

231102

🕒 작성 일시	@2023년 11월 2일 오전 9:42
🏷 태그	

언어⇒ 컴퓨터에서 사용자가 원하는 결과를 만들어 내기 위해

기계적, 프로그램적 작동 명령에 집합된 CODE

저급→ 기계어(0.1) → 컴퓨터만 사용한다

고급(자연어) - 인간이 사용하는 언어 ⇒ 영어

1. 기호 등을 사용 <코볼 = 사무용, 파스칼 = 과학, 포트란 = 수학, 통계>
2. 언어만<C>

OS : C언어 베이스로 제작한

user —coding → 컴파일 —compile → 기계어(0.1)(1.0.0)

coding : c / c++ / java / pyhon (자연어(글자) ⇒ 영어)

compile : 자연어 → 기계어 (한번에 번역)

진입 장벽이 높다

진입 장벽이 낮다

Basic → 인터프리터 → 기계어

인터프리터 : 줄 단위 번역(실시간), 오류를 찾기 쉽다.

↳ 용량이 커질 경우 : 소요 시간이 오래 걸리고 오류 찾기가 어렵다.

```
> a=5;
< 5
> a=5:
✖ Uncaught SyntaxError: Unexpected token ':'
> a+a
< 10
```

언어의 구조

1. 구조적 언어 : C, 포트란, 코볼 : 문법이 매우 어렵다. (시스템 프로그램 개발)

2. 객체 지향 언어 : C++, Java, : 상속 ⇒ 부모, 자식 [ex)PK, FK] 클래스
3. 스크립트 언어 : C#, Python, Java script, ~.Js :컴파일러가 없다, 인터프리터

Java script : Java의 자료형이 없이 간략화한 언어

JS save 하지 않을 시 터미널에 적용 되지않는다

```

JS exm01.js
1  a=1;
2  b=3;
3  console.log("a=" +a);
4  console.log("b=" +b);
5  console.log(a+b);

터미널

PS C:\work\javascript> node exm01.js
a=1
b=3
4

```

크롬 consol 에서 적용가능

```

> a=1;
   b=3;
   console.log("a=" +a);
   console.log("b=" +b);
   console.log(a+b);

a=1
b=3
4

```

크롬에는 자바스크립트를 인식
할수 있는 인터프리터 내장

a=1	VM41:3
b=3	VM41:4
4	VM41:5

package.json

```

{
  "name": "javascript", 프로젝트명
  "version": "1.0.0", 버전
  "description": "", 설명
  "main": "index.js", 프로젝트 실행시 처음 기준 파일
  > Debug 오류 발생시 처리
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [], TAG
  "author": "", 작성자
  "license": "ISC", 라이선스 정보
  "devDependencies": { 라이브러리 목록
    "eslint": "^8.52.0" (작업을 도와주는 들의 집합체)
  }
}

```

	반복	시작값	종료값	증가값	
1	for	(let i = 0;	i < 10;	i++)	1씩 더하라
2	console.log(i);		0~9	배후법	
java int i = 0; (int= 자료형) java let i = 0; javascript에는 자료형이 존재 하지 않는다			++i		i=x j=y
			전역법		

- > 크다 초과
- >= 크거나 같다 이상
- < 작다 미만
- <= 작거나 같다 이하
- == 같다
- != 다르다

기준: 왼쪽

a > b a가 b보다 크다

b < a b가 a보다 작다

```
JS exm01.js > ...      변수명
1  let s="hellow papawon";   갯수를 기억해라
2  for(let i=0; i<s.length; ++i){   {}기준값이
3      let ch = s[i];   i부터 시작   돌아 갈때 실행 할 내용
4      console.log(ch);
5  };
```

```
JavaScript
1  let s="hellow papawon";
2  for(let i=0; i<s.length; ++i){
3      let ch = s[i];
4      console.log(ch);
5  };      ||
6  JAVA
7  public class for {
8      public static void main(string[] args){
9          string s="hellow papawon";
10         for(int i=0; i<s.length(); ++i){
11             char ch = s.charAt(i);
12             System.out.println(ch);
13         }
14     }
15 }
```

자바 스크립트

반복문 : for, while, do-while

```

7   let i=0;
8   for (i=0; i<10; ++i)
9       console.log(i);
10
11
12  let j=0;
13  while (j<10){
14      console.log(j);
15      ++j;
16  }
17
18  let k=0;
19  do {
20      console.log(k);
21      ++k;
22  } while (k<10);

```

→

```

0
1
2
3
4
5
6
7
8
9
0
1
2
3
4
5
6
7
8
9

```

가장 직관적

```

let i=0;

for(i=0; i<100; ++i){
    if(i%2 ==1) continue; /* if조건 %나누기 continue 이어서 작업*/
    console.log(i);
    if(i>=20) break;      /* i의 값이 20이면 break멈처라 */
}

```

```

let i=0;
for( ; ; ){
    ++i;
    if(i%2 ==1) continue;
    console.log(i);
    if(i>=20) break;
}

```

PS C:\work\javascript> node exm01.js

```

0
2
4
6
8
10
12
14
16
18
20

```

```
let s1="papawon";  
const s2="papawon"; /* CONST 사용시 동일 값이어도 변화 없음 */
```

```
console.log(s1)  
console.log(s2)
```

```
PS C:\work\javascript> node exm01.js  
papawon  
papawon
```

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

||

```
let sum=0;    50까지 수를 전부 더한 값  
let l=0;  
for(l=1; l<=55; l++){  
    sum += l    변경 될때마다  
}              sum에 더한다  
console.log(sum);
```

```
PS C:\work\javascript> node exm02.js  
1540  
1540
```

```

100이하의 3의 배수의
let a = 0; 수를 더한값
for(i=1; i<=100; i++){
  if(i%3 ==0) {a= a+i;}
} console.log(a);

PS C:\work\javascript> node exm02.js
1683

```

-자바스크립트 ⇒ 약타입 언어:자바 스크립트는 자료형에 대하여 엄격하게 구분 하지 않는다

<pre> console.log(typeof a); console.log(typeof b); console.log(typeof c); console.log(typeof d); </pre>	<pre> PS C:\work\javascript> node exm03.js number 자료 타입 확인 string boolean undefined 자료 지정 X </pre>
--	--

```

JS exm03.js > ...
1  let a=3, b="3";
2  console.log(a);
3  console.log(b);
4  console.log(a === b); /* == 엄격하게 X ===엄격하게O 확실하게 구분하여야 할때*/
5  console.log(a !== b); /* != 엄격하게X !== 엄격하게O 확실하게 구분하여야 할때*/

PS C:\work\javascript> node exm03.js
3
3
true      == !=의 경우
false

PS C:\work\javascript> node exm03.js
3
3
false     === !==의 경우
true

```

Java 자료형 (DATA Type)

1.기본(primitive type)

number 수치

string 문자

boolean 불(True,False)

undefined

2.복잡 (complex type)

function ⇒ 기능 ⇒ 실행 문장

object ⇒ 객체 ⇒ 자료,처리 방법의 모음

```
a=3, b="3", c=true,  
console.log(a== c);  
false
```

```
let a = 3.141592;  
console.log(a);  
console.log(typeof a);  
  
let s=a.toString() /* toString문자열로 변경 해라 */  
console.log(s);  
console.log(typeof s);
```

PS C:\work\javascript> node exm03.js
3.141592
number
3.141592
string

```
let n=Number(s); /* Number 숫자열로 변경해라 */  
console.log(n);  
console.log(typeof n);
```

3.141592
number

부트스트랩 예시


```

for(let a=0; a<30; ++a){ /* 10 ~ 20 사이에 수를 30개 나오게 */
    let b =Math.floor(Math.random()*11+10)
    console.log (b);
}

```

```

JS exm04.js > ...
1  for(let i=0; i<5; ++i ){ /* 시작값; 종료값; 증가값;*/
2      let a = Math.floor(Math.random()*3+1) /* 10일경우 1자리 100일 경우 2자리*/
3      console.log (a); /* 임의의 수를 발생시킨다. (난수) */
4  }

```

가치수, 최소값

```

PS C:\work\javascript> node exm04.js
2
2
3
1
1

```