

Online Decompiler ver.2

Developing~! 😊

2019.07.08
김두형, 김성준

Clean JS & Decompiler

- 김두형 : 웹 플랫폼 개발, Clean JS 개발
- 김성준 : 디컴파일 작동원리 및 로직 개발

- 1. Clean JS

- Javascript 보기 쉽게
- React.js 사용
- 타 플랫폼과 비교
- 프론트 개발 후 시작

- 2. Assembly Decompiler

- Assem -> C
- APK -> Java
- 디컴파일 원리 이해
- 변환 원리
- 토큰
- 개발 시작

Clean JS

Online JavaScript Beautifier (v1.10.0)

Beautify, unpack or deobfuscate JavaScript and HTML, make JSON/JSONP readable, etc.

All of the source code is completely free and open, available on [GitHub](#) under MIT licence, and we have a command-line version, python library and a [node package](#) as well.

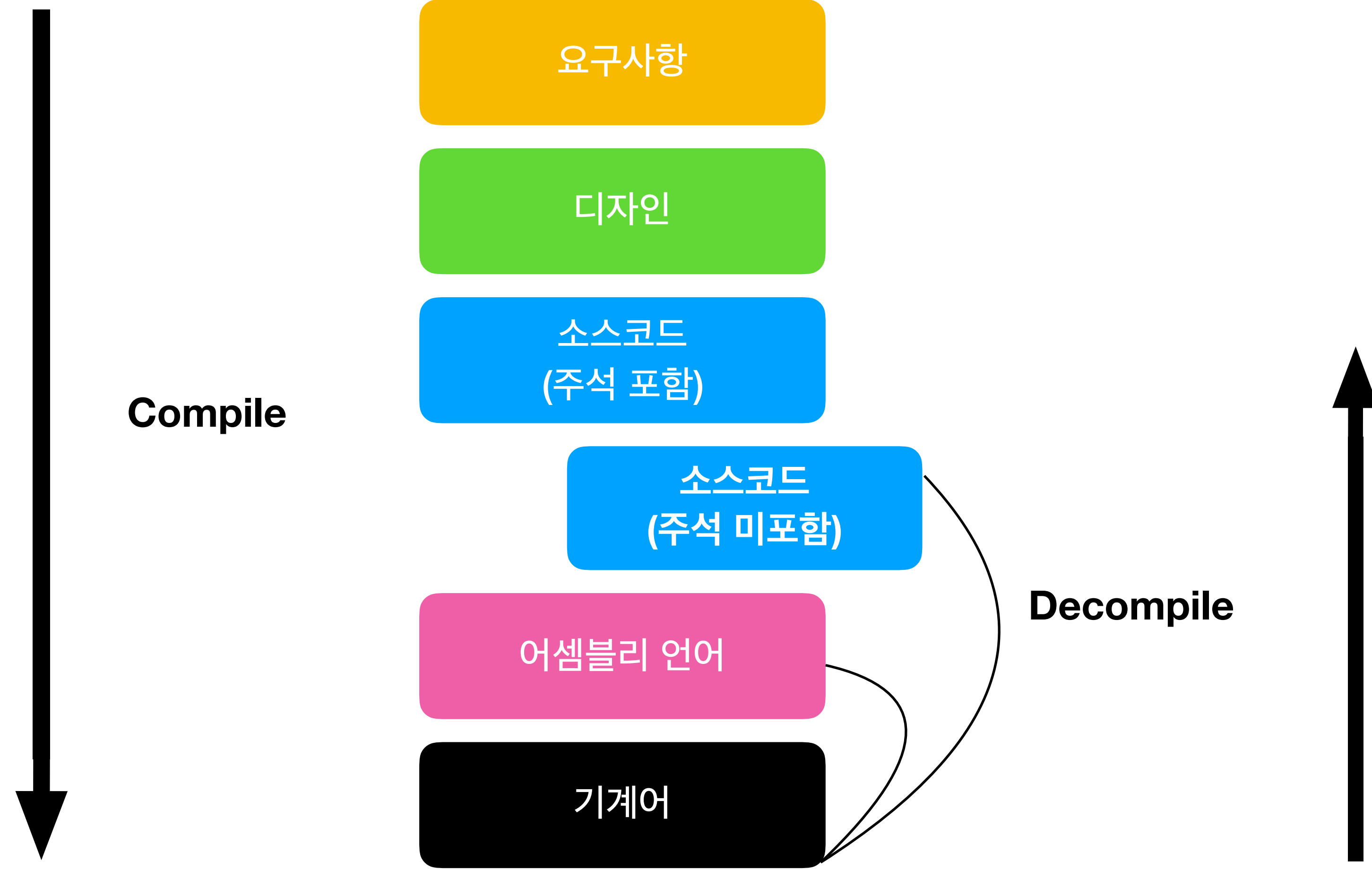
Beautify Code (ctrl-enter)

```
1 // This is just a sample script. Paste your real code (javascript or HTML) here.
2
3 if ( 'this_is'==/an_example/){of_beautifier();}else{var a=b?(c%d):e[f];}
```

```
1 // This is just a sample script. Paste your real code (javascript or HTML) here.
2 if ( 'this_is' == /an_example/ ) {
3   of_beautifier();
4 } else {
5   var a = b ? (c % d) : e[f];
6 }
```

- 중간길이의 코드는 깔끔하게 정리해주지만, 긴코드는 정리가 안되어있는부분이 많음
- Javascript에 한정되어있음
=> 프레임워크의 소스들도 포함시킬 생각
- 거의 Node.js를 사용
- & 만들면 당연히 오픈소스
- 기존의 디자인보다 깔끔하게 할 생각

Rev




```
#include <stdio.h>

int main() {
    int i;
    for ( i = 0; i < 10; i++)
        printf("%d", i);
    return 0;
}
```

```

01251820  push      ebp
01251821  mov       ebp,esp
01251823  sub       esp,0CCh
01251829  push      ebx
0125182A  push      esi
0125182B  push      edi
0125182C  lea       edi,[ebp-0CCh]
01251832  mov       ecx,33h
01251837  mov       eax,0CCCCCCCCh
0125183C  rep stos  dword ptr es:[edi]
0125183E  mov       ecx,offset _ACC03E1B_主主@cpp (0125C003h)
01251843  call      @__CheckForDebuggerJustMyCode@4 (01251217h)
-----
01251848  mov       dword ptr [i],0
0125184F  jmp       main+3Ah (0125185Ah)
01251851  mov       eax,dword ptr [i]
01251854  add       eax,1
01251857  mov       dword ptr [i],eax
0125185A  cmp       dword ptr [i],0Ah
0125185E  jge       main+53h (01251873h)
-----
01251860  mov       eax,dword ptr [i]
01251863  push      eax
01251864  push      offset string "%d" (01257B30h)
01251869  call      _printf (0125104Bh)
0125186E  add       esp,8
01251871  jmp       main+31h (01251851h)
01251873  xor       eax,eax

```

```
0101010101010111101011011000101010100101010101010010101  
101010101010101011110101101100010101010010101010101001010  
001010101010101011110101101100010101010010101010101001010  
101000111111111111111100000000101010100000011001010101  
0101010101010111101011011000101010100101010101010010101  
101010101010101011110101101100010101010010101010101001010  
001010101010101011110101101100010101010010101010101001010  
010101010101011110101101100010101010010101010101010010101  
101010101010101011110101101100010101010010101010101001010  
001010101010101011110101101100010101010010101010101001010  
1010001111111111111111100000000101010100000011001010101  
010101010101011110101101100010101010010101010101010010101  
101010101010101011110101101100010101010010101010101001010  
001010101010101011110101101100010101010010101010101001010  
010101010101011110101101100010101010010101010101010010101  
101010101010101011110101101100010101010010101010101001010  
001010101010101011110101101100010101010010101010101001010  
1010001111111111111111100000000101010100000011001010101  
010101010101011110101101100010101010010101010101010010101
```