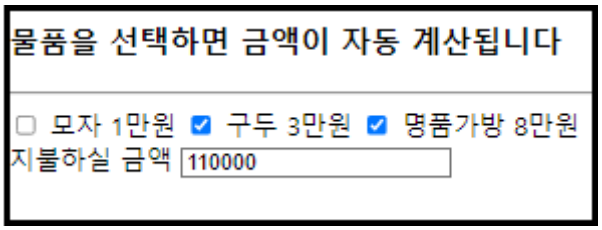
[11/10] Javascript : parseInt, Audio & Video, control list, onloadedmetadata, Worker

1) parseInt : 문자열에서 숫자를 추출, 입력받은 문자열을 실제 숫자값으로 바꾸어 반환

```
<body>
    <h3>물품을 선택하면 금액이 자동 계산됩니다</h3><hr>
    <form>
    <input type="checkbox" name="cap" value="10000" onclick="a(this)"> 모자 1만원
    <input type="checkbox" name="shoes" value="30000" onclick="a(this)"> 구두 3만원
    <input type="checkbox" name="bag" value="80000" onclick="a(this)"> 명품가방 8만원<br>
    지불하실 금액 <input type="text" id="sumtext" value=""><br>
    </form>
</body>
<script>
    let sum = 0;
    function a(b){
        if(b.checked){sum += parseInt(b.value);}
        else{sum -= parseInt(b.value);}
        document.getElementById("sumtext").value = sum;}
</script>
```



클릭으로 넣고 뺀 가격이 즉시 계산됨

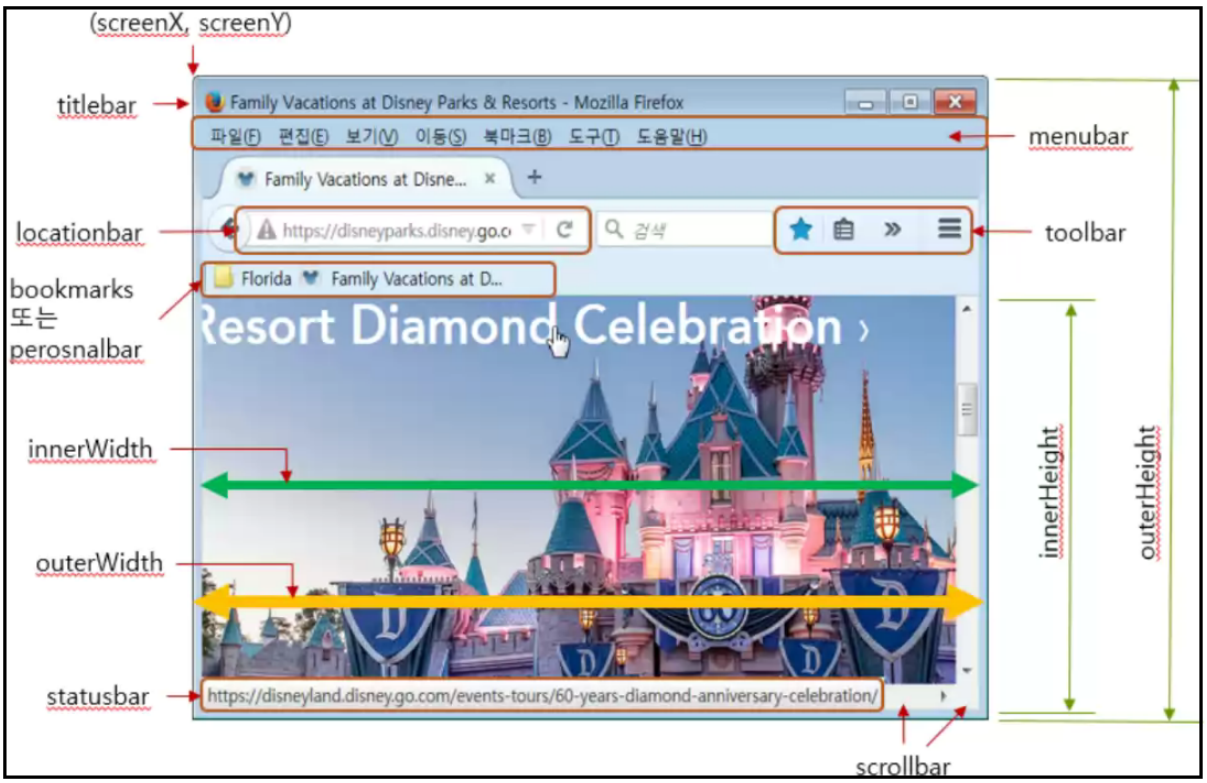
- parseInt + if 문

```
function a(b){
    if(b.checked){sum += parseInt(b.value);}
    else{sum -= parseInt(b.value);}
    document.getElementById("sumtext").value = sum;}
```

  - 파라미터 b 를 if의 의 인자 값으로 받음
  - .checked : 체크된 값들만
  - sum += : sum의 값에서 증가
  - sum -= : sum의 값에서 감소
  - parseInt(b.value) : b의 값을 계산

  - for(i=0 ~) 같은 배열로 받으면 코드가 복잡해진다.
  - input type = “checkbox” 가 아닌 radio 로 받으면 사칙연산이 적용 안됨

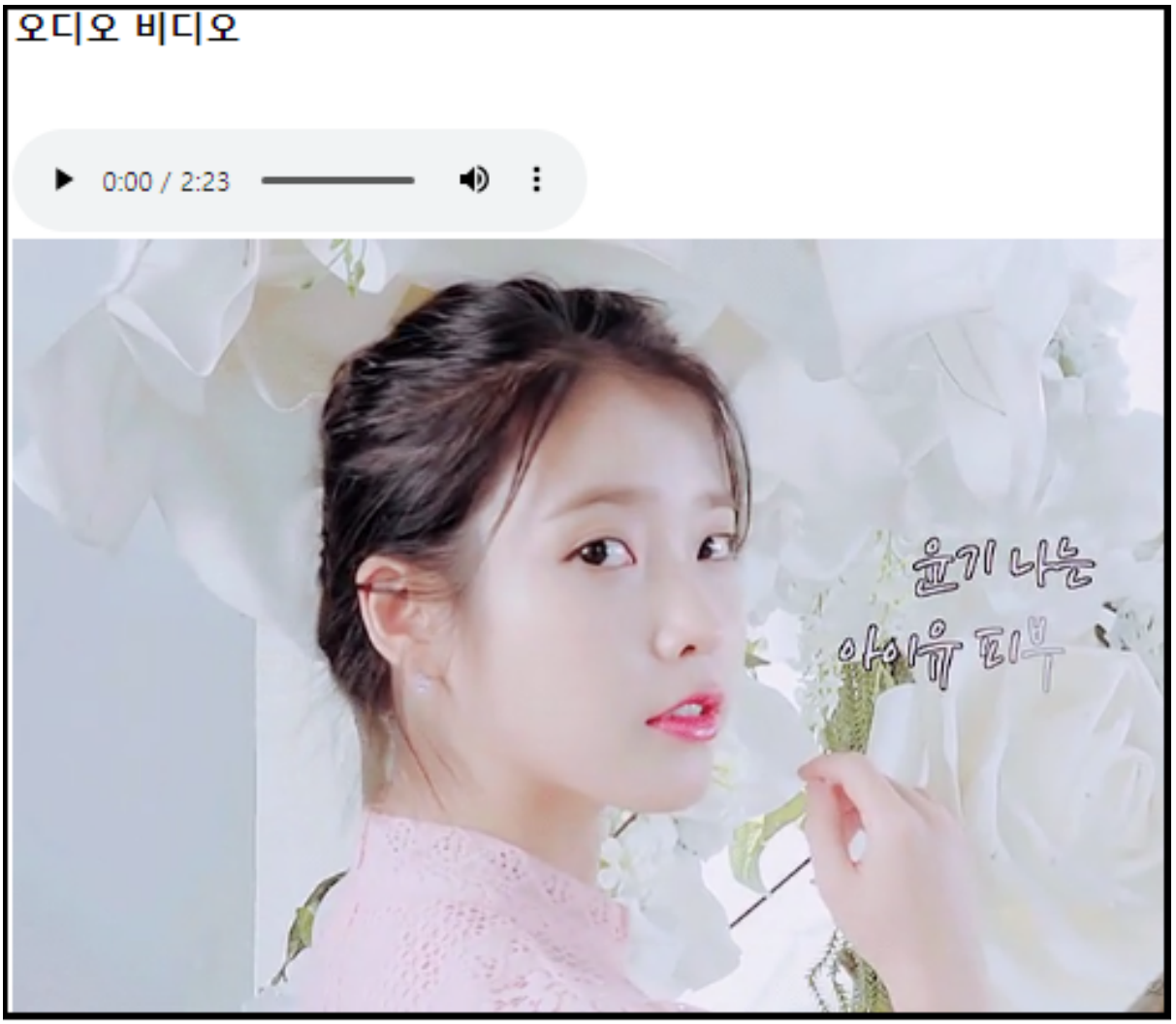
2) HTML DOM 트리


(screenX, screenY)
titlebar
Family Vacations at Disney Parks & Resorts - Mozilla Firefox
menubar
locationbar
toolbar
bookmarks
또는
perosnalbar
Florida
Family Vacations at D...
innerWidth
outerWidth
innerHeight
outerHeight
statusbar
https://disneyland.disney.go.com/events-tours/60-years-diamond-anniversary-celebration/
scrollbar

3) Audio / Video

```
<body>
    <h3>오디오 비디오</h3><br>
    <audio id="audio" src="img\1.mp3" loop controls>태그 미지원</audio>
    <video id="video" src="img\1.mp4" width="100%" height="100%" loop controls></video>
</body>
```

이렇게 작성해도 현 시점에서 잘 작동된다.



안구 정화의 시간

- controls : 영상 및 음악의 시작과 종료를 선택 가능
- loop controls : 영상, 음악을 시장시 무한 반복
    - 컨트롤이 없을 경우 영상이 아닌 스냅샷만 등장

```
<body>
    <h3>오디오 비디오</h3><br>
    <audio id="audio" src="img\1.mp3" loop controls>태그 미지원</audio>
    <video id="video" width="100%" height="100%" loop controls>
     <source src="img\1.mp4" type="video/mp4"> 태그 미지원</video>
</body>
```

강사님이 작성한 코드

- 제어 목록 구현

```
<body>
    <h3>아이유.mp4</h3>
    <hr>
    <video id="video" src="img/1.mp4"></video>
    <div id="msg">아이유짤 제어 목록</div>
    <button type="button" id="play" onclick="control(event)">play</button>
    <button type="button" id="pause" onclick="control(event)">pause</button>
    <button type="button" id="replay" onclick="control(event)">replay</button>
    <button type="button" id="vol+" onclick="control(event)">vol+</button>
    <button type="button" id="vol-" onclick="control(event)">vol-</button>
    <button type="button" id="mute on/off" onclick="control(event)">mute on/off</button>
<script>
    let div = document.getElementById("msg");
    let video = document.getElementById("video");
    function control(e){let id = e.target.id;
        if (id == "play"){video.play(); div.innerHTML = video.src + "재생"}
        else if(id == "pause"){video.pause();div.innerHTML = "일시정지";}
        else if(id == "replay"){video.load(); video.play(); div.innerHTML = video.src + "처음부터 재생";}
        else if(id == "vol-"){video.volume -= 0.1;
            if(video.volume < 0.1) video.volume = 0; div.innerHTML = "음량 0.1 감소" + "현재" + video.volume;}
        else if(id == "vol+"){video.volume += 0.1;
            if(video.volume > 0.9) video.volume = 1.0 ;div.innerHTML = "음량 0.1 증가" + "현재" + video.volume;}
        else if(id == "mute on/off"){video.muted = !video.muted;
            if(video.muted) {div.innerHTML = "muted on";}
        else {div.innerHTML = "muted off";}}}
    </script>
</body>
```

  - toggle : 클릭시 on/off 기능
    - video.mute =! video.muted
    - video.mute = video.mute =! video.muted 가 원론
  - replay : 클릭시 재시작
    - video.replay() 보다는 video.load(); audio.play() 로 작성

  - type="video/mp4" : 다른 고화질을 적용이 안될수도 있으므로 지정

- onloadedmetadata : 비디오의 원본 파일 호출

```
<body>
    <h3>비디오 원본 출력</h3><hr>
    <video id="video"  width="0" height="0" controls>
        <source src="img\1.mp4" type="video/mp4">
    </video>
</body>
<script>
    let video = document.getElementById("video");
    video.onloadedmetadata = function f(e){
        alert(video.videoWidth + "x" + video.videoHeight);
        video.width = video.videoWidth;
        video.height = video.videoHeight;}
</script>
```

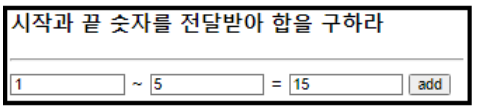
- video.width = video.videoWidth; : 원본 크기로 출력
  video.height = video Height;
  - video.videowidth : 비디오 파일의 가로축

4) Worker : https://developer.mozilla.org/ko/docs/Web/API/Web_Workers_API

```html
<body>
    <h3>시작과 끝 숫자를 전달받아 합을 구하라</h3><hr>
        <input id="from" type="text" size="10"> ~
        <input id="to" type="text" size="10"> =
        <input id="sum" type="text" size="10">
        <button type="button" id="add" onclick="a()">add</button>
    <script>
        let b = new Worker("add.js");
        function a(){let c = {
            from : document.getElementById("from").value,
            to : document.getElementById("to").value};
            b.postMessage(c);}
        b.onmessage = function (e){document.getElementById("sum").value = e.data;}
    </script>
</body>
```

html

```js
onmessage = function (e){
    let sum = 0;
    let from = parseInt(e.data.from);
    let to = parseInt(e.data.to);
    for(let i=from; i<=to; i++) sum += i ;
    postMessage(sum);}
```



js / 브라우저 화면

- Worker : Web Worker 은 main thread(HTML) 와 별도의 worker thread(JavaScript) 를 가진다.
    - 싱글스레드 기반으로 작동하는 javascript는 싱글스레드의 단점을 보완하기 위해 코드들이 비동기로 실행된다. 그러나 API비동기 실행이 너무 많이 쌓이게 되면 모든 작업의 실행 속도가 느려질 수 있다. 이 문제를 해결하기 위해 나온 것이 바로 워커(Worker) 이다.
    - 양측 모두 postMessage() 메서드를 사용해 전송하고, onmessage 이벤트 처리기(메시지는 Message의 data 속성에 들어있습니다)를 사용해 수신합니다. 전송하는 데이터는 복사하며 공유하지 않습니다.

- 작성 과정
    - HTML(main thread) 에서 new **Worker**('add.js') 로 add.js라는 자바스크립를 만듬
    - b.onmessage 에 **Worker** 의 메시지를 전달받기 위한 이벤트 핸들러를 등록
    - 자바스크립트에서 add.js 의 작업 처리후 postMessage() 로 데이터를 HTML의 b.onmessage 에게 전달한다.
    - 만약 더이상add.js(worker thread) 가 필요 없거나, 작업을 종료해야 한다면, b.terminate() 를 호출한다.

- let b = new Worker("add.js") : Worker 생성자 생성, add.js 와 연결
- function a() ~ : from 의 값과 to 의 값을 b.postMessage() 를 통해 add.js 로 전송
- b.onmessage : add.js 의 onmessage 의 결과값(e.data)을 id="sum" 에 전달, 화면에 출력됨
- postMessage() : html 과 js 간의 메시지 수신, 전송하는 데이터는 복사하며 공유하지 않음

5) Guess My Number!

```js
'use strict';


let secretNumber = Math.trunc(Math.random() * 100) + 1;
let score = 20;
let highscore = 0;

const displayMessage = function (message) {
  document.querySelector('.message').textContent = message;
};

document.querySelector('.check').addEventListener('click', function () {
  const guess = Number(document.querySelector('.guess').value);
  console.log(guess, typeof guess);

  // When there is no input
  if (!guess) {
    // document.querySelector('.message').textContent = '⛔ No number!';
    displayMessage('⛔ No number!');

    // When player wins
  } else if (guess === secretNumber) {
    // document.querySelector('.message').textContent = '🎉 Correct Number!';
    displayMessage('🎉 Correct Number!');
    document.querySelector('.number').textContent = secretNumber;

    document.querySelector('body').style.backgroundColor = '#60b347';
    document.querySelector('.number').style.width = '30rem';

    if (score > highscore) {
      highscore = score;
      document.querySelector('.highscore').textContent = highscore;
    }

    // When guess is wrong
  } else if (guess !== secretNumber) {
    if (score > 1) {
      // document.querySelector('.message').textContent =
      // guess > secretNumber ? '📈 Too high!' : '📉 Too low!';
      displayMessage(guess > secretNumber ? '📈 Too high!' : '📉 Too low!');
      score--;
      document.querySelector('.score').textContent = score;
    } else {
      // document.querySelector('.message').textContent = '💥 You lost the game!';
      displayMessage('💥 You lost the game!');
      document.querySelector('.score').textContent = 0;
    }
  }

});

document.querySelector('.again').addEventListener('click', function () {
  score = 20;
  secretNumber = Math.trunc(Math.random() * 10) + 1;

  // document.querySelector('.message').textContent = 'Start guessing...';
  displayMessage('Start guessing...');
  document.querySelector('.score').textContent = score;
  document.querySelector('.number').textContent = '?';
  document.querySelector('.guess').value = '';

  document.querySelector('body').style.backgroundColor = '#222';
  document.querySelector('.number').style.width = '15rem';
});
```

js

- use strict : 함수호출에 있어서 정확한 값이 사용되어 빠른 수행, 에러 예방, 디버깅에 편리

- const : 블록 범위의 상수를 선언 contents를 의미
    - const 는 let 과 유사한 편

- querySelector : 파라미터에 일치하는 첫 번째 Element를 반환

```
<body>
    <h1>h1</h1>
    <h2 class="h2">h2</h2>
    <h3 id="h3">h3</h3>
    <h3>h3_2</h3>
    <div>
        <span>Span1</span>
        <span>Span2</span>
    </div>
    <script src="main.js"></script>
</body>
```

h1

h2

h3

h3_2

Span1 Span2

html / 브라우저 화면

```
const a = document.querySelector("h1");
a.style.color = "red";
```

js

    - tag : document.querySelector("h1")
    - class : document.querySelector(".class")
    - id : document.querySelector("#h3")
    - div , span : document.querySelector("div span")
        - 첫 번째 element만 반환, "div span:first-child"
        - 두 번째 element에 적용시, “nth-child” 혹은 “last-child” 를 사용[1]

- Math.trunc : 사이의 값, 소수점 버림
- displayMessage : 화면에 나오는 메시지
- @import url(‘’) : 외부주소를 가져옴, //css

[1] [JS] 자바스크립트 querySelector 함수 사용법 및 예제

for Code Newbie

getElementByid 는 주로 function 의 기능값{ } 내에서 지정되어 사용된다

1) let c = document.getElementByid("c")

- 변수(let c) 와 id("c") 를 통한 연결

```
let c = document.getElementById("c");
```

- 변수(let c) 와 특정 영역 id="c" 를 연결, 이때 c 는 <select> 전체를 의미

```
<select id="c" onchange="a()">
    <option value="img\1.png">apple</option>
    <option value="img\2.png">banana</option>
    <option value="img\3.png">mango</option>
</select>
```

2) let kc = document.getElementByName("city")

- 변수(let kc)와 Name("city") 를 통한 연결

```
let kc = document.getElementsByName("city");
```

- 변수(let kc)와 특정 인자 name="city" 를 연결, form 내의 일부와의 연결을 의미

```
<input type="radio" name="city" value="seoul" checked>서울
<input type="radio" name="city" value="busan">부산
<input type="radio" name="city" value="suwon">수원
<input type="button" value="fine checked" onclick="fineChecked()">
```

3) document.getElementByid("c").onmouseover = a

- document.getElementByid("c") 와 .기능의 형태로 사용
  - 항상 변수로 설정해야 하는 것은 아니다. function 내에서 한줄로 작성
  - document.getElementByid("c") 와 onmouseover 의 형태로도 확인된다.
- 구문 전체가 필요한 편

```
<body>
    <p id="p">마우스를 올리면 이벤트 객체와 메시지 출력</p>
    <button onclick="a(event)">클릭</button>
</body>
<script>
    function a(b){alert(b.type);}
    document.getElementById("p").onmouseover=a;
</script>
```

4) document.getElementByid(name).focus()

- document.getElementByid(name) 과 .기능의 형태로 사용
- function 내부가 아닌 <body>의 태그 속성으로 지정되었다.

```
<body onload="document.getElementById(name).focus();">
    <form>
        이름 : <input type="text" id="name" onblur="a(this)"><p>
        학번 : <input type="text" id="name">
    </form><br>
</body>
<script>
    function a(b){if(b.value == ""){b.focus();}}
</script>
```

5) document.getElementById("checkbox").addEventListener("click", function(event){ event.preventDefault()}; window.alert('test)

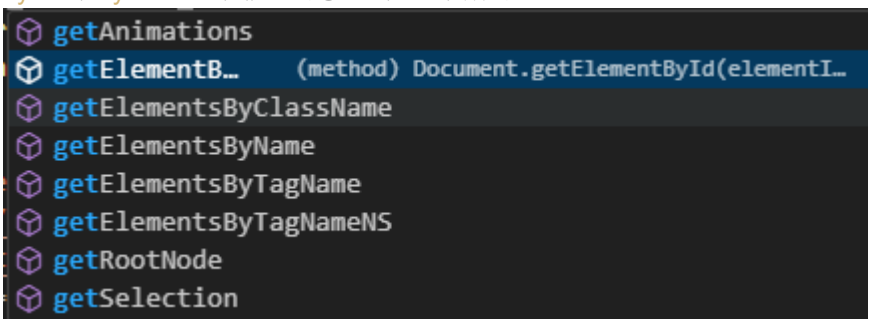
```
<script>
    window.onload = function(){
        document.getElementById("checkbox").addEventListener("click", function(event){
            event.preventDefault();
            window.alert('test');});}
</script>
    <input type="checkbox" id="checkbox"><hr>
```

- document.getElementById("checkbox") : check 박스와 연결
- .addEventListener : addEventListener 의 기능으로써
- "click", function(event) : 클릭하면 function(event)를 시행
- event.preventDefault : 해당 이벤트는 클릭불가
- window.alert('test) : 알림창과 메시지 test 출력
- 즉, checkbox 를 유저가 누르면, 클릭이 안되면 대신 test 메시지 출력

[정리] document.getElementById(“ ”) 는 다양한 형태로 작성할 수 있다.

A. function 내부와 태그 속성에서도 작성된다.
   a. 1),2),3),5) 는 function 내부에서 document.getElementById(“ ”) 의 형태를 가졌다.
   b. 4) 는 태그 속성으로 document.getElementById(“ ”) 의 형태로 작성되었다.

B. Byid 과 ByName 외에도 다양한 구문이 있다.



C. Byid(“ ”) 로 지정된 곳은 태그 전체를 의미하거나 특정 인자를 지칭할 수 있다.
   a. 1) 에서 변수(let c) 와 <select> 태그 전체를 연결 했다.
   b. 2) 에서 변수(let kc)와 Name(“city”) 만을 연결했다.

D. 변수로 설정하거나 .기능의 형태로도 작성된다.
   a. 1),2) 는 let c = document.getElementByid(“c”) 의 변수로 설정했다.
   b. 3),4),5) 는 document.getElementByid(“c”).onmouseover 의 형태로 .기능이 붙어서 작성되었다.
   c. .기능은 onmouseover, focus, addEventListener 등 다양한 구문이 있을 것

E. . 기능은 더욱 세부적인 설정으로 가능해진다.
   a. 5) 의 코드처럼 document.getElementById("checkbox").addEventListener("click", function(event){event.preventDefault()}; window.alert(‘test) 긴 형태를 취한다.