LEARNING SPOONS

LEARNING SPOONS ONLINE

자바스크립트 기초

DOM 프로그래밍

Copyright ⓒ LEARNING SPOONS Corp. All Rights Reserved. 무단전재 및 배포금지

LEARNING SPOONS

Unit

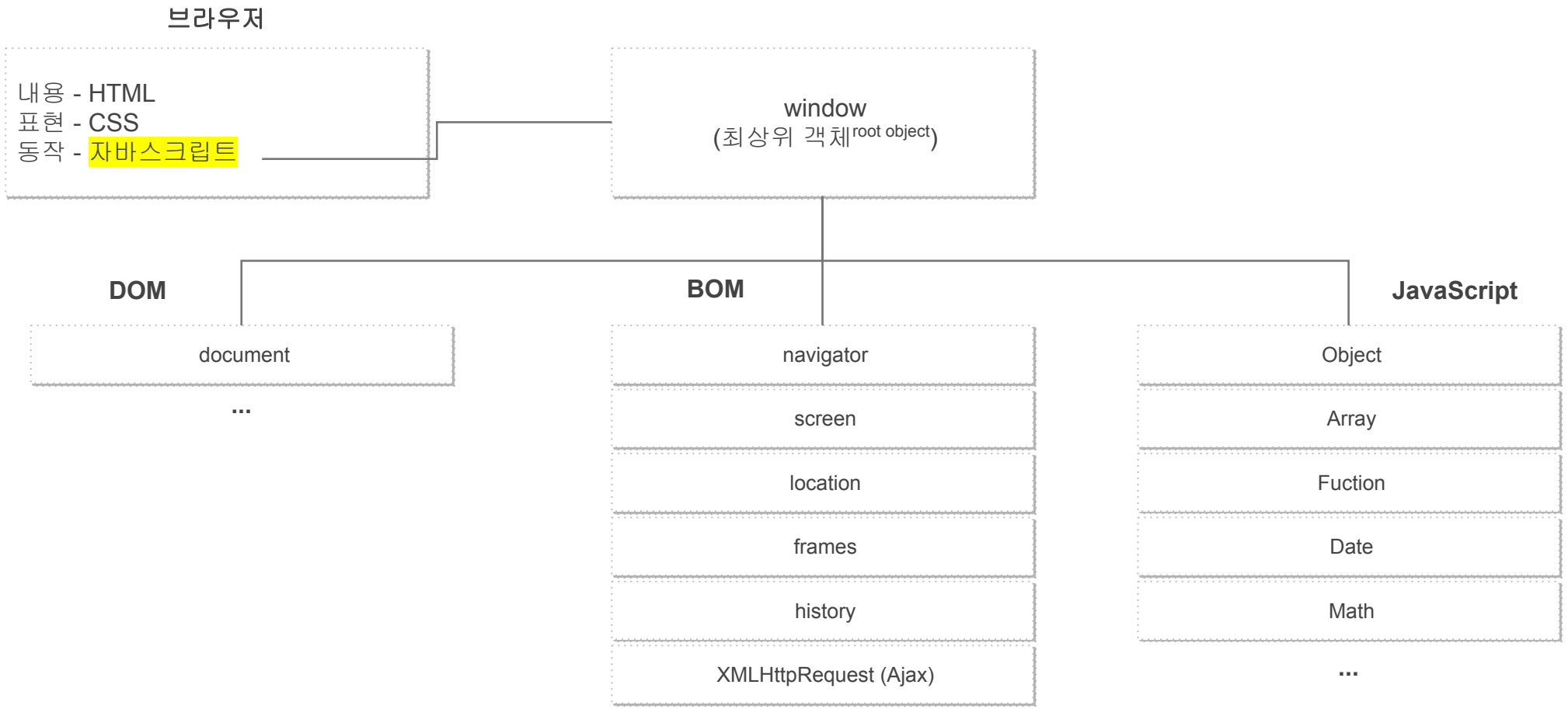
1. DOM 프로그래밍 시작하기
2. DOM 이벤트 처리하기
3. [실습] To-do 앱 클론코딩 final

Unit 5-1

문서객체모델DOM 프로그래밍 시작하기

LEARNING SPOONS

웹 브라우저와 자바스크립트

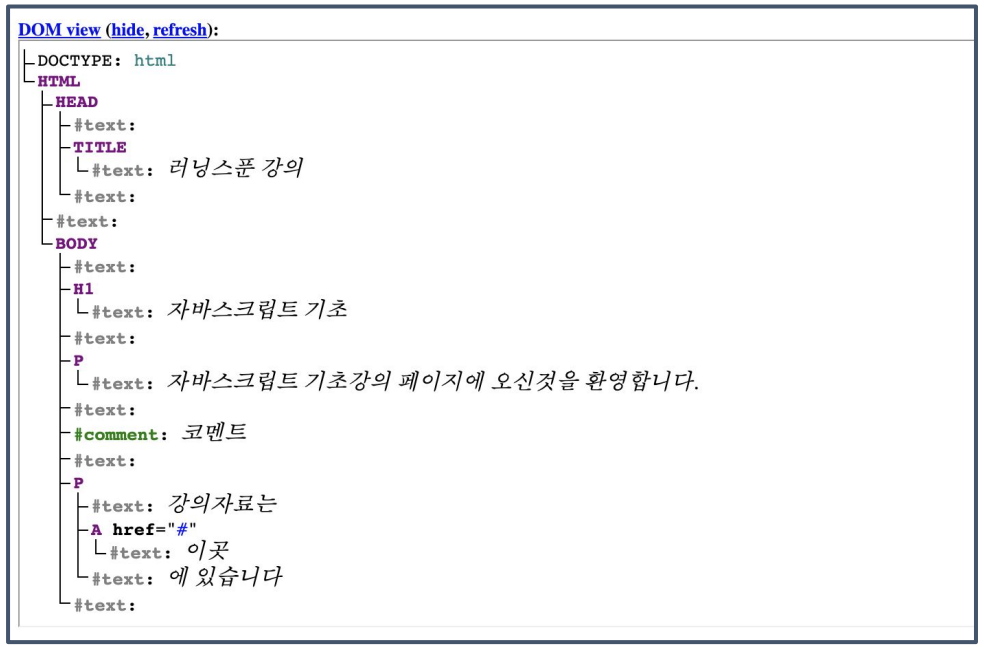


LEARNING SPOONS

문서객체모델 - DOM 이란?

- 문서객체모델Document Object Model은 HTML과 XML 문서를 위한 프로그래밍 인터페이스
- 문서의 구조, 내용과 시각적 표현을 자바스크립트를 통해 바꾸는 것을 가능하게 한다.
- DOM은 노드 객체들의 트리 구조를 가진다.
- Document Object Model Level 1, 2, 3의 명세를 가진다. (https://www.w3.org/DOM/DOMTR#dom3)

```
<!doctype html>
<html>
 <head>
   <title>러닝스푼 강의</title>
 </head>
 <body>
   <h1>자바스크립트 기초</h1>
   <p>자바스크립트 기초강의 페이지에 오신것을
환영합니다.</p>
   <!-- 코멘트 -->
   <p>강의자료는 <a href="#">이곳</a>에 있습니다
   </p>
 </body>
</html>
```



http://software.hixie.ch/utilities/js/live-dom-viewer/

LEARNING SPOONS

노드 객체 타입

- https://dom.spec.whatwg.org/#node에 정의되어 있다.
- 다음은 주요 노드 타입 상수이다.
  - DOCUMENT_NODE (예: window.document)
  - ELEMENT_NODE (예: <body>, <a>, <p>, <script>, <style>, <html>, <h1>)
  - ATTRIBUTE_NODE (예: class="funEdges")
  - TEXT_NODE (예: HTML문서에서의 텍스트들이고 케리지 리턴과 화이트 스페이스를 포함한다.)
  - DOCUMENT_FRAGMENT_NODE (예: document.createDocumentFragment())
  - DOCUMENT_TYPE_NODE (예: <!DOCTYPE html>)
- 각 노드타입은 숫자값이다.

§ 4.4. Interface Node

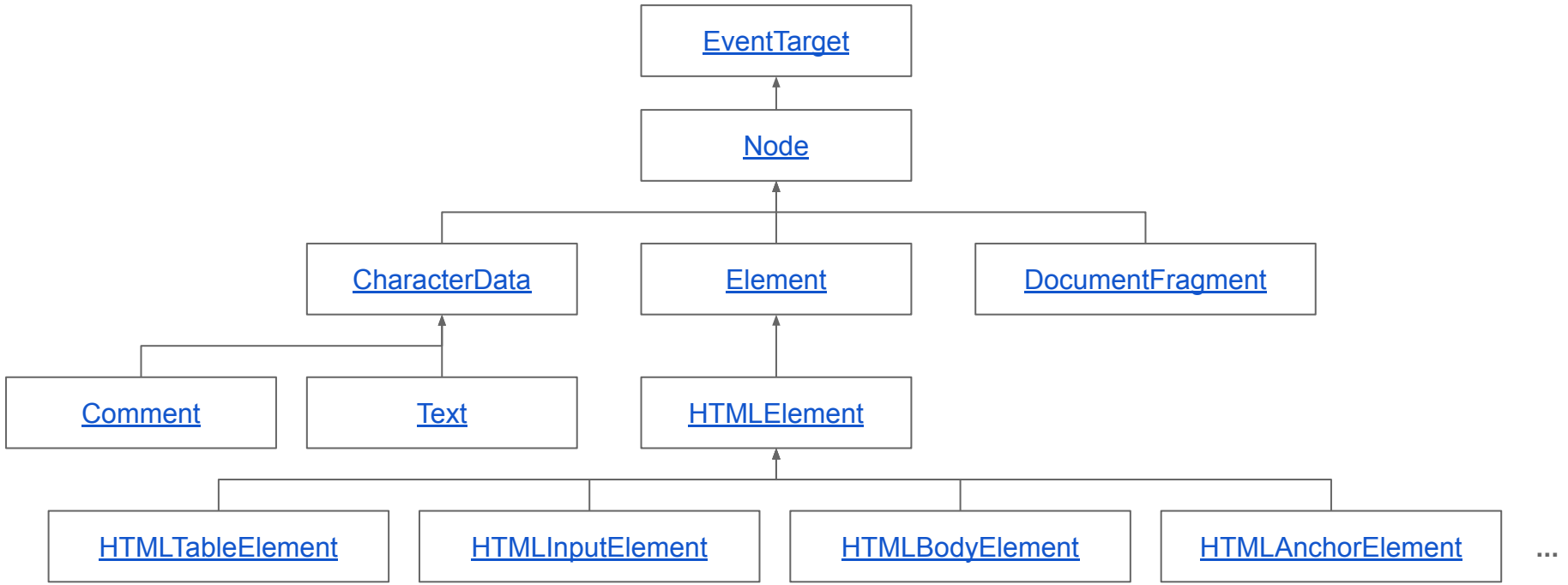
IDL

```
[Exposed=Window]
interface Node : EventTarget {
  const unsigned short ELEMENT_NODE = 1;
  const unsigned short ATTRIBUTE_NODE = 2;
  const unsigned short TEXT_NODE = 3;
  const unsigned short CDATA_SECTION_NODE = 4;
  const unsigned short ENTITY_REFERENCE_NODE = 5; // historical
  const unsigned short ENTITY_NODE = 6; // historical
  const unsigned short PROCESSING_INSTRUCTION_NODE = 7;
  const unsigned short COMMENT_NODE = 8;
  const unsigned short DOCUMENT_NODE = 9;
  const unsigned short DOCUMENT_TYPE_NODE = 10;
  const unsigned short DOCUMENT_FRAGMENT_NODE = 11;
  const unsigned short NOTATION_NODE = 12; // historical
  readonly attribute unsigned short nodeType;
```

LEARNING SPOONS

노드 인터페이스

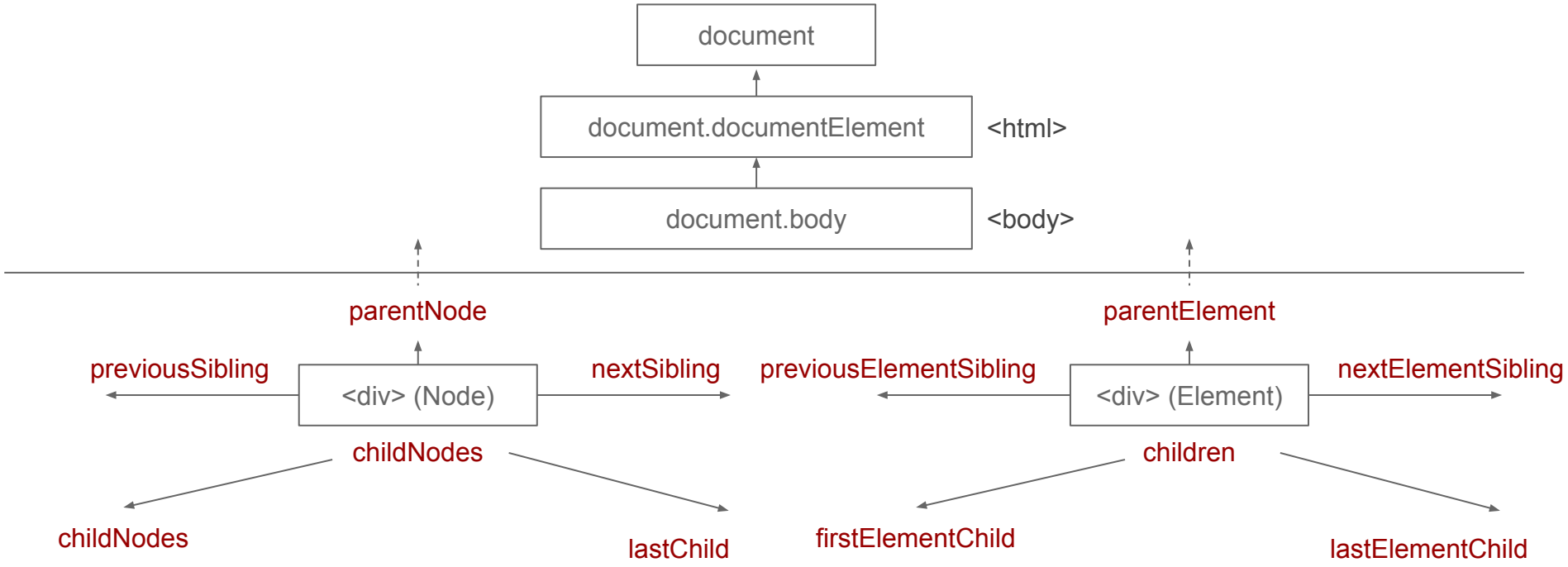
- DOM 노드들은 IDL[interface description language]로 정의
  (참고: DOM Level 3 Core 문서에서 정의된 Node IDL)
- 각 인터페이스들은 상속구조를 가진다.



LEARNING SPOONS

DOM 네비게이션

- Node타입과 Element타입의 특정 속성들에 의하여 연결되어 있다.
- elem.childNodes 는 상태가 변경되면 바로 반영된다. (Live)



DOM 검색

- getElement* 와 querySelector*

메소드	검색방법	element 메소드로 존재	배열로 반환
getElementById	아이디 속성값	-	-
getElementsByName	name 속성값	-	✔
getElementsByTagName	태그명 또는 *	✔	✔
getElementsByClassName	CSS 클래스 이름	✔	✔
querySelector	CSS 선택자	✔	-
queyrSelectorAll	CSS 선택자	✔	✔

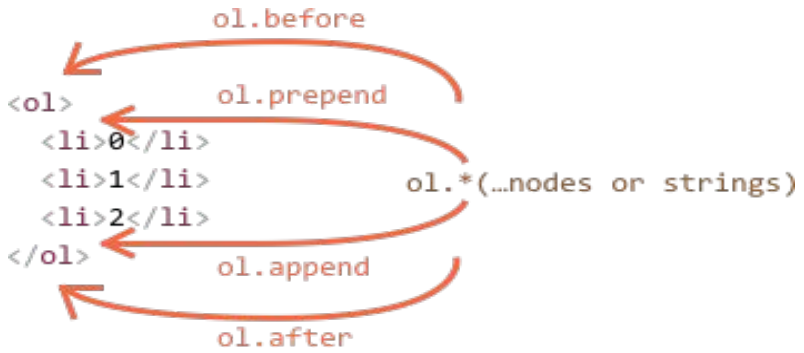
DOM Attribute와 Properties

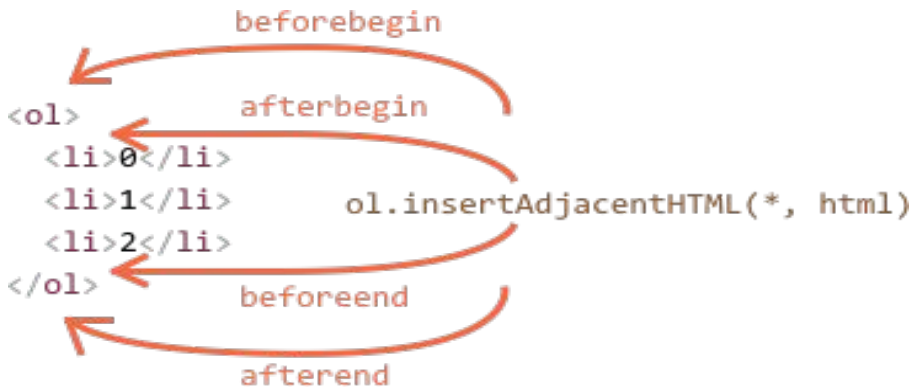
- DOM properties는 자바스크립트 객체의 속성이고 HTML attributes는 태그의 속성이다.
- 모든 attributes는 다음 메소드와 속성으로 접근이 가능하다.
  → elem.hasAttribute(name)
  → elem.getAttribute(name)
  → elem.setAttribute(name, value)
  → elem.removeAttribute(name)
  → elem.attributes
- DOM properties가 문자열이 아닐 수 있다. ex) checkbox의 checked는 불린이다.
- data-* HTML attributes는 DOM properties의 dataset으로 사용이 가능하다.
- 특정 attribute는 대응하는 속성이 없고 반대로 특정 속성은 대응하는 attribute가 없다.
- ex) HTML attribute colspan, DOM property textContent

LEARNING SPOONS

DOM 변경하기

- 요소 생성하기
  → *document.createElement(tag), document.createTextNode(value)*
- 요소 삽입하기
  → *perentElm.appendChild(node)*
  → *perentElm.insertBefore(node, nextSibling)*
  → *perentElm.replaceChild(node, oldChild)*
  → *elem.append(...nodes)*
  → *elem.insertAdjacentHTML(위치, html 문자열)*
- 요소 복제하기
  → *elem.cloneNode(deep 여부)*
- 요소 삭제하기
  → *parentElem.removeChild(node)*
  → *node.remove()*





LEARNING SPOONS

Unit 5-2

DOM 이벤트 처리하기

DOM 이벤트

- HTML 요소에 대한 사건의 발생을 의미한다. 주로 마우스 및 입력폼에의해 발생하고 문서가 브라우저에서의 상태가 변경될 때마다 발생하는 이벤트들도 있다.

이벤트	설명
click	요소를 클릭하였을때 발생한다.
dblclick	요소를 더블클릭 하였을때 발생한다.
change	입력폼의 요소의 내용이 변경되었을때 발생한다. 주로 <input> <select>에서 많이 사용된다.
focus	요소가 포커스가 되었을때 발생한다.
load	문서의 로드가 완료되었을 때 발생한다.
keydown	키보드 키를 눌렀을때 발생한다.

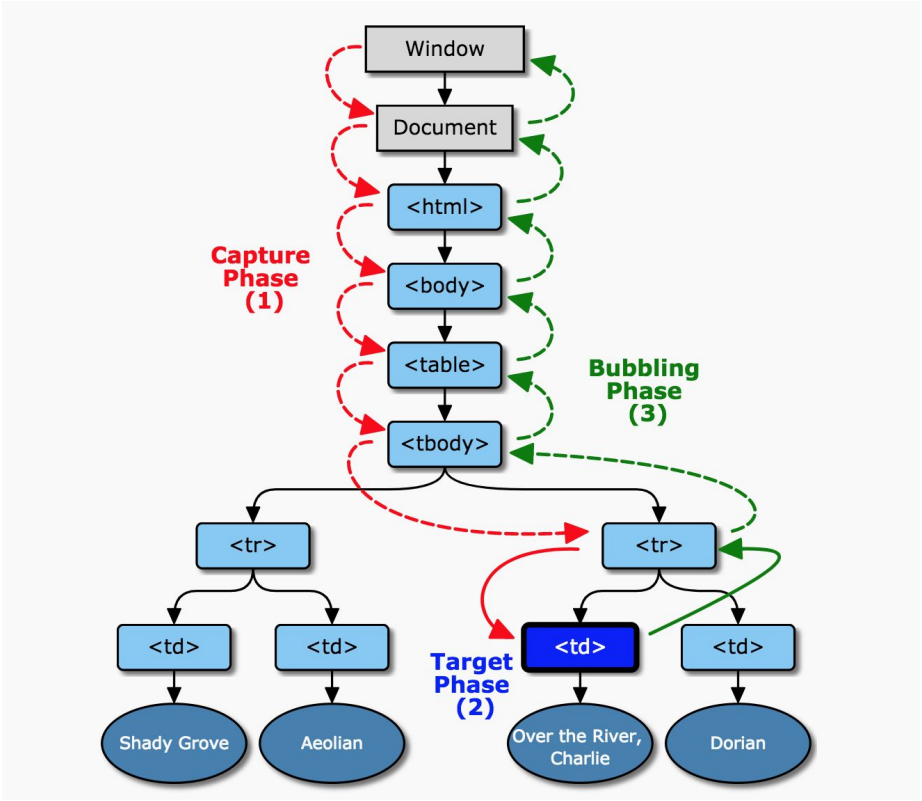
이벤트 핸들러

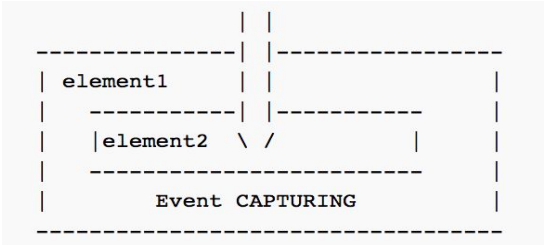
- 각 노드에 addEventListener 메서드를 이용하여 이벤트 핸들러를 등록하면 해당 이벤트 발생시 등록된 이벤트 핸들러(함수)가 호출 된다.

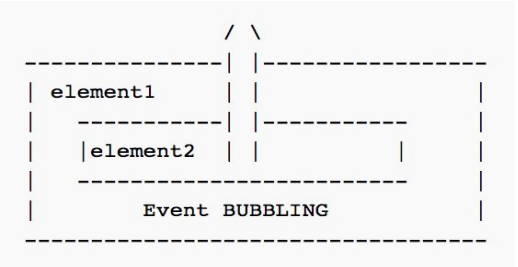
```
<button>클릭하세요!</button>
<p>핸들러가 없는 요소</p>

<script>
 var button = document.querySelector("button");
 button.addEventListener("click", function() {
   console.log("버튼이 클릭되었습니다.");
 });
</script>
```

LEARNING SPOONS

이벤트 발송 및 DOM 이벤트 플로우








```
element1.addEventListener('click', doSomething2, true)
element2.addEventListener('click', doSomething, false)
```

1. 클릭이벤트는 캡쳐링 단계에서 시작한다. 요소2를 클릭하면 요소2의 부모 요소들에 캡쳐링에 등록된 이벤트 핸들러를 찾는다.
2. 요소1의 doSomething2() 호출
3. 타겟까지 내려온 후 버블링 단계가 되며 요소2의 버블링 단계로 등록된 이벤트 핸들러인 doSomething()을 호출한다.
4. 최상위까지 올라가며 등록된 이벤트 핸들러를 찾아서 호출한다. 위 코드에서는 element1에 캡쳐링에 등록한 핸드러만 있기 때문에 doSomething외에 버블링시 더 호출되는 핸들러는 없다.

LEARNING SPOONS

이벤트 전파propagation 제어

```
<p>
 일반적인 글
 <span id="prevent">이벤트를 막은 글</span>
 <br>
 일반적인 글 2
</p>
<script>
 var para = document.querySelector("p");
 var prevent = document.querySelector("#prevent");
 para.addEventListener("contextmenu", function() {
  console.log("p태그 클릭");
 });
 prevent.addEventListener("contextmenu", function(event) {
  console.log("이벤트 막은 글 클릭");
  event.stopPropagation();
  event.preventDefault();
 });
</script>
```

이벤트 위임처리

- 각 노드에 addEventListener 메서드를 이용하여 이벤트 핸들러를 등록하면 해당 이벤트 발생시 등록된 이벤트 핸들러(함수)가 호출 된다.
- 이벤트가 버블링될때 부모 요소에서 자식 요소들의 이벤트들을 처리하는 방식
- 개별 요소에 이벤트 리스너들을 매번 등록할 필요도 없어 메모리 사용량이 줄어들고 누수 가능성도 줄어들수있다.
  - 예제코드 1
- 트위터 부트스트랩의 data-api에서 위임처리를 한 것을 볼 수 있다.
  - Github 소스코드 링크 (Alert 컴포넌트)
- data-* 활용 예제코드

실습

To-do 앱 클론코딩 final