bunseokbot@UpRoot

- 세종대학교 정보보호학과 15
- UpRoot Core-System Developer
- 비오비 3기 포렌식
- EnScript 로 집체교육때 과제 이득 몇번 본 적 있음

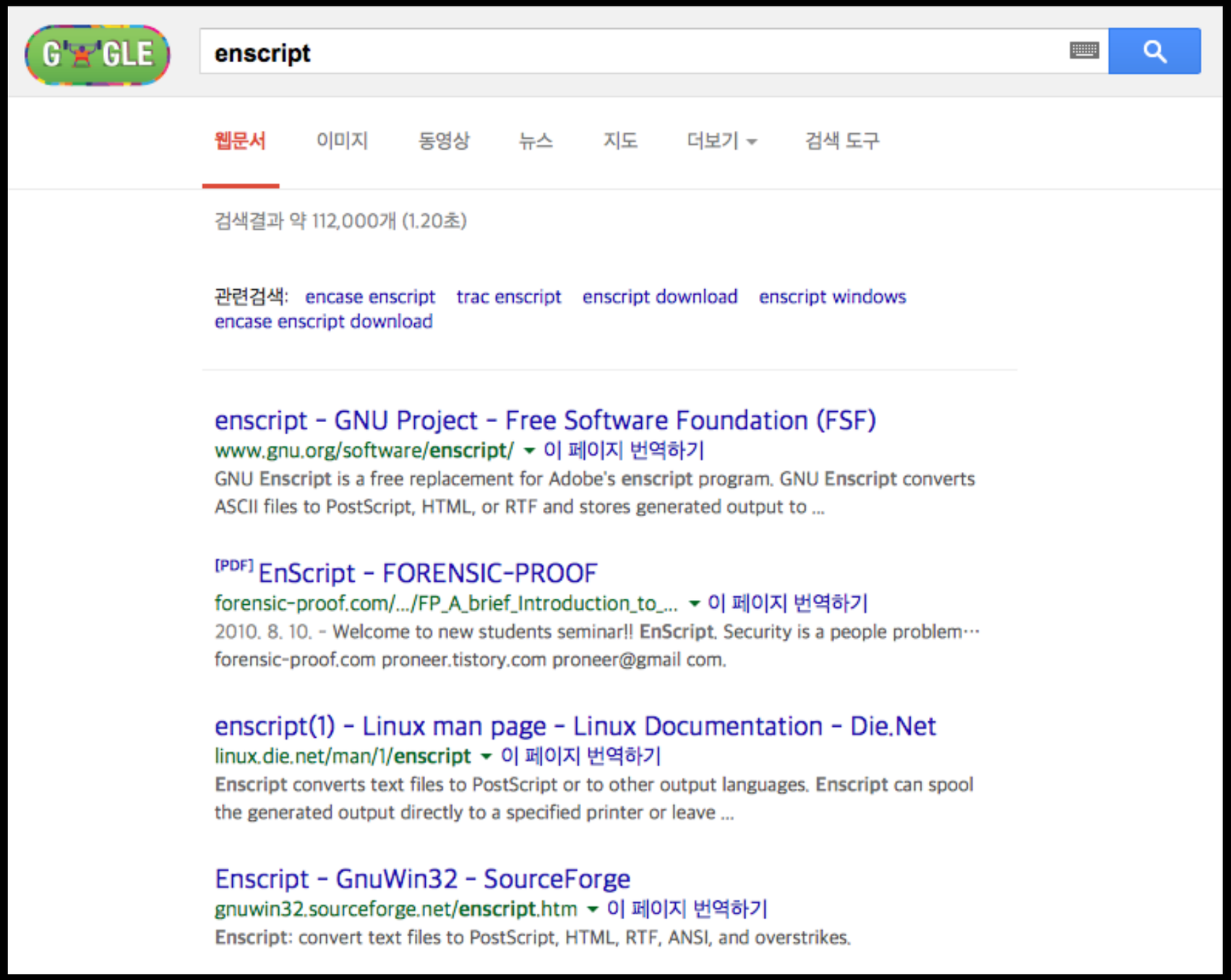
EnScript?

- runnable script language specified for “EnCase Forensic Tool”
- but, not.. like python, ruby..
- most likely C#..? (Why “script”?)

Why EnScript?

- easy to extract “EVIDENCE”
- User-ability
- No-Ga-Da work to script?!

Search from Google..?



Search from Google..?

enscript

웹문서 이미지 동영상 뉴스 지도 더보기 검색 도구

검색결과 약 112,000개 (1.20초)

관련검색: encase enscript trac enscript enscript download enscript windows encase enscript download

enscript - GNU Project - Free Software Foundation (FSF)
www.gnu.org/software/enscript/ ▾ 이 페이지 번역하기
GNU Enscript is a free replacement for Adobe's enscript program. GNU Enscript converts ASCII files to PostScript, HTML, or RTF and stores generated output to ...

[PDF] EnScript - FORENSIC-PROOF
forensic-proof.com/.../FP_A_brief_Introduction_to_... ▾ 이 페이지 번역하기
2010. 8. 10. - Welcome to new students seminar!! EnScript. Security is a people problem…
forensic-proof.com proneer.tistory.com proneer@gmail com.

enscript(1) - Linux man page - Linux Documentation - Die.Net
linux.die.net/man/1/enscript ▾ 이 페이지 번역하기
Enscript converts text files to PostScript or to other output languages. Enscript can spool the generated output directly to a specified printer or leave ...

Enscript - GnuWin32 - SourceForge
gnuwin32.sourceforge.net/enscript.htm ▾ 이 페이지 번역하기
Enscript: convert text files to PostScript, HTML, RTF, ANSI, and overstrikes.

찬양하라

Umm..

« Back to Courses

CPE Credits:	32**
Course Level:	Expert
Course Type:	Recommended
Delivery Method:	Group-Live, Classroom
Tuition:	$2,995.00 USD*

Prerequisite:

Expert

EnCase® EnScript® Programming

This hands-on course introduces the student to the EnScript language, which is designed to allow users to fully tap into the data processing power of EnCase® software (EnCase), automate tasks, and create fully functional applications that can be shared with other EnCase® users. The class is designed for students who have fundamental programming skills and wish to enhance their investigative techniques through the use of EnScript programming.

Instructors and students will write EnScript® applications together. Practical exercises will be used to reinforce the tuition given during the course. Students will leave with the ability to write intermediate-level EnScript® programs that automate searching, interpretation, extraction, bookmarking, and external reporting of data encountered during the examination of computer systems.

교육생 40명 한달 교육비

However?

우리는 “차세대 보안리더” 이다

아 시바 이거 말고..

우리는 언제나 답을 찾을 것이다.

Pull requests Issues Gist

Search

enscript encase

Search

Repositories 5

Code 60

Issues

Users

Languages

C# 1

Advanced search Cheat sheet

geoffblack/ **EnScript**

C# ★5 ⑂3

EnScripts for... **EnCase**

Updated on May 18

jhellin/ **EnScript**

★0 ⑂0

EnScript is a proprietary programming language and application programming interface (API) that exists within the **EnCase** program environment, which means **EnCase** must be running to run **EnScripts**. **EnScript** adheres to the ANSI C++ and Java standards for

Updated on Feb 4, 2014

kevthehermit/ **EnScripts**

★0 ⑂0

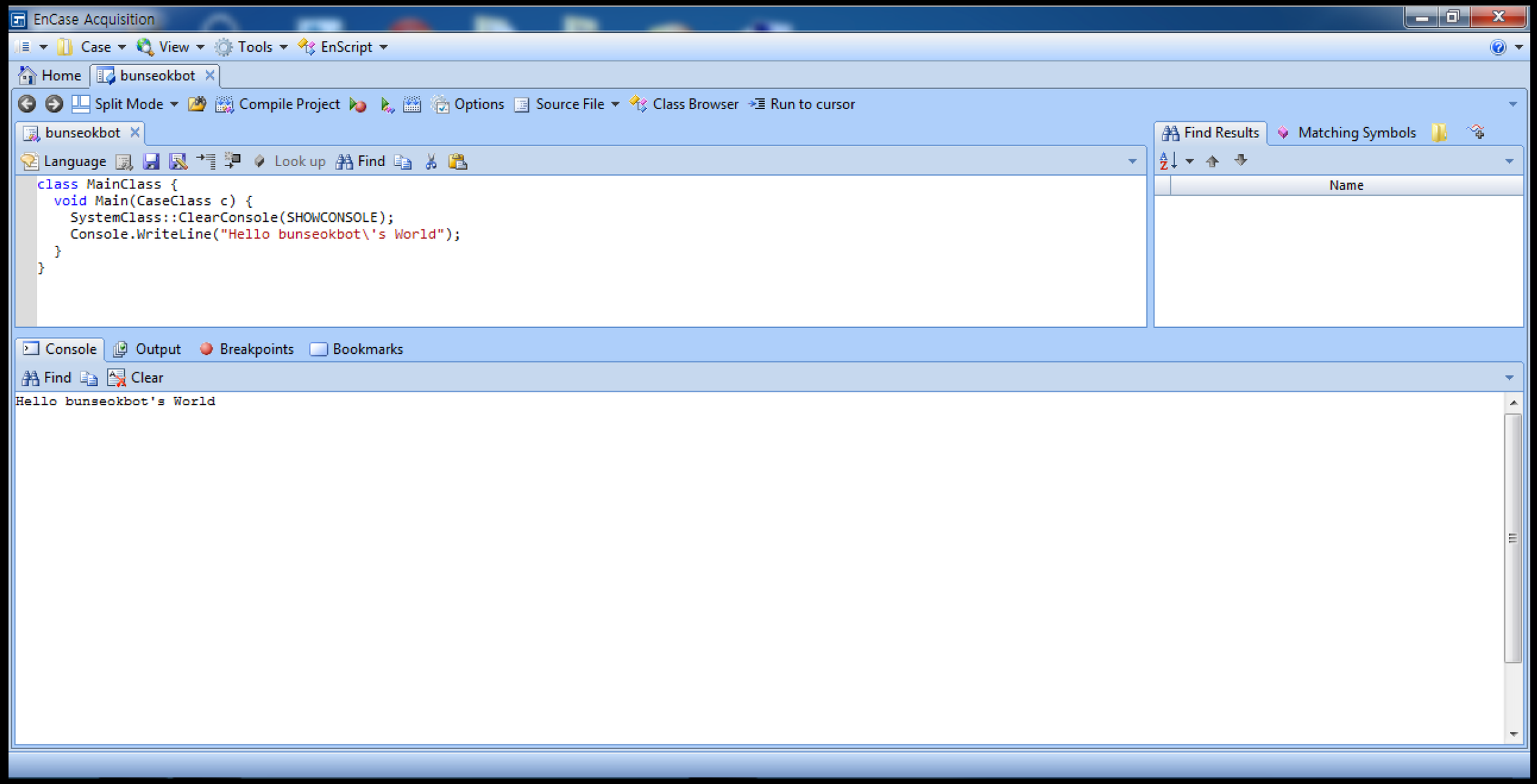
EnCase Scripts

Updated on Sep 28, 2014

EnScript 코드 구경

```
class MainClass {

    void Main(CaseClass c) {

        SystemClass::ClearConsole(SHOWCONSOLE);

        Console.WriteLine("Hello bunseokbot\'s World");

    }

}
```

Running..


EnCase Acquisition
Case
View
Tools
EnScript
Home
bunseokbot
Split Mode
Compile Project
Options
Source File
Class Browser
Run to cursor
bunseokbot
Language
Look up
Find
class MainClass {
void Main(CaseClass c) {
SystemClass::ClearConsole(SHOWCONSOLE);
Console.WriteLine("Hello bunseokbot\'s World");
}
}
Find Results
Matching Symbols
Name
Console
Output
Breakpoints
Bookmarks
Find
Clear
Hello bunseokbot's World

Code Review

```
class MainClass { //entry class
        void Main(CaseClass c) { //main function of class
                SystemClass::ClearConsole(SHOWCONSOLE); //clear the console
                Console.WriteLine("Hello bunseokbot₩'s World”); //write to the console
        }
}
```

Code Review

```
class MainClass { //entry class
        void Main(CaseClass c) { //main function of class
                SystemClass::ClearConsole(SHOWCONSOLE); //clear the console
                Console.WriteLine("Hello bunseokbot₩'s World”); //write to the console
        }
}
```

Time is most important!

```
class MainClass {

        void Main(CaseClass c) {

                SystemClass::ClearConsole(SHOWCONSOLE);

                DateClass date;
                date.Now();
                TimeClass tc(date);

                Console.WriteLine(tc.GetDate());

        }

}
```

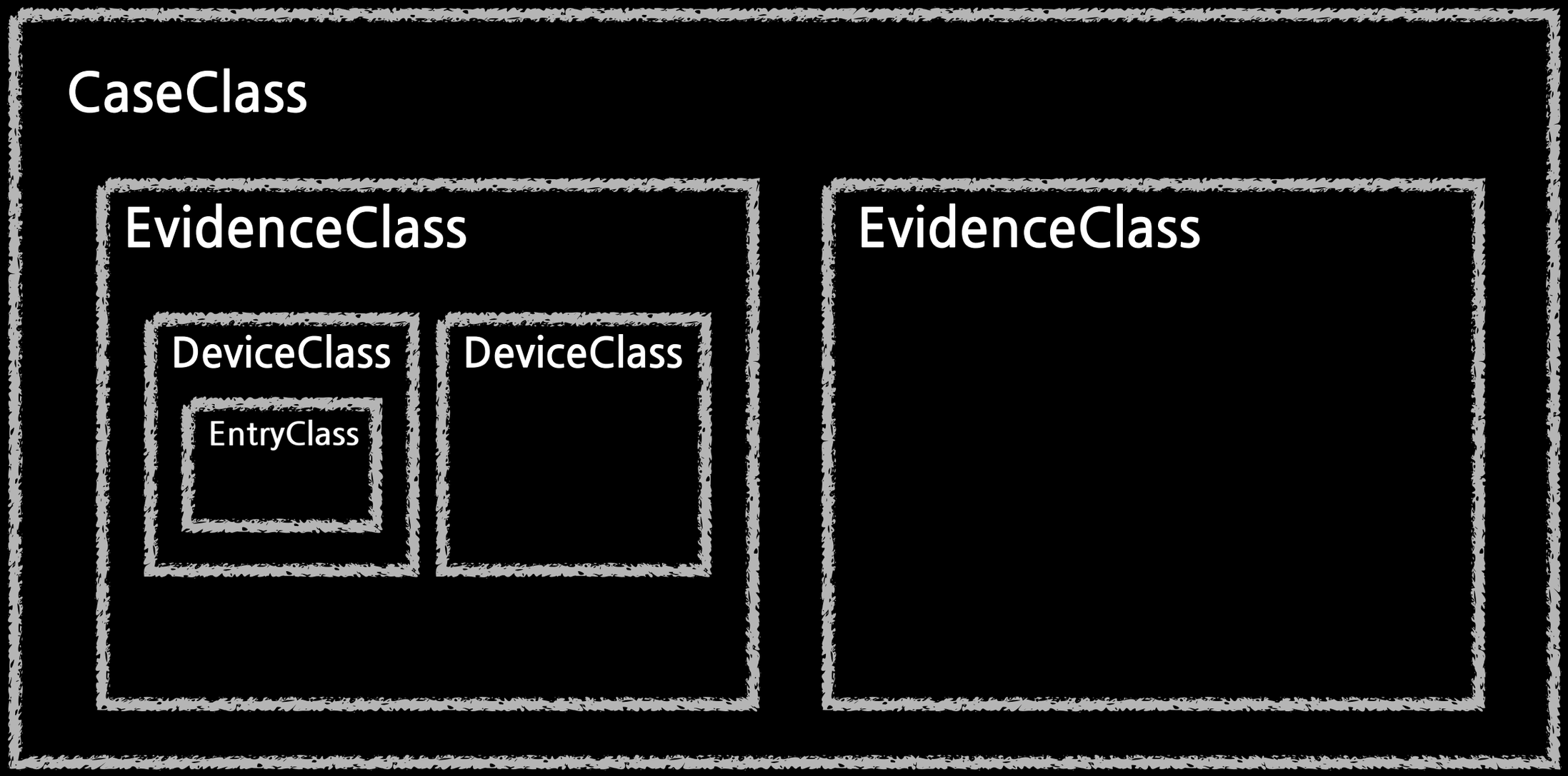
Time is most important!

```
class TimeClass {

  DateClass Date;

  TimeClass(DateClass date = DateClass::Null) :
    Date = date { }

  String GetDate() {
    if( Date != DateClass::Null) {
      return String::Format("Current time is {0}", Date.GetString());
    } else {
      return "Invaild DateTime";
    }
  }
}
```

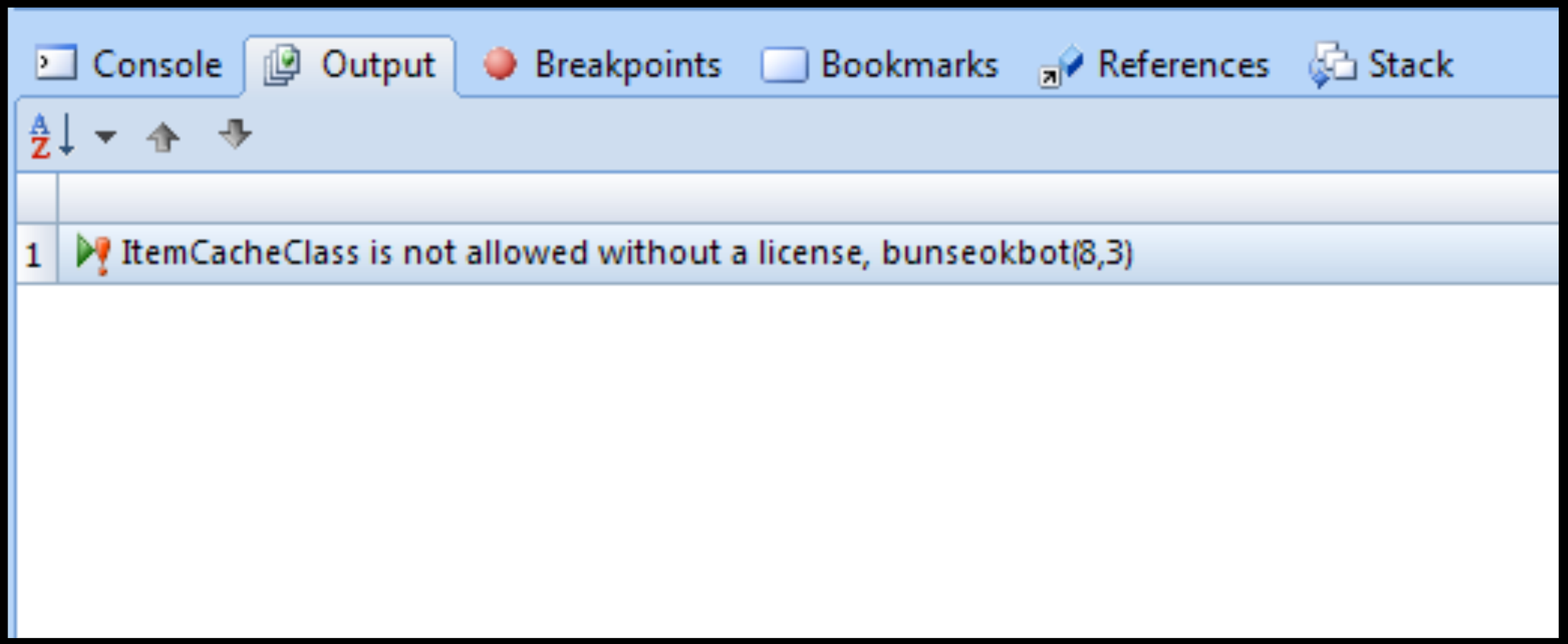
I want to handle the Devices!

```
class MainClass {
        void Main(CaseClass c) {
                foreach(EvidenceClass ec in c.EvidenceRoot()) {
                        Console.WriteLine(ec.Name());
                        DeviceClass dc = new DeviceClass();
                        dc = ec.GetDevice(c, new EvidenceOpenClass);
                        forall(EntryClass entry in dc.GetRootEntry()) {
                                Console.WriteLine(entry.Name());
                        }

                }
        }
}
```

DeviceClass, EvidenceClass



그러나 쓰지 못하는 현실이여..

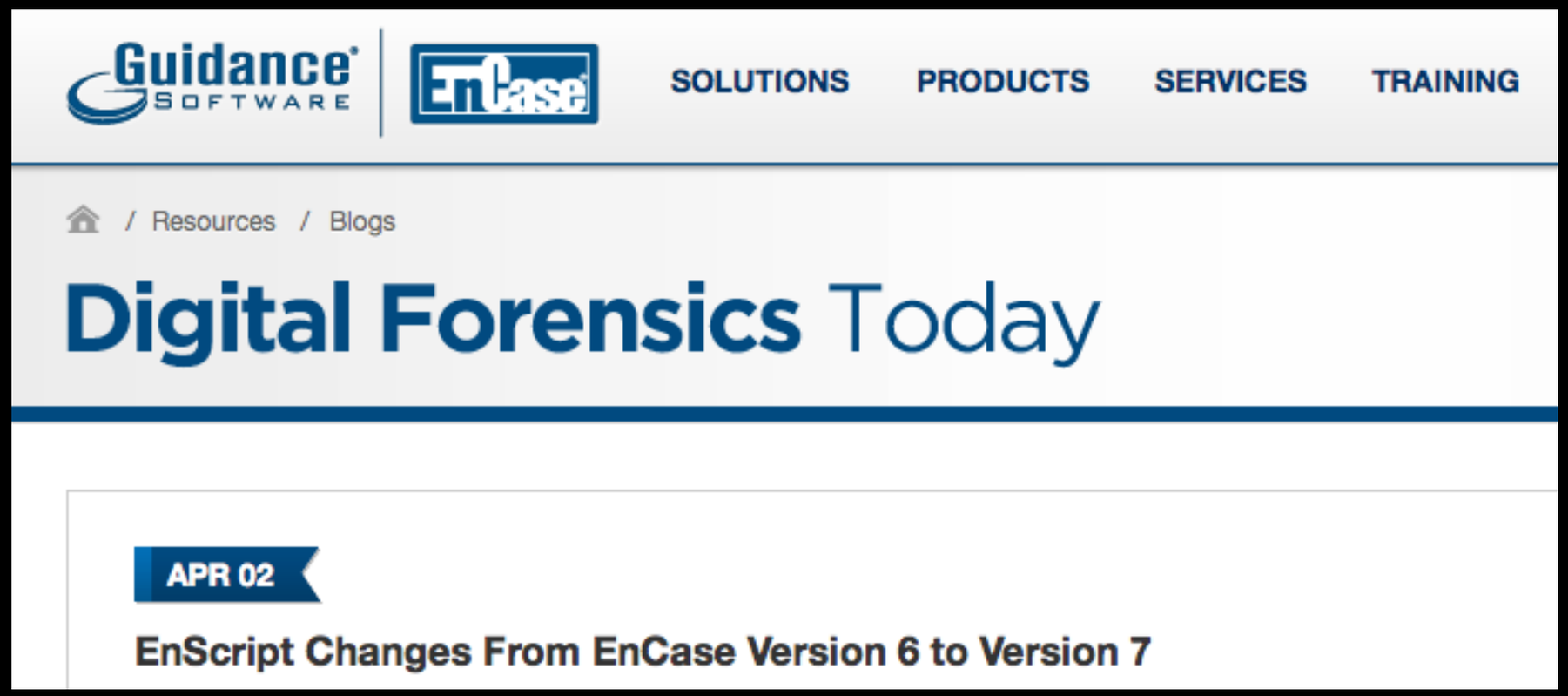


하하하하하하하하하하할

실습할꺼면 워룸 컴퓨터로 하세요

인증 하하하하하하하하하하황ㅎ ㅎ ㅎ ㅎ
아니면 연구원님 멱살잡..고.. 협박하면 가능할겁니다

우리는 언제나 답을 찾을 것이다.



엔스크립트 문법 변함

EnScript Changes From EnCase Version 6 to Version 7

You may know that Version 6 of EnCase keeps the majority of data in memory, which gives you fast access to the evidence items in a case, but is not conducive to handling large data sets. In addition, keeping most data in memory requires that records and entries be handled separately.

EnCase Version 7 behaves in a similar way to a database in that working through multiple evidence items is accomplished using an iterator. This makes for more stable processing and allows the EnScript programmer to handle both entries and records in a more streamlined way. It is possible, for instance, to iterate through all of the evidence items in a case (entries and e-mail attachments, for instance), quickly identifying those items that are pictures or documents.

It's very common for EnScript programmers to want to migrate their EnCase v6 workflows to EnCase v7. In doing so, it's good to consider the fact that the EnCase v7 evidence processor was designed to reduce the number of additional steps you have to take (hash and signature analysis, thumbnail creation, Registry pre-processing, etc.) before a review of the evidence in a case can begin. Taking this into account, some tasks might be better performed using a custom evidence processor module, an example of which is to be found in the following folder –

어..?

프로그래머들이 개발할 때 관리를 더 잘하고, 세분화하기 위해
그리고 빠른 로딩을 위해서.. 라는데..

```
V6
forall (EntryClass entry in c.EntryRoot()) {
  Console.WriteLine(entry.Name());
}
V7
ItemIteratorClass iter
            (c, ItemIteratorClass::NORECURSE |
                ItemIteratorClass::NOPROXY,
                ItemIteratorClass::ALL);
while (EntryClass entry = iter.GetNextEntry()) {
  Console.WriteLine(entry.Name());
}
```

장점

- 증거 분석이 엄청나게 빨라짐 (과제 시 엄청난 이득)
- 자신만의 특기로 엔스크립트 스페셜리스트 등극
- EnCase에서 지원하지 않는 기능을 본인이 제작 가능
- 로컬라이제이션 가능

단점

- 해보면 단점이 뭔지 알게됨

받는다 질문
안해도 된다
그냥 해본다

자러간다