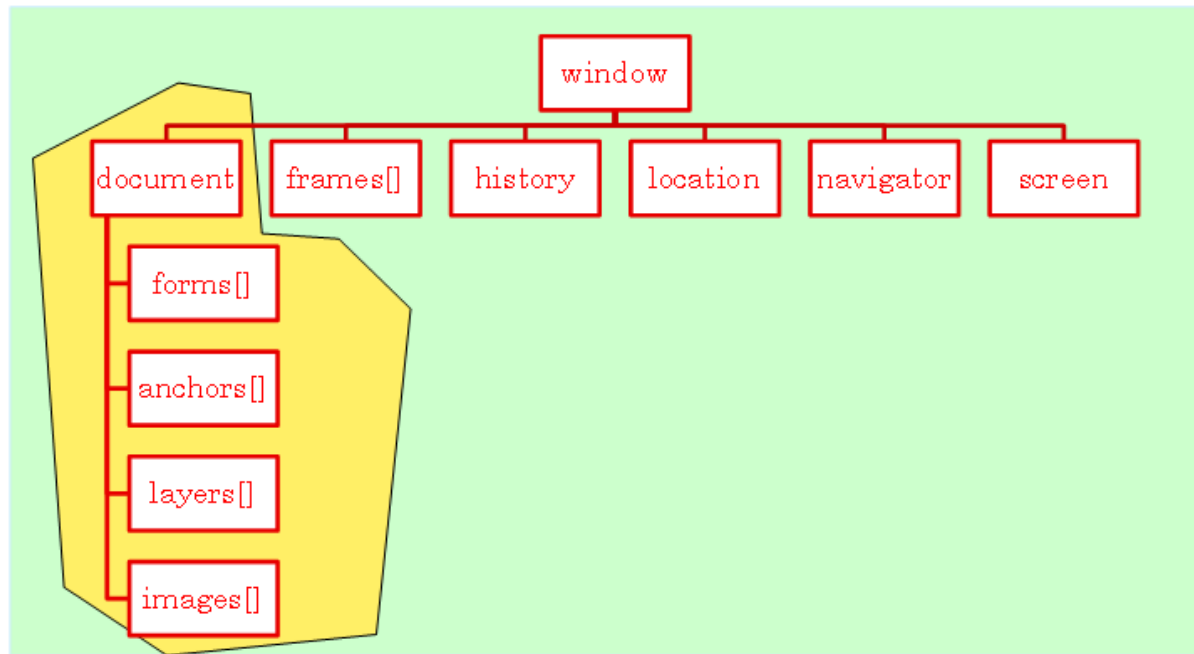


DOM

DOM과 BOM

- HTML 문서를 객체로 표현한 것 - DOM
- 웹브라우저를 객체로 표현한 것 - BOM(Browser Object Model)

BOM(Browser Object Model)



DOM(Document Object Model)

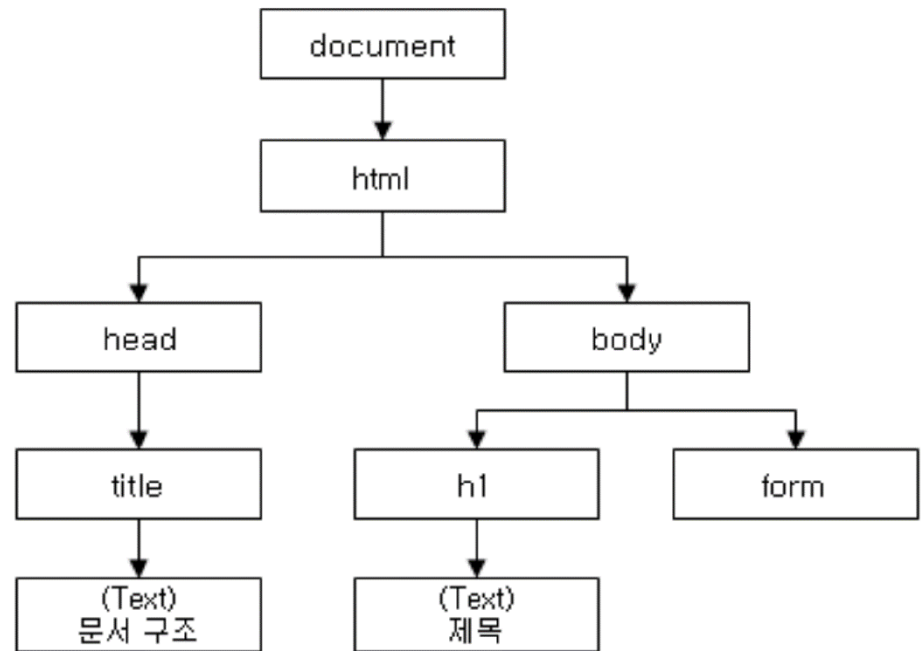
문서 객체 모델(DOM)

- Document Object Model
 - 문서를 객체로 표현하기 위한 표준(W3C)
 - 모든 웹브라우저에서 지원
 - 대부분의 언어에서 API지원
 - HTML 문서의 계층적인 구조를 트리(tree)로 표현
 - 서버의 XML 응답 문서로부터 데이터 추출
- DOM의 사용이유?
 - 사용자의 동작에 바로 응답하고,
 - 필요없는 리로드를 제거
 - 보다 역동적인 웹 애플리케이션 제작이 용이

HTML & DOM

- HTML 문서: 하나의 노드-트리로 간주
- 그 트리에 있는 모든 노드들은 서로 간에 계층적 관계를 가짐

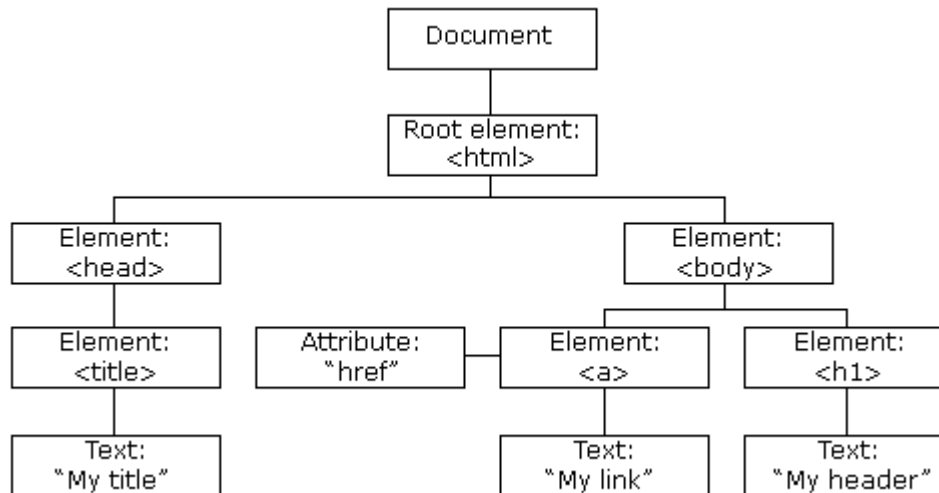
```
<html>  
<head>  
  <title>문서구조</title>  
</head>  
<body>  
  <h1>제목</h1>  
  <form></form>  
</body>  
</html>
```



DOM Tree

HTML DOM

- HTML 문서에 접근하고 조작하기 위한 표준 방법을 정의(API 정의)
- DOM: HTML 문서를 트리-구조로 표시



DOM Tree



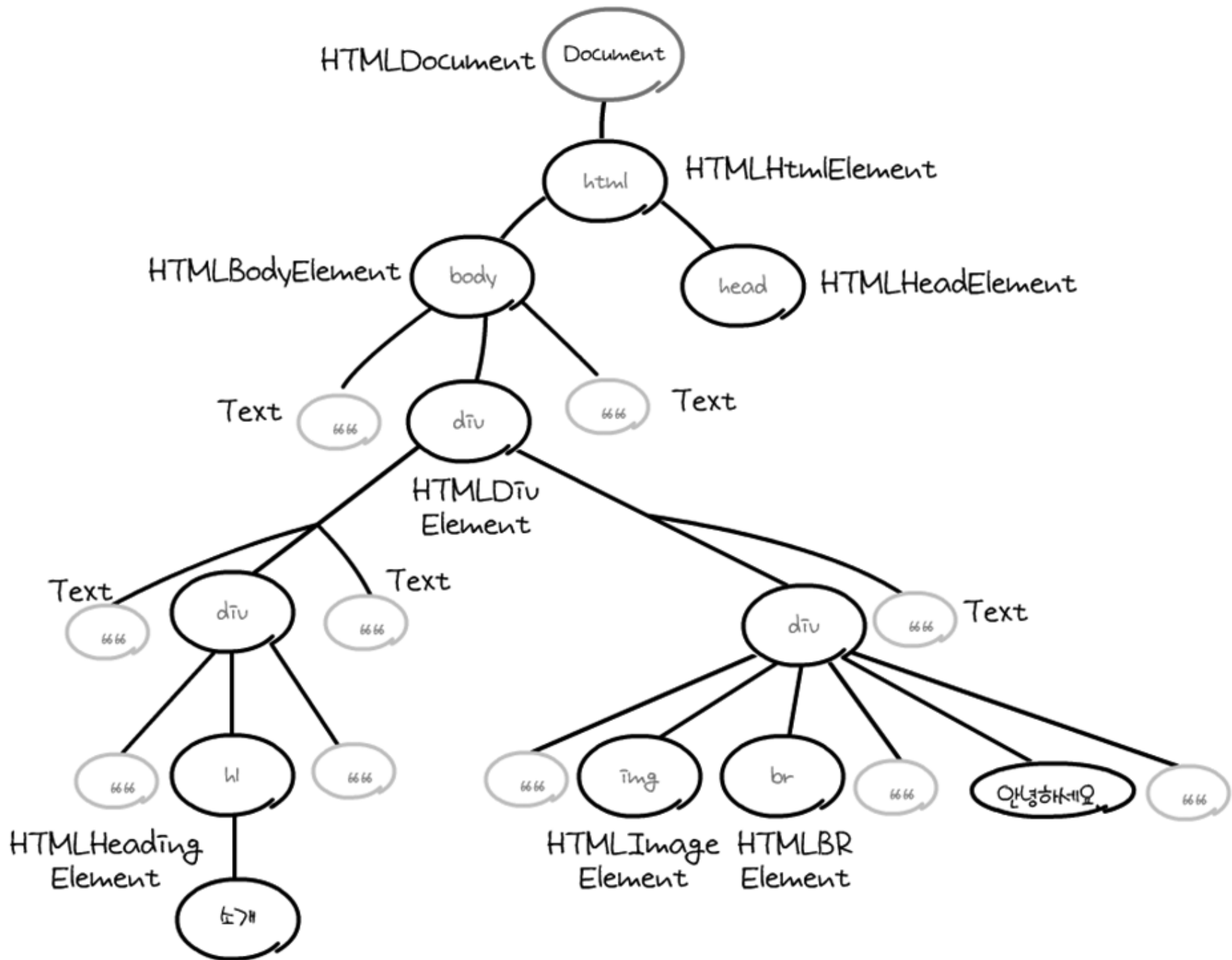
1. HTML 페이지
구독

2. 파싱단계에서 거쳐
웹페이지에
작성된 태그와 1:1
대칭되는 DOM
객체 생성

3. 브라우저
화면에 출력

```
<html>
<head>
<title>DOM과 HTML 페이지와의 관계</title>
</head>
<body>
<div id="intro">
<div class="header">
<h1>소개</h1>
</div>
<div class="content">
<br>
안녕하세요.
</div>
</div>
</body>
</html>
```





HTML 요소 찾기

- 동적인 웹페이지를 작성하려면 원하는 요소를 찾아야 한다.
 - id로 찾기
 - getElementById
 - 태그 이름으로 찾기
 - getElementsByTagName

HTML 요소 찾기

- "document" 객체: 자바스크립트에서 웹 브라우저의 DOM 트리를 사용하도록 해줌
- 사용예:
 - var cashE1 =
document.getElementById("cash");
 - var phone =
document.getElementById("phone").value

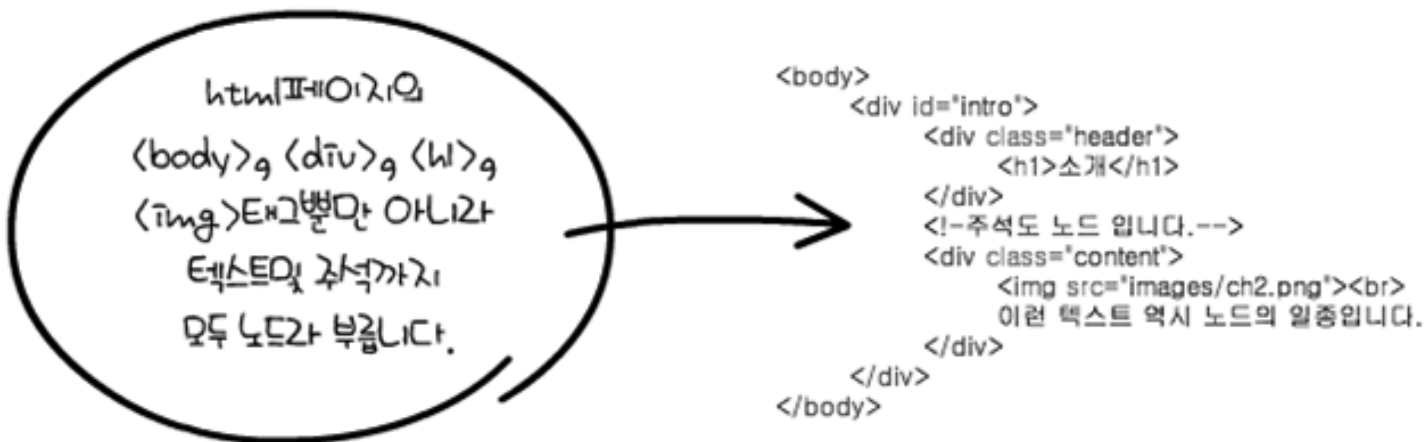
DOM Nodes

- HTML 문서의 모든 것이 하나의 DOM 노드이다.
 - 전체 문서: 하나의 문서 노드(document)
 - HTML 태그: 하나의 엘리먼트 노드
 - HTML 속성: 애트리뷰트 노드
 - text: 텍스트 노드

```
<html>                                     ← 루트 노드 (2개의 자식 노드를 가짐)
  <head>
    <title>DOM Tutorial</title>
  </head>
  <body>
    <h1>DOM Lesson one</h1>
    <p id="hello">Hello world!</p>
  </body>
</html>
```

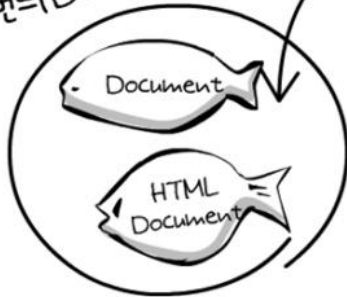
DOM Nodes

- Document node(문서 노드)
 - 전체 HTML 문서 그 자체를 의미
- Element node(엘리먼트 노드)
 - HTML 문서내의 각 태그(tag)를 표현, <a>,
- Attribute node(속성 노드)
 - HTML 태그의 속성을 표현하는 노드, href, src
- Text node(텍스트 노드)
 - 각 태그가 포함하는 텍스트 값(문자열)을 의미



DOM Nodes

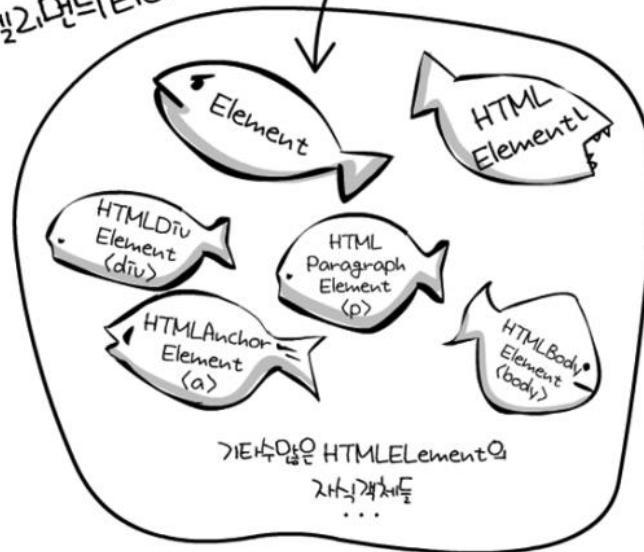
도큐먼트(Document) 노드



문서 노드 타입에는
객체가 많지 않군요.

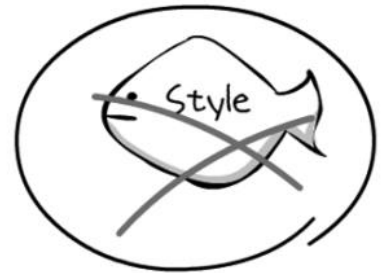
이 객체들의
집합을 우리는
엘리먼트 노드이라
부릅니다.

엘리먼트(Element) 노드



주의!

스타일은 노드 타입이
아닙니다.



속성(Attribute) 노드

```
<div id="content" data-role="content">
</div>
```

id, data-role 등은 각각
속성 노드로 만들어져
있습니다.

또한, 속성 노드는 엘리먼트의
프로퍼티입니다.

텍스트 노드

이 내용이 바로
텍스트 노드입니다.



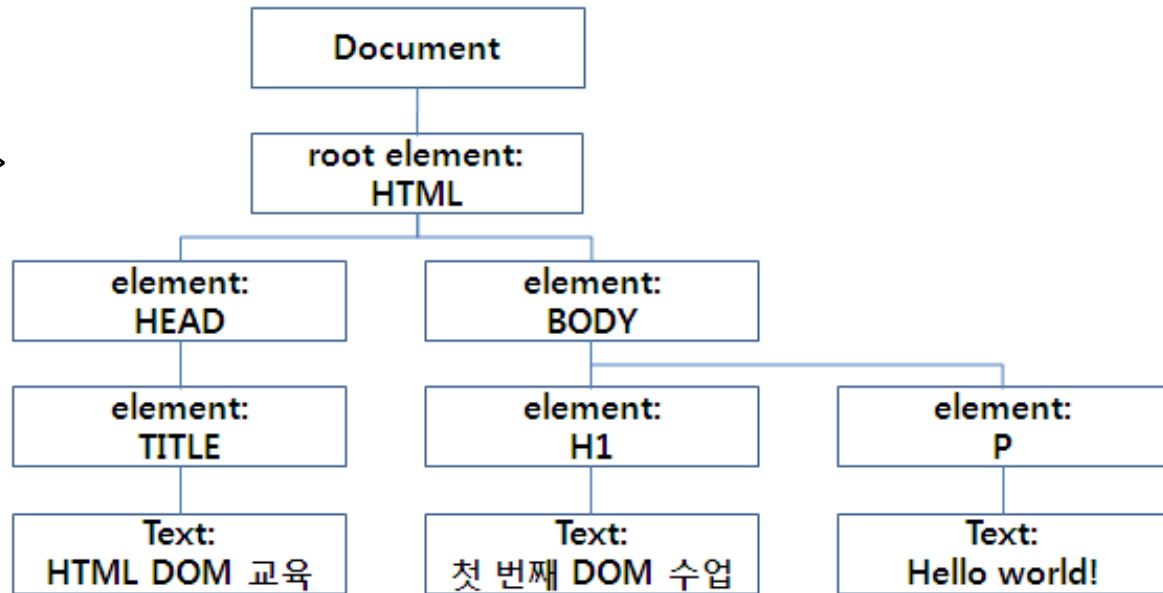
DOM Nodes

- DOM 처리에서 가장 흔히 범하는 오류:
 - 엘리먼트 노드가 텍스트를 포함하고 있다고 오해
 - 엘리먼트 노드의 텍스트: 텍스트 노드에 저장됨
 - `<title>DOM Tutorial</title>`
 - `<title>` 엘리먼트 노드는 "DOM Tutorial"이라는 값을 가지는 텍스트 노드를 가짐
 - "DOM Tutorial"은 `<title>` 엘리먼트 노드의 값이 아님
- HTML DOM에서 텍스트 노드의 값
 - `innerHTML` 프로퍼티를 통해 읽기/쓰기 가능

DOM Tree – HTML File

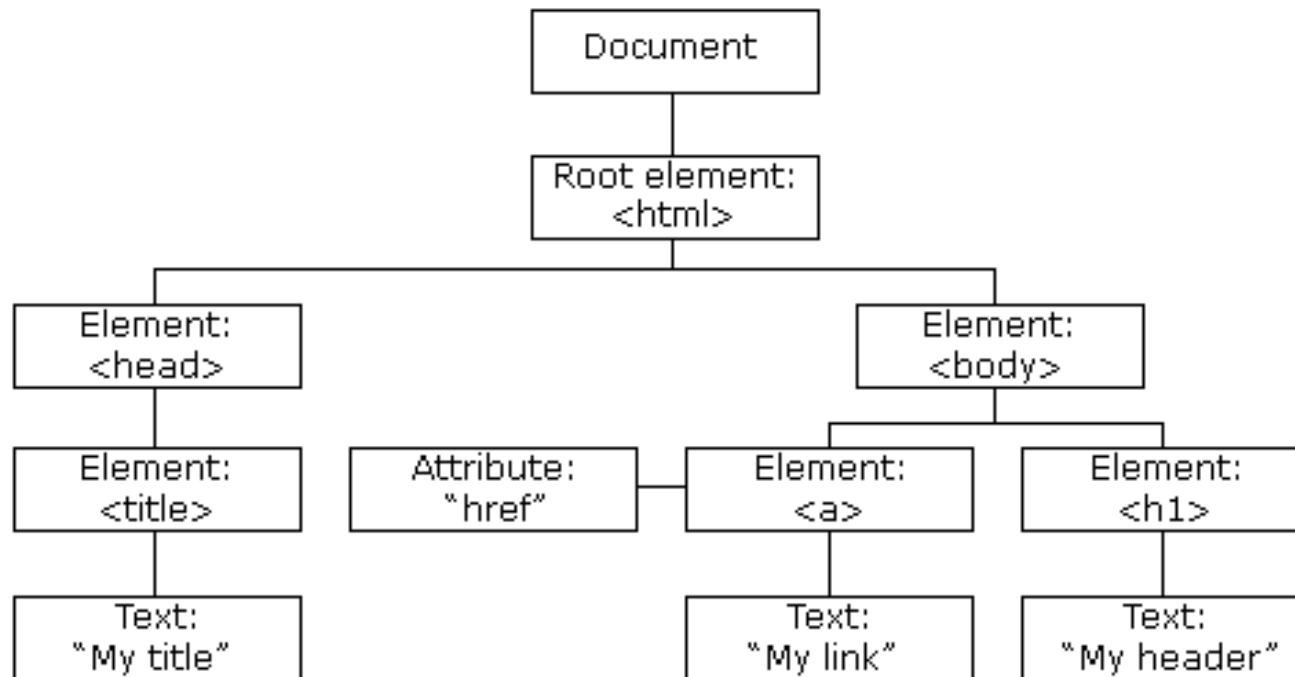
- 다음 HTML File의 DOM Tree ?

```
<html>
<head>
<title>HTML DOM 교육</title>
</head>
<body>
<h1>첫 번째 DOM 수업</h1>
<p>Hello world!</p>
</body>
</html>
```



DOM Tree – HTML File

- 다음 DOM Tree의 HTML File내용은?



Tree 관련 용어

- root node, child nodes, parent node, sibling nodes, firstChild, lastChild, previousSibling, nextSibling

<html>

← root node (2개의 child node를 가짐)

<head>

<title>DOM Tutorial</title>

</head>

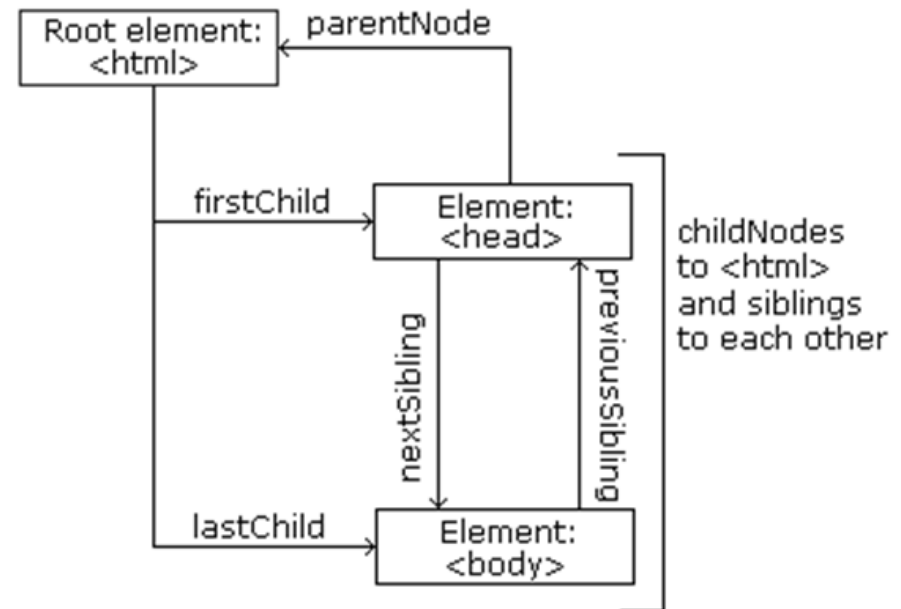
<body>

<h1>DOM Lesson one</h1>

<p id="hello">Hello world!</p>

</body>

</html>



Tree 관련 용어

- <html> 노드: 부모 노드를 가지지 않는 root node
- <head>와 <body> 노드의 parent node: <html> node
- "Hello world!" text node의 parent node : <p> node
- <html> 노드의 두 개의 child nodes: (<head>, <body>)
- <head> 노드의 하나의 child node: <title> node
- <title> 노드의 하나의 child node: "DOM Tutorial" text node
- <h1>노드와 <p> 노드: sibling node, <body>의 child nodes

```
<html>
  <head>
    <title>DOM Tutorial</title>
  </head>
  <body>
    <h1>DOM Lesson one</h1>
    <p>Hello world!</p>
  </body>
</html>
```

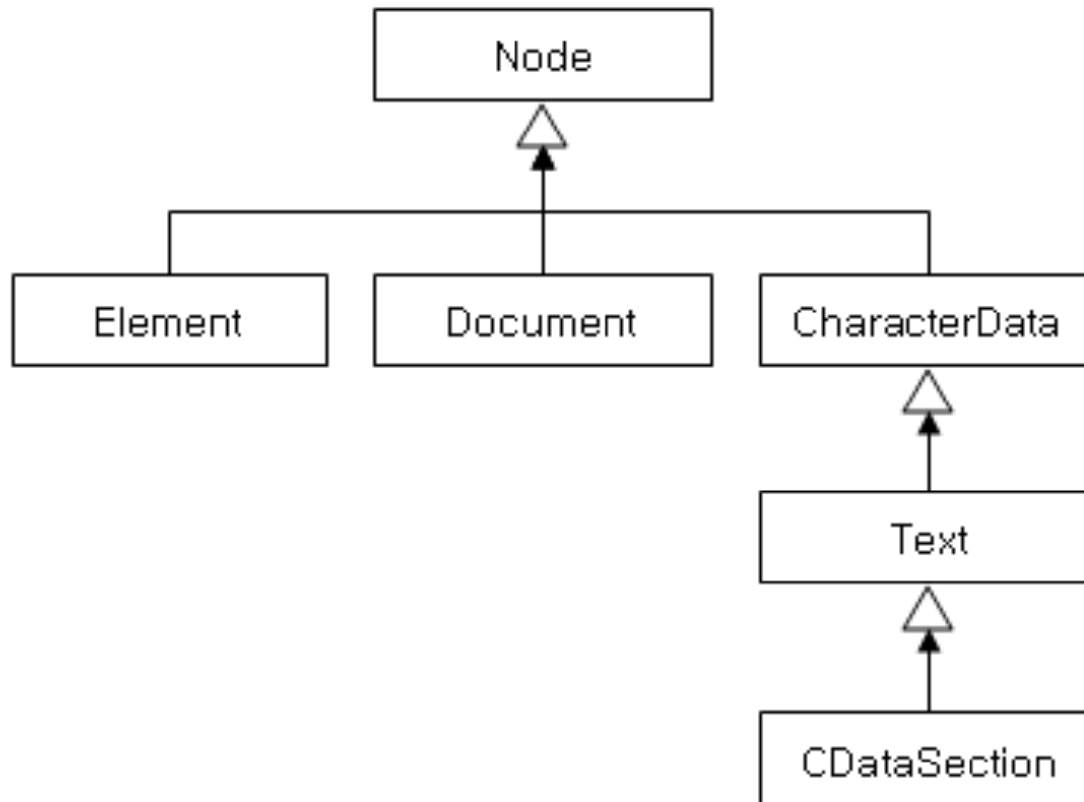
- <head> 노드: <html> 노드의 first child node
- <body> 노드: <html> 노드의 last child node
- <h1> 노드: <body> 노드의 첫 번째 자식
- <p> 노드: <body> 노드의 마지막 자식

DOM Tree 순회

- `childNodes`: 한 요소의 모든 자식 요소에 접근가능, 배열
- `firstChild`: 'childNodes' 배열의 첫번째 요소-`childNodes[0]`
- `lastChild`: 'childNodes' 배열의 마지막 요소-`childNodes[childNodes.length-1]`
- `parentNode`: 현재 node의 parent node
- `previousSibling`: 현재 node의 이전 형제 node
- `nextSibling`: 현재 node의 다음 형제 node

DOM API 인터페이스

- DOM API의 인터페이스



DOM API 인터페이스

- Node: 문서 Tree상의 각 노드를 의미
- Document
 - 전체 문서, document객체의 타입
- Element: 각 태그, <books>, <book>
- Text: 문자열 테이터 노드
 - <title>웹 표준</title>에서 ‘웹 표준’
- CDataSection
 - XML 문서의 CDATA 영역의 문자열 값

DOM API 인터페이스

- Node 인터페이스의 주요 프로퍼티

프로퍼티 타입	프로퍼티 이름	설명
String	nodeName	노드의 이름
String	nodeValue	노드의 값
unsigned short	nodeType	노드 타입
Node	parentNode	부모 노드
NodeList	childNodes	자식 노드 목록
Node	firstChild	첫번째 자식 노드
Node	lastChild	마지막 자식 노드
Node	previousSibling	현재 노드와 같은 부모를 갖는 자식 노드들 중 현재 노드 이전의 자식노드
Node	nextSibling	현재 노드와 같은 부모를 갖는 자식 노드 중 현재 노드 다음의 자식노드
Document	ownerDocument	이 노드가 포함된 Document 객체
NodeList	attributes	엘리먼트 노드의 애트리뷰트 목록
String	innerHTML	노드의 텍스트 값(HTML DOM)

DOM API 인터페이스

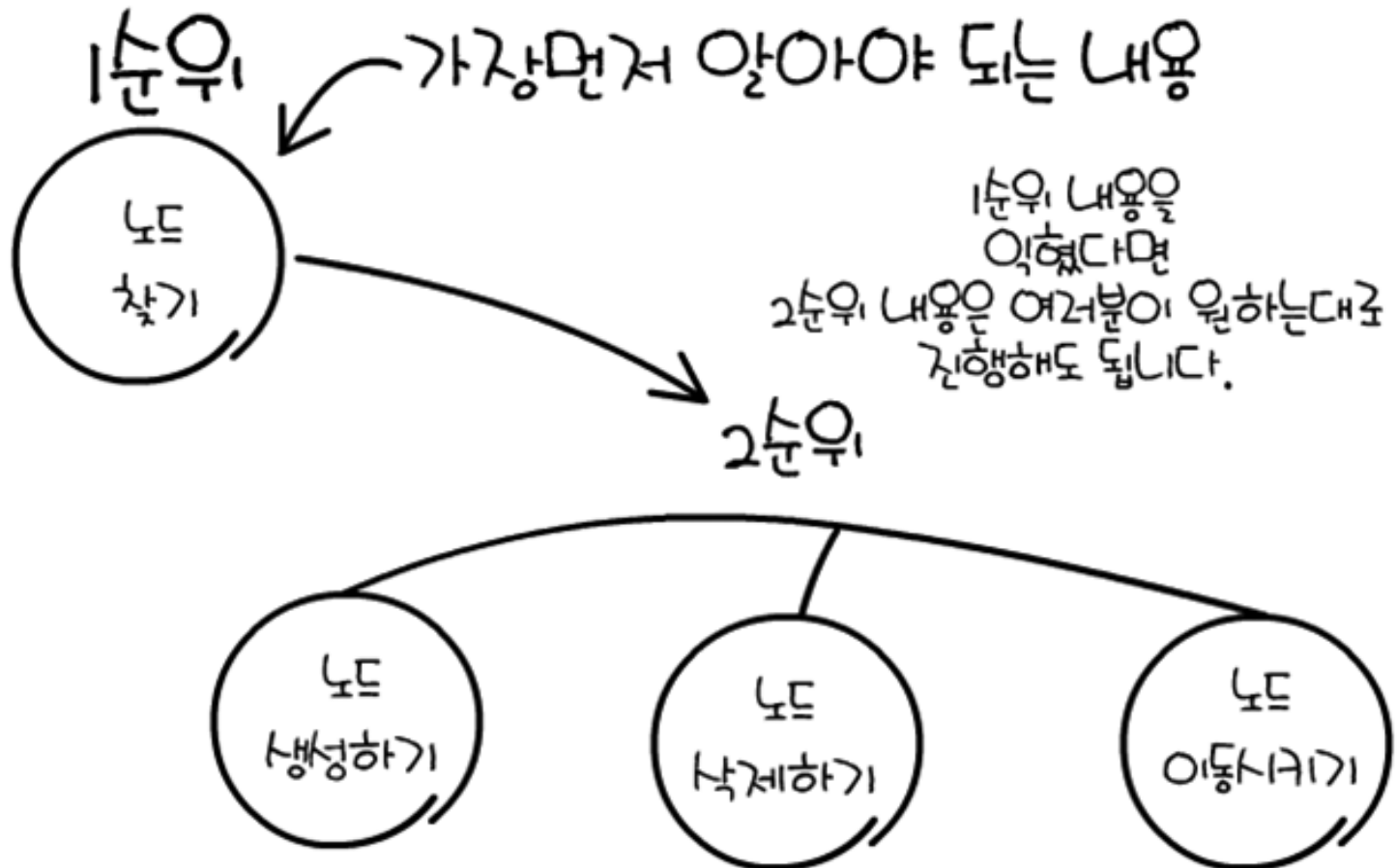
- `childNodes`
 - `NodeList`
- `NodeList` 프로퍼티&함수
 - `length`: `NodeList`에 저장된 노드의 개수
 - `item(i)`: 인덱스 `i`에 저장된 노드를 반환(`i`는 0부터 시작)

nodeType, nodeName, nodeValue

구분	nodeType	nodeName	nodeValue
Element	ELEMENT_NODE	태그명	null
Attribute	ATTRIBUTE_NODE	속성명	속성값
Text	TEXT_NODE	#text	문자열
CDATA	CDATA_SECTION_NODE	#cdata-section	문자열
Entity	ENTITY_NODE	참조된 엔티티명	null
Entity Reference	ENTITY_REFERENCE_NODE	선언된 엔티티명	null
Processing Instruction	PROCESSING_INSTRUCTION_NODE	PI 이름	PI이름을 제외한 전체내용
comment	COMMENT_NODE	#comment	주석내용
Document	DOCUMENT_NODE	#document	null
Document Type	DOCUMENT_TYPE_NODE	루트 태크명	null
Notation	NOTATION_NODE	Notation선언명	null
DOCUMENT FRAGMENT	DOCUMENT_FRAGMENT	#document-fragment	null

Node 다루기

노드 다루기 핵심내용 진행단계



Node 찾기

<u><code>document.getElementById()</code></u>	Returns the element that has the ID attribute with the specified value
<u><code>document.getElementsByClassName()</code></u>	Returns a NodeList containing all elements with the specified class name
<u><code>document.getElementsByName()</code></u>	Returns a NodeList containing all elements with a specified name
<u><code>document.getElementsByTagName()</code></u>	Returns a NodeList containing all elements with the specified tag name

form 태그의 node 찾기

형식	document.폼객체명(form태그의name속성값).속성 document.폼객체명(form태그의name속성값).메소드
속성	action: 서비스 요청 url elements: form태그의 자식 노드들의 배열 name: form태그의 name속성값 target: form태그의 target속성값 method: form태그의 method속성값
메소드	submit(): 입력 내용을 서버로 전송 reset(): 입력 내용의 초기화

<form name='userinfo' ...> ... </form>인 경우

document.userinfo.elements[1]

document.userinfo.name

document.userinfo.submit()

form 태그의 node 찾기

- <input type=''>: text or password, <textarea>

형식	document.폼객체명.text객체명.속성 document.폼객체명.text객체명.메소드
속성	value: 입력창에 입력한 문자열 defaultValue: 기본적으로 설정되어 있는 문자열
메소드	blur(): 입력창에 focus를 제거 focus(): 입력창에 focus를 주어 입력창에 커서를 표시 eval(): 입력창의 수식을 평가(계산, 실행)함 select(): 입력창에 입력한 문자열을 모두 선택

```
<form name='userinfo' ...>  
  <input type='text' name='id' ... />
```

</form>인 경우

```
document.userinfo.id.focus()  
document.userinfo.id.value  
document.userinfo.id.select()
```

form 태그의 node 찾기

- <input type=''>: checkbox

형식	document.폼객체명.checkbox객체명.속성 document.폼객체명.checkbox객체명.메소드
속성	value: input 태그의 value 속성값 checked: 체크여부를 설정하거나 반환
메소드	blur(): 입력창에 focus를 제거 focus(): 입력창에 focus를 주어 입력창에 커서를 표시 click(): 체크박스를 클릭

```
<form name='userinfo' ...>
```

```
  <input type='checkbox' name='interest' ... />
```

```
</form>인 경우
```

```
document.userinfo.interest.focus()
```

```
document.userinfo.interest.checked
```

```
document.userinfo.interest.click()
```

form 태그의 node 찾기

- <input type='': radio

형식	document. 폼객체명 .radio 객체명 .속성 document. 폼객체명 .radio 객체명 .메소드
속성	value: input 태그의 value 속성값 checked: 체크여부를 설정하거나 반환 length: 같은 그룹 안에 있는 라디오 버튼의 갯수
메소드	blur(): 입력창에 focus를 제거 focus(): 입력창에 focus를 주어 입력창에 커서를 표시 click(): 체크박스를 클릭

<form name='userinfo' ...>

 <input type='radio' name='gender' ... />

</form>인 경우

 document.**userinfo**.**gender**.focus()

 document.**userinfo**.**gender**.checked

 document.**userinfo**.**gender**.click()

form 태그의 node 찾기

- <select>

- 콤보박스 내의 값을 다루기 위해서는 `options[]` 배열 사용

형식	document.폼객체명.select객체명.속성 document.폼객체명.select객체명.메소드
속성	name: name속성 selectedIndex: 선택된 항목의 index번호 options: <option>태그 객체들의 배열 length: 콤보박스내의 리스트 항목 개수
메소드	blur(): 입력창에 focus를 제거 focus(): 입력창에 focus를 주어 입력창에 커서를 표시

```
<form name='f1' >
  <select name="s1" >
    <option value="menu1" selected>메뉴 1</option>
    <option value="menu2">메뉴 2</option>
    <option value="menu3">메뉴 3</option>
  </select>
</form>
```

document.f1.s1.length
document.f1.s1.options[document.f1.s1.selectedIndex].value