1. 자바스크립트의 소개

* 웹 개발자가 필수적으로 학습해야 하는 세 가지 요소
  + HTML: 웹페이지의 콘텐츠를 정의
  + CSS: 웹페이지의 스타일 정의
  + 자바스크립트: 동작
* 자바 언어와 자바스크립트의 차이점

|  |  |  |
| --- | --- | --- |
| 특징 | 자바 언어 | 자바스크립트 |
| 언어 종류 | 컴파일언어 | 인터프리터 언어 |
| 실행 방식 | 자바 가상 기계 위에서 시행 | 브라우저 위에서 실행 |
| 작성 위치 | 별도의 소스 파일로 작성 | HTML 파일 안에 삽입 가능 |
| 변수 선언 | 변수의 타입 선언 | 변수 타입 선언 안함 |

* 1. 자바 스크립트의 역사

자바 스크립트 동적의 처리나 사용자의 이벤트 처리 등을 위해 웹브라우저에서 많이 사용하는 프로그래밍 언어이다.

* 1995년 모카라는 이름으로 탄생 (브랜든 아이크)
* 이후 라이브스크립트로 이름을 변경
* 1996년 넷스케이프사에서 자바스크립트 버전 1.0버전 발표
* 2010년 6월 발표한 1.8.5 버전이 가장 최근 버전

자바 스크립트 표준은 유럽컴퓨터 제조업자협회(European Computer Manufactures Association)의 규격을 구현하는 ECMAScript이다. ECMAScript의 가장 최근 버전은 6.4이다.

* 1. 자바스크립트의 특징

자바스크립트는 기술적인 측면에서 볼 때 객체 지향적이고 느슨한 타입의 언어이다. 다음은 자바 스크립트 언어의 특징이다.

1. 객체기반 언어이다. => 프로그래밍에서 사용하는 모든 변수는 객체이다. 함수, 배열도 모두 객체이다. 또한 모든 객체는 프로퍼티(property: 속성)와 메서드(method)를 갖는다.
2. 타입에 관대하다. (동적 타이핑) => 변수와 데이터가 쉽게 다른 타입으로 바뀔 수 있다. 타입에 엄격하지 않아 타입 변환이 자동으로 일어나며 유연성을 갖는다.
3. 스크립트 언어이다. => 스스로 실행되는 형태가 아니라 웹브라우저의 스크립트 엔진을 통해 실행된다.
4. 인터프리터 언어이다. => 컴파일 과정을 거치지 않고 바로 실행시킬 수 있는 언어이다.
5. 더 이상 클라이언트 측만의 기술이 아니다. 서버 측 자바스크립트가 개발(Node.js)되었다.
6. 프로토타입-기반의 언어이다. => 자바스크립트는 상속을 위해 클래스 개념 대신에 프로토타입을 사용한다.
7. 배우기 쉽다.
8. 보안처리에 부적합하다.
   1. 자바스크립트의 장점

자바스크립트의 장점 중 가장 확실한 부분은 유용하다는 점이다.

* 서버 부하 감소
* 동적으로 변경 가능
* 모바일에 그대로 적용
* 높은 동작 신뢰도
  1. ㅇ

1. 내부 자바스크립트
2. 외부 파일에 저장한 후 src속성을 사용해 불러오기
3. HTML 태그 내에 자유롭게 섞어서 사용하는 인라인 방법

자바스크립트를 사용하기 전에 알아둬야 할 주의 사항

* 하나의 HTML 페이지 안에 다수의 스크립트 인스턴스를 사용할 수 없다.
* 하나의 스크립트 태그에서 외부 자바 스크립트 파일도 지정하고 인라인 형태의 코드도 작성할 수 없다.
* 스크립트는 <head>나 <body> 어디에도 올 수 있으며, <body>를 닫기 직전 스크립트를 삽입하는 경우 페이지 로딩 속도를 향상시킬 수 있다.
* 너무 많은 스크립트를 하나의 페이지 안에 삽입할 경우 HTML 페이지의 성능에 영향을 줄 수 있다.
* thow문장

개발자가 자신이 기준을 정하고 이 기준에 맞지 않으면 사용자에게 어떤 경고 메시지를 줄 수 있는데, 이때 사용하는 문장이 throw이다. 개발자는 다음과 같은 문장으로 예외 객체를 발생시킬 수 잇따.

Throw “예외메시지”

1. 자바스크립트의 소개
   1. 기본용어

* 표현식과 문장

값을 만들어내는 간단한

예) 10+20+30\*2

표현식 끝에 ‘;’를 붙이면 문장인된다.

* 식별자

변수명과 함수 명처럼 자바스크립트에서 이름을 붙일 때 사용하는 단어이다. 식별자를 만드는 규칙은 다음과 같다.

* 키워드를 사용할 수 없다.
* 숫자로 시작하지 않는다.
* 특수문자는 \_와 $만 허용한다.
* 공백문자를 포함할 수 없다.
* 대소문자를 구분한다

Alpha, $alpha, alpha10, break 등은 모두 식별자로 사용할 수 있다.

식별자에 대한 개발자 관례는 다음과 같다.

* 생성자 함수의 이름은 항상 대문자로 한다.
* 변수와 인스턴스, 함수, 메서드의 이름은 항상 소문자로 시작한다.
* 여러 단어로 이루어진 식별자는 각 단어의 첫 글자를 대문자로 한다.

식별자의 특징을 정리하면 다음과 같다.

|  |  |  |
| --- | --- | --- |
| 구분 | 단독사용 | 다른 식별자와 사용 |
| 식별자 뒤에 괄호 있음 | 함수 | 메서드 |
| 식별자 뒤에 괄호 없음 | 변수 | 속성 |

* 주석

HTML의 주석은 <!-- -->이고 자바스크립트의 한줄 주석은 //, 여러줄 주석은 /\* \*/이다. 이는 프로그램의 실행내용과는 관계가 없는 내용으로 참고할 내용을 기재한다.

* 1. 출력

자바스크립트의 기본적인 출력 방법은 alert()함수를 이용하거나 document.write 메서드를 이용하는 방법이 있다.

* 1. Df
  2. 정수
  3. 불

참(true,1)과 거짓(false,0)값을 표현하는 자료형이다. 불 관련 비교 연산자는 다음과 같다.

|  |  |
| --- | --- |
| 연산자 | 설명 |
| >= | 크거나 같다 |
| <= | 작거나 같다 |
| > | 크다 |
| < | 작다 |
| == | 같다 |
| != | 다르다 |

* 1. 변수
* 변수의 생성과 사용

변수는 값을 저장할 때 사용하는 식별자이다. 변수를 선언하고, 값을 할당하는 과정을 거쳐 사용하게 되는데 자바스크립트는 변수를 선언하지 않고 사용할 수 있다.

|  |  |  |
| --- | --- | --- |
| var quantity, color;  quantity=20;  color=”blue” //문자열로 초기화 | 은 다음과 같은 의미이다 | var quantity=20;  var color=”blue”;  or  quantity=20;, color=”blue” |

var키워드를 사용하는 경우와 그렇지 않은 경우의 차이는 변수의 유효범위(scope)이다. 즉, 일반적으로 var라고 선언한 변수는 지역변수이며, 그렇지 않은 경우는 전역변수로 사용된다. 함수를 만들어서 사용하는 경우는 함수 외부에 변수를 var로 선언해도 전역변수가 된다.

* 변수의 범위

인터프리터가 구동되면 코드 실행 전 생성하는 객체를 전역객체(global object)라고 한다. 브라우저에서는 window 객체가 전역객체이다. 자바스크립트 프로그램이 실행되면 HTML 엘리먼트, 전역변수, 함수 등이 전역객체(global object)의 속성으로 정의된다.

전역객체(global object)에서 함수가 호출되면 새로운 실행 컨텍스트(호출객체: Call object)를 생성한다. 함수 호출 시 필요한 전달인자, 지역변수와 관련된 컨텍스트들을 저장하고 함수 호출이 완료되면 호출객체는 가비지(Garbage) 대상이 된다. Scope chain은 스택(stack)으로 동작하게 된다.

String:문자열

Boolean:true/false

Object: 객체

Undefined:값이 정해지지 않은 상태

|  |  |
| --- | --- |
| 변수의 타입 | 출력하는 값 |
| number | number |
| string | string |
| Boolean | Boolean |
| 호출이 가능한 객체 | object |
| 호출이 불가능한 객체 | function |
| null | object |
| undefined | undefined |

* 1. 그 이외 연산자
* 복합대입 연산자

|  |  |
| --- | --- |
| 연산자 | 설명 |
| += |  |
| -+ |  |
| \*= |  |
| /= |  |
| %= |  |

* 1. 입력

자바스크립