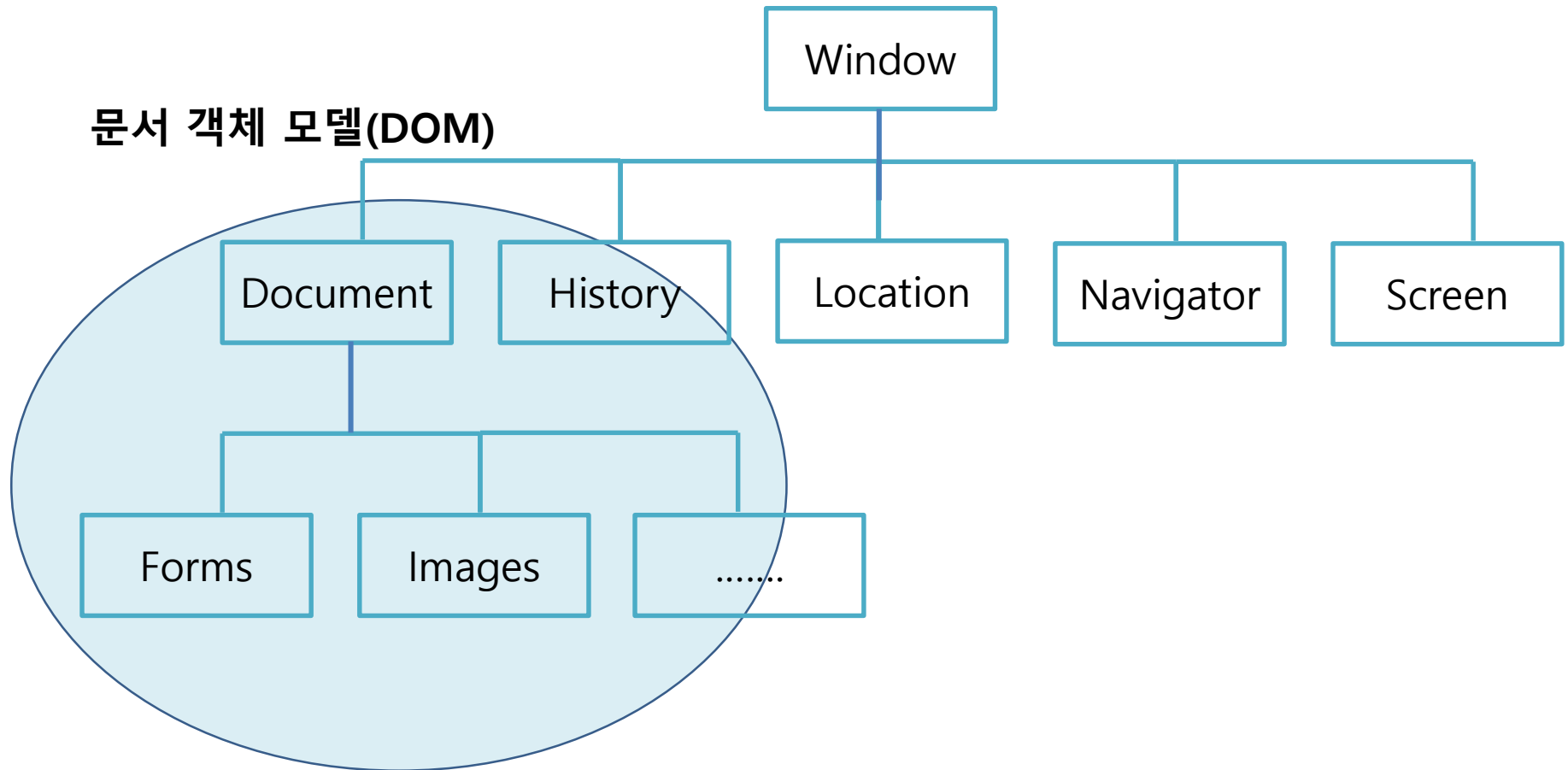


# 8강. DOM(Document Object Model)



# DOM(Document Object Model)

## 브라우저 객체 모델(BOM)



# DOM(Document Object Model)

## DOM이란?

웹 페이지(HTML 문서)를 자바스크립트로 조작할 수 있도록 객체 모델 형태로 표현한 것을 말합니다.

DOM은 HTML 문서를 객체 트리로 바꾼 것이고, 자바스크립트를 통해 DOM을 조작하면 웹 페이지의 구조, 내용, 스타일, 동작을 동적으로 바꿀 수 있다.

## DOM 주요 객체

- **document** : 웹 페이지 전체를 대표하는 객체
- **element** : HTML 태그 하나하나를 객체로 표현
- **attribute** : 태그의 속성(id, class 등)
- **text** : 태그 안의 글자

# 문서 객체 모델(DOM)

## DOM 요소에 접근하기

### 1. DOM 요소를 id 선택자로 접근하는 함수

`document.getElementById("display")`

속성	설명
<b>innerHTML</b>	태그 요소의 내용(텍스트)을 바꾸는 속성
<b>style</b>	태그 요소의 스타일을 바꾸는 속성
<b>src</b>	파일의 경로

# 문서 객체 모델(DOM)

## DOM 요소에 접근하기

2. DOM 요소를 태그, id, class 선택자로 접근하는 함수

```
document.querySelector("img")
```

```
document.querySelector("#pic")
```

```
document.querySelector(".pic")
```

3. 객체.querySelectorAll(리스트) – 여러 개의 값을 반환

```
document.querySelectorAll(".city")
```

# 문서 객체 모델(DOM)

## ➤ DOM 요소에 접근하기

**Hello, DOM!**

강조된 문장

```
<h1 id="title">안녕하세요</h1>  
<p class="desc">DOM 학습 중입니다.</p>  
  
<script src="js/dom.js"></script>
```

# 문서 객체 모델(DOM)

## ➤ DOM 요소에 접근하기

```
// ID 선택
let title = document.getElementById("title");

//title.textContent = "Hello, DOM!"; // 텍스트 변경
title.innerText = "안녕, DOM!";

// CSS 선택자 방식
let p = document.querySelector(".desc"); // 첫 번째 요소

p.innerHTML = "<b>강조된 문장</b>"; // HTML 포함 변경
```

dom.js

```
▼<body>
  <h1 id="title">Hello, DOM!</h1>
  ▼<p class="desc">
    <b>강조된 문장</b>
  </p>
  <script src="js/dom.js"></script>
```

# 이벤트 다루기

## 이벤트 효과

이벤트(Event)란 웹 브라우저나 사용자가 실행하는 어떤 동작을 말한다.

### 마우스 이벤트

속성	설명
onclick	사용자가 HTML 요소를 클릭할 때 이벤트가 발생
mouseover	마우스 포인터가 요소 위로 옮겨질때 이벤트 발생
mouseout	마우스 포인터가 요소를 벗어날 때 이벤트 발생

### 키보드 이벤트

속성	설명
keypress	키를 눌렀을때 이벤트가 발생
keydown	키를 누르고 있는 동안 이벤트 발생
keyup	키에서 손을 뗄 때 이벤트 발생



# 이벤트 다루기

## 이벤트 효과

### 문서로딩 이벤트

속성	설명
load	문서 로딩이 끝나면 이벤트가 발생
abort	웹 문서가 로딩되기 전 멈췄을때 이벤트 발생
scroll	문서 화면이 스크롤 되었을때 이벤트 발생

### 폼 이벤트

속성	설명
change	목록이나 체크 상태 등이 변경되었을때 이벤트 발생
focus	폼 요소에 포커스(커서)가 놓였을때 이벤트 발생
submit	submit 버튼을 눌렀을때 이벤트 발생

## 태그 요소에 접근

- 마우스를 클릭하여 글자색을 바꾸기

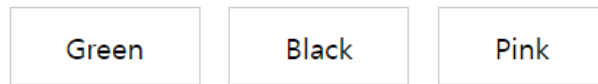
Click on this text! -----> Click on this Text!!  
onclick

```
<h1 id="switch" onclick="changeText()">Click on this Text!!</h1>  
<script>  
  function changeText(){  
    //document.getElementById("switch").innerHTML = "Good!"  
    document.getElementById("switch").style.color = "blue";  
  }  
</script>
```

changeText.html

## 태그 요소에 접근

- 버튼을 클릭하여 배경색 넣기



Green 클릭 -> 배경색 녹색

Black 클릭 -> 배경색 검정색

Pink 클릭 -> 배경색 분홍색

# 태그 요소에 접근

```
<!-- 버튼 메뉴 만들기 -->
<div class="menu">
  <ul>
    <li>
      <a href="#" onclick="changeBg('green')">Green</a>
    </li>
    <li>
      <a href="#" onclick="changeBg('black')">Black</a>
    </li>
    <li>
      <a href="#" onclick="changeBg('pink')">Pink</a>
    </li>
  </ul>
</div>
<!-- 배경색을 넣을 사각형 -->
<div id="rect" class="box"></div>

<script>
  function changeBg(color){
    let rect = document.querySelector("#rect");
    rect.style.backgroundColor = color;
  }
</script>
```

changebg.html

# 태그 요소에 접근

```
.menu{width: 400px; margin-left: 80px;}
.menu ul{list-style: none; padding: 0;}
.menu ul li{
  display: inline-block;
  margin: 10px;
  border: 1px solid #ccc;
  border-radius: 3px;
  padding: 10px 20px;
}
.menu ul li a{text-decoration: none; color: #000;}
.box{
  width: 400px;
  height: 300px;
  border: 1px solid #ccc;
  margin-left: 50px;
}
```

bgColor.css

# 숨기기와 보이기 이벤트 효과

- 버튼을 클릭하여 이미지와 텍스트를 숨기고 보이는 효과 만들기

설명글 숨기기



설명글 보이기



민들레

어디서나 매우 흔하게 보이는 잡초로써 바닥에 딱 붙어서 꽃봉오리 하나가 쑥 올라온다 톱니 모양의 잎새와 눈에 확 띄는 노란 꽃이 인상적이다. 특히 꽃이 지고 나면 솜털모양의 깃을 가진 씨앗들이 나오는데 바람을 타고 날아가서 널리 퍼진다.

상세 설명 닫기

# 숨기기와 보이기 이벤트 효과

showHide.html

```
<div id="item">
  
  <button onclick="showText()" class="btnShow" id="show">상세 설명 보기</button>
  <div class="text" id="detail">
    <h4>민들레</h4>
    <p>어디서나 매우 흔하게 보이는 잡초로써 바닥에 딱 붙어서 꽃봉오리 하나가
      쑥 올라온다.톱니 모양의 잎새와 눈에 확 띄는 노란 꽃이 인상적이다.
      특히 꽃이 지고 나면 솜털 모양의 깃을 가진 씨앗들이 나오는데 바람을
      타고 날아가서 널리 퍼진다.</p>
    <button onclick="hideText()" id="hide">상세 설명 닫기</button>
  </div>
</div>

<script src="js/event.js"></script>
```

# 숨기기와 보이기 이벤트 효과

```
#item{
  width: 500px;
  margin: 0 auto;
  position: relative; /* 부모 기준 */
}
.btnShow{
  top: 260px;
  left: 20px;
  position: absolute;
}
.text{
  width: 400px;
  display: none; /* 시작시 숨기기 */
  line-height: 1.5;
}
```

event.css



# 숨기기와 보이기 이벤트 효과

```
//보기 버튼을 누르면 설명 텍스트가 보이고,  
//닫기 버튼을 누르면 설명 텍스트가 숨기짐  
let button = document.getElementById("show");  
let text = document.getElementById("detail");  
  
function showText(){  
    text.style.display = "block"; //텍스트 보이기  
    button.style.display = "none"; //버튼 숨기기  
}  
  
function hideText(){  
    button.style.display = "block"; //버튼 보이기  
    text.style.display = "none"; //텍스트 숨기기  
}
```

event.js

# 자바스크립트 객체의 onclick 속성

## 이벤트 처리1

- 자바스크립트 객체의 onclick 속성 사용하기



```
<!-- 이미지를 클릭하여 바꾸기 -->
<div id="content">
  
</div>

<script src="js/changeImg.js"></script>
```

# 자바스크립트 객체의 onclick 속성

## 이벤트 처리1

### - 자바스크립트 객체의 onclick 속성 사용하기

```
let pic = document.querySelector("#pic") //이미지 요소에 접근

function changePic(){
  pic.src = "images/cup-2.png";
}

//객체의 onclick 속성 사용 - 호출된 함수의 ()는 생략함
pic.onclick = changePic;

//실행 함수로 구현
pic.onclick = () => {
  pic.src = "images/cup-2.png";
}
```

changelmg.js

# addEventListener() 함수

## 이벤트 처리2

– addEventListener(이벤트, 함수)



```
<!-- 이미지를 클릭하여 바꾸기 -->
<div id="content">
  
</div>

<script src="js/changeImg2.js"></script>
```

# addEventListener() 함수

## addEventListener() 함수 사용하기

한 요소에 여러 이벤트가 발생했을때 이벤트 처리기를 연결해 주는 함수이다.

### Syntax

```
element.addEventListener(event, function, useCapture)
```

false가 기본임

### Parameter Values

이벤트 유형('on'을 붙이지 않음)

Parameter	Description
<i>event</i>	Required. A String that specifies the name of the event.  <b>Note:</b> Do not use the "on" prefix. For example, use "click" instead of "onclick".  For a list of all HTML DOM events, look at our complete <a href="#">HTML DOM Event Object Reference</a> .
<i>function</i>	Required. Specifies the function to run when the event occurs.  When the event occurs, an event object is passed to the function as the first parameter. The type of the event object depends on the specified event. For example, the "click" event belongs to the MouseEvent object.

# addEventListener() 함수

```
let pic = document.querySelector("#pic") //이미지 요소에 접근

//객체.addEventListener(이벤트, 함수){}
pic.addEventListener("click", () => {
  pic.src = "images/cup-2.png";
});

//토글(toggle) 구현
let sw = false; //상태 변수
pic.addEventListener("click", () => {
  if(!sw){
    pic.src = "images/cup-2.png";
    sw = true;
  }else{
    pic.src = "images/cup-1.png";
    sw = false;
  }
});
```

changeImg2.js

# addEventListener() 함수

```
//mouseover, mouseout 사용
pic.addEventListener("mouseover", () => {
  pic.src = "images/cup-2.png";
});

pic.addEventListener("mouseout", () => {
  pic.src = "images/cup-1.png";
});
```

changeImg2.js

```
//event 정보 보기
pic.addEventListener("click", (e) => {
  console.log(e);
  alert(`클릭한 위치: ${e.pageX}, ${e.pageY}`)
});
```

# DOM 요소 선택자

- `querySelectorAll("선택자")`

## 세계의 수도(Capital)

- 베이징
- 도쿄
- 파리
- 워싱턴

`querySelectorAll.html`

```
<!-- class 선택자 모두 선택 -->
<h2>세계의 수도(Capital)</h2>
<ul>
  <li class="city">베이징</li>
  <li class="city">도쿄</li>
  <li class="city">파리</li>
  <li class="city">워싱턴</li>
</ul>

<script src="js/city.js"></script>
```



# DOM 요소 선택자

- `querySelectorAll("선택자")`

```
let cityList = document.querySelectorAll(".city");  
  
//console.log(cityList);  
//console.log(cityList[0]);  
//console.log(cityList.length);  
  
cityList[0].style.color = "■ #ff0000";  
cityList[1].style.color = "■ #0000ff";  
cityList[2].style.fontSize = "20px";  
cityList[3].style.fontStyle = "italic";
```

city.js

# DOM 요소 선택자

- 속성을 가져오거나 설정하는 함수

`getAttribute(속성명)` – 태그 요소의 속성 가져오기

`setAttribute(속성명, 값)` – 태그 요소의 속성 설정하기

OK

입력상자를 버튼으로 변경하려면 클릭하세요.

Try it

# DOM 요소 선택자

- 속성을 가져오거나 설정하는 함수

```
<!-- 입력상자를 버튼 타입으로 바꾸기 -->
<input type="text" value="OK" class="input">
<p>입력상자를 변경하려면 버튼을 클릭하세요</p>
<button onclick="myFunction()">Try it</button>
```

setAttribute1.html

```
<script>
  function myFunction(){
    let input = document.querySelector(".input");

    console.log(input.getAttribute("type")); //type 속성값 얻기
    console.log(input.getAttribute("value")); //value 속성값 얻기
    input.setAttribute("type", "button"); //type 속성 변경하기
  }
</script>
```

# DOM 요소 선택자

- 마우스로 선택한 사진으로 바꾸기



# DOM 요소 선택자

- 마우스로 선택한 사진으로 바꾸기

```
<section id="content">
  <h2>커피의 세계</h2>
  <div id="photo">
    
    <div class="smallPic">
      
      
      
    </div>
  </div>
</section>

<script src="./js/showbig.js"></script>
```

showBig.html

# DOM 요소 선택자

## 브라우저 Console 입력, 출력

```
> document.querySelector("#big")
< 
> document.querySelector("#big").getAttribute("src")
< './images/coffee-gray.jpg'
> document.querySelector("#big").getAttribute("alt")
< '커피잔'
```

```
> document.querySelectorAll(".small")
< ▶ NodeList(3) [img.small, img.small, img.small]
> document.querySelectorAll(".small")[2].src
< 'http://192.168.219.103:5501/day4/images/coffee-pink.jpg'
> document.querySelectorAll(".small")[2].getAttribute("src")
< './images/coffee-pink.jpg'
> document.querySelector("#big").setAttribute("src", './images/coffee-pink.jpg')
```

# DOM 요소 선택자

```
#content{  
  width: 600px;  
  margin: 0 auto;  
  text-align: center;  
}  
.smallPic{margin: 16px 5px;}  
.small{  
  width: 100px;  
  height: 100px;  
}
```

showBig.css

# DOM 요소 선택자

```
//큰 이미지 선택
let bigPic = document.querySelector("#big");

//작은 이미지를 큰 이미지로 변경
function showBig(){
  let newPic = this.src; //클릭된 작은 이미지 경로
  console.log(bigPic.getAttribute("src"));
  bigPic.setAttribute("src", newPic);
}

//모든 작은 이미지 선택
let smallPics = document.querySelectorAll(".small");

//smallPics[1].onclick = showBig; //1번 이미지 이벤트

//모든 작은 이미지에 이벤트 등록
for(let i=0; i<smallPics.length; i++){
  smallPics[i].onclick = showBig;
}
```

showBig.js



# classList 속성

- classList 속성

요소의 CSS 클래스 이름을 반환합니다 – 클래스 이름을 추가, 삭제함

## classList Properties and Methods

Name	Description
<u>add()</u>	Adds one or more tokens to the list
<u>contains()</u>	Returns true if the list contains a class
<u>entries()</u>	Returns an Iterator with key/value pairs from the list
<u>forEach()</u>	Executes a callback function for each token in the list
<u>item()</u>	Returns the token at a specified index
<u>keys()</u>	Returns an Iterator with the keys in the list
<u>length</u>	Returns the number of tokens in the list

# 반응형 header

- **classList** 속성

요소의 CSS 클래스 이름을 반환합니다.


## classList 속성

Add

Remove

Toggle

오늘도 좋은 하루되세요~

```
<style>
  .myStyle {
    background-color:  coral;
    padding: 16px;
    margin-top: 8px;
  }
</style>
```

# 반응형 header

```
<h1>classList 속성</h1>

<button onclick="classAdd()">Add</button>
<button onclick="classRemove()">Remove</button>
<button onclick="classToggle()">Toggle</button>

<div id="myDIV">
  <p>오늘도 좋은 하루되세요~</p>
</div>

<script>
  function classAdd() {
    const list = document.getElementById("myDIV").classList;
    list.add("myStyle");
  }
  function classRemove() {
    const list = document.getElementById("myDIV").classList;
    list.remove("myStyle");
  }
  function classToggle() {
    const list = document.getElementById("myDIV").classList;
    list.toggle("myStyle");
  }
</script>
```

classList.html

# DOM에서 노드 추가.삭제하기

## DOM 트리에 새로운 노드 추가

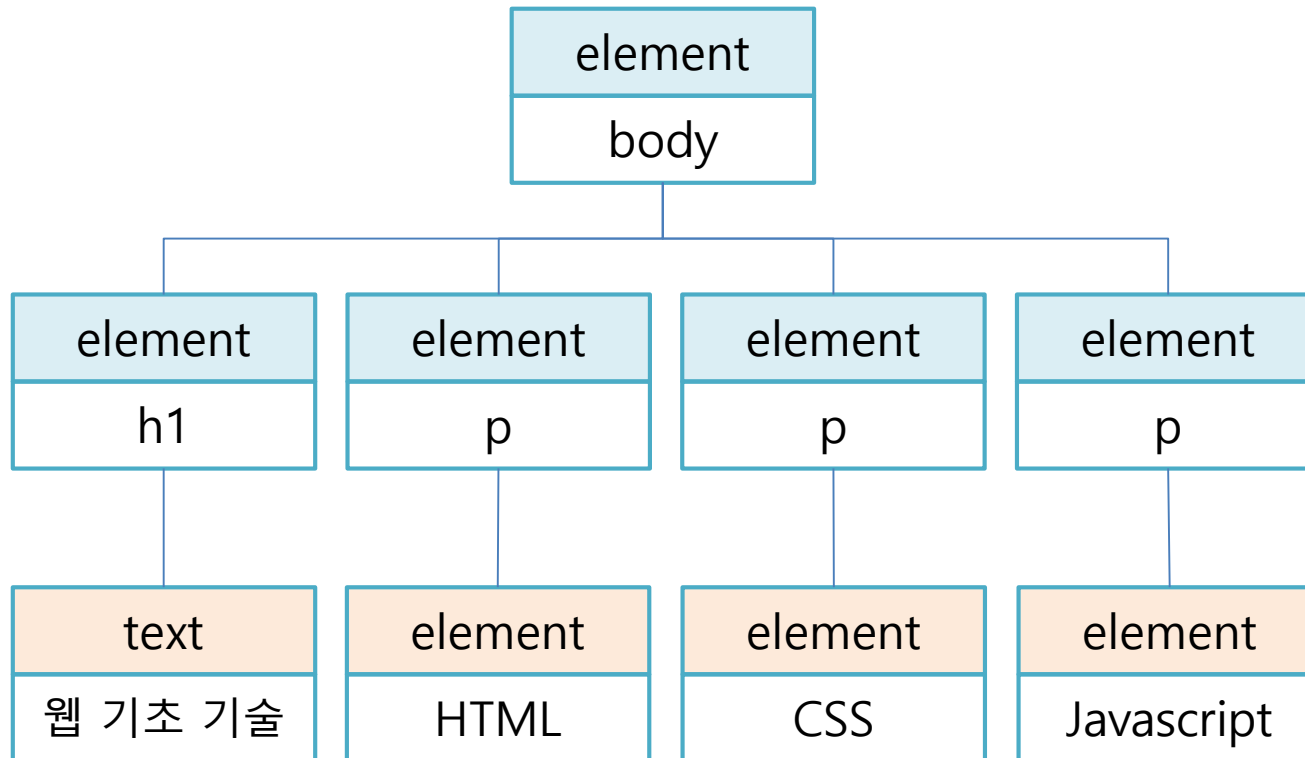
특정 이벤트가 발생했을때 DOM 트리에 새로운 노드를 추가하여 화면에 특정한 내용이 표시되도록 할 수 있다.

## 노드 구성 내용

- 모든 HTML 태그는 **요소 노드**가 된다.
- HTML 태그에서 사용하는 텍스트 내용은 **자식 노드인 텍스트 노드**가 된다.
- HTML 태그에 있는 속성은 모두 **자식 노드인 속성 노드**가 된다.

# DOM에서 노드 추가.삭제하기

## DOM 트리에 새로운 노드 추가



# DOM에서 노드 추가하기

- 텍스트가 있는 노드 추가하기

1. 요소 노드 만들기

```
document.createElement(요소명)
```

2. 텍스트 노드 만들기

```
document.createTextNode(요소명)
```

3. 자식 노드 연결하기

```
부모_노드.appendChild(자식_노드)
```

# DOM에서 노드 추가하기

- 텍스트 추가하기

```
<h1>웹 기초 기술</h1>
<p>HTML</p>
<p>CSS</p>
<p>Javascript</p>

<script>
  let newP = document.createElement("p");
  let textNode = document.createTextNode("Typescript");
  newP.appendChild(textNode);

  document.body.appendChild(newP);
</script>
```

# DOM에서 노드 추가하기

- 텍스트 추가하기

## 웹 기초 기술

HTML

CSS

Javascript

Typescript

```
<h1>웹 기초 기술</h1>
```

```
<p>HTML</p>
```

```
<p>CSS</p>
```

```
<p>Javascript</p>
```

```
▶ <script>...</script>
```

```
<p>Typescript</p>
```



# DOM에서 노드 추가하기

- 숨겨진 텍스트 보기

DOM 노드 추가하기

[더 보기](#)



DOM 노드 추가하기

[더 보기](#)

안녕하세요~ 숨겨진 텍스트입니다.

# DOM에서 노드 추가하기

node1.html

```
<h2>DOM 노드 추가하기</h2>
<!-- 주의! href="#" #을 반드시 표기할 것 -->
<a href="#" onclick="addP(); this.onclick='' ">더보기</a>
<div id="info"></div>

<script>
  function addP(){
    let p = document.createElement("p"); //p태그 생성
    let text = document.createTextNode("안녕하세요~ 숨겨진 텍스트입니다.");
    p.appendChild(text); //p태그에 내용 붙이기

    let div = document.getElementById("info");
    div.appendChild(p); //div 하위에 p태그 연결
  }
</script>
```

# DOM에서 노드 추가하기

- 속성값이 있는 노드 추가하기

1. 요소 노드 만들기

```
document.createElement(요소명)
```

2. 속성 노드 만들기

```
document.createAttribute(속성명)
```

```
노드명.value = 속성값
```

3. 속성 노드 연결하기

```
요소_노드.setAttribute(속성_노드)
```

# DOM에서 노드 추가하기

- 이미지 추가하기

## 웹 기초 기술

HTML

CSS

Javascript



# DOM에서 노드 추가하기

- 이미지 추가하기

```
<h1>웹 기초 기술</h1>
<p>HTML</p>
<p>CSS</p>
<p>Javascript</p>
<!-- 이미지 추가하기 -->

<script src="js/node.js"></script>
```

```
<h1>웹 기초 기술</h1>
<p>HTML</p>
<p>CSS</p>
<p>Javascript</p>
<!-- 이미지 추가하기 -->
<script src="js/node.js"></script>

```

# DOM에서 노드 추가하기

- 이미지 추가하기

```
//속성이 있는 노드 추가하기
let newImg = document.createElement("img");

//속성값 연결하기
newImg.setAttribute("src", "images/doit.jpg");
newImg.setAttribute("alt", "학생");

document.body.appendChild(newImg); //body에 연결하기
```

node.js

# 폼 요소에 접근하는 방법

## 체크 박스(CheckBox)의 checked 속성

Checkbox: ☒ ☐ Check Checkbox

```
Checkbox: <input type="checkbox" id="myCheck">

<button onclick="check()">Check Checkbox</button>
<button onclick="uncheck()">Uncheck Checkbox</button>

<script>
function check() {
    document.getElementById("myCheck").checked = true;
}

function uncheck() {
    document.getElementById("myCheck").checked = false;
}
</script>
```

# 폼 요소에 접근하는 방법

## 배송정보 자동으로 입력하기

온라인 쇼핑몰에서 상품 주문시  
주문한 사람의 정보와 배송을 받  
는 사람의 정보가 일치할 경우,  
**체크 상자만 누르면** 주문한 사람  
의 정보를 그대로 배송정보에 입  
력해 주는 것을 구현.

### 주문 정보

이름 장그래

연락처 01012345678

주소 서울시 구로구

### 배송 정보

☒ 주문 정보와 배송 정보가 같습니다

이름 장그래

연락처 01012345678

주소 서울시 구로구

결제하기



# 폼 요소에 접근하는 방법

## 배송정보 자동으로 입력하기 – order.html

```
<div id="container">
  <form name="order" action="order.jsp" method="post">
    <fieldset>
      <legend>주문 정보</legend>
      <ul>
        <li>
          <label class="field" for="billingName">이름</label>
          <input type="text" class="input-box" id="billingName" name="billingName">
        </li>
        <li>
          <label class="field" for="billingTel">연락처</label>
          <input type="text" class="input-box" id="billingTel" name="billingTel">
        </li>
        <li>
          <label class="field" for="billingAddr">주소</label>
          <input type="text" class="input-box" id="billingAddr" name="billingAddr">
        </li>
      </ul>
    </fieldset>
  </form>
```

# 폼 요소에 접근하는 방법

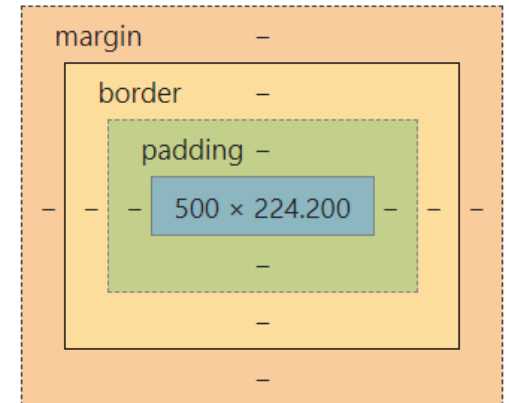
```
<form name="ship">
  <fieldset>
    <legend>배송 정보</legend>
    <ul>
      <li>
        <input type="checkbox" id="shippingInfo" name="shippingInfo">
        <label for="shippingInfo" class="check">주문 정보와 배송 정보가 같습니다</label>
      </li>
      <li>
        <label class="field" for="shippingName">이름</label>
        <input type="text" class="input-box" id="shippingName" name="shippingName">
      </li>
      <li>
        <label class="field" for="shippingTel">연락처</label>
        <input type="text" class="input-box" id="shippingTel" name="shippingTel">
      </li>
      <li>
        <label class="field" for="shippingAddr">주소</label>
        <input type="text" class="input-box" id="shippingAddr" name="shippingAddr">
      </li>
    </ul>
  </fieldset>
  <button type="submit" class="order">결제하기</button>
</form>
</div>
<script src="js/order.js"></script>
```

# 폼 요소에 접근하는 방법

## 배송정보 자동으로 입력하기 – order.css

```
#container{  
  width: 500px;  
  margin: 30px auto;  
}  
fieldset{  
  border: 1px solid #ccc;  
  margin-bottom: 35px;  
}  
legend{  
  font-size: 1.2em;  
  font-weight: bold;  
  margin-left: 10px;  
}  
ul{ list-style: none; }  
li{ margin: 20px; }
```

```
.field{  
  width: 60px;  
  float: left;  
  line-height: 2em;  
  text-align: right;  
  margin-right: 10px;  
}  
.input-box{  
  width: 250px;  
  height: 30px;  
  border: 1px solid #aaa;  
  border-radius: 5px;  
}  
.order{  
  width: 100%;  
  padding: 10px;  
  font-size: 1em;  
  border: 1px solid #ccc;  
}
```



# 폼 요소에 접근하는 방법

## 배송정보 자동으로 입력하기 – order.js

```
//체크 박스 속성에 이벤트 처리하기
var check = document.querySelector("#shippingInfo");

check.addEventListener("click", checkBox);
//체크박스에 click 이벤트 발생했을 때 함수 호출

var billingName = document.querySelector("#billingName");
var billingTel = document.querySelector("#billingTel");
var billingAddr = document.querySelector("#billingAddr");

var shippingName = document.querySelector("#shippingName");
var shippingTel = document.querySelector("#shippingTel");
var shippingAddr = document.querySelector("#shippingAddr");
```

# 폼 요소에 접근하는 방법

## 배송정보 자동으로 입력하기 – order.js

```
function checkBox(){  
    if(check.checked==true){ //체크박스에 체크 되었다면  
        shippingName.value = billingName.value;  
        //주문 정보(이름)을 배송 정보(이름)에 대입  
        shippingTel.value = billingTel.value;  
        shippingAddr.value = billingAddr.value;  
    }else{  
        shippingName.value = "";  
        shippingTel.value = "";  
        shippingAddr.value = "";  
    }  
}
```

# 선택한 옵션 항목 찾아내기

## onchange 속성 – HTML Reference > HTML Events > Form Events

### HTML Reference

HTML by Alphabet

HTML by Category

HTML Browser Support

HTML Attributes

HTML Global Attributes

HTML Events

HTML Colors

HTML Canvas

HTML Audio/Video

HTML Character Sets

## Form Events

Events triggered by actions inside a HTML form (applies to form elements):

Attribute	Value	Description
<u>onblur</u>	<i>script</i>	Fires the moment that the element loses focus.
<u>onchange</u>	<i>script</i>	Fires the moment when the value of the element changes.
<u>oncontextmenu</u>	<i>script</i>	Script to be run when a context menu is invoked.

# 선택한 옵션 항목 찾아내기

## onchange 속성 예제.

Select a new car from the list.

BMW ▼

When you select a new car, a function is triggered which outputs the value of the selected car.

You selected: BMW

```
<p>Select a new car from the list.</p>

<select id="mySelect" onchange="myFunction()">
  <option value="Audi">Audi
  <option value="BMW">BMW
  <option value="Mercedes">Mercedes
  <option value="Volvo">Volvo
</select>

<p>When you select a new car, a function is triggered which outputs the value of
the selected car.</p>

<p id="demo"></p>

<script>
function myFunction() {
  var x = document.getElementById("mySelect").value;
  document.getElementById("demo").innerHTML = "You selected: " + x;
}
</script>
```

# 선택한 옵션 항목 찾아내기

## 선택 목록에서 사용자가 선택한 옵션 항목 찾아내기

수

127.0.0.1:5500 내용:  
[전기전자공학과]를 선택했습니다.

확인

이름

장그래

학과

산업공학과 ▼

신청 과목

이 달에 신청할 과목을 선택하세요.  
☐ 회화 ☒ 문법 ☐ 작문

메일링

메일로 받고 싶은 뉴스 주제를 선택해 주세요  
☒ 해외 단신 ☐ 5분 회화 ☒ 모닝팝스



# 선택한 옵션 항목 찾아내기

```
<div id="container">
  <h1>수강신청</h1>
  <form name="testForm">
    <fieldset>
      <legend>신청인</legend>
      <ul>
        <li>
          <label class="reg" for="userName">이름</label>
          <input type="text" id="userName" name="userName" autofocus required>
        </li>
        <li>
          <label class="reg" for="class">학과</label>
          <select name="major" id="major" onchange="displaySelect()">
            <option>---- 학과 선택 ----</option>
            <option value="archi">건축학과</option>
            <option value="mechanic">기계공학과</option>
            <option value="elec">전기전자공학과</option>
            <option value="computer">컴퓨터공학과</option>
            <option value="chemical">화학공학과</option>
          </select>
        </li>
      </ul>
    </fieldset>
  </form>
</div>
```

major.html

# 선택한 옵션 항목 찾아내기

major.html

```
<fieldset>
  <legend>신청 과목</legend>
  <p>이 달에 신청할 과목을 선택하세요.</p>
  <label><input type="radio" name="subject" value="speaking">회화</label>
  <label><input type="radio" name="subject" value="grammar">문법</label>
  <label><input type="radio" name="subject" value="writing">작문</label>
</fieldset>
<fieldset>
  <legend>메일링</legend>
  <p>메일로 받고 싶은 뉴스 주제를 선택해 주세요</p>
  <label><input type="checkbox" name="mailing1" value="news">해외 단신</label>
  <label><input type="checkbox" name="mailing2" value="dialog">5분 회화</label>
  <label><input type="checkbox" name="mailing3" value="pops">모닝팝스</label>
</fieldset>
</form>
</div>
```

# 선택한 옵션 항목 찾아내기

```
#container {  
    width:500px;  
    margin:10px auto;  
}  
form fieldset{margin-bottom:25px;}  
form legend{  
    font-size:15px;  
    font-weight:700;  
    margin-left: 10px;  
}  
form ul li{  
    list-style:none;  
    margin: 15px 0;  
    font-size:14px;  
}  
label.reg {  
    width:80px;  
    float:left;  
    text-align:right;  
    margin-right:10px;  
    font-size:15px;  
    color:■ #390;  
    font-weight:bold;  
}
```

major.css

# 선택한 옵션 항목 찾아내기

## 선택 목록에서 사용자가 선택한 옵션 항목 찾아내기

```
//select 태그 이름에 폼 이름으로 접근
var selectMenu = document.testForm.major;

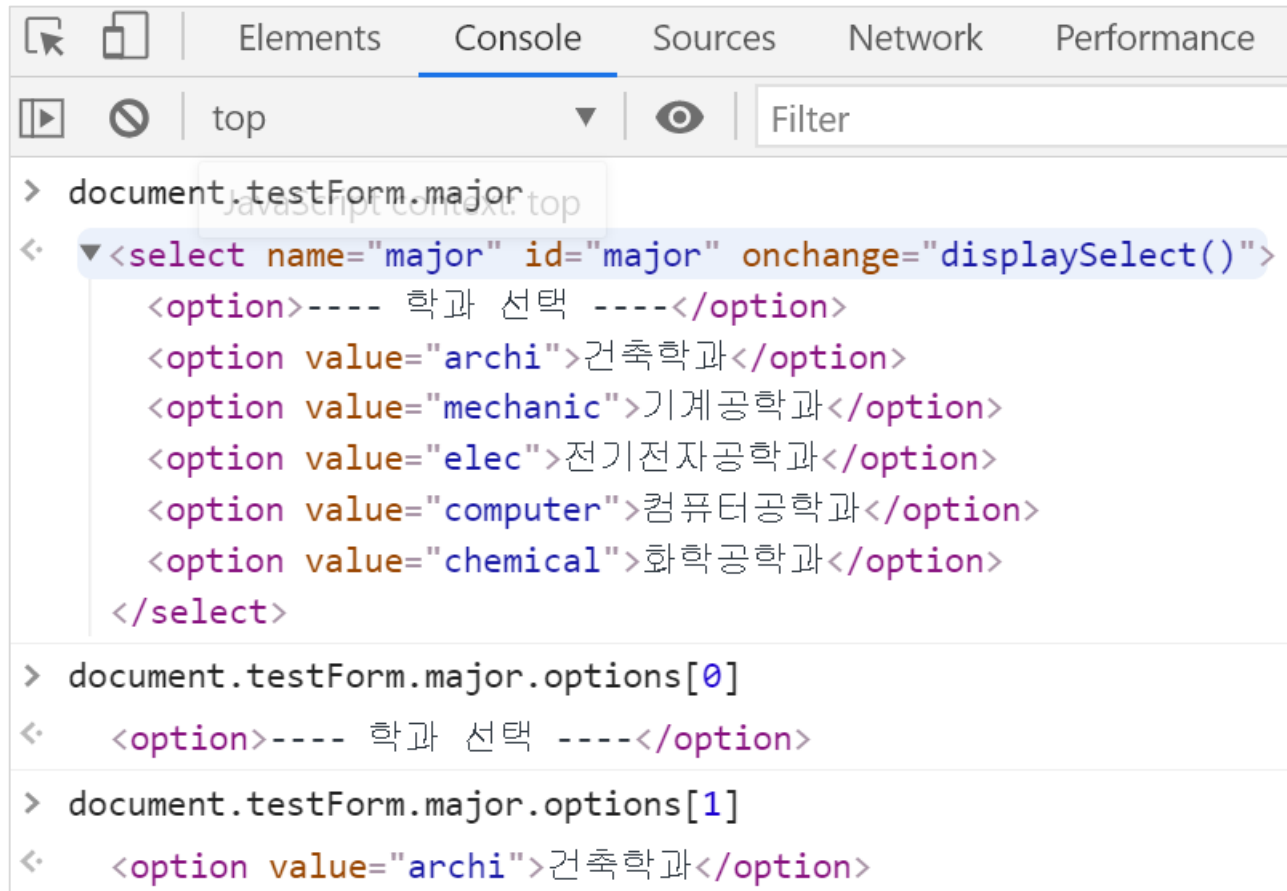
function displaySelect(){
    //select option속성의 인덱스로 선택
    // var selectedText = selectMenu.options[selectMenu.selectedIndex].innerText;
    // alert(selectedText + "가 선택되었습니다.");

    //id 선택자로 처리
    var selectedText = document.getElementById("subj").value;
    alert(selectedText + "가 선택되었습니다.");
}
```

major.js

# 선택한 옵션 항목 찾아내기

## 콘솔 창에서 보기



The screenshot shows a web browser's developer console with the 'Console' tab selected. The console displays the following code and its output:

```
> document.testForm.major
< <select name="major" id="major" onchange="displaySelect()">
  <option>---- 학과 선택 ----</option>
  <option value="archi">건축학과</option>
  <option value="mechanic">기계공학과</option>
  <option value="elec">전기전자공학과</option>
  <option value="computer">컴퓨터공학과</option>
  <option value="chemical">화학공학과</option>
</select>

> document.testForm.major.options[0]
< <option>---- 학과 선택 ----</option>

> document.testForm.major.options[1]
< <option value="archi">건축학과</option>
```

# Todo-list 만들기

- Todo List

## 할 일 목록

자바스크립트 연습하기

일본어 공부하기

클래식 연주 듣기

# Todo-list 만들기

## ● Todo List

```
<h1>할 일 목록</h1>

<div id="controls">
  <input type="text" id="todoInput" placeholder="할 일을 입력하세요">
  <button id="addBtn">추가</button>
</div>

<ul id="todo-list">
  <li>자바스크립트 연습하기</li>
  <li>일본어 공부하기</li>
</ul>

<script src="js/todo-list.js"></script>
```

# Todo-list 만들기

## ● Todo List

```
#todo-list {  
  padding: 0;  
  list-style-type: none;  
}  
#todo-list li {  
  background: □ #f9f9f9;  
  margin: 5px 0;  
  padding: 10px;  
  border-radius: 5px;  
  border: 1px solid □ #ddd;  
}  
#controls {  
  margin: 10px 0;  
}
```

style.js

```
input {  
  padding: 5px;  
  font-size: 14px;  
}  
button {  
  padding: 6px 12px;  
  margin-left: 5px;  
  cursor: pointer;  
}
```



# Todo-list 만들기

## ● Todo List

```
//태그 요소 선택
const todoInput = document.getElementById("todoInput");
const addBtn = document.getElementById("addBtn");
const todoList = document.getElementById("todo-list");

//버튼 클릭 이벤트
addBtn.addEventListener("click", () => {
  const todoText = todoInput.value; //입력값 가져오기

  //새로운 <li> 생성
  const newItem = document.createElement("li");
  newItem.textContent = todoText;

  todoList.appendChild(newItem); //리스트에 추가

  //입력창 초기화
  todoInput.value = "";
  todoInput.focus();
})
```

todo-list.js

# Todo-list 만들기

- Todo List

```
//엔터로 할 일 추가
todoInput.addEventListener("keydown", (event) => {
  if(event.key == "Enter"){
    addBtn.click(); //버튼 클릭 실행
  }
})
```

# 여행 준비물 프로그램 만들기

## 여행 준비물 점검 목록

<input type="text"/>	추가
칫솔	X
비누	X
치약	X

```
<div id="wrapper">
  <h2>여행 준비물 점검 목록</h2>
  <form action="addItem.jsp" method="post">
    <input type="text" name="item" id="item" autofocus>
    <button type="button" id="add" class="btnAdd">추가</button>
  </form>
  <div id="itemList"></div>
</div>

<script src="./js/travel.js"></script>
```

# 여행 준비물 프로그램 만들기

```
/* 폼 스타일 */
#wrapper{width: 600px; margin:0 auto;}
h2{text-align: center;}
form{background: #0094ff; padding: 30px 40px;}
input{width: 420px; padding: 10px;}
.btnAdd{width: 60px; padding: 10px;}

/* itemList 스타일*/
ul{margin:0; padding: 0; }
ul li{
    list-style: none;
    padding: 10px 10px 10px 40px;
    background: #eee;
    position: relative;
}
ul li:hover{background: #ddd;}
.close{
    position: absolute;
    top:0; right:0;
    padding: 10px 20px;
}
.close:hover{background: rgb(30, 100, 200); color: #fff;}
```

item.css

# 여행 준비물 프로그램 만들기

## 1. 물품(아이템) 추가

```
// 아이템 리스트
let itemList = []

let addBtn = document.getElementById("add");
addBtn.addEventListener("click", addList);

//아이템 추가 함수
function addList(){
    let item = document.getElementById("item").value; //아이템 입력

    itemList.push(item); //아이템 추가
    document.getElementById("item").value = ""; //아이템 초기화
    document.getElementById("item").focus(); //커서 위치

    //console.log(itemList);

    showList(); //목록 조회 호출
}
```

```
> itemList
< ▶ []

> itemList
< ▶ ["비누"]

> itemList
< ▶ (2) ["비누", "치솔"]
```

# 여행 준비물 프로그램 만들기

## 2. 물품 목록 조회

```
//아이템 목록 조회 함수
function showList(){
    let list = "<ul>"; //목록 리스트 변수 할당
    for(var i = 0; i < itemList.length; i++){
        //document.getElementById("itemList").innerHTML = itemList[i];
        list += "<li>" + itemList[i] + "<span class='close' id=" + i + ">X</span></li>";
    }
    list += "</ul>"
    document.getElementById("itemList").innerHTML = list;

    //아이템 삭제 - 'X' 클릭
    let removeList = document.querySelectorAll(".close");

    for(var i = 0; i < removeList.length; i++){
        removeList[i].addEventListener("click", remove);
    }
}
```

# 여행 준비물 프로그램 만들기

## 3. 물품(아이템) 삭제

추가

비누	X
칫솔	X
치약	X

Elements Console Sources Network Performance Memory Application Security

top Filter Default levels

```
<span class="close" id="0">X</span>  
<span class="close" id="1">X</span>  
<span class="close" id="2">X</span>
```

```
//삭제 함수  
function remove(){  
  console.log(this); //클릭 이벤트가 일어난 대상  
  let id = this.getAttribute("id"); //인덱스 가져오기  
  itemList.splice(id, 1); //인덱스가 id인 요소 1개 삭제  
  
  showList();  
}
```