**[JAVA VS C]**

**1) JAVA**

자바는 썬 마이크로시스템즈의 제임스 고슬링과 연구원들이 개발하여 1996년 1월에 공식적으로 발표한 객체지향 프로그래밍 언어이다.

**자바의 역사**

자바의 역사는 1991년 썬의 엔지니어들에의해서 고안된 오크(Oak)라는 언어에서부터 시작되었다. 제임스 고슬링과 아서 밴 호프와 같은 썬의 엔지니어들의 원래 목표는 가전제품에 탑재될 소프트웨어를 만드는 것이었다. 처음에 C++을 확장해서 사용하려 했지만 부족함을 깨닫고 C++의 장점과 단점을 보완한 새로운 언어를 개발한다. 당시에 여러 종류의 운영체제를 사용하는 컴퓨터들이 통신하는 인터넷이 등장하자 운영체제에 독립적인 Oak가 이에 적합하다고 판단하여 Oak를 자바라는 이름으로 바꾸었다. 이후 자바로 개발한 웹브라우저 핫 자바를 발표하고 다음해 자바의 정식버전을 발표했다. 예전 자바로 작성된 Applet은 정적인 웹페이지에 멀티미디어적인 요소들을 제공할 수 있는 유일한 방법이었기 때문에 많은 인기를 얻고 단 기간에 많은 사용자층을 확보할 수 있었다. 현재는 보안상의 이유로 더 이상 Applet을 지원하지 않고 스마프폰 운영체제인 안드로이드, 소규모 가전제품, 기업 등에 다양하게 사용되어지고 있다.

**자바의 특징**

1. **운영체제에 독립적이다.**

기존의 언어들은 한 운영체제에 맞게 개발되었기 때문에 다른 종류의 운영체제에 적용하기가 힘들었다. 하지만 자바에서는 일종의 에뮬레이터인 자바가상머신(JVM) 통해 가능하다. 자바 응용프로그램은 OS, HW가 아닌 JVM하고만 통신하고 JVM이 자바 응용프로그램으로부터 받은 것을 운영체제가 이해할 수 있도록 변환하여 전달한다. 그러므로 자바는 운영체제에 독립적이고 JVM은 종속적이라고 볼 수 있다. JVM은 여러 운영체제에 설치할 수 있는 서로 다른 버전의 JVM을 제공한다.

1. **객체지향언어**

자바는 객체지향프로그래밍언어 중 하나로 상속, 캡슐화, 다형성이 잘 적용된 객체지향언어이다.

1. **배우기 쉽고 simple하다.**

하나의 소프트웨어가 제작될 때는 제작 비용도 많이 들지만 유지 보수 비용도 어마어마하다. 이러한 점을 고려해 자바는 디자인되었다. 자바의 연산자와 기본구문은 C++에서 객체지향구문은 small talk이라는 객체지향 언어에서 가져왔는데 이들의 장점만 가져오고 혼란을 일으키는 부분은 제거했다. 예를 들면 C++의 다중 상속, 연산자 오버로딩 그리고 기억 장소 경영 문제를 garbage collection을 사용하여 극복하므로 개발자가 신경 쓸 필요가 없어졌다.

1. **Garbage Collection(자동 메모리 관리)**

garbage collector는 자바 프로그램 실행시 자동적으로 메모리를 관리해준다. 따라서 개발자는 따로 메모리영역에 대한 관리할 필요가 없다. 자동으로 메모리를 관리하는 것이 비효율적인 면도 있지만 프로그래머가 보다 프로그래밍에 집중할 수 있도록 도와준다.

1. **네트워크와 분산처리를 지원한다.**

자바는 소스 코드를 컴파일하여 바이트 코드로 만들어내는데 이 바이트 코드는 다양한 플랫폼에 설치된 자바 인터프리터에 의해 해석되기 때문에 어떠한 플랫폼에서도 실행 가능하다.

다양한 네트워크 프로그래밍 라이브러리(Java API)를 통해 네트워크 관련 프로그램을 쉽게 개발할 수 있도록 지원한다.

자바는 원래부터 [네트워크](https://terms.naver.com/entry.nhn?docId=843904&ref=y) [분산 처리](https://terms.naver.com/entry.nhn?docId=849136&ref=y) 환경에서 사용하기 위해 디자인된 언어이다. 네트워크 환경은 다른 환경보다 보안의 측면이 강조되는 환경인 만큼 자바는 보안에 중점을 두고 있다. 자바는 [바이러스](https://terms.naver.com/entry.nhn?docId=842459&ref=y)가 침투하지 못하는 구조를 가지고 메모리에서 [데이터 접근](https://terms.naver.com/entry.nhn?docId=845118&ref=y)을 제한할 수 있다. 접근을 허용하지 않으면, 애플리케이션의 [데이터 구조](https://terms.naver.com/entry.nhn?docId=820011&ref=y) 또는 데이터에 대한 접근은 불가능하다.

1. **멀티쓰레드를 지원한다.**

2개 이상의 쓰레드를 병렬적으로 처리하는 형태를 멀티쓰레드라고 한다. 처리 시간을 분할하여 동시에 수행되는 것처럼 보인다. 멀티쓰레드의 지원은 사용되는 OS에 따라 구현방법도 상이하고 처리하는 방식도 다르다. 그러나 자바에서는 시스템과 관계없이 구현가능하며, 관련된 라이브러리가 제공되므로 구현이 쉽다.

1. **동적 로딩을 지원**

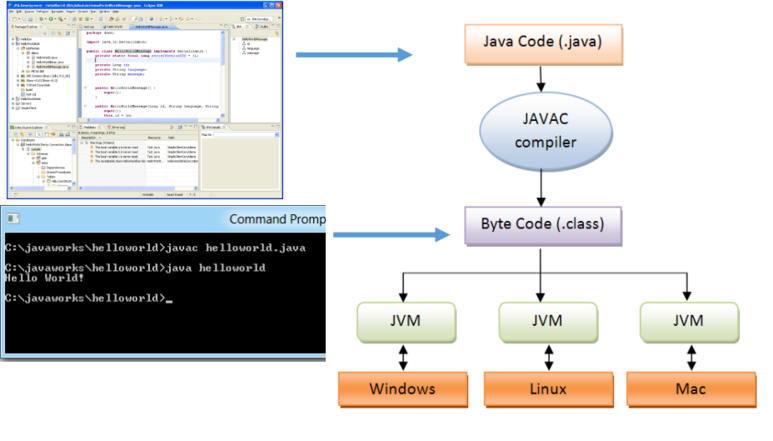
자바로 작성된 어플리케이션은 여러 개의 클래스로 구성되어 있다. 자바는 동적 로딩을 지원하기 때문에 실행 시 필요한 클래스만 로딩하여 사용할 수 있다. 그 외에도 일부 클래스 변경시 컴파일하지 않아도 되며, 변경사항이 발생해도 비교적 적은 작업만으로도 처리할 수 있는 유연한 애플리케이션 작성할 수 있다.

1. **JVM(Java Virtual Machine)**

JVM은 자바 컴파일러가 변환한 바이트코드를 실행시키는 프로그램이다. 또한 자바와 OS 사이에서 중개자 역할을 수행하여 자바가 특정 OS에만 제한되지 않고 재사용을 가능하게 해주는 장점이 있다. JVM은 메모리관리, Garbage Collection을 수행한다.

자바 프로그램을 컴파일한 후의 결과인 바이트코드는 JVM의 해석단계를 거쳐 실행되기 때문에 최초의 가상 머신은 인터프리티 방식만을 써서 실행속도가 느리다는 단점이 있었지만 현재는 JIT 컴파일 도입과 하드웨어 발전으로 성능이 개선되었다.

JVM은 자바 가상 머신이라고 해서 자바 바이트코드만 인식하는 것은 아니다. 자바가 아닌 다른 언어들을 가지고도 생성할 수 있기 때문이다.



**2) C**

벨 연구소에서 1971년경부터 데니스 리치와 켄 톰슨 등에 의해서 설계 개발된 시스템 기술용의 프로그래밍 언어이다. UNIX 오퍼레이팅 시스템의 기술에 사용할 것을 목적으로 설계한 언어로 UNIX OS의 대부분이 이 언어로 개발되었다. [컴퓨터](https://terms.naver.com/entry.nhn?docId=1149677&ref=y)의 구조에 밀착한 기초 기술이 가능한 것과 간결한 표기가 될 수 있는 것 등을 특징으로 하고 있다.

**C언어 역사**

켄 톰슨이 당시의 BCPL 언어를 개조해서 "B" 언어를 만들었고, 데니스 리치가 "B"를 개선해서 "C"라고 명명합니다. 그러나 C 언어에 대한 공식적인 표준이 만들어지지 않은 상태에서 1978년 데니스 리치가 자신의 저서 부록에 C 언어의 문법을 기술했는데 그 부록이 C 언어의 표준 문서 역할을 합니다. 이후 ANSI(American National Standard Institute)와 ISO(International Organization for Standardization)에 의해 다음과 같이 C 언어의 표준이 발표된다.

**C 언어 특징**

1. **시스템 프로그래밍에서 널리 사용된다.**

유닉스 운영체제가 완벽하게 C언어로 작성되었으며, 이후 리눅스 운영체제와 RDBMS인 MySQL 등이 C언어로 개발되었다. 또한 많은 프로그래밍 언어의 컴파일러와 라이브러리가 C언어로 작성되어 컴퓨터 과학의 중추적인 역할을 하고 있다.

1. **시스템 이식성이 높다.**

C언어는 운영체제나 하드웨어에 따라 달라지는 기능들을 언어에 고정적으로 정의해놓지 않고 별도로 런타임 라이브러리에 제공한다. 따라서 특정 플랫폼에서 C언어로 작성한 프로그램이 다른 플랫폼에 이식되어도 라이브러리만 교체되면 되므로 프로그램을 수정하지 않고 사용할 수 있다.

1. **프로그램을 구조화된 형식으로 작성할 수 있다.**

C언어는 함수라는 코드 단위를 기준으로 프로그램을 구성하기 대문에 프로그램을 구성하는 코드들을 재사용하거나 공유할 수 있게 되어 프로그램의 개발은 물론 유지보수가 효율적이다.

1. **프로그램이 작고 실행속도가 빠르다.**

C언어는 기계어와 유사한 기능을 가지고 있어 코드의 실행속도가 빠르다. 모든 프로그램은 최종적으로 0과 1로 구성된 기계어 상태로 실행되기 때문에 기계어에 가까울수록 실행속도가 빠르다.

**Java와 C의 차이점**

1. **절차지향, 객체지향**

C는 절차지향으로 위에서부터 순서대로 읽으면서 실행된다.

자바의 경우 객체지향언어로 클래스라는 개념을 사용하여 공통되는 메소드와 속성을 묶어서 분류하여 유지보수와 가독성이 좋음

1. **컴파일 차이**

C언어는 컴파일러, 자바는 컴파일 후 JVM에 의해 실행된다.

1. **문법의 차이**

기본적인 문법은 비슷하나 배열, ArrayList, 제네릭 등 차이점이 있다.

1. **Garbage Collection**

자바의 경우 메모리의 효율적인 관리를 위해 가비지 콜렉션을 자동으로 실행하지만 C의 경우 추가로 라이브러리를 사용해서 사용해야 함

1. **메모리에 접근하는 방식의 차이**

C언어의 경우 포인터를 사용해서 메모리에 접근한다. 이 경우 실수로 중요한 영역을 건드린다면 치명적인 오류를 발생시킬 수도 있다. 자바의 경우 reference를 사용한다. 직접적인 접근이 아니기 때문에 C언어 비해 안전하다.

**[JAVA VS JAVASCRIPT]**

**자바스크립트란?**

우리가 매일 접속하는 웹사이트는 크게 3가지 요소로 구성된다. ‘HTML(Hyper Text Markup Language)’, ‘CSS(Cascading Style Sheets)’, ‘자바스크립트(Javascript)’다. HTML은 웹페이지의 큰 뼈대를 제공하고, CSS는 색깔이나 글씨체와 같은 디자인 요소를 관리한다. 자바스크립트는 크로스 플랫폼(cross platform), 객체지향 스크립트 언어로 웹페이지의 동작을 담당한다. 예를 들어 자바스크립트를 이용하면 ‘버튼을 클릭하면 밑에 날짜를 보여줘’라는 식의 명령을 내릴 수 있다.

자바스크립트는 사용자와 상호작용할 수 있도록 돕는다. HTML과 CSS와 달리 자바스크립트는 독립적인 컴퓨터 프로그램이라는 측면에서 ‘완전한’ 프로그램 언어다. 코더가 원하는 대부분의 일들을 수행하는 소프트웨어다. 자바스크립트는 루프, 변수, If 문 같은 프로그래밍 툴을 사용할 수 있다. 특정 텍스트를 클릭하면 사라지게 하는 것부터 구글 문서를 위한 오피스 프로그램 같은 완전한 응용프로그램을 생성하는 것까지 전반적인 업무에 사용할 수 있다. 페이지 자체를 새로 띄우지 않으면서 쌍방향 기능을 제공하는 모든 웹사이트는 자바스크립트를 사용한다.

**자바스크립트 역사**

자바스크립트는 1995년 5월에 초창기 브라우저 중 하나였던 넷스케이프에서 일하던 브렌던 아이크(Brendan Eich)가 10일 만에 만들었다. 처음에는 모카(Mocha)로, 이후에는 라이브스크립트(Livescript)로 불렸다. 흥미롭게도 다른 프로그래밍 언어인 자바와는 이름이 비슷함에도 불구하고 아무런 관련이 없다. 넷스케이프는 선(Sun, 자바 언어의 소유권을 가진)으로 부터 자바스크립트라는 이름을 사용할 수 있다는 양해를 받았다. 하지만 이는 자바가 그 당시 인기가 많은 언어였던 관계로 마케팅 전략의 하나였을 뿐이다. 자바스크립트는 숱한 개발 과정을 거쳤고 버전을 바꿔 발표됐다. 최신의 표준 버전(우리가 여기서 배울)은 모든 브라우저에서 호환되고 데스크톱과 모바일 플랫폼 모두에서 사용할 수 있다.

**자바스크립트 특징**

* **웹 사이트를 동적으로 만들 수 있다.**

HTML과 CSS를 들어 본 적 있나요? 웹 사이트를 건물에 비유하면 HTML은 건물의 기본 골격을 담당하고 CSS는 건물의 외부·내부 인테리어를 담당합니다. 즉 HTML과 CSS는 눈에 보이는 외관만을 담당합니다. 따라서 메뉴에 마우스 포인터를 올렸을 때 하위 메뉴가 펼쳐지는 것과 같이 웹 사이트에 움직임을 주기 위해서는 다른 언어를 사용해야 합니다. 이때 사용하는 언어가 바로 자바스크립트입니다.

* **웹 브라우저에서 실행되는 프로그램을 만들 수 있다.**

자바스크립트로 웹을 동적으로 만드는 것뿐 아니라 웹에서 작동하는 프로그램도 만들 수 있습니다. 다음은 웹 사이트에서 버스 정보를 검색하는 프로그램입니다. 이 프로그램의 핵심인 버스 노선이나 정류장 정보를 저장하는 기능부터 지도에 표시하는 기능은 모두 자바스크립트로 만들 수 있죠.

* **서버를 구성하고 서버용 프로그램을 만들 수 있다.**

과거에는 서버 프로그램을 만들기 위해 PHP, ASP.NET, 자바 같은 언어가 필요했습니다. 이 언어를 공부하려면 시간이 아주 많이 필요하죠. 이제는 자바스크립트만 알아도 서버 프로그램을 만들 수 있습니다. 예를 들어 노드제이에스(Node.js) 자바스크립트 프레임워크를 사용하면 서버 프로그램을 쉽게 개발할 수 있습니다.

**자바스크립트 장단점**

* 자바스크립트는 컴파일 과정이 없기 때문에 다른 언어와 비교했을 때 빠른 시간안에 스크립트 코드를 작성할 수 있다.
* C, 자바와 달리 단순한 구조와 원칙을 가지고 있기 때문에 초보자들이 쉽게 배우고 이해할 수 있다.
* 자바스크립트는 웹에 특화된 기술이기 때문에 운영체제나 플랫폼에 상관없이 잘 적용되고 확장성도 높다.
* 단점은 성능이나 보안 측면이다. 일단 내부에서 제공되는 기능이 제한적이고, 관련된 개발도구도 적은 편이다. 또한 자바스크립트는 HTML 소스코드와 함께 작성되면서 외부로 공개되어지는데 이 과정에서 보안문제가 발생할 수 있다.

**자바와 자바스크립트 차이점**

|  |  |
| --- | --- |
| **자바** | **자바스크립트** |
| 객체지향언어 | 객체지향언어(상속, 클래스는 존재하지 않음) |
| HTML과 독립적으로 사용가능 | HTML과 함께 사용 |
| 스트롱 타입(항상 변수형을 선언해야함) | 루즈 타입(변수형 선언할 필요가 없다.) |
| 정적 연결(static binding)을 취하여 컴파일시 객체에 대한 참조가 이루어짐 | 동적 연결(dynamic binding)이기 때문에 객체에 대한 참조는 실행시에만 함 |
| 컴파일 과정을 거친다. | 인터프리터 언어로서 컴파일 과정을 거치지 않는다. |
| 서버측 언어 | 클라이언트용 언어 |
| 하드 디스크에 자동으로 작성 가능 | 하드 디스크에 자동으로 작성 불가 |