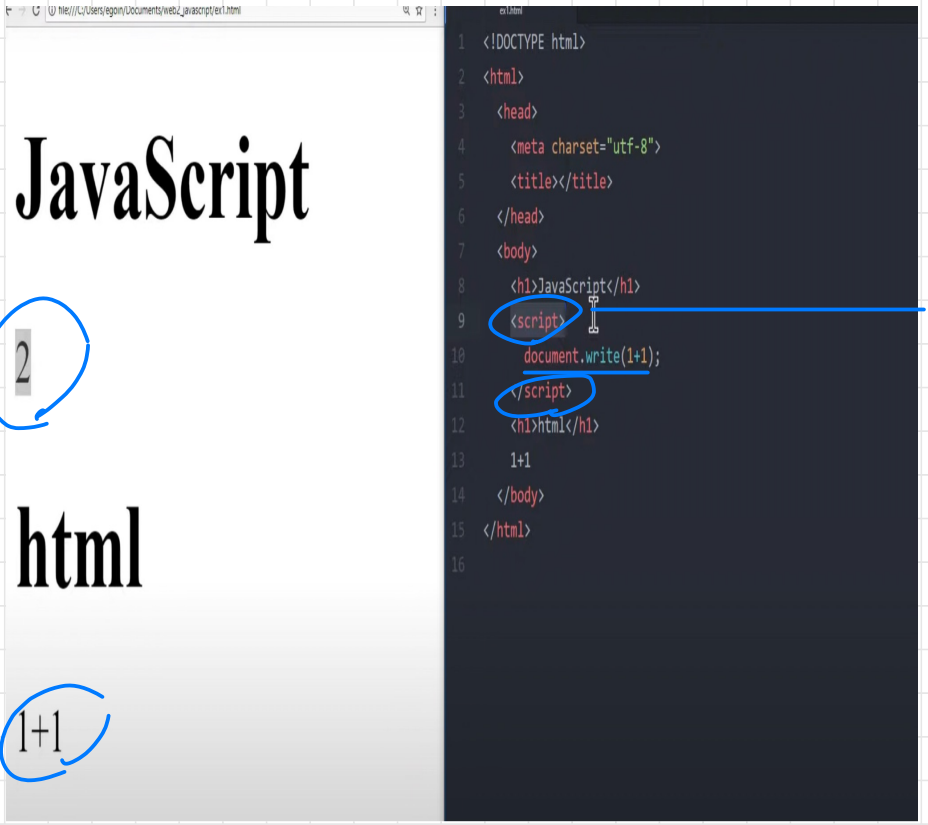
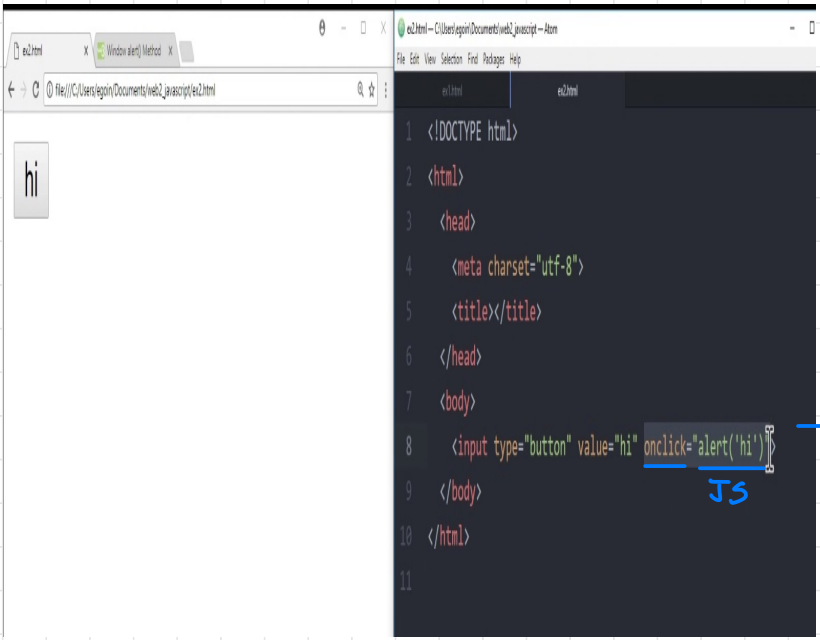
JS ? - 사용자와 상호작용 할 수 있다.



script 태그
Web browser에게
Js코드임을 알려줌.



onclick 속성을 Web browser가 기억
event (클릭)가 일어났을때 JS의 문법에 따라 해석하여 동작.

※ event? Web browser 위에서 일어나는 일들 (ex) 클릭)

기념할만한 event 몇가지 있음 약 10개 ~ 20개

onclick - 클릭
onchange - 입력값 변화
onkeydown - 키다운

콘솔을 이용해 간단한 JS 실행

콘솔에서 실행하는 JS는 Webpage 안에 삽입되어 있는 JS 인것처럼 동작 한다.
(web page를 대상으로 실행)

Data Type (자료형) - 문자열, 숫자 → 내가 아는 그거..

"1" 문자열 + - * / 연산자

1 숫자

변수와 대입 연산자

x=1; (x: 변수(variable), =: 대입연산자)

y=1;

x+y; ⇒ 2

1=2; ⇒ error (1은 항상 1이므로)

↳ 상수 (항상 상, 바뀌지않음) (constant)

변수를 왜 쓰는가? ⇒ 수많은이유 확실 강하다.

변수를 선언할때 Var을 가급적 앞에 붙일 것 variable

웹브라우저 제어

배경색 바꾸기 (css 이용)

```
</head>
<body style="background-color:black;color:white;">
  <h1><a href="index.html">WEB</a></h1>
```

⇒ dark mode를 선택했을 때 body TAG의 style 속성을 바꿀수 있다면..

⇒ html은 한번화면에 표시되면 자기를 바꾸는 능력이 없는 정적인 언어

⇒ JS를 이용하면 event 발생에 따라 body TAG를 바꿀 수 있다.

이벤트

```
<div id="mode">
  <input type="button" value="Night" onclick="
 document.querySelector('body').style.backgroundColor = 'black';
 document.querySelector('body').style.color = 'white';
" />
  <input type="button" value="Day" onclick="
 document.querySelector('body').style.backgroundColor = 'white';
 document.querySelector('body').style.color = 'black';
" />
```

] JS

Javascript → 프로그래밍언어 html → 그냥 컴퓨터 언어

↳ 순서에 따라서 여러기능을 실행가능
(사용자와 상호작용)

조건, 반복, 단순화 등등 발전 가능..

조건문

조건에 따라서 다른 순서의 기능들이 실행된다.

night 상태에서 클릭시 day로 , day 클릭시 night로 변경하고싶다.(토글)

비교연산자 Boolean 조건문.. 사용

```
<input id="darklight" type="button" value="Dark" onclick="
if(document.querySelector('#darklight').value === 'Dark'){
document.querySelector('body').style.backgroundColor = 'black';
document.querySelector('body').style.color = 'white';
document.querySelector('#darklight').value = 'Light'
} else{
document.querySelector('body').style.backgroundColor = 'white';
document.querySelector('body').style.color = 'black';
document.querySelector('#darklight').value = 'Dark'
}
```

darklight 의 value값이 Dark 일때 (현재 Light)

] 배경, 본문색 바꾸고
- value 값을 Light로

↳ 그외 value 일때 (value 가 Light 일때)

] - 색바꾸기
↳ value 를 Dark로

※ Console 에서 value값을 찾을 수 있음

리팩토링

비효율적인 코드를 효율적으로 만들어 가독성을 높이고 유지보수 편하게함. 중복 제거

소프트웨어가 규정에 따라 등등히 해줘야 good program

ex) 같은 버튼을 여러개 만들 때 id 값 매번 바꿔야함.

```
<input id="darklight" type="button" value="Dark" onclick="
if(document.querySelector('#darklight').value === 'Dark'){
document.querySelector('body').style.backgroundColor = 'black';
document.querySelector('body').style.color = 'white';
document.querySelector('#darklight').value = 'Light'
} else{
document.querySelector('body').style.backgroundColor = 'white';
document.querySelector('body').style.color = 'black';
document.querySelector('#darklight').value = 'Dark'
}
```

이 구간만에서 코드가 속해 있는 tag를 가리키도록 약속된 keyword = this

⇓ refactoring

```
<input type="button" value="DarkMode" onclick="
var target = document.querySelector('body')
if(this.value === 'DarkMode'){
target.style.backgroundColor = 'black';
target.style.color = 'white';
this.value = 'LightMode'
} else{
target.style.backgroundColor = 'white';
target.style.color = 'black';
this.value = 'DarkMode'
}
```

변수를 선언해 중복 간소화

배열과 반복문

Dark 모드, Light 모드에 따라 글씨 색 변경하고 싶음.

```
<input type="button" value="DarkMode" onclick="
var target = document.querySelector('body')
if(this.value === 'DarkMode'){
target.style.backgroundColor = 'black';
target.style.color = 'white';
this.value = 'LightMode'

var alist = document.querySelectorAll('a');
var i = 0;
while(i< alist.length){
  console.log(alist[i]);
  alist[i].style.color = 'powderblue';
  i = i+1;
}
} else{
target.style.backgroundColor = 'white';
target.style.color = 'black';
this.value = 'DarkMode'

var alist = document.querySelectorAll('a');
var i = 0;
while(i< alist.length){
  console.log(alist[i]);
  alist[i].style.color = 'DarkSlateblue';
  i = i+1;
}
}
```

→ 콘솔확인용: 필요X.

→ 모든 A TAG list를 alist 변수에 저장

→ i가 alist의 길이보다 작으면 반복 (list 만큼반복)

→ i번째 element의 color 변경

→ i+1

Light mode로 바꿀때

함수 함수선언 매개변수

```
<script>
  function LightDarkHandler(self) {
    var target = document.querySelector('body')
    if (self.value === 'DarkMode') {
      target.style.backgroundColor = 'black';
      target.style.color = 'white';
      self.value = 'LightMode'

      var alist = document.querySelectorAll('a');
      var i = 0;
      while (i < alist.length) {
        alist[i].style.color = 'powderblue';
        i = i + 1;
      }
    } else {
      target.style.backgroundColor = 'white';
      target.style.color = 'black';
      self.value = 'DarkMode'

      var alist = document.querySelectorAll('a');
      var i = 0;
      while (i < alist.length) {
        alist[i].style.color = 'DarkSlateblue';
        i = i + 1;
      }
    }
  }
</script>
```

여기에 this로 써버리면 (독립된 함수) 전역 객체를 가리키게 된다.
input 값을 받기위해 매개변수 사용해 this로 변경

```
<input type="button" value="DarkMode" onclick="
  LightDarkHandler(this)
```

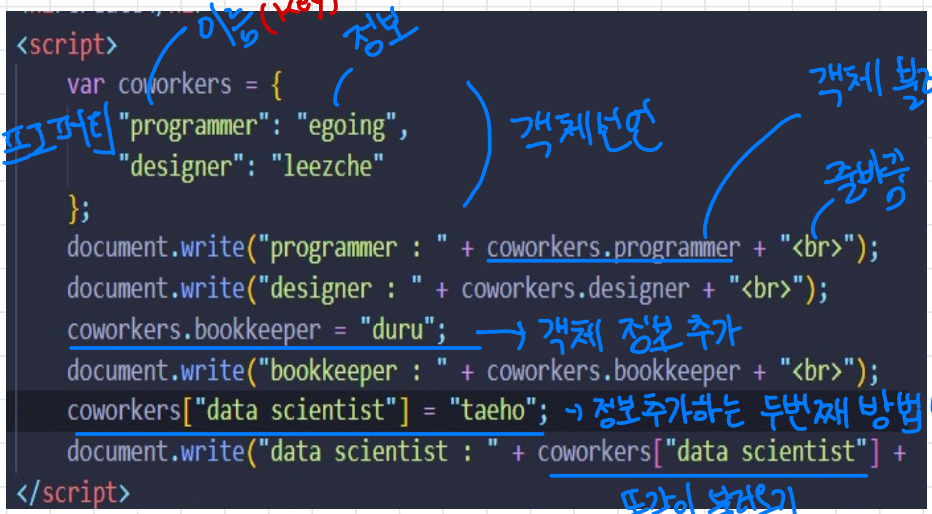
매개변수 this로 (self → this)

객체 정리정돈의 수단 (이름이 있는)

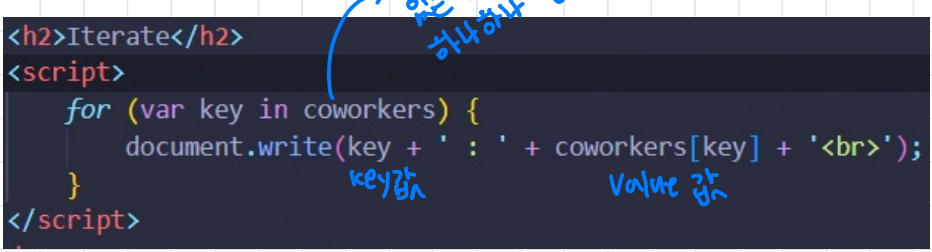
서로 연관된 함수와 연관된 변수를 같은 이름으로 Grouping 해서 정리

객체에 속한 함수 → method

객체 쓰기와 읽기



반복문이용



프로퍼티와 메서드

객체 - 여러 data를 담을 수 있음
객체에 소속된 함수 - method
변수 - property

객체에서 함수 선언

메서드

```
<script>
    coworkers.showAll = function () {
        for (var key in this) {
            document.write(key + ' : ' + this[key] + '<br>');
        }
    }
    coworkers.showAll()
</script>
```

함수사용

객체 활용

Links 객체

```
var Links = {
  SetColor: function (color) {
    var alist = document.querySelectorAll('a');
    var i = 0;
    while (i < alist.length) {
      alist[i].style.color = color;
      i = i + 1;
    }
  }
}
```

a태그 글씨색 메서드

Body 객체

```
var Body = {
  SetColor: function (color) {
    document.querySelector('body').style.color = color;
  },
  SetBackgroundColor: function (color) {
    document.querySelector('body').style.backgroundColor = color;
  }
}
```

배경색, 본문색 메서드

파일로 쪼개서 정리정돈 하기

연관된 code를 파일로 Grouping

Script TAG 안의 내용들을 *.js 파일안에 넣고

Script TAG의 SRC 속성을 이용해 파일 load

라이브러리와 프레임워크

↓ 땡겨와서 쓰기 ↓ 들어가서 쓰기

유명한 라이브러리 : jQuery ⇒ 가장 안정적, 오래됨
직접 Coding 보다 생산성↑

CDN 사용. 편하다.

마치며

Project 할때 모든 개념을 총동원 하려하지 말고
필수 불가결한 최소한의 도구로 문제를 해결하자 순서대로 실행되는 것에 집중.

⇒ 그것만으로는 도저히 해결 불가할 때가 온다. 그때 추가로 객체, 함수, 조건을 신중하게 도입, 개념에 익숙해지기

⇒ 또 한계, 필요할때 공부 document, DOM, window, ajax, cookie, offline web application, web RTC, Speech, webGL, webVR