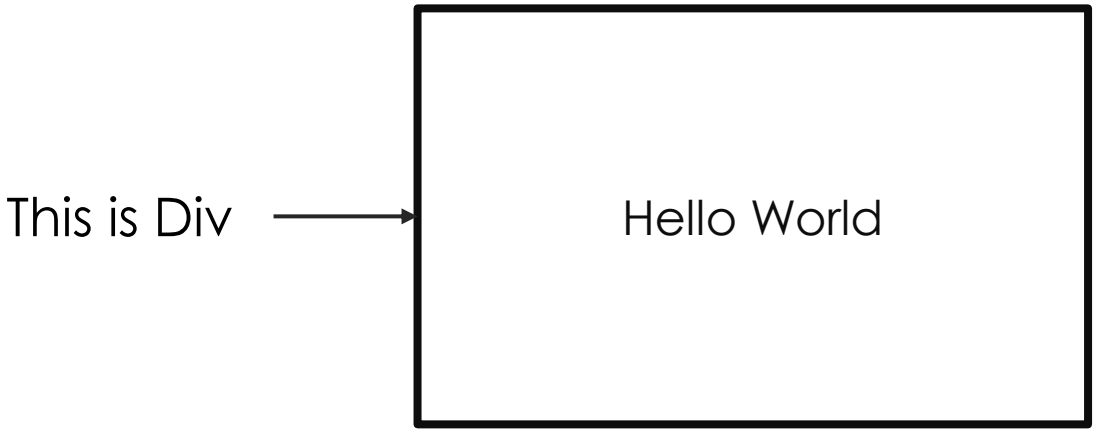
textContent

Getting the HTML inside an Element

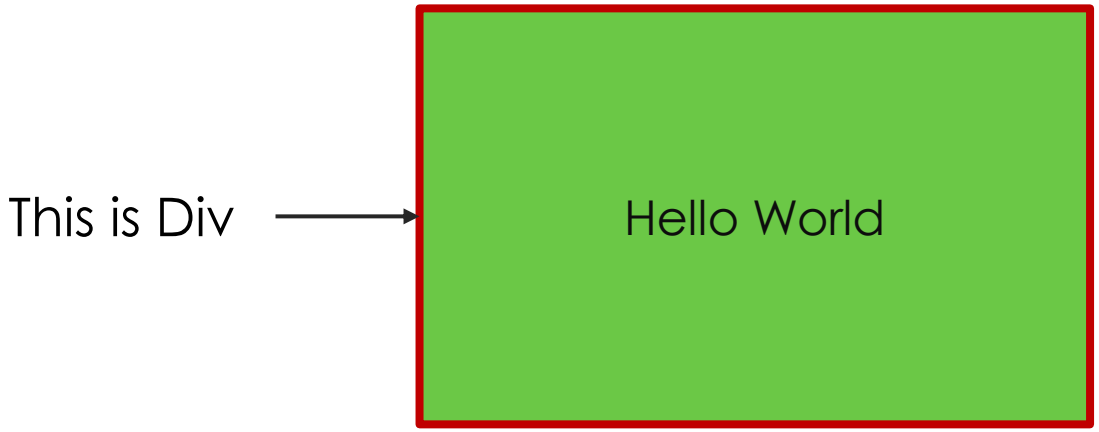
```
var html = document.getElementById('myDiv').textContent;

console.log(html);
```

Setting the HTML inside an Element

```
document.getElementById('myDiv').textContent ='hi there';
```




```
var div = document.getElementById('myDiv');
```




```
var div = document.getElementById('myDiv');

div.style.backgroundColor = 'green';

div.style.borderColor = 'red';
```

TREATING WITH ATTRIBUTES I

Getting an attribute value of an Element

```
var imgSource = document.getElementById('myImg').src;

console.log(imgSource); //lemon-juice.png
```

TREATING WITH ATTRIBUTES I

Getting an attribute value of an Element

```
var imgSource = document.getElementById('myImg').src;

console.log(imgSource); //lemon-juice.png
```

Setting an attribute value of an Element

```
document.getElementById('myImg').src ='orange-juice.png';
```

TREATING WITH ATTRIBUTES II

Getting an attribute value of an Element

```
var img = document.getElementById('myImg');
var imgSource = img.getAttribute('src');
console.log(imgSource); //lemon-juice.png
```

TREATING WITH ATTRIBUTES II

Getting an attribute value of an Element

```
var img = document.getElementById('myImg');
var imgSource = img.getAttribute('src');
console.log(imgSource); //lemon-juice.png
```

Setting an attribute value of an Element

```
img.setAttribute('src','orange-juice.png');
```

My Div

```
<div id="my-div" class="blue square"> My Div</div>
```

Getting class list of an Element

```
var div = document.getElementById('my-div');
var classes = div.classList;
console.log(classes); //[blue, square]
```


```
<div id="my-div" class="square"> My Div</div>
```

Setting class list of an Element

```
var div = document.getElementById('my-div');
div.classList.add('blue');
div.classList.remove('blue');
div.classList.toggle('red');
div.classList.toggle('red');
```

CREATING AND FORMING ELEMENTS

```
var article = document.createElement('p');
```

CREATING AND FORMING ELEMENTS

```
var article = document.createElement('p');
var content = document.createTextNode("I'm an article");
article.appendChild(content);
```

```
<p >
I'm an article
</p>
```

CREATING AND FORMING ELEMENTS

```
var article = document.createElement('p');
var content = document.createTextNode("I'm an article");
article.appendChild(content);
var myAttr = document.createAttribute('class');
myAttr.value = 'make-me-bold';
article.setAttributeNode(myAttr);
```

```
<p class='make-me-bold'>
I'm an article
</p>
```

BOM

The **Browser Object Model** (BOM) allows **JavaScript** to talk to the browser.

WINDOW OBJECT

The **window** object represents the browser's window.

- **All** global JavaScript objects, functions, and variables automatically become members of the window object.

```
alert( "Hello")      ===   window.alert("hello")

document   ===   window.document
```

- **Global variables** are properties of the window object.

- **Global functions** are methods of the window object.

THANK YOU

ahmedmowd@gmail.com