

1. 계기

HTML은 정적으로 한번 화면에 출력되면 그 상태를 유지한다.

우리는 동적으로 사용자와의 상호작용을 원한다. 이것이 JavaScript가 등장한 계기가 된다.

HTML로 웹 페이지를 만들고 JavaScript로 상호작용을 하는 것이다.

2. 목적

HTML 문서의 구조는 각각의 `<head>` 영역과 `<body>` 영역 등으로 나누어져 있다.

```
<html>
<head>
  <title>WEB1 - Welcome</title>
  <meta charset="utf-8">
</head>
<body>
  <h1><a href="index.html">WEB</a></h1>
  <ol>
    <li><a href="1.html">HTML</a></li>
    <li><a href="2.html">CSS</a></li>
    <li><a href="3.html">JavaScript</a></li>
  </ol>
  <h2>WEB</h2>
  <p>The World Wide Web (abbreviated WWW or the Web) is an information space where documents
and other web resources are identified by Uniform Resource Locators (URLs), interlinked by hypertext
links, and can be accessed via the Internet.[1] English scientist Tim Berners-Lee invented the World
Wide Web in 1989. He wrote the first web browser computer program in 1990 while employed at
CERN in Switzerland.[2][3] The Web browser was released outside of CERN in 1991, first to other
research institutions starting in January 1991 and to the general public on the Internet in August 1991.
  </p>
</body>
</html>
```

사용자가 상호작용을 하면 ex) button 클릭, alert 발생, Link 클릭등

전체 혹은 일부의 영역에 ex) `<head>` , `<body>` 영역 등

어떠한 이벤트가 발생하는 것이다. ex) 화면이 검게 변하거나, 데이터 입출력등

HTML의 정적인 내용이 JavaScript로 인해 (일시적으로) 동적으로 변하게 되는 것이다.

3. HTML과 JavaScript - <Script>

1) <Script> 태그에는 JavaScript 가 작성되고 실행되도록 정의되어 있다.

```
<script>
  document.write('hello world')
</script>
```

웹 브라우저에서 hello world가 출력된다.

2) HTML과 JavaScript 의 차이

HTML은 작성된 그대로 출력되지만 (정적)

Javascript는 작성된 결과, 산출출력을 실행한다.(동적)

<pre><body> <h2>HTML</h2> 1+1 <h2>JavaScript</h2> <script> document.write(1+1) </script> </body></pre>	HTML 1+1 JavaScript 2
--	--

4. HTML과 JavaScript - 이벤트

HTML은 태그와 속성으로 작성된다.

<input> 은 태그이다. type, value, onclick, onchange 등이 속성에 해당한다.

```
<input type="button" value="hi" onclick="alert('hi')">
<input type="text" onchange="alert('changed')">
```



1) **onclick** 속성의 속성값으로는 자바스크립트가 반드시 와야한다고 정의되어 있다.

또한 **onclick** 속성의 속성값은 웹브라우저가 기억하고 있다가 **onclick** 속성이 위치하고 있는 태그에 사용자가 이벤트를 발생시, 속성값에 따라 웹브라우저가 동작하게 된다.

2) **onchange** 속성은 변화가 감지될 경우 실행된다. 두번째 input 박스는 text 를 입력할 수 있게 되고, ㄱㅇㄷ같이 어떤 내용을 입력시 alert 가 등장한다. ㄱㅇㄷ이후에 무언가를 쓰고 지워서 변화가 없으면 alert 가 뜨지 않도록 감지된다.

이외에도 javascript keydown event attribute 등으로 검색해보자

5. [HTML과 JavaScript - Console](#)

페이스북을 콘솔창으로 열어서 랜덤추출 코드를 통해 출력가능
콘솔창은 동적으로 출력된다.

6. 자료형(DataType) - 문자열과 숫자

JavaScript의 타입과 자료구조

7. 변수와 대입연산자

자바스크립트에서는 변수 선언시에 데이터타입을 지정하지않고, 변수에 담기는 대상에 따라 알아서 바뀐다

9. CSS 기초 : <div>, . Style 속성

1) <div>,

CSS를 적용하기 위해 모든 영역에서 사용가능한 무색무취의 태그이다.

<div> 는 block level element , 자동 줄바꿈

 은 inline element, 줄바꿈은 되지 않는다.

2) style 속성의 속성값으로는 css가 와야한다고 정의되어 있다. style 이라는 속성 안에 css를 넣으면 디자인을 할 수 있다. 즉, 특정 태그를 css 언어로 디자인하고 싶을 때는 style이라는 속성을 쓰고 그 안에 속성의 문법을 쓰면 된다.

```
<h2 style="background-color:coral ; color:powderblue">JavaScript</h2>
```

3) 선택자 : 속성을 누구에게 적용할 것인가 ex) tag, class, id

특정 문단의 글자(ex)Javascript)에 볼드체를 전부 씌운다고 가정해보자.
하나씩 하드코딩한다면 가능할 것이다.

```
<span style="font-weight: bold;">JavaScript</span>
```

```
<span style="font-weight:bold;">JavaScript</span> ('dʒɑ:vəˌskɹɪpt/[6]),  
often abbreviated as JS, is a high-level, dynamic, weakly typed,  
prototype-based, multi-paradigm, and interpreted programming language.  
Alongside HTML and CSS, <span style="font-weight:bold;">JavaScript</span>  
is one of the three core technologies of World Wide Web content production.  
It is used to make webpages interactive and provide online programs,  
including video games. The majority of websites employ it, and all modern  
web browsers support it without the need for plug-ins by means of a  
built-in <span style="font-weight:bold;">JavaScript</span> engine. Each of  
the many <span style="font-weight:bold;">JavaScript</span> engines  
represent a different implementation of <span  
style="font-weight:bold;">JavaScript</span>, all based on the ECMAScript  
specification, with some engines not supporting the spec fully, and with  
many engines supporting additional features beyond ECMA.
```

그러나 만약 적용할 개체수가 상당히 많다면 문제가 발생한다.

그래서 이때는 CSS 가 적용되는 <style> 태그를 통해 적용될 부분을 지정해준다.

```
<!doctype html>
<html>
<head>
  <style>
    .js{
      font-weight: bold;
    }
  </style>
</head>
<body>
  <h1><a href="index.html">WEB</a></h1>
  <h2 style="background-color:coral;color:powderblue">JavaScript</h2>
  <p>
    <span class="js">JavaScript</span> (/ˈdʒɑːvəˌskript/[6]), often abbreviated
    as JS, is a high-level, dynamic, weakly typed, prototype-based,
    multi-paradigm, and interpreted programming language. Alongside HTML and
    CSS, <span class="js">JavaScript</span> is one of the three core
    technologies of World Wide Web content production. It is used to make
    webpages interactive and provide online programs, including video games.
    The majority of websites employ it, and all modern web browsers support it
    without the need for plug-ins by means of a built-in <span
    class="js">JavaScript</span> engine. Each of the many <span
    class="js">JavaScript</span> engines represent a different implementation
    of <span class="js">JavaScript</span>, all based on the ECMAScript
    specification, with some engines not supporting the spec fully, and with
    many engines supporting additional features beyond ECMA.
```

이때의 .js 가 바로 선택자이다. class 는 . 을 id 는 # 을 앞에 붙인다.

```
1  <!doctype html>
2  <html>
3  <head>
4    <style>
5      .js{
6        font-weight: bold;
7        color:red;
8      }
9      #first{
10       color:green;
11     }
12   </style>
13 </head>
14 <body>
15   <h1><a href="index.html">WEB</a></h1>
16   <h2 style="background-color:coral;color:powderblue;">JavaScript</h2>
17   <p>
18     <span id="first" class="js">JavaScript</span> (/ˈdʒɑːvəˌskript/[6]), often
19     abbreviated as JS, is a high-level, dynamic, weakly typed, prototype-based,
20     multi-paradigm, and interpreted programming language. Alongside HTML and
21     CSS, <span class="js">JavaScript</span> is one of the three core
22     technologies of World Wide Web content production. It is used to make
23     webpages interactive and provide online programs, including video games.
24     The majority of websites employ it, and all modern web browsers support it
25     without the need for plug-ins by means of a built-in <span
26     class="js">JavaScript</span> engine. Each of the many <span
```

JavaScript (/ˈdʒɑːvəˌskript/[6]), often abbreviated as JS, is a high-level, dynamic, weakly typed, multi-paradigm, and interpreted programming language. Alongside HTML and CSS, JavaScript is one of the three core technologies of World Wide Web content production. It is used to make webpages interactive and provide online programs, including video games. The majority of websites employ it, and all modern web browsers support it without the need for plug-ins by means of a built-in JavaScript engine. Each of the many

우선순위 : 태그 < 클래스 < 아이디 순서이다. 클래스로 전체를 씌우고 특정 부분만 추가 적용하고 싶다면 아이디를 적용한다.

3) 지시자

```
<input type="button" value="night" onclick="
    document.querySelector('body').style.backgroundColor = 'black';
    document.querySelector('body').style.color = 'white';
">
```

```
<input type="button" value="night" onclick="
```

night 라는 버튼이 클릭된다면,

document 이 문서에서 **query** 질의(요청)한다. **Selector** CSS의 셀렉터에게 ('body') <body>태그에게
-> 이 문서에서 <body> 태그에게 CSS의 셀렉터가 질의한다.

style style(CSS)를 적용해라 **backgroundColor** 배경라운드색상을 = 'black'; 블랙으로
-> CSS로 배경라운드 색상을 블랙으로 적용해라.

즉. 이 문서의 <body> 태그에 배경라운드 컬러가 블랙생상인 **style**을 적용한다.

10. 프로그램, 프로그래밍, 프로그래머

program에서는 '순서'라는 개념이 아주 중요한데 (ex. 연주회 프로그램을 보면 순서대로 진행

programing은 순서를 만드는 행위로 볼 수 있고,

programer는 순서를 만드는 사람이라 정의할 수 있을 것이다.

HTML은 프로그래밍 언어가 아니다.

HTML은 웹페이지를 묘사하는 언어이기에 시간의 순서에 따라 일을 할 필요가 없기 때문이다.

JavaScript는 사용자와 상호작용하기 위해 고안된 컴퓨터 언어이기에 시간의 순서가 필요하다.

정리)

```
<!doctype html><html>
<head>
  <title>WEB1 - Welcome</title>
  <meta charset="utf-8">
</head>
<body>
  <h1><a href="index.html">WEB</a></h1>
  <ol>
    <li><a href="1.html">HTML</a></li>
    <li><a href="2.html">CSS</a></li>
    <li><a href="3.html">JavaScript</a></li>
  </ol>
  <h2>WEB</h2>
  <p>The World Wide Web</p>
</body>
```

WEB

1. [HTML](#)
2. [CSS](#)
3. [JavaScript](#)

WEB

The World Wide Web

11. 조건문

night_day 라는 버튼을 토글 기능으로 만들어 보자

```
<!doctype html>
<html>
<head>
  <title>WEB1 - Welcome</title>
  <meta charset="utf-8">
</head>
<body>
  <h1><a href="index.html">WEB</a></h1>
  <input id="night_day" type="button" value="night" onclick="
    if(document.querySelector('#night_day').value === 'night'){
      document.querySelector('body').style.backgroundColor = 'black';
      document.querySelector('body').style.color = 'white';
      document.querySelector('#night_day').value = 'day';
    }else{
      document.querySelector('body').style.backgroundColor = 'white';
      document.querySelector('body').style.color = 'black';
      document.querySelector('#night_day').value = 'night';
    }
  ">
  <ol>
    <li><a href="1.html">HTML</a></li>
    <li><a href="2.html">CSS</a></li>
    <li><a href="3.html">JavaScript</a></li>
  </ol>
  <h2>WEB</h2>
  <p>The World Wide Web
  </p>
</body>
</html>
```

버튼을 클릭하면 백그라운드가 바뀌는 토글 기능

WEB

night

1. [HTML](#)
2. [CSS](#)
3. [JavaScript](#)

WEB

The World Wide Web

WEB

day

1. [HTML](#)
2. [CSS](#)
3. [JavaScript](#)

WEB

The World Wide Web

12. 리팩토링

만약 `night_day` 라는 버튼이 아래에 하나 더 필요하다면 선택지가 발생한다.

1) 추가 버튼의 `id`를 `night_day2` 로 작성해서 `querySelector()` 에도 `night_day2` 작성하거나

```
<input id="night_day" type="button" value="night" onclick="
if(document.querySelector('#night_day').value === 'night'){
    document.querySelector('body').style.backgroundColor = 'black';
    document.querySelector('body').style.color = 'white';
    document.querySelector('#night_day').value = 'day';
}else{
    document.querySelector('body').style.backgroundColor = 'white';
    document.querySelector('body').style.color = 'black';
    document.querySelector('#night_day').value = 'night';
}
">
<input id="night_day2" type="button" value="night" onclick="
if(this.value === 'night'){
    document.querySelector('body').style.backgroundColor = 'black';
    document.querySelector('body').style.color = 'white';
    document.querySelector('#night_day2').value = 'day';
}else{
    document.querySelector('body').style.backgroundColor = 'white';
    document.querySelector('body').style.color = 'black';
    document.querySelector('#night_day2').value = 'night';
}
">
```

이렇게 중복적인 내용은 리팩토링이 필요하다

2) `this` 를 통해 작성하면 기존과 동일하게 적용된다.

현재의 input button은 `document.querySelector('#night_day')` 이 내용이다.

input button내의 if 문중 `document.querySelector('#night_day').value = 'day';` 이 부분을

`this.value = 'day';` 로 줄일 수 있게 된다.

```
<input id="night_day" type="button" value="night" onclick="
if(this.value === 'night'){
    document.querySelector('body').style.backgroundColor = 'black';
    document.querySelector('body').style.color = 'white';
    this.value = 'day';
}else{
    document.querySelector('body').style.backgroundColor = 'white';
    document.querySelector('body').style.color = 'black';
    this.value = 'night';
} "> //this 로 리팩토링해서 간결해진다
```

3) 변수로 퀴리셀렉터 담기

`document.querySelector('body')` 이 부분도 변수 `var target` 로 담고
기존의 `document.querySelector('body')` 부분을 `target` 으로 수정한다.

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

중간정리)

```
<!doctype html>
<html>
<head>
    <title>WEB1 - Welcome</title>
    <meta charset="utf-8">
</head>
<body>
    <h1><a href="index.html">WEB</a></h1>
    <input id="night_day" type="button" value="night" onclick="
        var target = document.querySelector('body');
        if(this.value === 'night'){
            target.style.backgroundColor = 'black';
            target.style.color = 'white';
            this.value = 'day';
        }else{
            target.style.backgroundColor = 'white';
            target.style.color = 'black';
            this.value = 'night';
        }
    ">
    <ol>
        <li><a href="1.html">HTML</a></li>
        <li><a href="2.html">CSS</a></li>
        <li><a href="3.html">JavaScript</a></li>
    </ol>
    <h2>WEB</h2>
    <p>The World Wide Web
</p></body></html>
```


13. 배열과 반복문

HTML	Result
<pre><h1>Loop & Array</h1> <script> var coworkers = ['egoing', 'leezche', 'duru', 'taeho']; </script> <h2>Co workers</h2> <script> var i = 0; while(i < coworkers.length){ document.write(''+coworkers[i]+''); i = i + 1; } </script> </pre>	<h2>Loop & Array</h2> <h3>Co workers</h3> <ul style="list-style-type: none">• egoing• leezche• duru• taeho

F12 콘솔창에서 **shift+enter** 를 하면 계속 작성할 수 있다.

1) 반복문을 통해 리스트들도 CSS를 적용할 수 있게 된다.

```
<!doctype html>
<html>
<head>
  <title>WEB1 - Welcome</title>
  <meta charset="utf-8">
</head>
<body>
  <h1><a href="index.html">WEB</a></h1>
  <input id="night_day" type="button" value="night" onclick="
    var target = document.querySelector('body');
    if(this.value === 'night'){
      target.style.backgroundColor = 'black';
      target.style.color = 'white';
      this.value = 'day';

      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';
      this.value = 'night';
      var alist = document.querySelectorAll('a');
      var i = 0;
      while(i < alist.length){
```

```

        alist[i].style.color = 'blue';
        i = i + 1;
    }
}
">
<ol>
  <li><a href="1.html">HTML</a></li>
  <li><a href="2.html">CSS</a></li>
  <li><a href="3.html">JavaScript</a></li>
</ol>
<h2>WEB</h2>
<p>The World Wide Web
</p>
</body>
</html>

```

WEB

night

1. [HTML](#)
2. [CSS](#)
3. [JavaScript](#)

WEB

The World Wide Web

WEB

day

1. [HTML](#)
2. [CSS](#)
3. [JavaScript](#)

WEB

The World Wide Web

14. 함수

1) 함수로 중복적인 부분을 요약

결국 리팩토링도 함수화 한다고 볼수 있다
익숙해지면 처음부터 함수화할 수 있게 짜보자

HTML	Result
<pre><h1>Function</h1> <h2>Basic</h2> <script> function two(){ document.write('2-1'); document.write('2-2'); } document.write('1'); two(); document.write('3'); two(); </script> <h2>Parameter & Argument</h2> <h2>Return</h2></pre>	<h1>Function</h1> <h2>Basic</h2> <ul style="list-style-type: none">• 1• 2-1• 2-2• 3• 2-1• 2-2

2) 매개변수(parameter)와 인자(argument)

```
<script>
  function onePlusOne(){
    document.write(1+1+'<br>');
  }
  onePlusOne();
  function sum(left, right){
    document.write(left+right+'<br>');
  }
  sum(2,3); // 5
  sum(3,4); // 7
</script>
```

3) return

<pre><h2>Return</h2> <script> function sum2(left, right){ return left+right; } document.write(sum2(2,3)+'
'); document.write('<div style="color:red">'+sum2(2,3)+'</div>'); document.write('<div style="font-size:3rem;">'+sum2(2,3)+'</div>'); </script></pre>	<h2>Return</h2> 5 5 5
--	---------------------------------------

15. 함수의 활용

반복적으로, 중복적으로 실행될 기능을 담은 코드가 있다면 이를 함수(){}에 담고,
<script> 함수(){self}</script> 스크립트 태그에 담는다.
이후 함수를 호출은 함수(this); 로 출력할 수 있다.

```
<script>
function nightDayHandler(self){
    var target = document.querySelector('body');
    if(self.value === 'night'){
        target.style.backgroundColor = 'black';
        target.style.color = 'white';
        self.value = 'day';
        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 = 'night';
        var alist = document.querySelectorAll('a');
        var i = 0;
        while(i < alist.length){
            alist[i].style.color = 'blue';
            i = i + 1;
        }
    }
}
</script>
<body>
    <h1><a href="index.html">WEB</a></h1>
    <input id="night_day" type="button" value="night" onclick="
    nightDayHandler(this);
    ">
    <ol>
        <li><a href="1.html">HTML</a></li>
        <li><a href="2.html">CSS</a></li>
        <li><a href="3.html">JavaScript</a></li>
    </ol>
</body>
```

이제 함수 `function nightDayHandler(self)` 에서 `self` 는 onclick 이 이루어진 input 태그 자신을 가리킨다.
그래서 `self.value = 'day';` 라고 하면 input 태그의 value 가 바뀌게 된다!

1) var alist 와 while 부분도 함수로 리팩토링 해보자

```
<script>
function LinkssetColor(color){
  var alist = document.querySelectorAll('a');
  var i = 0;
  while(i < alist.length){
    alist[i].style.color = 'color';
    i = i + 1;
  }
}
function nightDayHandler(self){
  var target = document.querySelector('body');
  if(self.value === 'night'){
    target.style.backgroundColor = 'black';
    target.style.color = 'white';
    self.value = 'day';
    LinkssetColor('powderblue');
  } else {
    target.style.backgroundColor = 'white';
    target.style.color = 'black';
    self.value = 'night';
    LinkssetColor('blue');
  }
}
</script>
```

아래 부분을 `function LinkssetColor(color)` 함수화한다.

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

`alist[i].style.color = 'color';` 로 변경하고

기존의 영역에 `LinkssetColor('powderblue');` 로 매개변수를 전달한다.

2) 이 부분도 함수화 `target.style.color = 'black';` 한다.

함수화 하고,

```
function BodySetColor(color){
    target.style.color = color;
}
```

기존의 `target.style.color = 'white';`는

이걸로 바꾼다. `BodySetColor('white');`

위 함수화 한부분의 `target` 도 변수화되었는데 다른 함수안에 있으므로 다시 설정해주자

```
function BodySetColor(color){
    document.querySelector('body').style.color = color;
}
```

3) 나머지 한 부분도 함수화

```
function BodySetBackgroundColor(color) {
    document.querySelector('body').style.backgroundColor = color;
}
```

기존의 `target.style.backgroundColor = 'black';` 도

이걸로 바꾼다. `BodySetBackgroundColor('black');`

4) 전반적으로 아래와 같이 바꾼다.

```
<!doctype html>
<html>
<head>
  <title>WEB1 - Welcome</title>
  <meta charset="utf-8">
</head>
<script>
function LinksetColor(color){
  var alist = document.querySelectorAll('a');
  var i = 0;
  while(i < alist.length){
    alist[i].style.color = color;
    i = i + 1;
  }
}
function BodySetColor(color){
  document.querySelector('body').style.color = color;
}
```

```

}

function BodySetBackgroundColor(color){
    document.querySelector('body').style.backgroundColor = color;
}

function nightDayHandler(self){
    var target = document.querySelector('body');
    if(this.value === 'night'){
        BodySetBackgroundColor('black');
        BodySetColor('white');
        this.value = 'day';
        LinksetColor('powderblue');

    }else{
        BodySetBackgroundColor('white');
        BodySetColor('black');
        this.value = 'night';
        LinksetColor('blue');
    }
}

</script>

<body>
<h1><a href="index.html">WEB</a></h1>
<input id="night_day" type="button" value="night" onclick="
    nightDayHandler(this);
">
<ol>
<li><a href="1.html">HTML</a></li>
<li><a href="2.html">CSS</a></li>
<li><a href="3.html">JavaScript</a></li>
</ol>
<h2>WEB</h2>
<p>The World Wide Web
</p>
</body>
</html>

```

여기까지가 Test3.HTML

16. 객체

1) 쓰기과 읽기

```
<script>
  var coworkers = {
    "programmer":"egoing",
    "designer":"leezche"
  };
  document.write("programmer : "+coworkers.programmer+"<br>"); //programmer : egoing
  document.write("designer : "+coworkers.designer+"<br>"); //designer : leezche
  coworkers.bookkeeper = "duru";
  document.write("bookkeeper : "+coworkers.bookkeeper+"<br>"); //bookkeeper : duru
  coworkers["data scientist"] = "taeho";
  document.write("data scientist : "+coworkers["data scientist"]+"<br>"); //data scientist : taeho
</script>
```

키와 밸류값 입력도 새로운 방식이 보이고,
띄워쓰기된 형태의 입력도 보이네

2) 객체와 반복문

[Object.entries\(\)](#) : Javascript의 [key, value] 쌍의 배열을 반환은 for...in 은 사용한다.

```
for(var key in coworkers){
}
```

```
<body>
  <h1>Object</h1>
  <h2>Create</h2>
  <script>
    var coworkers = {
      "programmer":"egoing",
      "designer":"leezche"
    };
    document.write("programmer : "+coworkers.programmer+"<br>");
    document.write("designer : "+coworkers.designer+"<br>");
    coworkers.bookkeeper = "duru";
    document.write("bookkeeper : "+coworkers.bookkeeper+"<br>");
    coworkers["data scientist"] = "taeho";
    document.write("data scientist : "+coworkers["data scientist"]+"<br>");
  </script>
  <h2>Iterate</h2>
  <script>
    for(var key in coworkers) {
```



```

    document.write(key+'<br>');
  }
</script>
<script>
  for(var key in coworkers) {
    document.write(coworkers[key]+'<br>');
  }
</script>
<script>
  for(var key in coworkers) {
    document.write(key+' : '+coworkers[key]+'<br>');
  }
</script>
</body>

```

programmer
designer
bookkeeper
data scientist

egoing
leezche
duru
taeho

programmer : egoing
designer : leezche
bookkeeper : duru
data scientist : taeho

3) 객체프로퍼티와 메소드

Javascript 의 객체에도 함수도 담을 수 있다.
객체명.함수명() = function(){} 의 형태로 선언한다.

```

coworkers.showAll = function(){ }
function showAll(){ } //두개는 같은 의미를 가진다.

```

기존 Iterate 부분을 함수로 옮겨서 담고, 함수를 호출한다.

```

<h2>Property & Method</h2>
<script>
  coworkers.showAll = function(){
    for(var key in coworkers) {
      document.write(key+' : '+coworkers[key]+'<br>');
    }
  }
  coworkers.showAll();
</script>

```

Property & Method

programmer : egoing
designer : leezche
bookkeeper : duru
data scientist : taeho
showAll : function(){ for(var key in coworkers) {
document.write(key+' : '+coworkers[key]+'
'; } }

기존의 객체명 **var coworkers** 가 변경될 경우가 생길 수 있으므로 this를 활용한다.

```

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

```

자바스크립트에서는 객체에 소속된 함수를 메서드, 객체에 소속된 변수를 프로퍼티 라고 한다.

17. 객체의 활용

기존의 함수를 객체로 넣으면서 수정한다.

이 부분들을

```
function BodySetColor(color){
    document.querySelector('body').style.color = color;
}
```

```
function BodySetBackgroundColor(color){
    document.querySelector('body').style.backgroundColor = color;
}
```

객체로 넣으면서 수정, 객체의 프로퍼티간에는 , 콤마가 필요함에 주의

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

이제 기존의 함수들은 **Body** 객체 안으로 들어갔으니 **nightDayHandler** 함수내부의 메서드들도 **Body** 뒤에 . 을 붙여준다.

아래들을

```
BodySetBackgroundColor('black');
BodySetColor('white');
```

. 을 찍어 **Body** 객체 안의 메서드로 찾고 인식할 수 있게 변경한다.

```
Body.SetBackgroundColor('black');
Body.SetColor('white');
```

중간정리

```
<!doctype html>
<html>
<head>
    <title>WEB1 - Welcome</title>
    <meta charset="utf-8">
</head>
<script>
function LinksetColor(color){
    var alist = document.querySelectorAll('a');
    var i = 0;
    while(i < alist.length){
        alist[i].style.color = color;
    }
}
```

```

        i = i + 1;
    }
}

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

function nightDayHandler(self){
    var target = document.querySelector('body');
    if(this.value === 'night'){
        Body.SetBackgroundColor('black');
        Body.SetColor('white');
        this.value = 'day';
        LinksetColor('powderblue');
    }else{
        Body.SetBackgroundColor('white');
        Body.SetColor('black');
        this.value = 'night';
        LinksetColor('blue');
    }
}
</script>

<body>
    <h1><a href="index.html">WEB</a></h1>
    <input id="night_day" type="button" value="night" onclick="
        nightDayHandler(this);
    ">
    <ol>
        <li><a href="1.html">HTML</a></li>
        <li><a href="2.html">CSS</a></li>
        <li><a href="3.html">JavaScript</a></li>
    </ol>
    <h2>WEB</h2>
    <p>The World Wide Web
    </p>
</body>
</html>

```

Links 부분도 객체화 해보자.

```
<!doctype html>
<html>
<head>
  <title>WEB1 - Welcome</title>
  <meta charset="utf-8">
</head>
<script>
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;
    }
  }
}

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

function nightDayHandler(self){
  var target = document.querySelector('body');
  if(this.value === 'night'){
    Body.SetBackgroundColor('black');
    Body.SetColor('white');
    this.value = 'day';
    Links.setColor('powderblue');

  }else{
    Body.SetBackgroundColor('white');
    Body.SetColor('black');
    this.value = 'night';
    Links.setColor('blue');
  }
}
</script>
```

```
<body>
  <h1><a href="index.html">WEB</a></h1>
  <input id="night_day" type="button" value="night" onclick="
    nightDayHandler(this);
  ">
  <ol>
    <li><a href="1.html">HTML</a></li>
    <li><a href="2.html">CSS</a></li>
    <li><a href="3.html">JavaScript</a></li>
  </ol>
  <h2>WEB</h2>
  <p>The World Wide Web
</p>
</body>
</html>
```

여기까지가 Test4.HTML

18. 파일로 쪼개서 정리 정돈하기

1) input 태그를 전체 페이지에 적용시킨다.

1,2,3 HTML에 전부 복붙

```
<input id="night_day" type="button" value="night" onclick="
    nightDayHandler(this);
">
```

2) color.js 파일을 만들고,

<script> </script> 사이의 소스를 복붙한다. <script> 태그가 들어가면 안된다. 저장할 것!

```
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;
    }
  }
}

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

function nightDayHandler(self){
  var target = document.querySelector('body');
  if(this.value === 'night'){
    Body.SetBackgroundColor('black');
    Body.SetColor('white');
    this.value = 'day';
    Links.setColor('powderblue');
  }else{
    Body.SetBackgroundColor('white');
    Body.SetColor('black');
    this.value = 'night';
    Links.setColor('blue');
  }
}
```

3) color.js 를 연동하기 위해

1,2,3.html 에 다음 소스를 복붙

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

고생했다!

19. 라이브러리와 프레임워크

[jQuery](#) 라이브러리가 있다.

필요한부분을 다운 받거나,

CDN(Content Delivery Network) 으로 script src를 걸어 연결하는 두 가지 방법이 있다.

1) [Google CDN](#) 을 눌러 3.x 스니펫 소스를 복사한다.

3.x 스니펫:

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
```

2) 3.html 의 color.js 소스 앞에 넣는다.

```
<head>
<title>WEB1 - JavaScript</title>
<meta charset="utf-8">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
<script src="color.js"></script>
</head>
```

3) color.js 로 이동

jQuery는 반복문을 손쉽게 작성할 수 있다.

```
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;
        // }
        $('a').css('color', color); //이 웹페이지의 모든 a 태그를 제이쿼리로 제어하겠다.
    }
}
```

4) color.js 정리

```
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;
        // }
        $('a').css('color', color); //이 웹페이지의 모든 a 태그를 제이쿼리로 제어하겠다.
    }
}
```

```

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

function nightDayHandler(self){
  var target = document.querySelector('body');
  if(this.value === 'night'){
    Body.SetBackgroundColor('black');
    Body.SetColor('white');
    this.value = 'day';
    Links.setColor('powderblue');

  }else{
    Body.SetBackgroundColor('white');
    Body.SetColor('black');
    this.value = 'night';
    Links.setColor('blue');
  }
}

```

추가 강의

- WEB1 - HTML & Internet : <https://opentutorials.org/module/3135>
- WEB2 - JavaScript : <https://opentutorials.org/course/3085>
- WEB2-CSS : <https://opentutorials.org/course/3086>