

javascript





개요



javascript

클라이언트 주요언어

- **HTML** : 웹페이지에 콘텐츠를 표현하는 기본언어로 웹페이지나 사이트를 제작할 때 사용하는 언어
→ 하이퍼텍스트 구현을 위한 뼈대
- **CSS** : 마크업 언어 사용시 작성한 웹 페이지 문서에 대한 스타일을 적용한 언어
→ 하이퍼텍스트를 꾸미기 위한 옷의 기능
- **javascript** : 로컬의 브라우저에서 실행되는 인터프리터 방식의 프로그래밍 언어
→ 하이퍼텍스트의 기능을 담당
- **jQuery** : 자바스크립트의 코드가 길어지면 사용하기가 복잡해지는 단점을 파격적으로 개선한, 존 레식(John Resig)이 창안한 자바스크립트 기반의 라이브러리 중 하나



javascript

javascript

- 웹브라우저에서 많이 사용하는 인터프리터 방식의 객체지향 프로그래밍 언어
- 자바스크립트는 ECMA Script 표준을 따르는 대표적인 웹기술
 - ※ ECMA(European Computer Manufacturers Association) : 표준화 기구
 - ※ ECMA Script : ECMA 표준화 단체에서 javascript를 기초하여 구성한 스크립트 표준

스크립트언어

- 매우 빠르게 배우고 작성하기 위해 고안되었고 짧은 소스코드 파일이나 REPL(Read Eval Print Loop)로 상호작용
- 기본 프로그램 동작을 사용자의 요구에 맞게 수행되도록 해주는 용도로 주로 사용

※ REPL : 하나의 입력을 받아서(Read single input), 처리하고(Evaluate), 결과를 반환하는(Print result) 환경으로 구형된 프로그램
자바처럼 전체 소스코드를 컴파일 하는 것이 아닌 한 줄 단위로 해석하여 바로바로 수행





작성 및 실행



javascript

브라우저 개발자 도구

- 브라우저별 개발자 도구(Develop Tools)가 있음
- 크롬/IE 브라우저 F12키 눌러서 실행
- 자바스크립트 소스코드 중 `console.log()`는 브라우저에 출력하는 것이 아니라 개발자 도구의 console 패널에 출력하는 것이며 스크립트 구문을 디버깅할 때 자주 사용



javascript

자바스크립트 선언

- `<script></script>` 태그 사이에 자바스크립트 코드 문장을 작성
- HTML에서 제공하는 `<script></script>`태그를 사용하여, 자바 스크립트 작성 영역을 설정
- `type`속성이 브라우저 호환성을 위해 사용되나 default값으로 생략이 가능

자바스크립트 작성방법

```
<script type="text/javascript">  
    자바스크립트 내용  
</script>
```

※ `script`태그에는 `type`외에 `language`, `charset` 속성이 추가로 존재하지만, `language`속성은 폐기되었고 `charset`속성은 `<meta>`태그가 적용되어 사용 할 필요가 없음



javascript

자바스크립트 위치

- `<script>`/`</script>`는 `<head>`,`<body>`안 어느 영역에나 작성 가능
- `<html>`태그 영역 밖에 작성 가능
- 웹 표준과 웹 접근성, 웹 호환성을 고려해서 `<head>`나 `<body>`에 작성

웹 표준 – 웹 기술을 표준화 하기 위한 일련의 단계와 요구사항으로 플러그인 기술과 같은 비표준 기술을 배제하고, W3C에서 나온 권고안을 사용하는 것을 말하며, 웹 문서의 구조와 표현, 동작을 구분해서 사용하는 것을 뜻한다. 또한 브라우저의 버전이나 종류에 상관없이 호환이 가능하도록 제시된 표준

웹 접근성 – 장애를 가진 사람과 장애를 갖지 않은 사람 모두가 웹사이트를 이용할 수 있게 하는 방식

웹 호환성 – 웹사이트가 작동하는 서비스 이용자 단말기의 하드웨어 및 소프트웨어 환경이 다른 경우에도 동등한 서비스를 제공하는 것



javascript

자바스크립트 작성 방식

- inline 방식 : 자바스크립트 양이 소량일 때 사용하며, 태그에 이벤트 핸들러 속성을 이용하여 직접 실행코드를 작성
`<input type="button" onclick="alert('클릭');" value="버튼">`
- internal 방식 : 가장 일반적인 방식으로 html파일 내 `<script>`태그 내부에 실행코드를 작성
`<script>`
 `alert("짠!");`
`</script>`
- external 방식 : 자바스크립트의 양이 많은 경우 자바스크립트 코드 부분을 외부 파일로 저장하여 작성
`<script src="파일경로">`태그를 이용하여 내용을 삽입 후 사용



javascript

자바스크립트 실행 방식

웹 브라우저에 내장되어 있는 자바스크립트 파서가 소스 코드를 한 줄씩 읽고 해석한다는 점에서 인터프리터 방식으로 소스를 해석하기 때문에 전체를 해석해 놓은 컴파일 언어와 상이

자바스크립트 실행은, 작성된 html 문서를 브라우저에서 읽어 들이면 바로 실행을 확인
→ html태그보다 자바스크립트 실행구문이 먼저 작성되어 있는 경우 화면 구성보다 먼저 수행





데이터 입/출력



javascript

데이터 출력 방법

코드	설명
<code>document.write(내용);</code>	브라우저 화면 상의 페이지에 값을 출력
<code>window.alert(내용);</code>	내용을 메시지창에 출력(window는 생략 가능)
<code>innerHTML = 내용;</code>	태그 엘리먼트의 내용을 변경
<code>console.log(내용);</code>	개발자 도구 화면의 콘솔에 출력



javascript

데이터 입력 방법

- 자바스크립트 내장 객체인 window 객체가 제공하는 confirm(), prompt() 메소드를 사용하여 입력 받는 방법
- HTML 태그에 접근하여 대상의 값을 읽는 방법
- HTML form 태그의 input 입력 양식을 통해 값을 입력 받는 방법



javascript

window.confirm()

- 어떤 질문에 대해 “예/아니오”의 결과를 얻을 때 사용
- 대화창에 메시지와 확인/취소 버튼 표시
- 확인 버튼 선택 시 true, 취소 버튼 선택 시 false 리턴

```
var 변수 = window.confirm(“질문내용”);
```

window.prompt()

- 텍스트 필드와 확인/취소 버튼이 있는 대화창 출력
- 텍스트필드에 입력한 메시지 내용이 리턴

```
var 변수 = window.prompt(“질문내용”);
```





html 태그 접근



javascript

`document.getElementById("아이디명")`

- 태그의 id 속성의 값을 이용해서 태그 엘리먼트 객체 정보를 가져옴
- id속성은 페이지 내에서 태그의 유일한 식별자 역할
- 리턴은 단일 엘리먼트(중복 id를 여러 개 사용 해도 1개만 리턴)

```
var 변수 = document.getElementById("아이디명");
```

```
console.log(변수.innerHTML);
```



javascript

`document.getElementsByClassName("클래스명")`

- 태그의 class 속성의 값을 이용해서 태그 엘리먼트 객체 정보를 가져옴
- 동일한 class 속성 값을 가진 엘리먼트들을 모두 가져옴
- 리턴은 엘리먼트 객체 배열(해당 class 속성의 엘리먼트가 1개여도 배열로 리턴)

```
var 변수 = document.getElementsByClassName("클래스명");
```

```
console.log(변수[0].innerHTML);
```



javascript

`document.getElementsByName("이름")`

- 태그의 name 속성의 값을 이용해서 태그 엘리먼트 객체 정보를 가져옴
- 동일한 name 속성 값을 가진 엘리먼트들을 모두 가져옴
- 리턴은 엘리먼트 객체 배열(해당 name 속성의 엘리먼트가 1개여도 배열로 리턴)

```
var 변수 = document.getElementById("이름");
```

```
console.log(변수[0].innerHTML);
```



javascript

`document.getElementsByTagName("태그명")`

- 태그의 태그명을 이용해서 태그 엘리먼트 객체 정보를 가져옴
- 태그 엘리먼트들을 모두 가져옴
- 리턴은 엘리먼트 객체 배열(해당 태그 엘리먼트가 1개여도 배열로 리턴)

```
var 변수 = document.getElementsByTagName("태그이름");
```

```
console.log(변수[0].innerHTML);
```





기본문법



javascript

주석처리

- 자바스크립트의 주석은 자바와 동일
 - 한 줄 주석 : //
 - 여러 줄 주석 : /* */
- HTML 주석 <!-- -->



javascript

변수 선언

구분	설명	변수 선언
멤버변수 (전역변수)	페이지 내 여러 함수에서 사용가능한 변수	변수명 = 값;
지역변수	선언한 함수 내부에서만 사용가능한 변수	var 변수명 = 값;

※ javascript에서는 자료형이 여러 개 존재 하지만 선언 할 때에는 var 로만 선언
대입하는 데이터에 맞춰서 자료형이 생성



javascript

변수 명명 규칙

1. 영어 대/소문자, 숫자, _, \$ 사용 가능
2. 첫글자 숫자 사용 불가
3. 공백을 포함한 특수문자 사용 불가(., \$ 만 가능)
4. 이름에 의미있는 단어 조합 권장
5. 예약어를 이름으로 사용 불가
6. 두 단어 이상 결합 시 카멜표기법 권장
7. 한글 사용 가능
8. 생성자 함수의 이름은 항상 대문자로 시작
9. 변수, 인스턴스, 함수, 메소드 이름은 항상 소문자로 시작



javascript

자료형 – 문자열

- “”, ‘’로 묶여 있는 리터럴
- 내장 객체로 String 객체 → 기본적인 메소드 존재

메소드	내용
toUpperCase()	모든 문자 대문자로 변환
toLowerCase()	모든 문자 소문자로 변환
length	글자 개수 조회용 멤버변수
indexOf()	찾는 문자의 위치 리턴
lastIndexOf()	뒤에서 부터 찾는 문자의 순번 리턴
charAt()	찾는 위치의 문자 리턴
substring()	값의 일부분만 리턴
split()	토큰 문자로 분리한 문자열 배열 리턴



javascript

자료형 – 숫자

- 정수형 숫자와 부동소수점 숫자로 구분
- 내장 객체로 Math 객체 제공, 기본 메소드 존재

메소드	내용
Math.abs()	절대값 리턴
Math.random()	임의의 난수 발생 리턴
Math.round()	반올림처리 후 리턴
Math.floor()	부동소수점 숫자를 정수로 리턴(버림)
Math.ceil()	소수점 자리에서 무조건 올림



javascript

자료형 – 기타자료형

자료형	내용
논리값(boolean)	true, false 두가지 값을 가짐
객체(Object)	new로 선언된 사용자 객체와 자바스크립트 내장 객체
undefined	변수명이나 함수명으로 선언되지 않은 식별자일때 지정
함수(function)	함수(메소드)를 가지는 자료형



javascript

자료형 – typeof()

- 값의 자료형을 확인하는 연산자
→ 선언 시 자료형을 지정하지 않아 자료형 확인 시 사용

구분	결과
typeof('문자열') or typeof("문자열")	String
typeof(숫자)	number
typeof(참/거짓)	boolean
typeof(객체)	object
typeof(초기값이 없는 변수)	undefined
typeof(function)	function



javascript

데이터 형변환

1. 숫자 → 문자열

- 숫자와 문자를 + 연산하게 되면 문자가 우선되어 숫자를 문자로 변환
- 강제 형변환 : `String()` 함수 사용
→ `String(숫자);`

2. 문자열 → 숫자

- 숫자, 문자 + 이외의 사칙연산시 숫자가 우선되어 문자를 숫자로 변환
- 강제 형변환 : `Number()`, `parseInt()`, `parseFloat()` 함수 사용
→ `Number(문자열)`, `parseInt(문자열)`, `parseFloat(문자열)`



javascript

연산자

종류	연산자
최우선 연산자	(), [], .
단항 연산자	++, --, +, -
산술 연산자	+, -, *, /, %
관계 연산자	>, <, >=, <=, ==, !=, ===, !==
논리 연산자	&&,
대입 연산자	=
복합대입 연산자	+=, -=, *=, /=, %=
삼항연산자	?;

연산자 우선순위 : 최우선 > 단항 > 산술 > 관계 > 논리 > 삼항 > 대입



javascript

제어문

종류	연산자
조건문	if, if ~ else, if ~ else if, switch, 짧은 조건문(,&&)
반복문	for, while, do~while, for in
분기문	continue, break

