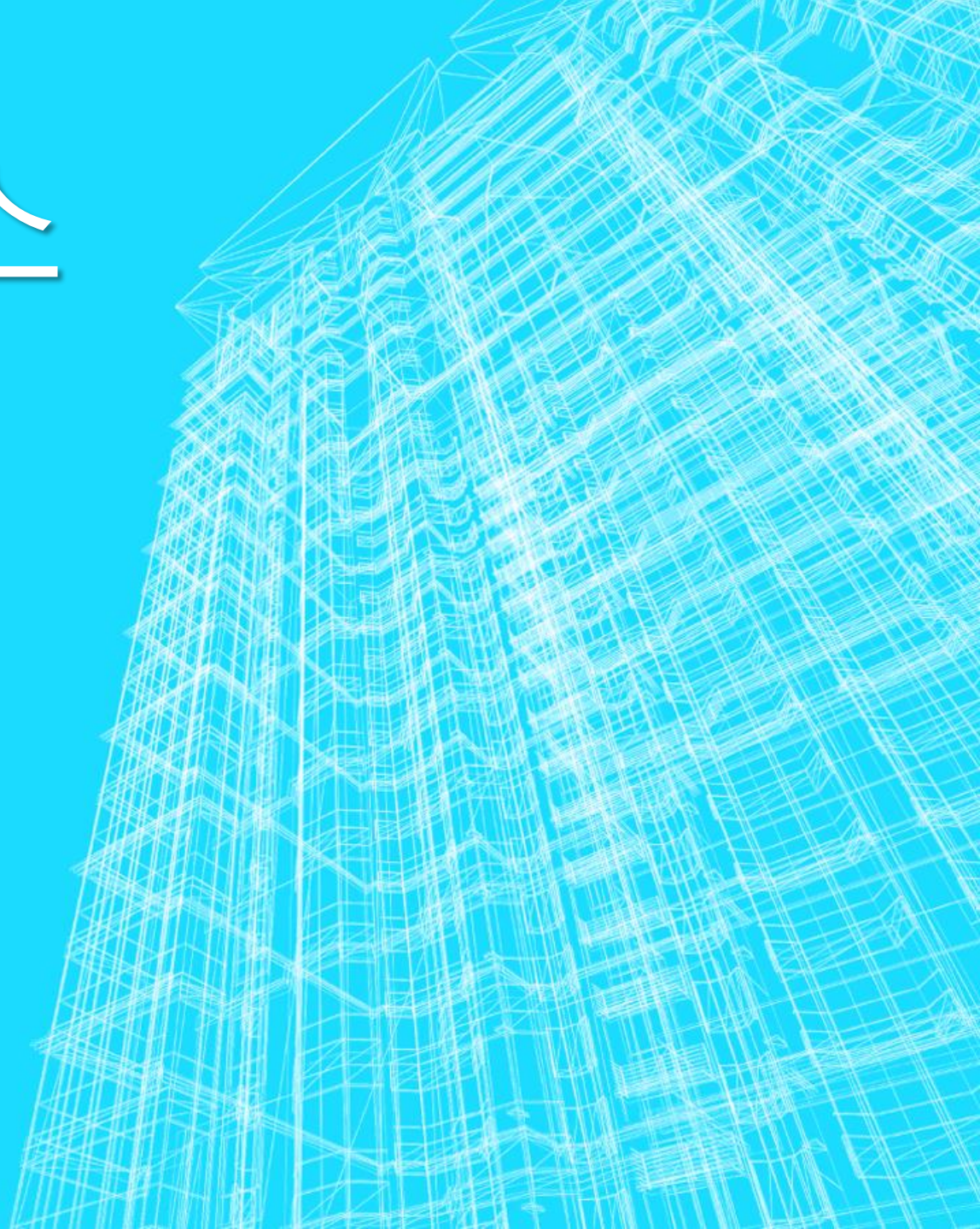


# 네트워크 서비스 프로토콜

## 3주차 2

소프트웨어학부  
김형균 교수



# 주차별 수업 계획

01주차	9월 3일	교과목 소개	09주차	10월 29일	몽고디비 1
	9월 5일	교과목 소개		10월 31일	몽고디비 2
02주차	9월 10일	노드 시작하기	10주차	11월 5일	익스프레스로 SNS 서비스 만들기 1
	9월 12일	추석 휴무		11월 7일	익스프레스로 SNS 서비스 만들기 2
03주차	9월 17일	알아두어야 할 자바스크립트 1	11주차	11월 12일	웹 API 서버 만들기 1
	9월 19일	알아두어야 할 자바스크립트 2		11월 14일	웹 API 서버 만들기 2
04주차	9월 24일	노드 기능 알아보기 1	12주차	11월 19일	웹 소켓으로 실시간 데이터 전송하기 1
	9월 26일	노드 기능 알아보기 2		11월 21일	웹 소켓으로 실시간 데이터 전송하기 2
05주차	10월 1일	개천절 휴무	13주차	11월 26일	실시간 경매 시스템 만들기 1
	10월 3일	패키지 매니저		11월 28일	실시간 경매 시스템 만들기 2
06주차	10월 8일	익스프레스 웹 서버 만들기 1	14주차	12월 3일	과제발표 1
	10월 10일	익스프레스 웹 서버 만들기 2		12월 5일	과제발표 2
07주차	10월 15일	MySQL 1	15주차	12월 10일	기말고사
	10월 17일	MySQL 2		12월 12일	보충수업
08주차	10월 22일	중간고사			
	10월 24일	보충수업			



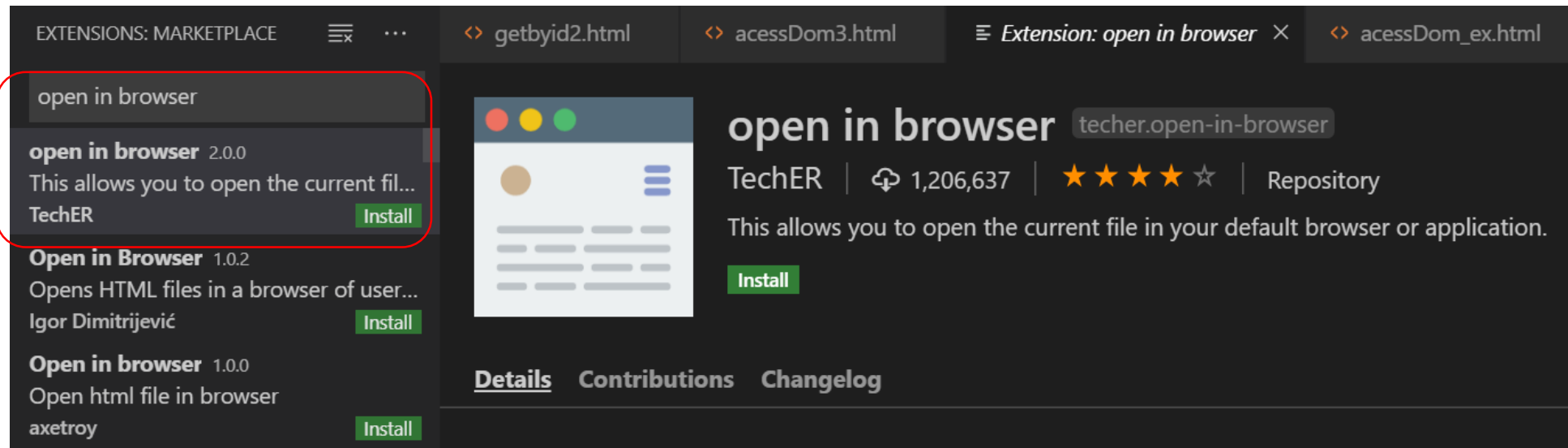
# 수업에 들어가며

- 복습문제
  - DOM을 사용해 이벤트처리기 등록
- 이번 시간 학습목표
  - <input>, <output> 태그 사용
  - HTML요소의 innerHTML 프로퍼티 사용
  - 타이머함수 활용
  - ES2015+
    - const, let
    - 템플릿 문자열
    - 객체리터럴
    - 화살표 함수



# VS CODE에 디터에서 브라우저 실행

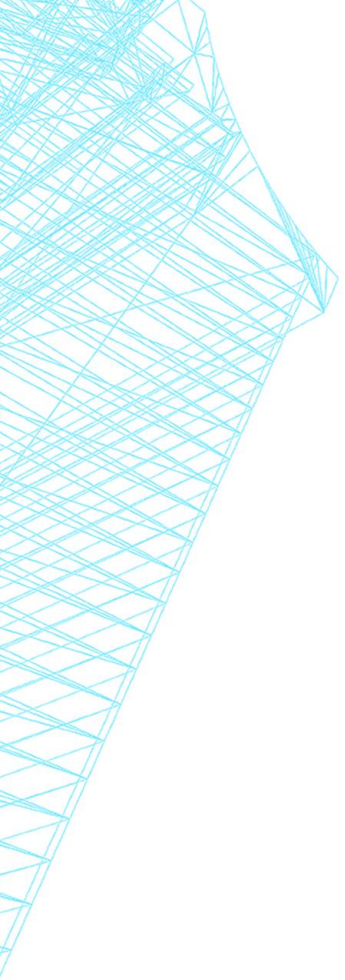
- View->extension
- open in browser 검색 후 설치
- 우클릭 Open In Default Browser



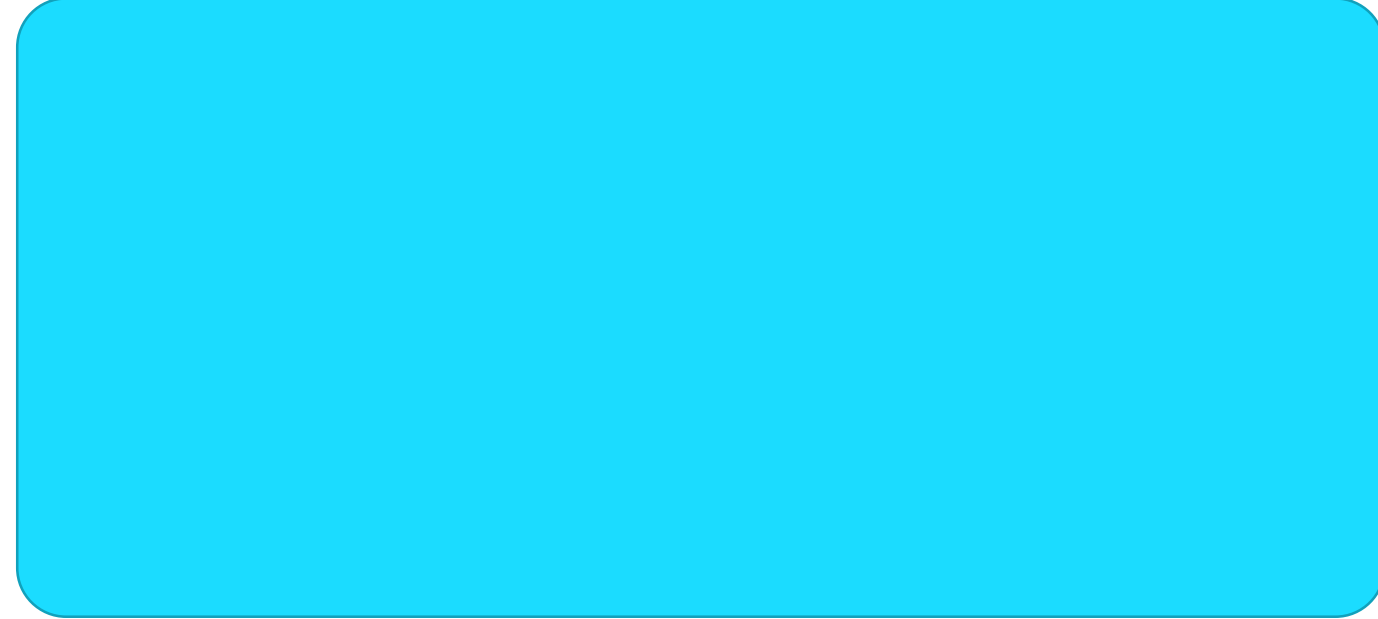
# 복습문제


- 다음 스크립트를 DOM객체를 이용한 이벤트 등록형태로 변경하시오

```
<html>
<head>
  <title>getElementById 예제</title>
  <script>
    function changeColor(newColor) {
      var elem = document.getElementById('para');
      elem.style.color = newColor;
    };
  </script>
</head>
<body>
  <p id="para">어떤 글</p>
  <button onclick="changeColor('blue');">blue</button>
  <button onclick="changeColor('red');">red</button>
</body>
</html>
```



```
<html>
<head>
  <title>getElementById 예제</title>
  <script>
```



```
    </script>
</head>
<body>
  <p id="para">어떤 글</p>
  
</body>
</html>
```

# 문제

- E캠퍼스 자료실에서 accessDom\_ex.html , dom.css, coffee-pink.jpg 다운로드
- 아래와 같이 "에디오피아 게덱" 문자에 mouseover시 red로 mouseout시 다시 black 으로 변경하도록 DOM객체를 이용해 구현하시오.

## 에디오피아 게덱



### Information

2차 세계대전 이후 설립된 **게덱농장**은 유기농 인증 농장으로 여성의 고용 창출과 지역사회 발전에 기여하며 3대째 이어져 내려오는 오랜 역사를 가진 농장입니다. 게덱 농장은 **SCAA 인증**을 받은 커피 품질관리 실험실을 갖추고 있어 철저한 관리를 통해 스페셜티커피를 생산합니다.

### Flavor Note

은은하고 다채로운 꽃향, 망고, 다크 체리, 달달함이 입안 가득.

## 에디오피아 게덱



### Information

2차 세계대전 이후 설립된 **게덱농장**은 유기농 인증 농장으로 여성의 고용 창출과 지역사회 발전에 기여하며 3대째 이어져 내려오는 오랜 역사를 가진 농장입니다. 게덱 농장은 **SCAA 인증**을 받은 커피 품질관리 실험실을 갖추고 있어 철저한 관리를 통해 스페셜티커피를 생산합니다.

### Flavor Note

은은하고 다채로운 꽃향, 망고, 다크 체리, 달달함이 입안 가득.

```
<!DOCTYPE html>
<html lang="ko">
<head> ...
</head>
<body>
  <div id="container">
    <h1 id="heading">에디오피아 게덱</h1>
    <div id="prod-img">
      
    </div>
    <div id="desc">
      <h2 class="bright">Information</h2>
      <p>2차 세계대전 이후 설립된 <span class="accent">게덱농장</span>은 유
      <h2>Flavor Note</h2>
      <p class="bright">은은하고 다채로운 꽃향, 망고, 다크 체리, 달달함이 입
    </div>
  </div>
</body>
</html>
```





```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
```

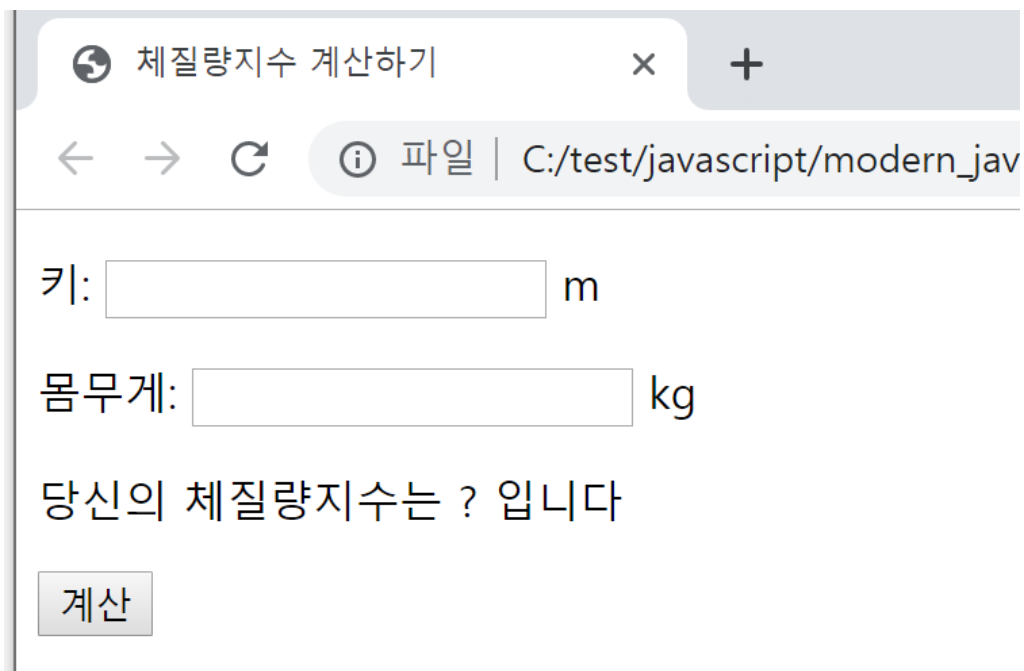
```
  <title>DOM에 접근하기</title>
```

```
  <link rel="stylesheet" href="dom.css">
```

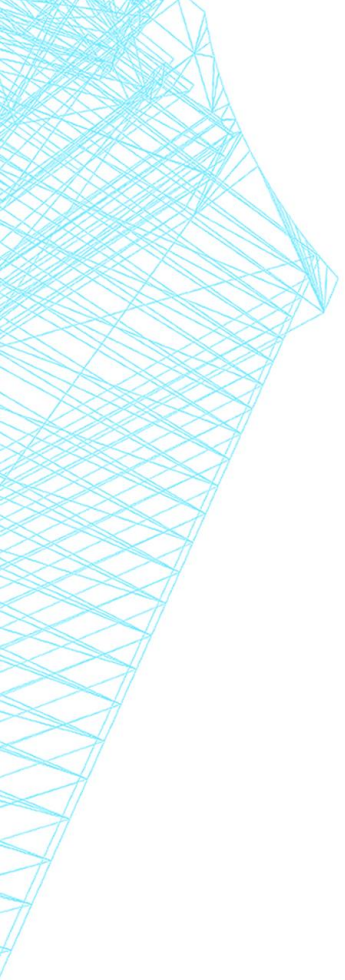
```
</head>
```

# HTML요소의 <INPUT>, <OUTPUT> 태그 사용

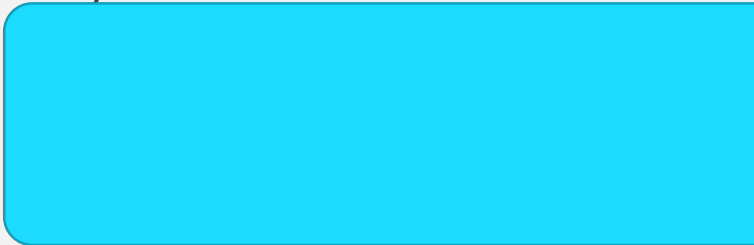
- 다음 화면과 같이 HTML 페이지를 구성해보자.



A screenshot of a web browser window. The browser has a single tab titled '체질량지수 계산하기' (BMI Calculator) with a close button 'x' and a plus sign '+'. The address bar shows navigation icons (back, forward, refresh) and a file path 'C:/test/javascript/modern\_jav'. The page content includes two input fields: '키: [ ] m' (Height) and '몸무게: [ ] kg' (Weight). Below these is the text '당신의 체질량지수는 ? 입니다' (Your BMI is ?) and a button labeled '계산' (Calculate).



```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>체질량지수 계산하기</title>
  <script>

  </script>
</head>
<body>
  
</body>
</html>
```

# document.getElementById( ).value 사용

- getElementById( ) 함수를 사용해 웹 페이지에 있는 엘리먼트의 객체를 가져옴
- 이 함수는 엘리먼트의 데이터를 직접 잡아내는 대신 자바스크립트 객체를 사용해서 HTML 필드 자체를 제공함
- 따라서 각 필드의 value 프로퍼티를 통해 데이터에 접근할 수 있음.
- 형식) document.getElementById(id명).value

```
var h = parseFloat(document.getElementById("height").value);
```

```
<body>
  <p>키: <input type="number" id="height"> m</p>
  <p>몸무게: <input type="number" id="weight"> kg</p>
  <p>당신의 체질량지수는 <output id="bmi">?</output> 입니다</p>
  <input type="button" id="button" value="계산">
</body>
```



# HTML요소의 innerHTML 프로퍼티 사용

- 요소객체의 **innerHTML** 프로퍼티
  - 그 HTML요소의 내용을 가리키며
  - 이로써 HTML요소의 내용을 읽거나 쓸수 있다.

```
var bmi = document.getElementById("bmi");  
bmi.innerHTML = (w/h/h).toFixed(1);
```

```
<body>  
  <p>키: <input type="number" id="height"> m</p>  
  <p>몸무게: <input type="number" id="weight"> kg</p>  
  <p>당신의 체질량지수는 <output id="bmi">?</output> 입니다</p>  
  <input type="button" id="button" value="계산">  
</body>
```

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>체질량지수 계산하기</title>
  <script>
    [Redacted JavaScript Code]
  </script>
</head>
<body>
  <p>키: <input type="number" id="height"> m</p>
  <p>몸무게: <input type="number" id="weight"> kg</p>
  <p>당신의 체질량지수는 <output id="bmi">?</output> 입니다</p>
  <input type="button" id="button" value="계산">
</body>
</html>
```

체질량지수 계산하기

← → ↻ ⓘ 파일 | C:/test/javascript/modern\_jav...

키:  m

몸무게:  kg

당신의 체질량지수는 29.4 입니다

# 문제

innerHTML 프로퍼티를 사용해 [English]버튼을 클릭시 "에디오피아 게덱 " 문자를 "Ethiopian Gedep"으로 변경하는 이벤트를 구현하시오

## 에디오피아 게덱



### Information

2차 세계대전 이후 설립된 **게덱농장**은 유기농 인증 농장으로 여성의 고용 창출과 지역사회 발전에 기여하며 3대째 이어져 내려오는 오랜 역사를 가진 농장입니다. 게덱 농장은 **SCAA 인증**을 받은 커피 품질관리 실험실을 갖추고 있어 철저한 관리를 통해 스페셜티커피를 생산합니다.

### Flavor Note

은은하고 다채로운 꽃향, 망고, 다크 체리, 달달함이 입안 가득.

## Ethiopian Gedep



### Information

2차 세계대전 이후 설립된 **게덱농장**은 유기농 인증 농장으로 여성의 고용 창출과 지역사회 발전에 기여하며 3대째 이어져 내려오는 오랜 역사를 가진 농장입니다. 게덱 농장은 **SCAA 인증**을 받은 커피 품질관리 실험실을 갖추고 있어 철저한 관리를 통해 스페셜티커피를 생산합니다.

### Flavor Note

은은하고 다채로운 꽃향, 망고, 다크 체리, 달달함이 입안 가득.

English

```
<script>
  window.onload = function() {
    document.getElementById("heading").onmouseover= function(){
      this.style.color="red";
    }
    document.getElementById("heading").onmouseout= function(){
      this.style.color="black";
    }
  }
};
```



```
</script>
```

```
<body>
  <div id="container">
    <h1 id="heading">에디오피아 게덱</h1>
    <button id="eng">English</button>
    <div id="prod-img">
      
    </div>
    <div id="desc">
      <h2 class="bright">Information</h2>
```





# 타이머함수 활용

- setInterval()함수 : 지정한 시간마다 반복실행하는 함수
- clearInterval()함수 : 타이머를 중지하는 함수

# 타이머 함수 예제

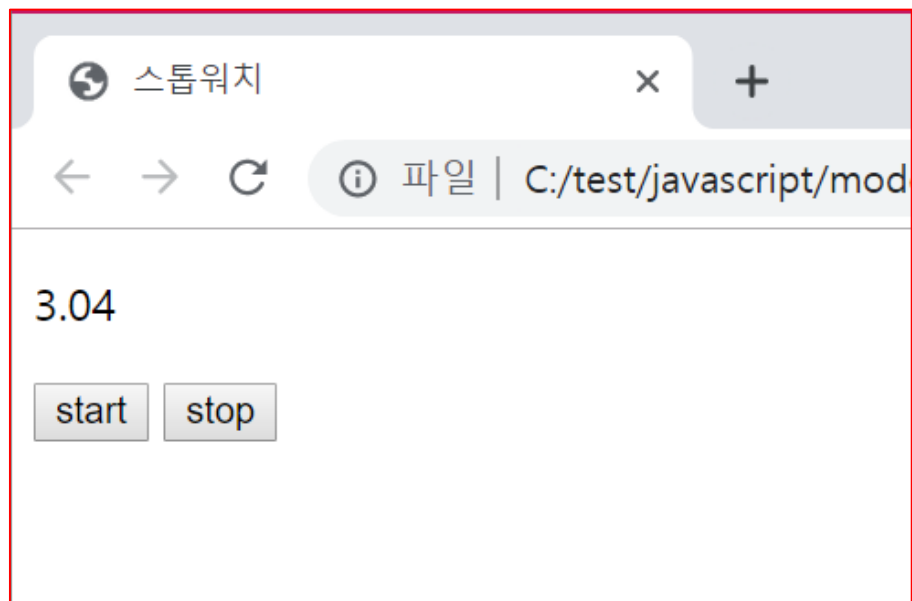
5초마다 경고 메시지를 콘솔에 출력하는 예제

```
var time = 0;
playAlert = setInterval(function() {
    time+=5;
    console.log(time,"초 경과");
    if(time==20) clearInterval(playAlert);
}, 5000);
```

```
C:\Program Files\nodejs\node.exe --ir
Debugger listening on ws://127.0.0.1:
For help, see: https://nodejs.org/en/
Debugger attached.
5 초 경과
10 초 경과
15 초 경과
20 초 경과
```

# 문제

- start버튼을 누르면 0.01 초마다 경과한 시간을 표시





```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <title>스톱워치</title>
  <script>
    [Redacted JavaScript Code]
  </script>
</head>
<body>
  <p id="display">0.00</p>
  <input id="start" type="button" value="start">
  <input id="stop" type="button" value="stop">
</body>
</html>
```



# ES2015+ P.56

ES2015 이전에는 var로 변수를 선언

- ES2015부터는 const와 let이 대체
- 가장 큰 차이점: 블록 스코프(var은 함수 스코프)
- 블록 {} 밖에서는 변수에 접근할 수 없음

```
if (true) {  
  var x = 3;  
}  
console.log(x); // 3
```

```
if (true) {  
  const y = 3;  
}  
console.log(y); // Uncaught ReferenceError: y is not defined
```

기존: 함수 스코프(function() {}이 스코프의 기준점)

- 다른 언어와는 달리 if나 for, while은 영향을 미치지 못함
- const와 let은 함수 및 **블록({})에도 별도의 스코프를 가짐**

# CONST, LET

## const는 상수

- 상수에 할당한 값은 다른 값으로 변경 불가
- 변경하고자 할 때는 **let** 으로 변수 선언
- 상수 선언 시부터 초기화가 필요함
- 초기화를 하지 않고 선언하면 에러

```
const a = 0;  
a = 1; // Uncaught TypeError: Assignment to constant variable.  
let b = 0;  
b = 1; // 1
```

```
const c; // Uncaught SyntaxError: Missing initializer in const declaration
```

# 템플릿 문자열

문자열을 합칠 때 + 기호때문에 지저분함

- ES2015부터는 ` (백틱) 사용 가능
- 백틱 문자열 안에 \${변수} 처럼 사용

```
var num1 = 1;  
var num2 = 2;  
var result = 3;  
var string1 = num1 + ' 더하기 ' + num2 + '는 \'' + result + '\'';  
console.log(string1); // 1 더하기 2는 '3'
```

```
const num3 =1;  
const num4=2;  
const result2=3;  
const string2=`${num3} 더하기 ${num4}는 '${result2}'`;  
console.log(string2);
```

1 더하기 2는 '3'



# 객체리터럴

- 자바스크립트에서 객체 생성법
- 객체리터럴 이용
  - `{ ... }`
  - `var p1 = {x:1, y:1};`



# 객체 리터럴

- ES5 시절의 객체 표현 방법
  - 속성 표현 방식에 주목

```
var sayNode = function() {  
  console.log('Node');  
};  
var es = 'ES';  
var oldObject = {  
  sayJS: function() {  
    console.log('JS');  
  },  
  sayNode: sayNode,  
};  
oldObject[es + 6] = 'Fantastic';  
  
oldObject.sayNode(); // Node  
oldObject.sayJS(); // JS  
console.log(oldObject.ES6); // Fantastic
```



- ES5+ 훨씬 간결한 문법으로 객체 리터럴 표현 가능
  - 객체의 메서드에 함수를 연결할때 :function을 붙이지 않아도 됨
  - 속성명과 변수명이 겹치는 경우 { sayNode: sayNode }와 같은 것을 { sayNode }로 축약 가능
  - 객체리터널내 안에서 [변수 + 값] 등으로 동적 속성명을 객체 속성명으로 사용 가능

```
const sayNode = function(){  
  console.log('Node');  
};  
const es = 'ES';  
const newobject = {  
  sayJS(){  
    console.log('JS');  
  },  
  sayNode,  
  [es + 6]: 'Fansatic',  
};  
newobject.sayNode();  
newobject.sayJS();  
console.log(newobject.ES6);
```

PROBLEMS OUTPUT DEBUG CONS

For help, see: <https://nodejs.org/en/docs/faq/troubleshooting/>  
Debugger attached.  
Node  
JS  
Fantastic

> Please start a debug session

```
var circle = {
  center : {x:1.0, y:2.0} ,
  radius : 2.5,
  area : function(){
    return Math.PI * this.radius * this.radius;
  },
  round : function(){
    return 2 * Math.PI * this.radius;
  }
};
```

```
console.log("원의 중심좌표는 (" + circle.center.x + ", " + circle.center.y + ")");
```

```
circle.translate = function(a,b){
  this.center.x = this.center.x + a;
  this.center.y = this.center.y + b;
};
```

```
circle.translate(1,2);
console.log("(1,2)이동한 원의 중심좌표는 (" + circle.center.x + ", " + circle.center.y + ")");
```

```
var circle = {
  center : {x:1.0, y:2.0} ,
  radius : 2.5,
  area(){
    return Math.PI * this.radius * this.radius;
  },
  round(){
    return 2 * Math.PI * this.radius;
  }
};
```

```
console.log("원의 중심좌표는 (" + circle.center.x + ", " + circle.center.y + ")");
```

```
circle.translate = function(a,b){
  this.center.x = this.center.x + a;
  this.center.y = this.center.y + b;
};
```

```
circle.translate(1,2);
console.log("(1,2)이동한 원의 중심좌표는 (" + circle.center.x + ", " + circle.center.y + ")");
```

# 화살표 함수

- add1, add2, add3, add4는 같은 기능을 하는 함수
  - function 선언 대신 => 기호로 함수 선언
  - add2: add1을 화살표 함수로 나타낼 수 있음
  - add3: 함수의 본문이 return만 있는 경우 return 생략
  - add4: return이 생략된 함수의 본문을 소괄호로 감싸줄 수 있음
  - not1과 not2도 같은 기능을 함

```
function add1(x, y) {  
  return x + y;  
}
```

```
const add2 = (x, y) => {  
  return x + y;  
};
```

```
const add3 = (x, y) => x + y;
```

```
const add4 = (x, y) => (x + y);
```

```
function not1(x) {  
  return !x;  
}
```

```
const not2 = x => !x;
```

# 학습참여평가 : E클래스 올리기

## 기존 객체 메서드를 외부 화살표함수형태로 변환해보자

```
var circle = {  
  center : {x:1.0, y:2.0} ,  
  radius : 2.5,  
};  
  
console.log("원의 중심좌표는 (" + circle.center.x + "," + circle.center.y + ")");  
  
:  
:  
console.log("원의 면적은 " + area(circle.radius).toFixed(2) + "입니다.");  
console.log("원의 둘레는 " + round(circle.radius).toFixed(2) + "입니다.");  
translate(1,2);  
console.log("(1,2)이동한 원의 중심좌표는 (" + circle.center.x + "," + circle.center.y + ")");
```