

설리번 프로젝트 - Javascript

"자바스크립트가 세상을 먹어치우고 있다(Javascript Is Eating The World)"

- Kevin Lacker

소개

자바스크립트는 객체 기반의 스크립트 프로그래밍 언어입니다.

웹에서 HTML, CSS가 웹을 설계하고, 꾸미는 역할을 한다면,

자바스크립트는 HTML을 동적으로 제어하는 역할을 합니다. (프로그래밍적으로 제어한다고도 하죠)

예를 들어, 여러분이 이러한 HTML코드를 코딩했다고 합시다.

```
1 | <button>버튼이다!</button>
```

이런 경우, 버튼을 눌렀을 경우, 아무런 반응이 없습니다.

이럴때, 자바스크립트가 필요한 것입니다.위의 코드를 이렇게 바꾸어 봅시다.

```
1 | <button onclick="alert('버튼이 눌렀당!')">버튼이다!</button>
```

이제, 버튼을 누르면 알림창이 나옵니다. 자바스크립트는 이런 용도 외에도 다양한 활용방법이 있습니다.

여러분들에게 이제부터 가르쳐드릴 것은 자바스크립트를 활용해서 동적인 웹 페이지를 구현할 수 있는 방법들 입니다.

자바스크립트

변수

모든 프로그래밍 언어에는 변수라는 것이 있습니다.

변수는, 이름 그대로 값이 변하는 수라는 것이 됩니다.

여러분은 변수에 이름을 정해준 다음(선언), 이름으로 변수에 접근해서 변수의 값을 바꾸어 줄 수 있습니다.

먼저, 변수의 선언에 대해서 알아보시다.

변수의 선언

자바스크립트에서 변수를 선언하는 것은 엄청나게 간단한 일입니다.

```
1 | var [변수이름] = [값];
```

바로 이렇게 써주기만 하면 바로 이름을 가진 변수가 생성된 것입니다.

변수에 접근하기

변수에 접근해서 값을 수정해 봅시다.

```
1 | var test = "Hello, World!";  
2 | test = 1;
```

이렇게 변수이름으로 변수에 접근하여 값을 수정할 수 있습니다.

자료형

만약 1과, 1.5가 있으면, 1은 정수이고, 1.5는 유리수이듯이, 프로그래밍 언어에서도 그것을 표현하는 분류가 있습니다. 이 분류가 바로 **자료형**이라고 합니다. 이제부터 자료형에 대해서 알아보시다.

자료형은 언어마다 다를 수 있는데, 자바스크립트에서의 자료형을 알아보시다.

자바스크립트에서의 자료형은 기본 자료형으로 6개가 존재합니다.

- Boolean
- Null
- Undefined
- Number
- String
- Symbol 이 중에서, 우리가 자주 보게될 것들은 Boolean, Undefined, Number, String입니다.

```
1 | // Boolean - true, false만 가진다.  
2 | var bool1 = true;  
3 | var bool2 = false;  
4 |  
5 | // Undefined - 선언될때 값을 지정해주지 않으면 undefined가 넣어진다.  
6 | var none = undefined;  
7 |  
8 | // Number - 숫자. 정수, 실수 등과 같은 모든 숫자를 가리킨다.  
9 | var number1 = 1;  
10 | var number2 = 1.5;  
11 |  
12 | // String - 문자열. 문자의 집합을 넣어주는 자료형입니다.  
13 | var str1 = "Hello, World!";  
14 | var str2 = "아주 좋습니다!";
```

연산자

현실에서 사칙연산과, 다양한 연산을 할 수 있듯이, 자바스크립트에서 여러가지 연산을 사용할 수 있습니다.

연산을 사용하려면 컴퓨터에 연산 작업을 지시해주어야 하는데, 이런 작업을 지시해주는 기호를 **연산자**라고 합니다.

이제 **연산자**를 사용하여 실습을 해 봅시다.

```
1 | var victim;  
2 | victim = 1 + 2;      // 3  
3 | victim = 1.2 - 0.2; // 1.0  
4 | victim = 5 * 6;      // 30  
5 | victim = 8 / 2;      // 4
```

이렇게 말이죠, 간단한 사칙연산 외에 제곱, 제곱근, 삼각비 등의 연산이 있습니다.

그러나, 자바스크립트에서는 **비교연산자**라는 것이 있습니다.

비교연산자는 연산자 양쪽의 값이 같은지 확인해주는 역할을 합니다.

그리고, Boolean타입의 결과를 반환합니다.

```
1 | var victim;  
2 | victim = 2 == 3;    // false  
3 | victim = "2" == 3;  // false  
4 | victim = 2 == 2;    // true  
5 | victim = "2" == 2;  // true  
6 |  
7 | victim = "2" === 2; // false  
8 | victim = 2 === 2;   // true
```

== 는 자료형이 달라도, 의미하는 값이 같으면 결과가 true로 나옵니다.

그러나, **===** 는 의미하는 값과, 자료형도 모두 일치해야 true가 나옵니다.

이런 비교연산자는 조금 뒤에 유용한 사용법을 소개해 주겠습니다.

조건문

컴퓨터 과학에서 조건문(條件文)이란 프로그래머가 명시한 불린 자료형 조건이 참인지 거짓인지에 따라 달라지는 계산이나 상황을 수행하는 프로그래밍 언어의 특징이다

Boolean타입은 true와 false만으로 이루어져 있어, 조건문에서 핵심적으로 동작합니다.

여기서 조건문은 조건에 따라서 실행을 다르게 할 수 있게 만드는 것입니다.

조건문은 이렇게 사용할 수 있습니다.

```

1 | if(1 + 1 == 2) {    // true
2 |     alert("계산 정말 잘했어요! 1000년에 한명 나올 천재인걸요?");
3 | }

```

당연하게도, $1 + 1 == 2$ 의 결과는 true이므로, 저기의 중괄호 안에 있는 alert가 실행이 되는것을 확인 할 수 있다. 그러나, if옆의 조건이 false가 되면, 중괄호 안에 있는 alert가 실행이 되지 않는다.

```

1 | if(1 + 1 == 3) {    //false
2 |     alert("계산 정말 잘했어요! 100년에 한명 나올 천재인걸요?");
3 | }

```

이렇게 말이다. 조건문에는 if 말고, 다른 것도 있다. 실습으로 확인해보자.

```

1 | if(false) {
2 |     alert("이것이 나오면 안되는거야!");
3 | } else if(1 + 1 == 2) {
4 |     alert("제대로 결과가 나오네요~");
5 | } else {
6 |     alert("컴퓨터가 1 + 1 == 2를 틀릴리가 없잖아요?");
7 | }

```

else if는 if와는 다르게 여러개 올 수 있는 조건문입니다.

if에서 조건이 충족되지 않는다면, else if를 여러개 사용함으로써, 조건 검사를 더욱 많이 할 수 있습니다.

조건문 말고, 반복문이라는 것도 있으나, 이번 수업에서는 거의 사용되지 않으므로 생략하고, 더 자세히 알고싶은 사람은 따로 찾아보길 바란다.

함수

함수는 소프트웨어에서 특정 동작을 수행하는 일정 코드 부분을 의미한다.

자바스크립트에서 함수는 이렇게 정의한다.

```

1 | function [함수이름] ( [인자 이름] ) {
2 |     // 로직 구현
3 | }

```

아직은 잘 모르겠다. 더 자세히 살펴보자.

```
1 | function plus (number1, number2) {  
2 |     return number1 + number2;  
3 | }
```

음.... 조금 더 좋아진 것 같지만, 아직은 모자란 것 같다.

```
1 | function plus (number1, number2) {  
2 |     return number1 + number2;  
3 | }  
4 |  
5 | var sum = plus(10, 50); // 60  
6 | alert(sum);
```

대충 감이 잡히시나?

웹에서의 사용법

드디어, 웹에서 자바스크립트를 사용하는 방법을 배워보도록 하자.

자바스크립트는 *CSS와 사용하는 방식이 거의 같습니다.*

CSS에서 HTML파일 내부에서 스타일링을 하는 방식과, CSS파일을 따로 만들어서 HTML에서 로딩해오는 방식이 있듯이, 자바스크립트도 같습니다.

HTML코드 내부에서 사용하는 방법

test.html

```
1 | a
```

외부에서 파일을 로딩하는 방법

test.html

```
1 | <button id="hello">버튼이다!</button>  
2 | <script type="text/javascript" src="script.js"></script>
```

script.js

```
1 | var hello = document.getElementById('hello');
2 |
3 | hello.addEventListener('click', function() {
4 |     alert('Hello, World!');
5 | })
```

어느 방법으로 사용하는지는 여러분의 몫이다.

주인과 대화해보기

컴퓨터도 하나의 지성체로써, 때때로는 주인과 의사소통을 해보고 싶어하는 경향을 보인다.
컴퓨터가 주인과 의사소통을 할 수 있게 자바스크립트를 이용해서 도와주자.

컴퓨터가 주인의 이름이 알고싶다고 한다.

```
1 | prompt("주인님은 이름이 뭐예요?");
```

컴퓨터가 주인에게 알려줄 것이 있다고 한다.

```
1 | alert("동영상 좀 지워줘요. 남은 용량이 없잖아요");
```

주인이 동영상을 지우지 않자. 컴퓨터가 강제로 지운다고 한다.
강제로 지우지 못하도록, 승인 기능을 달아주자.

```
1 | confirm("지울까요? 진짜??");
```

여담 :

- prompt는 입력창에 넣은 값을 내놓는다.
- confirm은 false와 true, 둘중에 하나를 내놓는다.

```
1 | alert(prompt("입력과 출력을 동시에!"));
2 | alert(confirm("오호!"));
```

DOM

Document Object Model 은 문서의 구조적 표현을 제공하고, 당신이 그 내용과 시각적 표현을 바꾸는 것을 가능하게 한다. 본질적으로, **DOM**은 스크립트 또는 프로그래밍 언어로 웹페이지를 연결한다.

먼저, 특정 태그를 **객체**라고 한다.

html, body, section, article, h1, p 모두 다 객체라고 할 수 있다.

DOM과 자바스크립트로 **객체**의 속성과 값을 바꿀 수 있다.

심지어는 CSS 스타일 마저 바꾸어버릴 수 있다!

객체 조회하기

- id의 이름으로 객체를 찾는다.

```
1 | getElementById('id이름');
```

- class의 이름으로 객체를 찾는다.

```
1 | getElementsByClassName('class이름');
```

- 태그 이름으로 객체들을 찾는다.

```
1 | getElementsByTagName('태그이름');
```

객체 조작하기

위에서 받아온 객체를 victim이라는 변수에 넣었다고 가정하자.

- 속성 바꾸기

```
1 | victim.setAttribute("이름", "값");
```

- 값 바꾸기

```
1 | victim.innerHTML = 값;
```

- 값 받아오기

```
1 | victim.value;  
2 | victim.innerHTML;
```

- 스타일 바꾸기

```
1 | victim.style.[css property] = 값;
```

마지막으로, DOM을 활용하여 div의 스타일링을 자바스크립트로 해보자.

```
1 | <div id="victim"></div>
2 | <script>
3 |     var victim = document.getElementById('victim');
4 |
5 |     victim.style.width = 300 + "px";
6 |     victim.style.height = 300 + "px";
7 |
8 |     // 조건문
9 |     if(confirm("핫핑크 싫으세요?")) {
10 |         victim.style.background = "skyblue";
11 |         victim.setAttribute("class", "skyblue");
12 |     } else {
13 |         victim.style.background = "hotpink";
14 |         victim.setAttribute("class", "hotpink");
15 |     }
16 |
17 |     alert("원하시는대로.");
18 | </script>
```

이벤트

객체에 특정 이벤트가 발생하였을때, 자바스크립트 코드를 실행시키는 방법이다.

밑에 설명된 것들은 HTML tag의 attribute부분에 넣어주면 된다.

Ex) `<button onclick="">버튼이다!</button>`

클릭 이벤트.

```
1 | onclick=""
```

더블클릭.

```
1 | ondblclick=""
```

포커스.


```
1 | onfocus=""
```

블러.

```
1 | onblur=""
```

로딩.

```
1 | onload=""
```

벗어남.

```
1 | onunload=""
```

내용의 바뀜.

```
1 | onchange=""
```

내용의 추가.

```
1 | oninput=""
```

키가 눌림

```
1 | onkeydown=""  
2 | onkeypress=""
```

키에서 손을 땀.

```
1 | onkeyup""
```

마우스가 객체 위에 올라감.

```
1 | onmouseover=""
```

마우스가 객체 위에서 빠짐.

여기까지 배운 내용들을 하나의 예제로 실습해보자.

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>이벤트 최종예제</title>
6      <style>
7          #wrapper {
8              font-family: sans-serif;
9              margin : auto;
10             text-align: center;
11         }
12
13         #color {
14             width : 300px;
15             height : 300px;
16             margin : auto;
17             background-color: gold;
18         }
19     </style>
20 </head>
21 <body>
22     <div id="wrapper">
23         <div id="color"
24             onmouseout="changeColor('hotpink')"
25             onmouseover="changeColor('green')">
26             버튼을 누르거라.
27         </div>
28         <button onclick="changeColor('skyblue')">바꾸기</button>
29     </div>
30     <script type="text/javascript">
31         function changeColor(colorName) {
32             var elem = document.getElementById('color');
33
34             elem.style.background = colorName;
35             elem.innerHTML = "현재 색 : " + colorName;
36         }
37     </script>
38 </body>
39 </html>

```

어때요, 참 쉽죠??

