DOM

요소 노드 취득하기

❖ ID 값으로 노드 취득하기

- document.getElementById(id)
  - id : 취득하고 싶은 요소의 id

```
현재 시간: <span id="result"></span>
```

```
var current = new Date();
var result = document.getElmentById('result');
result.textContent = current.toLocaleString();
```

- textContent : 노드의 text(HTML 태그 해석하지 않음)

요소 노드 취득하기

❖ 태그명으로 노드 취득하기

- document.getElementByTagName(name)
  - name : 태그명
  - 요소의 집합을 리턴 : HTMLCollection

```
<ul>
    <li><a href="https://google.com">google</a></li>
    <li><a href="https://naver.com">naver</a></li>
    <li><a href="https://daum.com">daum</a></li>
</ul>
```

```
var list = document.getElementByTagName('a')
for(var i=0, len=list.length; i < len; i++) {
    console.log(list.itme(i).href;
}
```

요소 노드 취득하기

❖ HTMLCollection

- 주요 멤버
  - length : 리스트에 포함된 요소의 수
  - itme(i) : iq번째의 노드를 취득
  - namedItem(name) : id 또는 name 속성이 일치하는 요소를 취득

요소 노드 취득하기

❖ name 속성을 키로 요소 취득하기

- document.getElementByName(name)
  - name: name 속성의 값
  - NodeList 리턴
  - 주로 폼 요소 접근시 사용(복수의 체크박스, 라디오 버튼 등)

```
<form>
    <label for="time">시간:</label>
    <input id="time" name="time" type="text" size="10"/>
</form>
```

```
var current = new Date();
var nam = document.getElementByName('time');
nam[0].value = current.toLocaleTimeString();
```

요소 노드 취득하기

❖ class 속성으로 요소 취득하기

- document.getElementByClassName(clazz)
  - clazz : class 속성의 값

```
<ul>
    <li><a href="https://google.com" class="link">google</a></li>
    <li><a href="https://naver.com" class="link">naver</a></li>
    <li><a href="https://daum.com">daum</a></li>
</ul>
```

```
var list = document.getElementByClassName('link')
for(var i=0, len=list.length; i < len; i++) {
    console.log(list.itme(i).href;
}
```

요소 노드 취득하기

❖ 셀렉터 식에 일치하는 요소 취득하기

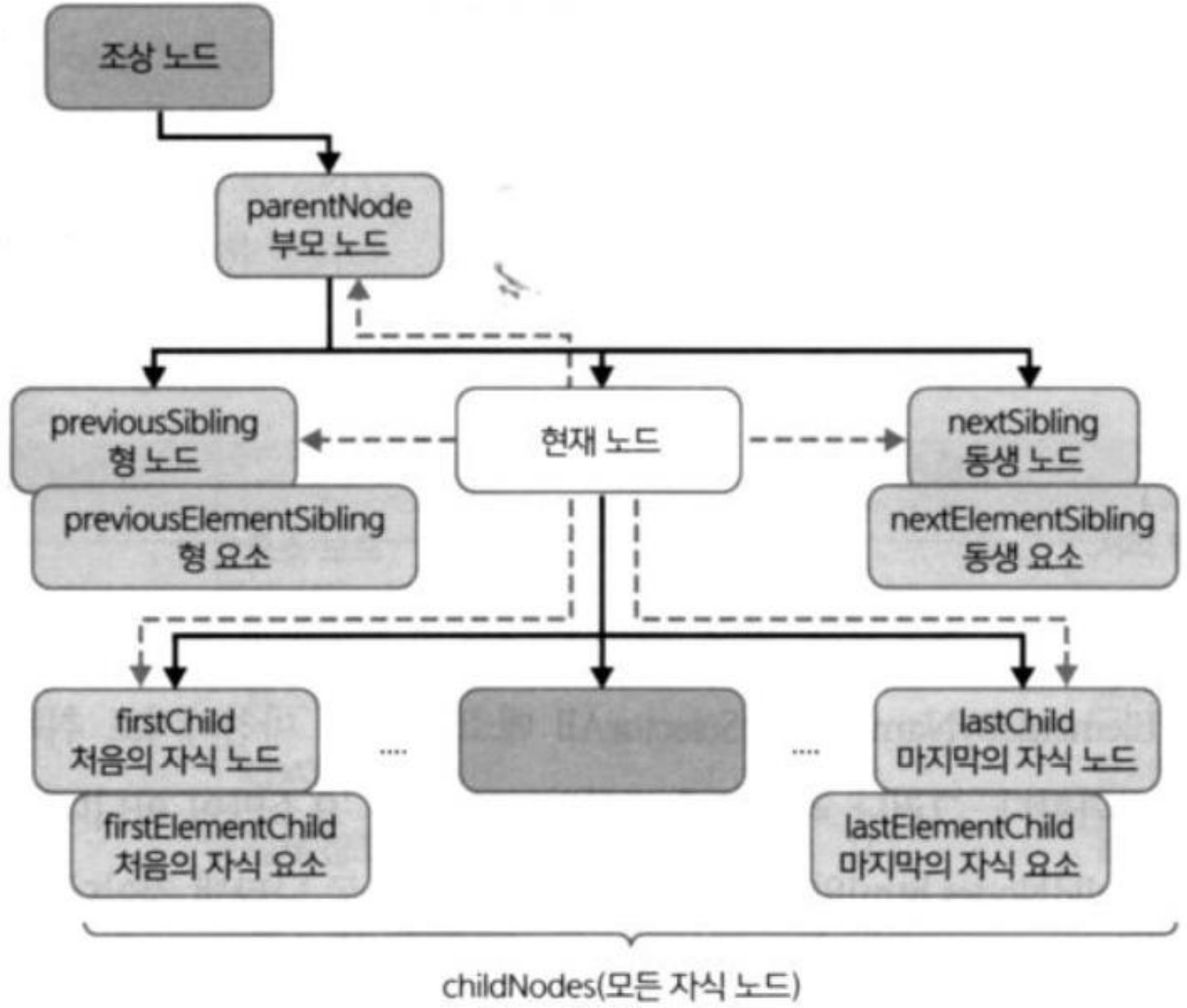
- document.querySelector(selector)
  - 발견된 첫 번째 요소 하나만 리턴
- document.querySelectorAll(selector)
  - NodeList 리턴
  - selector : css 선택자 표현식

```
<ul id="list">
    <li><a href="https://google.com" class="link">google</a></li>
    <li><a href="https://naver.com" class="link">naver</a></li>
    <li><a href="https://daum.com" class="texternal">daum</a></li>
</ul>
```

```
var list = document.getElementByClassName('#list .external')
for(var i=0, len=list.length; i < len; i++) {
    console.log(list.itme(i).href;
}
```

문서 트리 사이 오가기 - 노드워킹

❖ 노드 워킹



문서 트리 사이 오가기 - 노드워킹

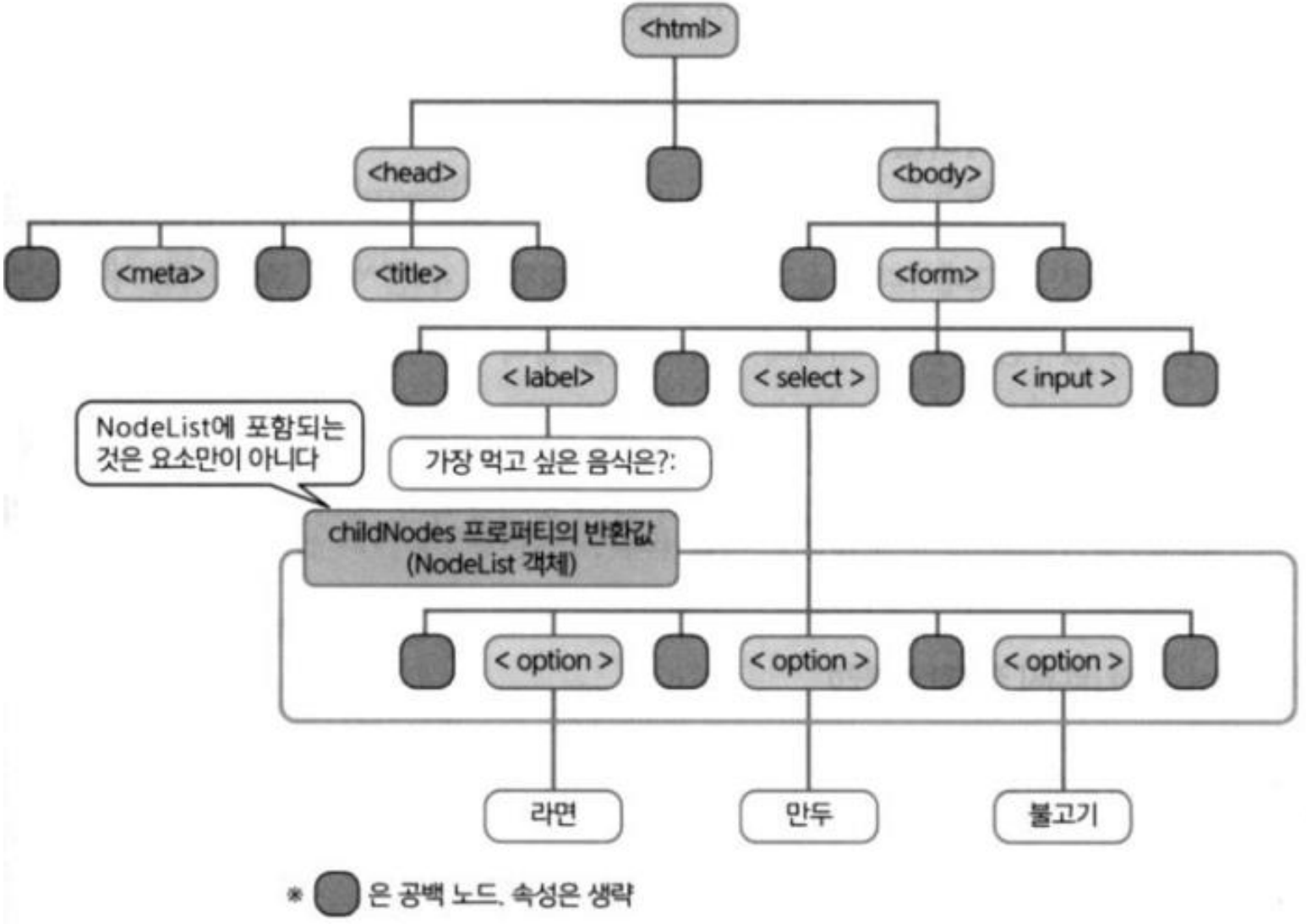
❖ 노드 워킹

```
<form>
  <label for="food">가장 먹고 싶은 음식은?:</label>
  <select id="food">
    <option value="라면">라면</option>
    <option value="만두">만두</option>
    <option value="불고기">불고기</option>
  </select>
 <input type="submit" value="송신" />
</form>
```

```
var s = document.getElementById('food');
var opts = s.childNodes;
for (var i = 0, len = opts.length; i < len; i++) {
  var opt = opts.item(i);
  if (opt.nodeType === 1) {
    console.log(opt.value);
  }
}
```

문서 트리 사이 오가기 - 노드워킹

❖ 노드 워킹



문서 트리 사이 오가기 - 노드워킹

❖ 노드의 종류

반환값	개요
1	요소 노드
2	속성 노드
3	텍스트 노드
4	CDATA 섹션
5	실제 참조 노드
6	실제 선언 노드

반환값	개요
7	처리 명령 노드
8	주석 노드
9	문서 노드
10	문서형 선언 노드
11	문서의 단편(fragment)
12	기법 선언 노드

문서 트리 사이 오가기 - 노드워킹

❖ 자식 요소 리스트를 취득하는 다른 방법

- firstChild/nextSibling 프로퍼티

```
var s = document.getElementById('food');
var child = s.firstChild;
while (child) {
  if (child.nodeType === 1) {
    console.log(child.value);
  }
  child = child.nextSibling;
}
```

문서 트리 사이 오가기 - 노드워킹

❖ 자식 요소 리스트를 취득하는 다른 방법

- firstElementChild/nextElementSibling 프로퍼티
- lastElementChild/previousElementSibling 프로퍼티

```
var s = document.getElementById('food');
var child = s.firstElementChild;
while (child) {
  console.log(child.value);
  child = child.nextElementSibling;
}
```

❖ 이벤트로 잠시 ....

속성값 취득/설정하기

❖ 속성 접근

- 요소 노드 속 같은 이름의 프로퍼티로 접근 가능

```
var url = link.href;
link.href = 'http://www.wings.msn.to/';
```

- 키워드 충돌이 있는 경우
  - class 속성 : javascript의 class 키워드와 중복 → className으로

```
<p class="summary">샘플</a>

node.className = 'summary';
```

속성값 취득/설정하기

❖ 속성과 프로퍼티 접근

- elem.getAttribute(name)
- elem.setAttribute(name, value)

```
var url = link.getAttribute('href');
link.setAttribute('href', 'http://www.wings.msn.to/');
```

속성값 취득/설정하기

❖ 불특정 속성 취득하기

```
<img id="logo" src="http://www.wings.msn.to/image/wings.jpg"
  height="67" width="215" border="0"
  alt="WINGS ( Www INtegrated Guide on Server-architecture ) " />
```

```
document.addEventListener('DOMContentLoaded', function() {
  var logo = document.getElementById('logo');
  var attrs = logo.attributes;
  for (var i = 0, len = attrs.length; i < len; i++) {
    var attr = attrs.item(i);
    console.log(attr.name + ':' + attr.value);
  }
}, false);
```

- elem.attributes : NamedNodeMap 객체
  - index로도 추출 가능

속성값 취득/설정하기

❖ 텍스트 취득 및 설정하기

- textContent : HTML 태그를 해석하지 않음
- innerHTML : HTML 태그를 해석하고, 적용

```
<div id="result_text">
  <p style="color: Red;">설정되어 있지 않다!</p>
</div>
<div id="result_html">
  <p style="color: Red;">설정되어 있지 않다!</p>
</div>
```

```
document.addEventListener('DOMContentLoaded', function() {
  document.getElementById('result_text').textContent =
    '<a href="http://www.wings.msn.to/">WINGS프로젝트</a>';
  document.getElementById('result_html').innerHTML =
    '<a href="http://www.wings.msn.to/">WINGS프로젝트</a>';
}, false);
```

노드 추가/치환/삭제

❖ 노드 추가하기

메소드	생성하는 노드
createElement(요소명)	요소 노드
createAttribute(속성명)	속성 노드
createTextNode(텍스트)	텍스트 노드
createCDATASection(텍스트)	CDATA 섹션
createComment(텍스트)	주석 노드
createEntityReference(실체명)	실체 참조 노드
createProcessingInstruction(타깃명, 데이터)	처리 명령 노드
createDocumentFragment()	문서의 단편

노드 추가/치환/삭제

❖ 노드끼리 조립하기

- element.appendChild(node)
- element.insertBefore(node, base)

```
// 요소의 마지막 자식으로 추가하기
list.appendChild(anchor);
list.insertBefore(anchor, null);  // base가 null이면 마지막 자식으로 추가
```

```
// 요소의 첫 번째 자식으로 추가하기
list.insertBefore(br, list.firstChild)
list.insertBefore(anchor, br);
```

노드 추가/치환/삭제

❖ 속성 노드 추가

```
anchor.href = url.value;
```

```
var href = document.createAttribute('href');
href.value = url.value;
anchor.setAttributeNode(href);
```

노드 추가/치환/삭제

❖ 사용자 정의 속성 data-xxx

- 값 추출
  - `getAttribute("data-xxx")`
- 값 설정
  - `setAttribute("data-xxx", 값)`

노드 추가/치환/삭제

❖ 기존 노드의 치환/삭제

- 노드의 치환
  - elm.replaceChild(after, before)
    - elem : 요소 객체
    - after: 치환 후의 노드
    - before: 치환 대상 노드

- 노드의 삭제
  - elem.removeChild(node)
    - elem : 요소 객체
    - node : 삭제 대상 노드

노드 추가/치환/삭제

❖ 기존 노드의 치환/삭제

```
<ul id="list">
  <li><a href="JavaScript:void(0)" data-isbn="978-4-7981-3547-2">
    개정3판 기초PHP</a></li>
  <li><a href="JavaScript:void(0)" data-isbn="978-4-7741-8030-4">
    Java포켓 레퍼런스</a></li>
  <li><a href="JavaScript:void(0)" data-isbn="978-4-7741-7984-1">
    Swift포켓 레퍼런스</a></li>
  <li><a href="JavaScript:void(0)" data-isbn="978-4-7981-4402-3">
    개정5판 ASP.NET입문</a></li>
  <li><a href="JavaScript:void(0)" data-isbn="978-4-8222-9644-5">
    앱을 만들자! Android입문</a></li>
</ul>

<input id="del" type="button" value="삭제" disabled />

<div id="pic"></div>
```

노드 추가/치환/삭제

❖ 기존 노드의 치환/삭제

```
document.addEventListener('DOMContentLoaded', function() {
  var list = document.getElementById('list');
  var pic = document.getElementById('pic');
  var del = document.getElementById('del');

  list.addEventListener('click', function(e) {
    var isbn = e.target.getAttribute('data-isbn');

    if (isbn) {
      var img = document.createElement('img');
      img.src = `http://www.wings.msn.to/books/${isbn}/${isbn}.jpg`;
      img.alt = e.innerHTML;
      img.height = 150;
      img.width = 108;
      if(pic.getElementsByTagName('img').length > 0){
        pic.replaceChild(img, pic.lastChild);
      } else {
        del.disabled = false;
        pic.appendChild(img);
      }
    }
  }, false);
```

노드 추가/치환/삭제

❖ 기존 노드의 치환/삭제

```
  del.addEventListener('click', function() {
    pic.removeChild(pic.lastChild);
    del.disabled = true;
  }, false);
}, false);
```

노드 추가/치환/삭제

❖ HTMLCollection/NodeList 반복시 주의 점

- Live 객체임 → 요소의 추가/치환/삭제 시, 동적으로 요소가 변경됨

```
<ul id="list">
  <li>개정3판 기초PHP</li>
  <li>Java포켓 레퍼런스</li>
  <li>Swift포켓 레퍼런스</li>
  <li>개정5판 ASP.NET입문</li>
  <li>앱을 만들자! Android입문</li>
</ul>
```

```
document.addEventListener('DOMContentLoaded', function() {
  var li = document.getElementsByTagName('li');
  console.log('변경전:' + li.length);

  var ul = document.getElementById('list');
  ul.appendChild(document.createElement('li'));
  console.log('변경후:' + li.length);
}, false);
```

스타일 시트 조작하기

❖ 인라인 스타일 접근

- elem.style.prop [=value]
  - elem : 요소 객체
  - prop : 스타일 프로퍼티
    - -를 포함하는 프로터티 → CamelCase 표기
    - font-size → fontSize
  - value : 설정값

스타일 시트 조작하기

❖ 인라인 스타일 접근

```
<div id="elem">마우스 포인터를 올리면 색이 변합니다.</div>
```

```
document.addEventListener('DOMContentLoaded', function() {
  var elem = document.getElementById('elem');

  elem.addEventListener('mouseover', function() {
    this.style.backgroundColor = 'Yellow';
  }, false);

  elem.addEventListener('mouseout', function() {
    this.style.backgroundColor = '';
  }, false);
}, false);
```

스타일 시트 조작하기

❖ 스타일 클래스의 추가/삭제

- 단일 클래스

```
document.addEventListener('DOMContentLoaded', function() {
  var elem = document.getElementById('elem');

  elem.addEventListener('click', function() {
    this.className = (this.className === 'highlight' ? '' : 'highlight');
  }, false);
}, false);
```

스타일 시트 조작하기

❖ 스타일 클래스의 추가/삭제

- 여러 개의 클래스가 배정된 경우 문자열 조작 기법 활용

```
document.addEventListener('DOMContentLoaded', function() {
  var elem = document.getElementById('elem');

  elem.addEventListener('click', function() {
    var classes = this.className.split(' ');
    var index = classes.indexOf('highlight');
    if(index === -1) {
      classes.push('highlight');
    } else {
      classes.splice(index, 1);
    }
    this.className = classes.join(' ');
  }, false);
}, false);
```

스타일 시트 조작하기

❖ 스타일 클래스를 더 간단히 조작하기

- classList 프로퍼티

멤버	개요
length	리스트의 길이
item(index)	인덱스 번호의 클래스를 취득
contains(clazz)	지정한 클래스가 포함되어 있는가?
add(clazz)	리스트에 클래스 추가
remove(clazz)	리스트로부터 클래스 삭제
toggle(clazz)	클래스의 On/Off를 전환

스타일 시트 조작하기

❖ 스타일 클래스를 더 간단히 조작하기

- classList 프로퍼티

```
<div id="elem" class="line">클릭하면 색이 변한다.</div>
```

```
document.addEventListener('DOMContentLoaded', function() {
  var elem = document.getElementById('elem');

  elem.addEventListener('click', function() {
    this.classList.toggle('highlight');
  }, false);
}, false);
```