

2023-11-02

언어 → 컴퓨터에서 사용자가 원하는 결과를 만들어내기 위한 기계적, 프로그램적 작동 명령의 집합된 CODE

저급 → 기계어 (0, 1)/ 컴퓨터만 사용한다.

고급(자연어) → 기호 등을 사용하는 것과 언어만 사용하는 것으로 나뉜다.

자연어 → 인간이 사용하는 언어 (영어)

기호 사용 → 코볼(사무), 파스칼(과학), 포트란(수학, 통계)

언어만 → C

user ———> 컴파일러 ———> 기계어

coding

컴파일(기계어로 번역함)

C#, C++, Java, Python은 자연어(영어)로 이루어져 있어서

자연어를 기계어로 바꿔줘야 한다.

(컴파일은 한번에 함)

초보 BASIC ———> 인터프리터 ———> 기계어

컴파일러와 인터프리터의 차이 - 줄 단위 번역, 실시간

오류 찾기가 쉽다.

번역 실행속도가 느리다.

프로그램 전체 제어의 오류가 있는것을 찾기 어렵다.

```
> a=3;
< 3
> a=5:
✖ Uncaught SyntaxError: Unexpected token ':' VM67:1
> a=5;
< 5
>
```

인터프리터

[언어의 구조]

1. 구조적 언어 : C, 포트란, 코볼, 파스칼 - 문법이 매우 어렵다. (시스템 프로그램 개발)
2. 객체 지향 : C++, Java - 상속의 개념을 갖는다. (부모, 자식 - 클래스)
3. 스크립트 : C#, Python, Javascript, ~~.js - 컴파일러가 없고 인터프리터 한다.
 - a. Java Script → java의 자료형이 없이 간략화 한 언어다.

package.json

```

{
  "name": "javascript",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "devDependencies": {
    "eslint": "^8.52.0"
  }
}

```

Handwritten notes:
 - "name": "javascript" → 프로젝트 명
 - "version": "1.0.0" → 버전
 - "description": "" → 설명
 - "main": "index.js" → 프로젝트 실행시 기본 파일
 - "scripts": { ... } → 반복 실행할 명령어
 - "keywords": [] → 태그 (안내태그를 붙이는 자리)
 - "author": "" → 작성자
 - "license": "ISC" → 라이선스 정보
 - "devDependencies": { ... } → 라이브러리 목록 (작업을 도와주는 툴의 집합체)

Package.json의 내용

"main": "index.js" - 프로젝트 실행시 기본파일 (알맞게 이름 변경 해줘야 함)

debug - 오류시 어떻게 할건지

scripts - 반복

keywords - 태그 (안내태그를 붙이는 자리)

author - 작성자

"license": "ISC" - 라이선스 정보

devDependencies → 라이브러리 목록 (작업을 도와주는 툴의 집합체)

변수명 ← 1 a=1; → 변수값.
 2 b=3;
 3 console.log("a=" + a);
 4 console.log("b=" + b);
 5 console.log(a+b);

상수 ⇒ 변하지 않는 값
 변수 ⇒ 변하는 값

```
for (let i = 0; i < 10; i++)  
  console.log(i);
```

← 시작
 ← 시작값
 ← 종료값
 ← 증가값
 1 for (let i = 0; i < 10; i++) → ①씩 더하기
 2 console.log(i); | 0~9 배후반
 Java int i = 0;
 자바스크립트 ← let i = 0;
 ++i
 전위증가

for에 ;를 찍으면 돌기전에 끝나서 오류가 뜬다.

for - 반복
let i = 0; - 시작값
i < 10; - 종료값
i++ - 증가값(1씩더하라)

javascript에는 자료형이 존재하지 않는다.

i = x

j = y 로도 사용 됨

> 크다, 초과

>= 크거나 같다 이상

<= 작다 미만

<= 작거나 같다 이하

== 같다

i= 다르다

```
1 let s="hello papawon";
2 for(let i=0; i<s.length; ++i) {
3   let ch = s[i];
4   console.log(ch);
5 }
```

} - for 문이 돌아갈때 실행 할 내용

```
let i=0;
for (i=0; i<10; ++i){
    console.log(i);
}
```

```
let j=0;
while (j<10) {
    console.log(j);
    ++j;
}
```

```
let k=0;
do {
    console.log(k);
    ++k;
} while (k<10);
```

for, while, do while - 반복문

모든 언어에서 공통적으로 나옴

반복문을 돌리기위한 조건문도 필수다.

```
let s =0;
for(let i=1; i<=55; i++){
    s=s+i;
}
console.log(s);

/*
let s=0;
for (let i=1; i<=100; i++){
    if(i%3 ==0)
        s=s+i
};
console.log(s);
```

```
let a = 3,
    b = "3";
console.log(a);
console.log(b);
console.log(a === b);
console.log(a !== b);
```

java 자료형 (data type)

1. 기본 (prinitive type)

number - 수치

string - 문자

booleam - 불(true, false)

undefind

2. 복잡 (complex type)

function → 기능 → 실행 문장

object → 객체 → 자료, 처리 방법

```
let a=3.141592;  
console.log(a);  
console.log(typeof a);  
  
let s=a.toString();  
console.log(s);  
console.log(typeof s);  
  
let n=Number(s);  
console.log(n);  
console.log(typeof n);
```

숫자 형태 → 문자 형태 → 숫자 형태로 바꿀 수 있다.

```
1
2 <!doctype html>
3 <html lang="en" data-bs-theme="auto">
4   <head><script src="/docs/5.3/assets/js/color-modes.js"></script>
5
6     <meta charset="utf-8">
7     <meta name="viewport" content="width=device-width, initial-scale=1">
8
9     <title>papawon</title>
10
11   </head>
12   <body>
13
14   </body>
15 </html>
16
```

```
for (let i=0; i<=29; ++i){
  let a = Math.floor(Math.random()*11+10)
  console.log(a);}
```