**JavaScript**

* 비교
* HTML : 웹 문서의 내용 구성
* CSS : 웹 문서의 레이아웃이나 색상, 스타일 지정
* JavaScript : 웹 문서의 각 요소를 가져와서 필요에 따라 스타일을 변경하거나 움직임
* 웹 브라우저 + 자바스크립트
* <body>앞에 <script></script> 삽입
* 외부 스크립트 파일로 저장해서 웹 문서와 연결하는 방법 많이 사용

1. .js파일로 저장
2. HTML문서에서 <script src = ‘외부 스크립트 파일 경로’></script> 작성

* 해석 과정
* <!DOCTYPE html>
* <html lang="ko">
* <head>
* <meta charset="UTF-8">
* <meta name="viewport" content="width=device-width, initial-scale=1.0">
* <title>글자 색 바꾸기</title>
* <style>
* body{text-align: center;}
* #heading{color: blue;}
* #text{
* color: gray;
* font-size: 15px;
* }
* </style>
* </head>
* <body>
* <h1>HTML 문서 안에 자바스크립트 소스 작성</h1>
* <h1 id="heading">자바스크립트</h1>
* <p id="text">위 텍스트를 클릭해 보세요</p>
* <script>
* var heading = document.querySelector('#heading');
* heading.onclick = function(){
* heading.style.color = "red";
* }
* </script>
* </body>
* </html>

1. <!DOCTYPE html>을 보고 이 문서가 웹 문서라는 것을 인지
2. <head></head>태그, <body></body>태그 사이에 어떤 태그 있는지 확인,관계 파악
3. 태그 분석 후 스타일 정보 분석
4. <script>태그를 만나면 자바스크립트 해석기에 자바스크립트 소스 넘김
5. ②에서 분석한 HTML과 ③에서 분석한 CSS 정보에 따라 웹 브라우저 화면에 표시
6. 웹 브라우저에서 텍스트 클릭 시 분석해 놓은 자바스크립트를 실행하여 화면에 표시

* 자바스크립트 용어와 기본 입출력 방법

1. 식과 문

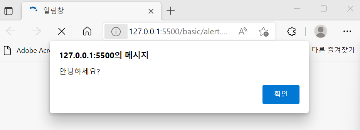
* 식 : 연산식, 문자열, 숫자 등 어떤 값을 만들어 낼 수 있으면 모두 식이 될 수 있음

식은 변수에 저장됨

* 문 : 명령, 끝에 세미콜론(;) 붙여서 구분

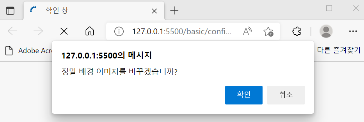
조건문, 제어문

1. 간단한 입출력 방법

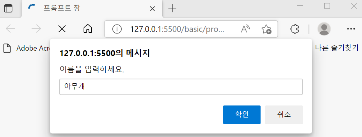
* 창
  + 알림 창 : 가장 많이 사용하는 간단한 대화 상자

작은 알림 창을 열어 메시지 표시

기본형 : alert(메시지)

* + 확인 창 : 사용자가 [확인]이나 [취소] 버튼 중 직접 클릭 가능, 결과에 따라 동작

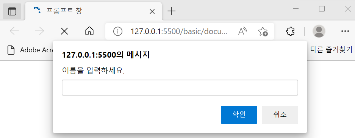
기본형 : confirm(메시지)

* + 프롬프트 창 : 텍스트 필드가 있는 작은 창

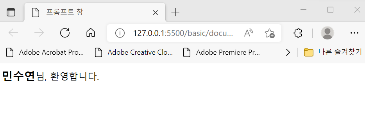
간단한 메시지 입력, 그 내용을 가져와 프로그램에 사용

기본값을 지정(힌트메시지 적혀 있음)하거나 지정하지 않을 수 있음

기본형 : prompt(메시지) 또는 prompt(메시지, 기본값)

* 문
  + document.write() 문 : 웹 브라우저 화면에 출력 담당

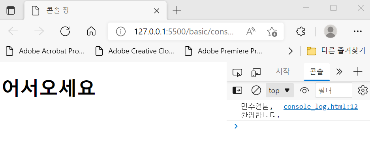
단순히 브라우저 화면에서 결과값을 활인하는 용도로 많이 사용

document : 웹 문서에서 괄호 안의 내용을 표시하는 명령문

( ) : 실제 웹 브라우저 화면에 표시할 내용이나 결과값이 저장된 변수, “ “

따옴표 안에는 HTML태그도 함께 사용 가능

연결연산자(+) : 내용과 변수 연결해주는 역할

* + console.log() 문 : 콘솔 창에 출력 담당

괄호 안의 내용을 콘솔 창에 표시

콘솔 창은 웹 브라우저의 개발자 도구 창에 포함되어 있는 공간

콘솔 창에서 소스 코드의 오류를 발견하거나 변수 값 확인 가능

( ) : 변수, 따옴표 사이 텍스트(HTML태그 사용 불가)

Ctrl + Shift + J

**자바스크립트 기본 문법**

* 연산자

1. 산술 연산자

|  |  |
| --- | --- |
| ++ | 1 증가 |
| -- | 1 감소 |

1. 비교 연산자

|  |  |  |  |
| --- | --- | --- | --- |
| == | 피연산자가 서로 같으면 true | 3 == ”3” | true |
| === | 피연산자와 자료형 모두 같으면 true | 3 === ”3” | false |
| != | 피연산자가 서로 같지 않으면 true | 3 != ”3 | false |
| !== | 피연산자와 자료형 모두 같지 않으면 true | 3 !== ”3 | true |

1. 논리 연산자

|  |  |  |
| --- | --- | --- |
| OR 연산자 | ¦¦ | 피연산자 중 하나만 true여도 true |
| AND 연산자 | && | 피연산자가 모두 true일 경우에만 true |
| NOT 연산자 | ! | 피연산자의 반대값 지정 |

* 조건문
* prompt( )로 입력 받은 값은 문자열로 저장됨
* parseInt( )함수를 사용하면 입력 받은 값을 숫자로 변환해서 저장됨 parseInt(prompt())

1. if~else 문

* 기본형 :

if(조건){

조건 결과값이 true일 때 실행할 명령

}else {

조건 결과 값이 false일 때 실행할 명령

}

1. 조건 연산자로 조건 체크

* 기본형 :

(조건) ? true일 때 실행할 명령 : false일 때 실행할 명령

1. switch 문

* 처리할 명령이 많다면 if ~ else 여러 개 사용하기보다 switch문이 더 편리
* 기본형 :

switch(조건){

case 값1 : 명령1

break

case 값2 : 명령2

break

:

default : 명령n}

* 반복문

1. for 문

* 기본형 : for (초기값; 조건; 증가식){ 실행할 명령 }
* 초기값 : 카운터 변수를 초기화함, 초기값음 0이나 1부터 시작
* 조건 : 명령을 반복하기 위해 조건 체크, 이 조건 만족해야 다음 명령 실행 가능
* 증가식 : 명령을 반복한 후 실행, 보통 카운터 변수를 1 증가시키는 용도로 사용

1. while 문과 do ~ while 문

* while 기본형 :

while(조건){ 실행할 명령 }

* do ~ while 기본형 :

do {

실행할 명령

} while (조건)

* do ~ while : 일단 명령을 실행한 후 조건을 체크

1. break 문 : 종료 조건이 되기 전에 반복문을 빠져나와야 할 경우
2. continue 문 : 주어진 조건에 해당하는 값을 만나면 해당 반복문 건너뜀

**이벤트**

* 이벤트와 이벤트 처리기

1. 마우스 이벤트

|  |  |
| --- | --- |
| click | HTML요소를 클릭할 때 이벤트 발생 |
| dblclick | HTML요소를 더블 클릭할 때 이벤트 발생 |
| mousedown | 요소 위에서 마우스 버튼을 눌렀을 때 이벤트 발생 |
| mousemove | 요소 위에서 마우스 포인터를 움직일 때 이벤트 발생 |
| mouseover | 마우스 포인터가 요소 위로 옮겨질 때 이벤트 발생 |
| mouseout | 마우스 포인터가 요소를 벗어날 때 이벤트 발생 |
| mouseup | 요소 위에 놓인 마우스 버튼에서 손을 뗄 때 이벤트 발생 |

1. 키보드 이벤트

|  |  |
| --- | --- |
| keydown | 키를 누르는 동안 이벤트 발생 |
| keypress | 키를 눌렀을 때 이벤트 발생 |
| keyup | 키에서 손을 뗄 때 이벤트 발생 |

1. 문서 로딩 이벤트

|  |  |
| --- | --- |
| abort | 문서가 완전히 로딩되기 전에 불러오기를 멈췄을 때 이벤트 발생 |
| error | 문서가 정확히 로딩되지 않았을 때 이벤트 발생 |
| load | 문서 로딩이 끝나면 이벤트 발생 |
| resize | 문서 화면 크기가 바뀌었을 때 이벤트 발생 |
| scroll | 문서 화면이 스크롤되었을 때 이벤트 발생 |
| unload | 문서에서 벗어날 때 이벤트 발생 |

1. 폼 이벤트

|  |  |
| --- | --- |
| blur | 폼 요소에 포커스를 잃었을 때 이벤트 발생 |
| change | 목록이나 체크 상태 등이 변경되면 이벤트 발생  <input>,<select>,<textarea>,<button>태그에서 사용 |
| focus | 폼 요소에 포커스가 놓였을 때 이벤트 발생  <label>,<select>,<textarea>,<button>태그에서 사용 |
| reset | 폼이 리셋 되었을 때 이벤트 발생 |
| submit | submit버튼을 클릭했을 때 이벤트 발생 |

1. 이벤트 처리기

* 이벤트를 처리하는 가장 기본적인 방법 : 이벤트가 발생한 HTML태그에 이벤트 처리기 직접 연결하기
* 기본형 : <태그 on이벤트명 = “함수명”>
* DOM을 이용한 이벤트 처리기
* 지금까지 : HTML이 주인이 되어 자바스크립트의 함수를 불러오는 방식
* DOM : 자바스크립트가 주인이 되어 HTML요소를 가져와서 이벤트 처리기를 연결
* 기본형 : 웹 요소.onclick = 함수;
* 자바스크립트에서는 웹 요소를 여러 방법으로 가져올 수 있는데 그 중 함수 querySelector( )를 가져오는 것이 쉬움

**자바스크립트와 객체**

* 내장 객체

1. Array 객체

* Array객체를 사용하려면 먼저 인스턴스를 만들어야 함
* 초기값이 없는 상태에서 단순히 객체의 인스턴스만 만든다면 new예약어 사용해 변수에 할당, 배열의 크기 지정가능
* Array 객체의 메서드

|  |  |
| --- | --- |
| concat | 기존 배열에 요소를 추가해 새로운 배열 만듦 |
| every | 배열의 모든 요소가 주어진 함수에 대해 참이면 true반환, 그렇지 않으면 false반환 |
| filter | 배열 요소 중에서 주어진 필터링 함수에 대해 true인 요소만 골라 새로운 배열을 만듦 |
| forEach | 배열의 모든 요소에 대해 주어진 함수를 실행 |
| indexOf | 주어진 값과 일치하는 값이 있는 배열 요소의 첫 인덱스 찾음 |
| join | 배열 요소를 문자열로 합침, 구분자를 지정할 수 있음 |
| push | 배열의 맨 끝에 새로운 요소를 추가한 후 새로운 length를 반환 |
| unshift | 배열의 시작 부분에 새로운 요소 추가 |
| pop | 배열의 마지막 요소를 꺼내 그 갑을 결과로 반환 |
| shift | 배열에서 첫 번째 요소를 꺼내 그 값을 결과로 반환 |
| splice | 배열에 요소를 추가할거나 삭제함  splice(2,1,”js”) : 인덱스 값이 2인 요소부터 요소 1개를 삭제 후 “js” 추가 |
| slice | 배열에서 특정한 부분만 잘라 냄 |
| reverse | 배열의 배치 순서를 역순으로 바꿈 |
| sort | 배열 요소를 지정한 조건에 따라 정렬 |
| toString | 배열에서 지정한 부분을 문자열로 반환, 각 요소는 쉼표로 구분 |

1. Date 객체

* Date 객체 메서드

|  |  |  |
| --- | --- | --- |
| 날짜, 시간 정보 가져오기 /설정하기 | getFullYear() / set | 연도를 4자리 숫자로 표시/ 설정 |
| getMonth() / set | 0 : 1월 ~ 11 : 12월 |
| getDate() / set | 1 ~ 31 |
| getDay() | 0 : 일요일 ~ 6 : 토요일 |
| getTime() / set | 1970년 1월 1일 자정 이후의 시간을 밀리 초(1/1000초)로 표시/ 설정 |
| getHours() / set | 0~23를 시로 표시/ 설정 |
| getMinutes() / set | 0~59를 분으로 표시/ 설정 |
| getSeconds() / set | 0~59를 초로 표시/ 설정 |
| getMilliseconds() / set | 0~999를 밀리 초로 표시/ 설정 |
| 날짜,시간 형식바꾸기 | toLocalString() | 현재 날짜와 시간을 현지 시간으로 표시 |
| toString() | Data 객체 타입을 문자열로 표시 |

1. Math 객체

* 브라우저와 관련된 객체

|  |  |
| --- | --- |
| window | 브라우저 창이 열릴 때마다 하나씩 만들어짐  브라우저 창 안의 요소 중 최상위 |
| document | 웹 문서마다 하나씩 있으며 <body>태그를 만나면 만들어짐  HTML 문서의 정보가 담겨있음 |
| navigator | 현재 사용하는 브라우저의 정보가 들어있음 |
| history | 현재 창에서 사용자의 방문 기록을 저장 |
| location | 현재 페이지의 URL 정보가 담겨 있음 |
| screen | 현재 사용하는 화면 정보를 다룸 |