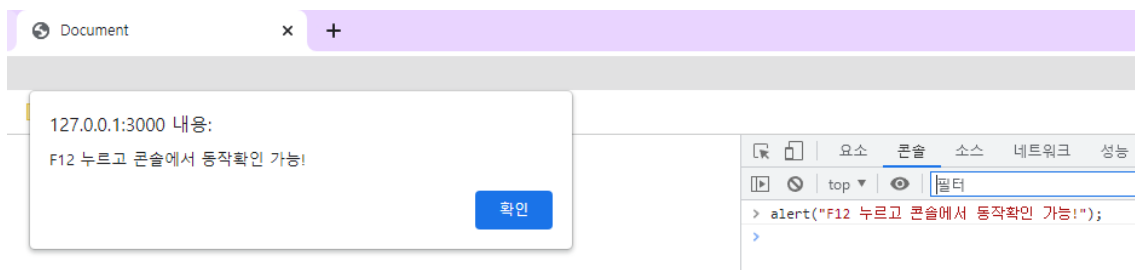


230509(화) CSS마무리 + JS 1일차

▼ Java Script 1일차

- **DOM**
 - Document Object Model
 - 웹페이지 내 모든 엘리먼트 정의&접근하는 방법 제공
- **Function 함수**
 - 자바에서의 method, 파이썬?에서의 procedure 등
- **네이티브 자바스크립트**
 - 타 라이브러리 혹은 프레임워크 사용X
 - 우리 수업에서는 기본적인 문법으로만 사용되는 자바스크립트 이용 예정
- **로그**
 - 프로그램의 흐름을 확인하기 위해서 남기는 기록
 - 개발전 가장 먼저 설계해야 할 목표
- **컴파일**
 - 사람이 작성한 소스코드 → 컴퓨터가 이해할 수 있는 기계어로의 번역과정
 - 모든 언어는 컴파일 과정을 거친다
- **인터프리트 언어**
 - 실시간 실행 언어도 있음
 - 자바스크립트 JavaScript, PHP, ASP, 파이썬 Python 등
 - 실행해보지 않아도 바로 에러를 확인할 수 있다는 장점.
 - 반대 예시는 C, C++, Java, Go 등...
- **기본 구조**
 - 인라인 추가 방식
 - embedded 방식
 - external 방식
- **꼭 VS에서 html을 작성하는 방식으로 JS 동작을 확인해야 할까?**
 - L L F12 개발자 모드



▼ 문법

- 자바와 달리 JS는 타입이 없는 언어이다. (정수, 문자, 문자열...)
- 타입이 없으면 **겉나 짜증난다...**

- var i = 10, var i = "hello"
- ??? var 는 쓰지 맙시다.

💡 그래서 나온 것이 let, const

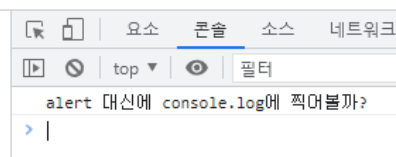
- let : 중복선언 가능
 - let i = 10;
 - let i = 20;
 - 연달아서.
- const : 중복선언 불가 (자바에서의 final 같은. 변경사항이 없을 때 쓸 것을 권장)
 - const i = 10;
 - const i = 20; << 에러발생
 - 단, 선언과 동시에 초기화 할 것. (const i; 안됨~~)

- console.log의 활용

```
function welcome() {
  // alert("환영해! from utils.js");
  console.log("alert 대신에 console.log에 찍어볼까?")
}
```

- 예시

인라인 방식 버튼이야
스크립트 방식으로 인사할까?
외부방식으로 인사할 수도 있겠지!



- 형식 선언
 - 별다른건 없고 할당되는 값이 숫자면 숫자, 텍스트면 텍스트가 되네.
- boolean
- 연산
- 텍스트 병합
- 타입확인typeof로 할당된 값의 타입 확인가능!
 - 자바에서의 instanceof랑 유사한 용도일까?

```
> typeof 안녕
< 'undefined'
> typeof "안녕"
< 'string'
> typeof 10
< 'number'
> typeof 10.0
< 'number'
> typeof null
< 'object'
```

- 함수

```
// 반환값return이 없고, 인자parameter도 없는 함수
function welcome() {
  // alert("환영해! from utils.js");
  console.log("alert 대신에 console.log에 찍어볼까?")
}

//
function sum(x, y) {
  let result = x + y;
  console.log(result);
}

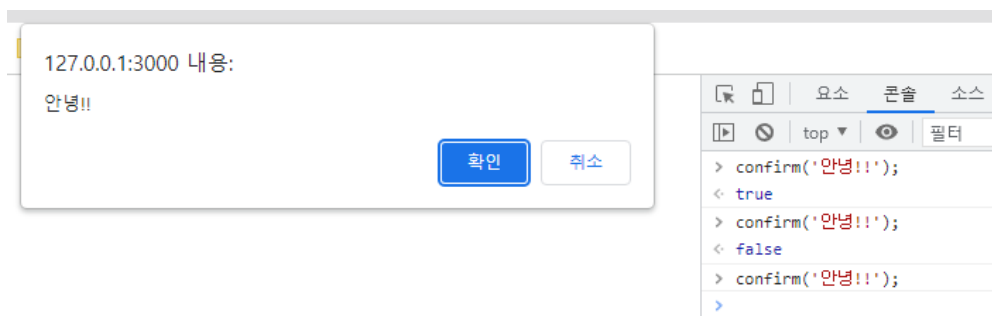
function total(x, y) {
  return x + y;
}
```

- 실행 결과

```
> sum(10, 20)
30
< undefined
> total(10, 20)
< 30
> let result = total(10, 20)
< undefined
> result
< 30
> total('Hi', ' 습');
< 'Hi 습'
> total(true, true)
< 2
> total(false, false)
< 0
> true + 1
< 2
> true + 'Hi'
< 'trueHi'
```

- confirm

- 확인 : true, 취소 : false 반환



- prompt

- 값 입력시 반환, 취소시 null
- 빈칸 입력시 “

```

> prompt('이름');
< '슬슬'

> prompt('이름');
< ''

> prompt('이름');
< null

>

```

- 비교 연산자(=)
 - == 값만 같은지 비교
 - === 타입과 값이 둘 다 같은지 비교

```

> true + 1
< 2

> true == 1
< true

> true === 1
< false

```

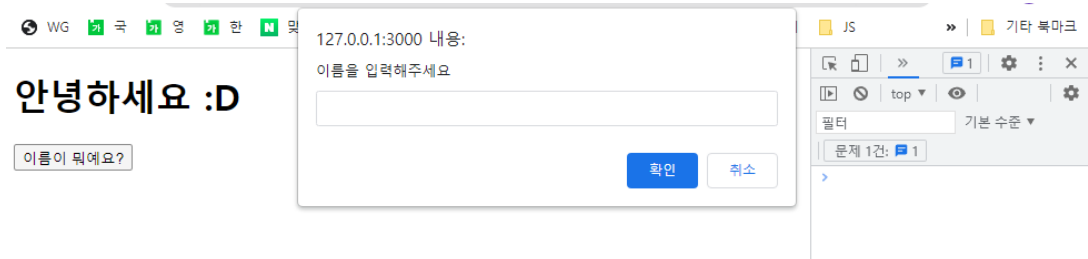
▼ 실습01 - 사용자명 + 환영alert

- 문제

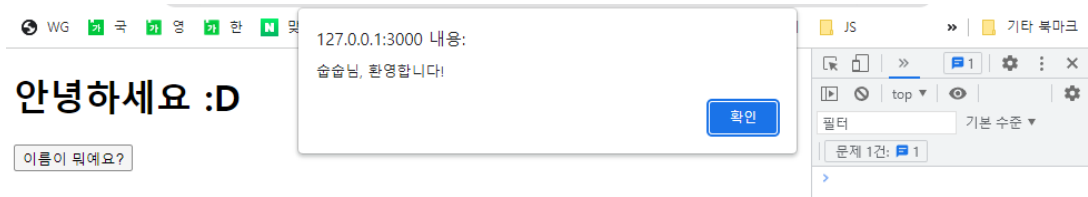
실습1

1. 이름을 입력 받고
2. 이름이 입력되지 않았을때 '입력해주세요' 라고 물어보고 다시 입력받고
3. 이름이 입력 되었을때 로그에 이름을 띄우고 경고창으로 '누구누구님 안녕하세요.' 를 띄워주세요.

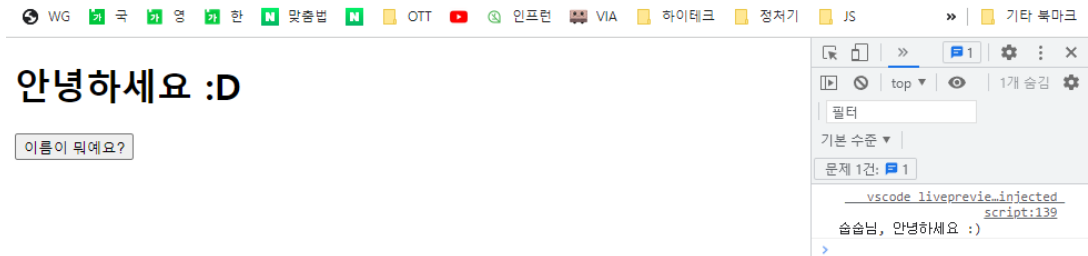
- 결과
 - 이름 입력받는 prompt 출력



- 입력한 이름 + 환영 alert 출력



- 사용자명 log 출력



- 코드

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script src="utils.js"></script>

  <h1>안녕하세요 :D</h1>
  <input type="button" value="이름이 뭐예요?" onclick="javascript:enterName()"></input>

</body>
</html>
```

```
function enterName() {
  const name = prompt('이름을 입력해주세요');
  if (name !== '' && name !== null) {
    alert(name + "님, 환영합니다!");
    console.log(name + "님, 안녕하세요 :)");
  } else {
    alert('이름을 다시 입력해주세요');
    enterName();
  }
}
```

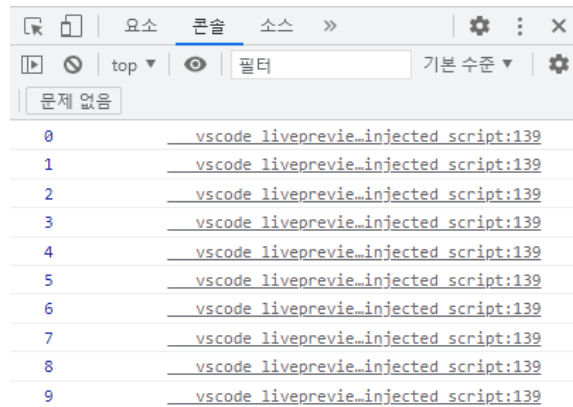
- 풀이

- 이름 제대로 입력됐는지 확인하는 방법
 1. name.length === 0 && name !== null
 2. name !== "" && name !== null

- 반복문

```
<script>
  for (let i = 0; i < 10; i++) {
    console.log(i);
  }
</script>
```

- 결과



- 추가 연산자 기능

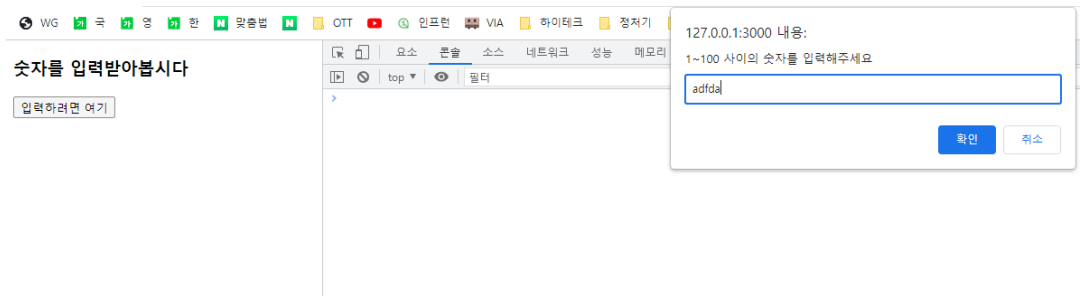
```
alert(`${name}님, 환영합니다!`);
```

- 데이터 타입 확인하는 기능이 있음!
 - falsy - 거짓같은 값

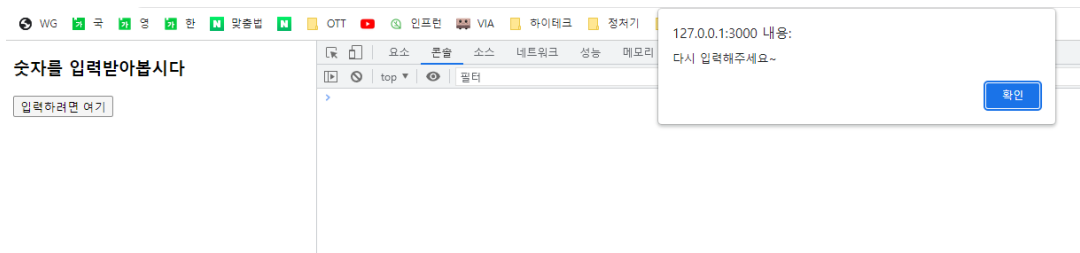
💡 그 중에 NaN - Not-A-Number, isNaN()에 사용해보면 알 수 있음

▼ 실습02 - 1~100 “사이”의 “숫자”만 입력받기 + for문

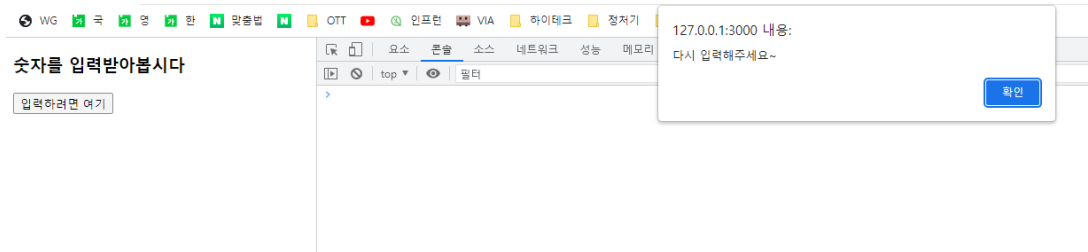
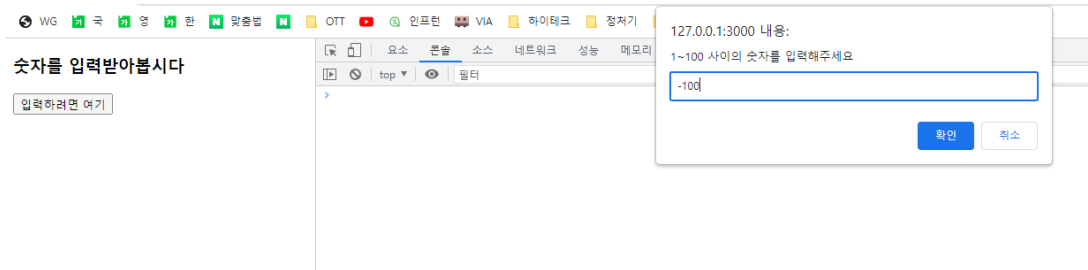
- 결과
 - prompt 입력창 출력



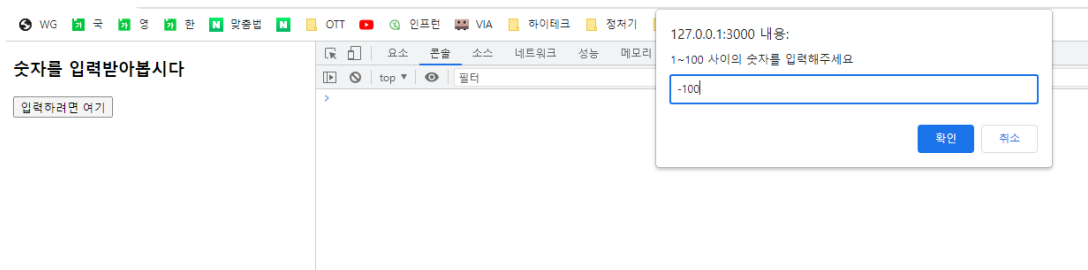
- 숫자가 아닌 값 입력시 재입력 alert



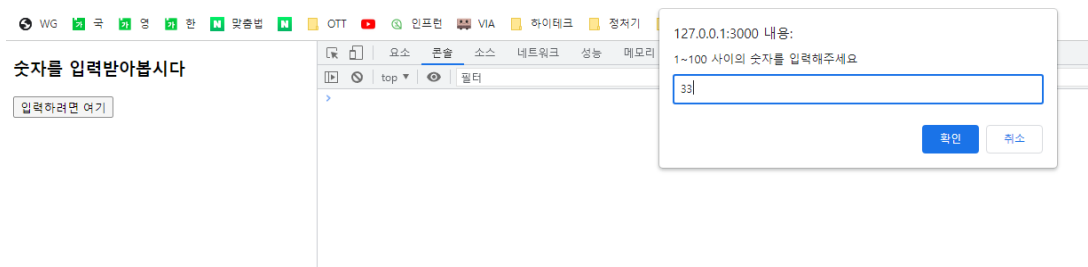
- 1~100 범위를 벗어난 숫자 입력시에도 재입력 alert



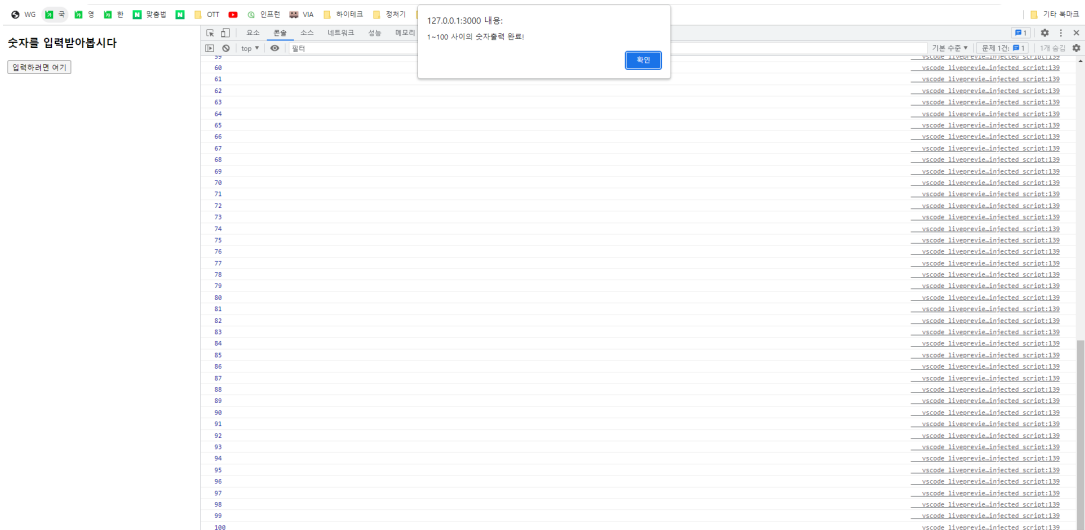
- 1~100 범위를 벗어난 숫자 입력시에도 재입력 alert



- 1~100 사이의 숫자 입력시



- 출력완료안내 alert + 1~100까지 콘솔 출력 완료!



• 코드

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>1~100 숫자 입력받기</title>
</head>
<body>
  <script src="utils.js"></script>

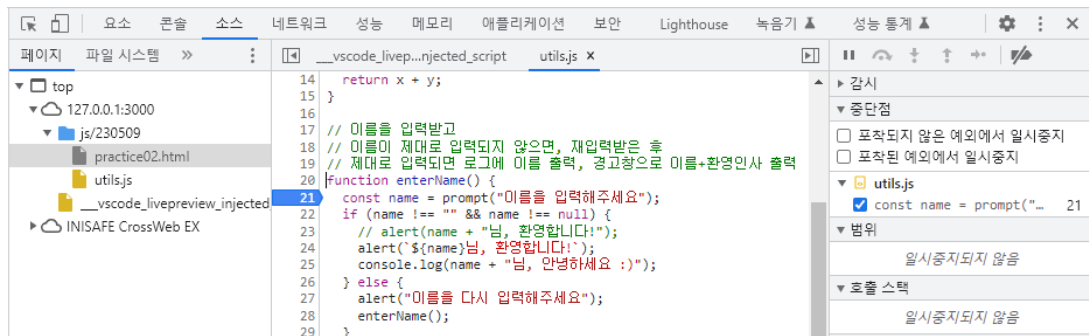
  <h3>1~100 사이의 숫자만 입력해주세요!</h3>
  <input type="button" value="입력하려면 클릭" onclick="javascript:enterNum()">
</body>
</html>
```

```
function enterNum() {
  //prompt를 통해 숫자를 입력받고,
  //입력받은 숫자를 num에 저장합니다
  const num = prompt("1~100 사이의 숫자를 입력해주세요");

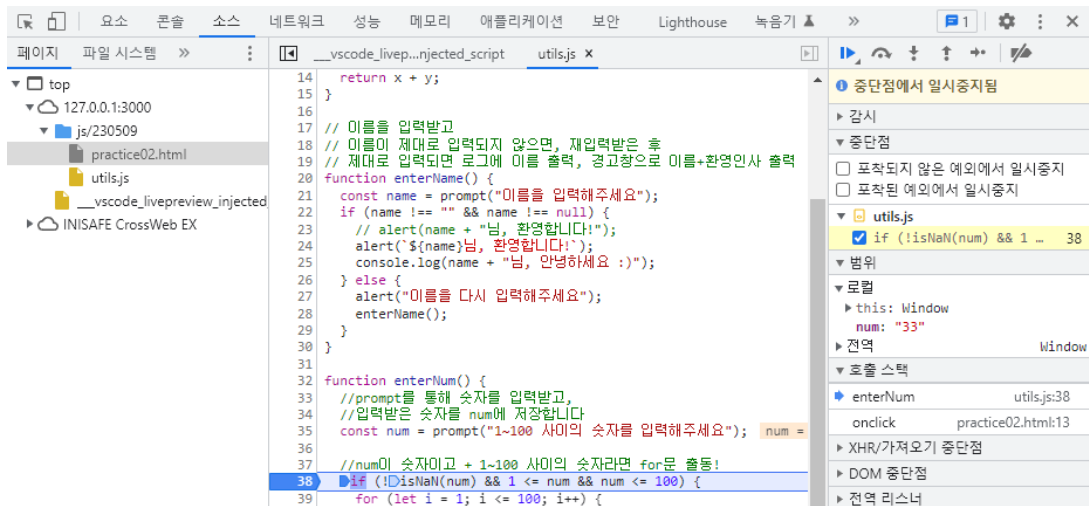
  //num이 숫자이고 + 1~100 사이의 숫자라면 for문 출동!
  if (!isNaN(num) && 1 <= num && num <= 100) {
    for (let i = 1; i <= 100; i++) {
      console.log(i);
    }
    alert("1~100 사이의 숫자출력 완료!");
    //숫자이긴 한데 범위만 벗어난거면
  } else if (!(1 <= num && num <= 100)) {
    alert("1부터 100 사이의 숫자여야 해요.");
    enterNum();
    //취소를 누르면
  } else if (num === null) {
    alert("입력을 취소했어요~");
    //그 밖의 비정상적인 입력들은 경고+함수 재호출
  } else {
    alert("잘못 입력했어요~");
    enterNum();
  }
}
```

• 콘솔에서 디버그 모드 실행하는 법!

- F12 - 소스 - 파일선택 - 중단점 선택



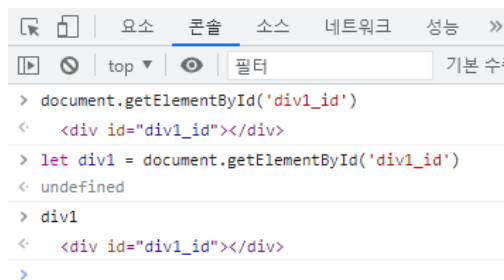
- 하나씩 실행해볼것



- 셀렉터

- html에서 작성한 class id 및 name 등을 가져오는 함수.

- `getElementsByld`



- `getElementsByClassName`

```

> document.getElementsByClassName('div1_class')
< HTMLCollection(3) [div.div1_class, div.div1_class, div.div1_class,
  div1_name: div.div1_class] {
  ▶ 0: div.div1_class
  ▶ 1: div.div1_class
  ▶ 2: div.div1_class
  ▶ div1_name: div.div1_class
  length: 3
  ▶ [[Prototype]]: HTMLCollection
}

> let nums = document.getElementsByClassName('div1_class')
< undefined

> nums[0]
< <div class="div1_class" name="div1_name">1</div>
>

```

• CSS를 조작해보자

- id, class, name 등을 변수에 할당한 후
- 변수.메소드명 하면 바로바로 적용됨.



```

> let div = document.getElementById('div1_id');
< undefined

> div.textContent = '나 여기있어'
< '나 여기있어'

> div.className = 'big-text'
< 'big-text'

> div.className = 'big-text blue-color'
< 'big-text blue-color'
>

```

- textContent로는 html 태그 안먹힘
- innerHTML 쓰면 적용가능



```

> let div = document.getElementById('div1_id');
< undefined

> div.textContent = '<h1>HELLO</h1>'
< '<h1>HELLO</h1>'

> div.innerHTML = '<h1>HELLO</h1>'
< '<h1>HELLO</h1>'
> |

```

• 이벤트Event

- mouseover : 마우스 커서 근접시, mouseleave : 커서 이탈시 속성 추가 가능
- this : 말 그대로 자기 자신을 가리키는 인자

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <style>
    #box {
      width: 100px;
    }
  </style>

```

```

        height: 100px;
        background-color: red;
    }
</style>

<script>
    function changeColor(color) {
        let box = document.getElementById('box');
        box.style.backgroundColor = color;
    }
    // function changeToBlue() {
    //     let box = document.getElementById('box');
    //     box.style.backgroundColor = 'blue';
    // }
    // function changeToYellow() {
    //     let box = document.getElementById('box');
    //     box.style.backgroundColor = 'yellow';
    // }

    // function showAlert2() {
    //     alert('Hello World');
    // }
    // function showAlert2() {
    //     alert('안녕하세요');
    // }

    function showAlert(element) {
        console.log(element.textContent);
        alert(element.textContent);
    }
</script>
</head>
<body>
    <div id="box" onmouseover="changeColor('blue')"
    onmouseleave="changeColor('yellow')"></div>

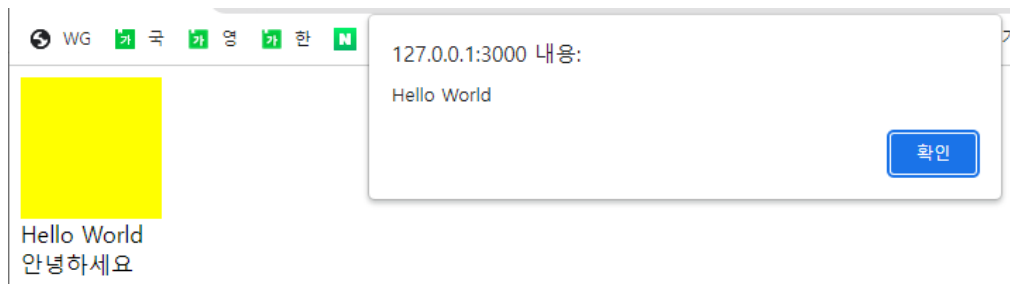
    <div class="big" onclick="showAlert(this)">Hello World</div>
    <div onclick="showAlert(this)">안녕하세요</div>

</body>
</html>

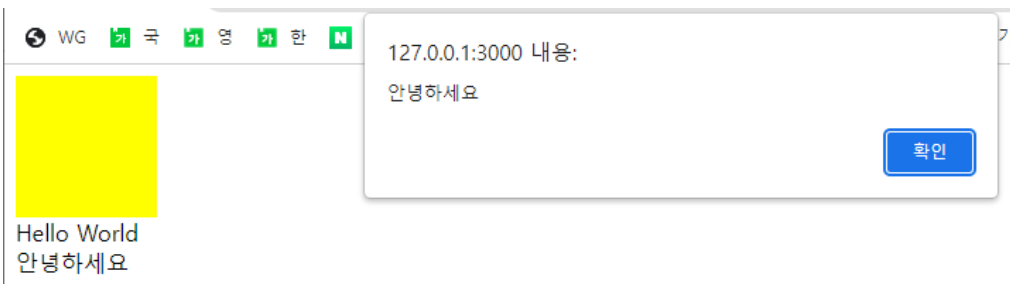
```

- 동일한 showAlert(**this**)를 사용하지만 다른 결과 출력

- Hello World 클릭시



- 안녕하세요 클릭시



- document.write

▼ 실습03 - 클릭으로 텍스트 색 변경하기

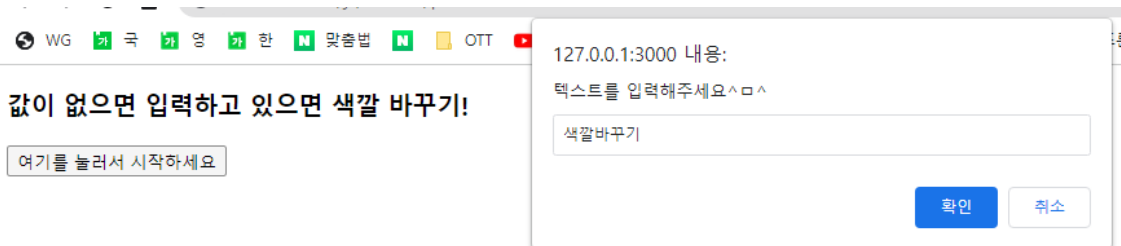
- 문제

실습3

1. 텍스트 값을 입력 받고
2. 값이 없으면 "텍스트값을 입력해 주세요" 경고창을 띄우고
3. 값이 있으면 가운데 크게 텍스트를 표시해 주고
4. 바로 아래에 빨강/파랑/초록 글자가 쓰여진 버튼이 생긴다.
5. 빨강 파랑 초록 이 쓰여진 버튼을 누르면
6. "무슨무슨 색으로 변경됩니다." 라고 경고창이 뜨면서
7. 입력된 텍스트의 색이 변경되게 해봅시다.

- 결과

- 텍스트 입력 프롬프트 출력



- 버튼 클릭시마다 색깔 변경

값이 없으면 입력하고 있으면 색깔 바꾸기!

여기를 눌러서 시작하세요

색깔바꾸기

파랑으로

빨강으로

초록으로

값이 없으면 입력하고 있으면 색깔 바꾸기!

여기를 눌러서 시작하세요

색깔바꾸기

파랑으로

빨강으로

초록으로

값이 없으면 입력하고 있으면 색깔 바꾸기!

여기를 눌러서 시작하세요

색깔바꾸기

파랑으로

빨강으로

초록으로

- 코드

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Change color what you want!</title>

  <style>
    .outer {
      /* background-color: beige; */
      /* display: inline-block; */
      position: relative;
      left: 400px;
    }
  </style>

  <script src="practice03.js"></script>
</head>

<body>
  <h3>값이 없으면 입력하고 있으면 색깔 바꾸기!</h3>
  <input type="button" value="여기를 눌러서 시작하세요" onclick="enterText()"></input>

  <div class="outer">
    <div id="textHere"></div>
    <div id="buttonHere"></div>
  </div>

</body>
```

</html>

```
function enterText() {
    let text = prompt('텍스트를 입력해주세요^o^');

    //텍스트가 0이면
    if (text.length === 0) {
        alert("다시 입력해주세요!");
        enterText();
    } //정상 입력이면 텍스트 출력, 버튼 나타내기
    } else {
        let div = document.getElementById('textHere');
        // div.innerHTML = '<h1>' + text + '</h1>';
        div.innerHTML = `<h1>${text}</h1>`;

        showButton();
    }
}

function showButton() {
    let btn = document.getElementById('buttonHere');
    btn.innerHTML = `<input type="button" class="btn" value="파랑으로" onclick="changeColor('blue')"></input>
    <input type="button" class="btn" value="빨강으로" onclick="changeColor('red')"></input>
    <input type="button" class="btn" value="초록으로" onclick="changeColor('green')"></input>`;
}

function changeColor(colorName) {
    let textColor = document.getElementById('textHere');
    textColor.style.color = colorName;
}
```

▼ 실습04 - 클릭한 메뉴 하이라이트

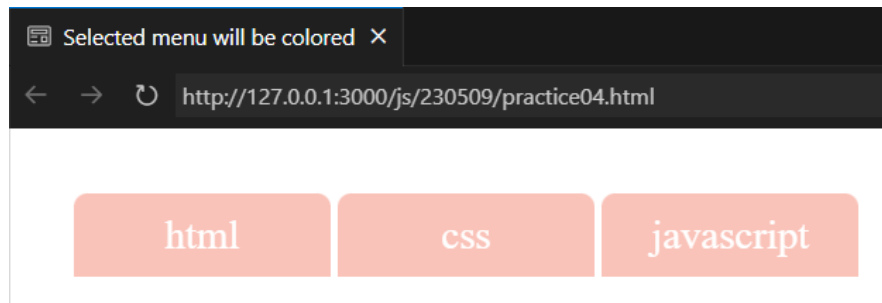
- 문제

실습4

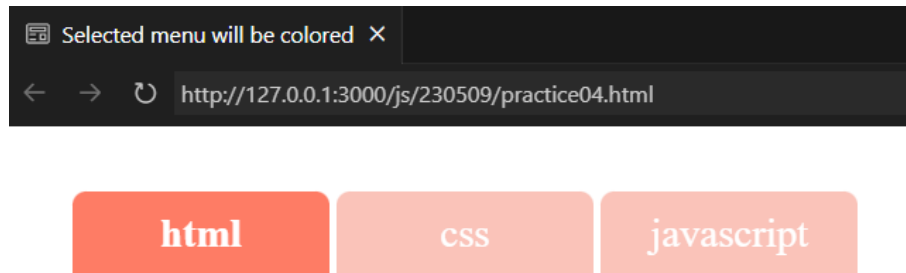
html	css	javascript
html	css	javascript
html	css	javascript

1. 배경이 회색이고 텍스트가 하얀 메뉴가 3개 있습니다.
2. 마우스로 메뉴를 클릭하면 한 메뉴만 배경이 하얗게 외곽선은 검게 텍스트는 검게 변합니다.
3. 다른메뉴를 클릭하면 아까 클릭한 메뉴는 원래대로 돌아오고 클릭한 메뉴가 변화합니다.

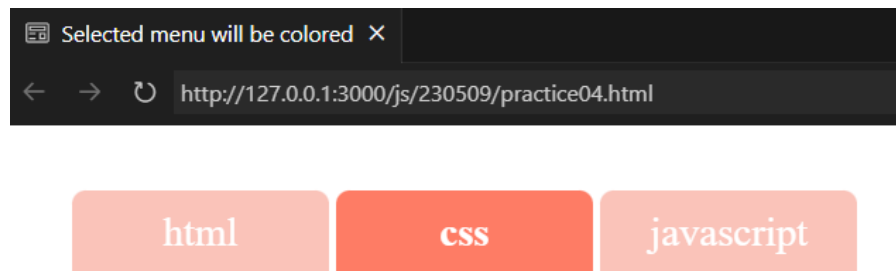
- 결과
 - 1단계
 -



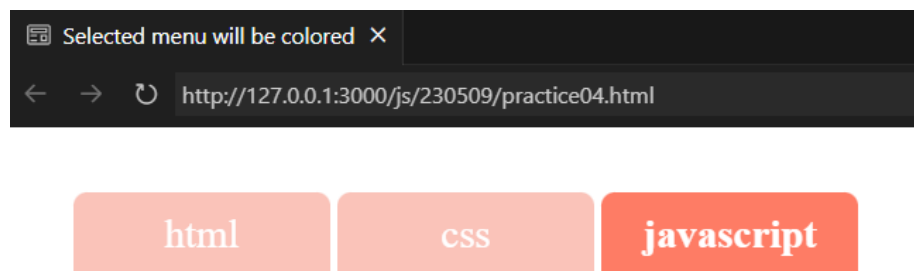
◦ 2단계



◦ 3단계



◦ 4단계



• 코드

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Selected menu will be colored</title>
```

```

<style>

    .outer {
        /* background-color: aquamarine; */
        padding: 30px;
    }

    .menu {
        background-color: rgb(250, 195, 185);
        border: 1px solid transparent;
        border-top-left-radius: 8px;
        border-top-right-radius: 8px;
        display: inline-block;
        top: 0;
        left: 0;
        width: 130px;
        text-align: center;
        font-size: x-large;
        margin: 0px;
        padding: 10px;
        text-decoration: none;
        color: white;
    }

</style>

<script src="practice04.js"></script>

</head>
<body>
    <div class="outer">
        <a class="menu" id="html" href="#html" onclick="markHtml()">html</a>
        <a class="menu" id="css" href="#css" onclick="markCss()">css</a>
        <a class="menu" id="js" href="#javascript" onclick="markJs()">javascript</a>
    </div>
</body>
</html>

```

```

function markHtml() {
    document.getElementById("html").style = "color : white; background-color : rgb(254, 124, 101); font-weight : 600;"

    document.getElementById("css").style = "color : white; background-color : rgb(250, 195, 185);"
    document.getElementById("js").style = "color : white; background-color : rgb(250, 195, 185);"
}
function markCss() {
    document.getElementById("css").style = "color : white; background-color :rgb(254, 124, 101); font-weight : 600;"

    document.getElementById("html").style = "color : white; background-color : rgb(250, 195, 185);"
    document.getElementById("js").style = "color : white; background-color : rgb(250, 195, 185);"
}
function markJs() {
    document.getElementById("js").style = "color : white; background-color : rgb(254, 124, 101); font-weight : 600;"

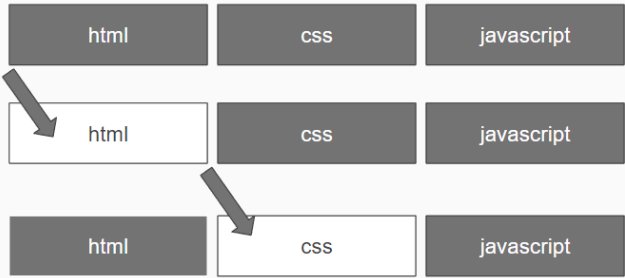
    document.getElementById("html").style = "color : white; background-color :rgb(250, 195, 185);"
    document.getElementById("css").style = "color : white; background-color : rgb(250, 195, 185);"
}

```

▼ 실습05 - 마우스오버 메뉴

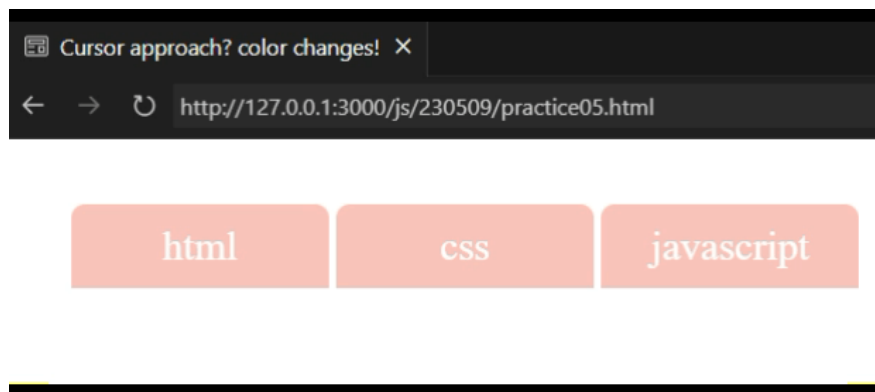
- 문제

실습5



1. 배경이 회색이고 텍스트가 하얀 메뉴가 3개 있습니다.
2. 마우스로 메뉴에 마우스를 대면 한 메뉴만 배경이 하얗게 외곽선은 검게 텍스트는 검게 변합니다.
3. 다른메뉴 대면 아까 댄 메뉴는 원래대로 돌아오고 클릭한 메뉴가 변화합니다.

- 결과(gif)



- 코드

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Cursor approach? color changes!</title>

  <style>

    .outer {
      /* background-color: aquamarine; */
      padding: 30px;
    }

    .menu {
      background-color: rgb(250, 195, 185);
      border: 1px solid transparent;
      border-top-left-radius: 8px;
      border-top-right-radius: 8px;

      display: inline-block;
      top: 0;
      left: 0;
      width: 130px;
      text-align: center;
      font-size: x-large;
    }
  </style>
</head>
<body>
  <div class="outer">
    <div class="menu">html</div>
    <div class="menu">css</div>
    <div class="menu">javascript</div>
  </div>
</body>
</html>
```

```

        margin: 0px;
        padding: 10px;
        text-decoration: none;
        color: white;
    }

    .outer a:hover {
        background-color: rgb(254, 124, 101);
        color: white;
        font-weight: 600;
    }
}

</style>

</head>
<body>
    <div class="outer">
        <a class="menu" id="html" href="#html" onclick="markHtml()">html</a>
        <a class="menu" id="css" href="#css" onclick="markCss()">css</a>
        <a class="menu" id="js" href="#javascript" onclick="markJs()">javascript</a>
    </div>
</body>
</html>

```