

javascript 기본문법





데이터 입·출력



javascript

주석처리

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



javascript

데이터 입·출력 함수

코드	설명
<code>console.log(내용);</code>	개발자 도구 화면의 콘솔에 출력
<code>window.alert(내용);</code>	내용을 메시지창에 출력
<code>window.confirm(질문);</code>	질문을 한 후 결과를 true/false로 리턴
<code>window.prompt(질문);</code>	질문을 한 후 사용자 입력값을 문자열형태로 리턴

※ `alert`, `confirm`, `prompt` 앞의 `window`는 생략가능



javascript

데이터 출력 함수

console.log();

- 개발 도중 확인을 위한 용도로 사용하며 브라우저 콘솔에 메시지를 출력
- 자바의 `System.out.println();`과 유사

```
console.log("Hello Javascript");  
console.log(1234);  
console.log("hi", 1234, true);
```

window.alert();

- 사용자에게 확인용 메시지를 전달하기 위해 사용
- 개발 시 값 확인용으로도 사용할 수 있으나 `alert`은 클라이언트에게 노출되므로 `console.log` 사용

```
window.alert("Hello Javascript");  
alert(1234);
```



javascript

데이터 입력 함수

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

```
console.log(window.confirm("질문내용"));
```

window.prompt()

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

```
console.log(window.prompt("질문내용"));
```





변수



javascript

변수 선언 타입

키워드	변수 선언
var	<ul style="list-style-type: none">- 과거에 변수를 선언할 때 사용하던 타입- 변수의 범위가 모호하여 코드가 길어지면 불편 ex) var 변수명; 또는 var 변수명 = 값;
let	<ul style="list-style-type: none">- 변수의 값이 변경 가능한 경우 사용하는 타입- 변수에 값이 대입된 후 값을 변경하여 사용하는 경우 사용 ex) let 변수명; 또는 let 변수명 = 값;
const	<ul style="list-style-type: none">- 변수에 값이 한번 들어간 뒤 변경이 불가능한 변수(상수형)- 변수에 값이 대입된 후 값을 변경할 수 없음 ex) const 변수명; 또는 const 변수명 = 값;

※ javascript에서는 자료형이 여러 개 존재 하지만 선언 할 때는 자료형을 지정하지 않고 데이터를 대입하면 자동으로 자료형이 설정



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





자료형



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
배열(Array)	다양한 타입의 여러 데이터를 인덱스로 관리하는 변수 모음
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)`





연산자 & 제어문



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

