C와 python에 관한 짤막한 예제

김용일

nalabi@formail.org

$Date: 2002/11/22 05:31:45 $

프로그램을 하면서 틈틈이 예제 프로그램을 작성하고, 다른 곳에 있는 예제를 모아 놓은 것입니다..

고친 과정
고침 0.12002-11-16고친이 김 용일
최초 작성

1. 소개

1.1. 이글은

프로그램을 하면서 틈틈이 예제 프로그램을 작성하고, 다른 곳에 있는 예제를 모아 놓은 것입니다..

주로 C와 python에 관련된 문서 입니다.

1.2. 덧글

한글을 살리고 나라를 살리는 길은, 과학적으로 으뜸가는 세벌식을 사용하는 것뿐입니다.

대한민국 과학기술을 살리는길은 쉬운 한글로 된 문서를 만드것입니다. 외국문서를 자기 혼자만 보고 말것이 아니라, 쉽게 번역, 보충해서 초중고생들로 볼수 있게 해야 합니다.

나라를 잘 살고, 부강하게 해야 하는 이유는 다른나라나, 인종에게 서러움을 받지 않기 하기 위함입니다.

1.3. 도와 주신분

관련서적과 여러 인터넷 사이트를 운영하고 계신 분들께 감사합니다.

1.4. 저작권

2. C

2.1. 컴파일 하기(make 사용함)

hello.c 소스

```
#include <stdio.h>

main()
{
  printf ("This is test program");
}
```

hello.c 소스

hello.c 컴파일하기

gcc -o hello hello.c

./hello

```
This is test program
```

※ gcc 옵션

c 언어는 기본적으로 컴파일을 먼저 해서 object 화일을 만들어 줍니다.

그리고 나서 link를 해주어서 실행 화일을 만들어 줍니다.

-o name : 실행 화일 이름을 정해서 만들어 준다.

-c : 컴파일만 한다.

make 사용하기

```
# all 명령을 실행한다.
all:   excute

# 오프젝트 화일을 만든다.
```

```
object:     hello.o

# 실행 화일 만든다.
excute :        hello

# 오프젝트 화일을 지운다.
clean:
            rm -f collage *.o
```

make 실행

```
cc    -c hello.c -o hello.o
cc   hello.o   -o hello
```

이때 오프 젝트 화일이 없어야 한다.(*.o)

hello 실행

```
This is test program
```

2.2. make 사용할수 있도록 만들기

autoscan

mv configure.scan configure.in

configure.in 수정

```
dnl Process this file with autoconf to produce a configure script.
AC_INIT()

AC_CONFIG_HEADER(config.h)

dnl Checks for programs.

.....
.....
.....

AC_OUTPUT( Makefile )
```

aclocal

autoheader

autoconf

automake (--add-missing)

./configure

2.3. 모듈(라이브러리) 기초

1. 정적 라이브러리

   .a

2. 공유 라이브러리

   .so, .sa

3. 정적 라이브러의 단점은 동시에 여러 프로그램을 띄웠을 경우 이들이 모두 같은 라이브러리의 함수들을 사용함으로써 메모리에 같은 함수들을 여러 번 복사하고, 프로그램 파일에도 같은 라이브러리를 여러 번 복사를 한다는 것이다.

   이는 메모리 용량과 디스크 용량을 많이 소모한다. 많은 유닉스 시스템에서는 이러한 불편함을 극복할 수 있도록 공유 라이브러리를 제공한다.

4. python 공유 모듈 만들기

```
#!/bin/bash

echo $1

if [ $# = 0 ] ;then
    echo "사용법 : 확장자(.c)를 뺀 소스 이름을 입력하세요."
    exit 0
fi
swig -python -module $1 $1.c
gcc -c $1.c $1_wrap.c -I/usr/include/python1.5 -
I/usr/lib/python1.5/config
ld -shared $1.o $1_wrap.o -o $1module.so
```

2.4. talk_server

대화방 서버 입니다.

컴파일 : **gcc -o talker_server talk_server.c**

실행 : **talk_server 3000**

소스

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <sys/types.h>
#include <signal.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>

#define MAXLINE 512

char * escapechar = "exit";

int main(int argc, char * argv[])
{
int server_fd, client_fd;
int clilen, num;

char sendline[MAXLINE], recvline[MAXLINE];

int size;
pid_t pid;
struct sockaddr_in client_addr, server_addr;

  if(argc != 2)
  {
    printf("사용법 : %s port  ", argv[0]);
    exit(0);
  }

  if((server_fd = socket(PF_INET, SOCK_STREAM, 0)) < 0)
  {
    printf("Server : Can't open stream socket ");
    exit(0);
  }

  bzero((char *) & server_addr, sizeof(server_addr));

  server_addr.sin_family = AF_INET;
  server_addr.sin_addr.s_addr = htonl(INADDR_ANY);
  server_addr.sin_port = htons(atoi(argv[1]));

  if(bind(server_fd, (struct sock-
addr *) & server_addr, sizeof(server_addr)) < 0)
  {
    printf("Server : Can't bind local address");
    exit(0);
  }
```

```
  printf("Server started. Waiting for client.. ");
  listen(server_fd, 1);

  clilen = sizeof(client_addr);
  if((client_fd = accept(server_fd, (struct sock-
addr *) & client_addr, &clilen)) < 0)
  {
    printf("Server : failed in accepting.");
    exit(0);
  }

  if(( pid = fork()) > 0)
  {
    while(fgets(sendline, MAXLINE, stdin ) != NULL)
    {
      size = strlen(sendline);
      if(write(client_fd, send-
line, strlen(sendline)) != size) printf("Error in write ");
      if(strstr(sendline, escapechar) != NULL)
      {
        printf("Good bye.");
        close(client_fd);
        exit(0);
      }
    }
  } else if (pid == 0)
  {
  while(1)
  {
    if((size = read(client_fd, recvline, MAXLINE)) < 0)
    {
      printf("Error if read ");
      close(client_fd);
      exit(0);
    }

    recvline[size] = ”;
    if(strstr(recvline, escapechar) != NULL) break;

    printf("%s", recvline);
    }
  }

  close(server_fd);
  close(client_fd);
}
```

2.5. talk_client

대화방 클라이언트 입니다.

컴파일 : **gcc -o talk_client talk_client.c**

실행 : **talk_client SERVERIP 3000**

소스

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <sys/types.h>
#include <signal.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>

#define MAXLINE 1024

char *escapechar = "exit";

int main(int argc, char *argv[])
{
  char line[MAXLINE], sendline[MAXLINE], recvline[MAXLINE+1];

  int n, size, comp;

  pid_t pid;

  static int s;

  static struct sockaddr_in server_addr;

  if(argc != 3)
  {
    printf("사용법 : %s server_IP port ", argv[0]);
    exit(0);
  }

  if((s = socket(PF_INET, SOCK_STREAM, 0)) < 0)
  {
    printf("Client: Can't open stream socket. ");
    exit(0);
  }

  bzero((char *) & server_addr, sizeof(server_addr));

  server_addr.sin_family = AF_INET;
  server_addr.sin_addr.s_addr = inet_addr(argv[1]);
```

```
  server_addr.sin_port = htons(atoi(argv[2]));

  if(connect(s, (struct sock-
addr *) & server_addr, sizeof(server_addr)) < 0)
  {
    printf("Client: can't connect to server. ");
    exit(0);
  }

  if(( pid = fork()) > 0)
  {
    while(fgets(sendline, MAXLINE, stdin) != NULL)
    {
      size = strlen(sendline);
      if(write(s, send-
line, strlen(sendline)) != size) printf("Error in write ");

      if(strstr(sendline, escapechar) != NULL)
      {
        printf("Good bye. ");
        close(s);

        exit(0);
      }
    }
  } else if(pid == 0)
  {
    while(1)
    {
      if((size = read(s, recvline, MAXLINE)) < 0)
      {
        printf("Error if read ");
        close(s);
      }
      recvline[size] = ";
      if(strstr(recvline, escapechar) != NULL) break;

      printf("%s", recvline);
    }
  }
  close(s);
}
```

2.6. 인자 처리 함수

프로그램 실행할때의 인자 처리 방법입니다.

```
#include <stdio.h>

char *programname; // usage()

char a;
char *b = "";
unsigned int d = 1;

void usage ();

main(int argc, char **argv)
{
char option;
extern char *optarg;
programname = argv[0]; // 프로그램 이름을 변수에 넣는다.

  if (argc < 2) usage (); // 인자가 없으면 usage() 함수 호출

  while ((option = getopt (argc, argv, "a:b")) != -1)
  { // 인자 처리 함수
    switch (option)
    {
      case 'a':
        a = 1;
        break;
      case 'b':
        b = optarg;
        break;
      default:
        usage ();
    }
  }

  printf("a : %d , b : %s", a, b);
}

void usage ()
{
    printf ("사용법 : %s [-a 인자] [-b 인자] [-c 인자] [-d 인
자]", programname);
    exit (1);
}
```

2.7. daytime.c 소스입니다.

daytime.c 소스입니다.

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#define BUF_LEN 128

int main(int argc, char *argv[])
{
int sockfd, n;

char *haddr;
struct sockaddr_in server_addr; /* 서버 주소를 넣을 구조체 */
char buf[BUF_LEN + 1];

  if(argc != 2)
  {
    printf("사용법 : %s ip_address", argv[0]);
    exit(0);
  }

  haddr = argv[1];

  if((sockfd = socket(AF_INET, SOCK_STREAM, 0)) < 0 )
  {
    printf("소켓을 만들수 없습니다.");
    exit(0);
  }

  bzero((char *) &server_addr, sizeof(server_addr)); /* 초기화를 해준다. */

  server_addr.sin_family = AF_INET;
  server_addr.sin_port = htons(13); /* daytime 서버 포트를 설정한다. */
  server_addr.sin_addr.s_addr = inet_addr(argv[1]);

  if(connect(sockfd, (struct sock-
addr *)&server_addr, sizeof(server_addr)) < 0)
  {
    printf("서버 포트에 접속을 할수 없습니다.");
    exit(0);
  }

  while((n = read(sockfd, buf, BUF_LEN)) > 0)
  {
    buf[n] = ”;
    if(fputs(buf, stdout ) == EOF) printf("aaaa");
```

```
  }

  close(sockfd);
}
```

2.8. 시그널 함수

콘트롤 키의 조합 할때 사용하는 함수 입니다.

시그널 함수 : /usr/include/bits/signum.h

컴파일 : **gcc -o signal_test signal_test.c**

실행 : **./signal_test**

소스

```
#include <sys/types.h>
#include <signal.h>
#include <stdio.h>
#include <unistd.h>

int my_signal(void); // 새로운 시그널 함수를 선언한다.
int count = 0;  // ctrl-c 입력 횟수 카운터

int main()
{
  int i = 0;

  if(signal(SIGINT, my_signal) == SIG_ERR)
  { // 시그널 함수를 작동 시킨다.
    printf("singal() error");
    exit(0);
  }

  while(count < 3)
  { // contrl-c 입력이 3번까지
    sleep(1); // 1초가 정지
    printf("%d", i++); //  ctrl-c 카운터
  }

}

int my_signal(void)
{
  printf("Ctrl-C pressed. ");
  count++;
  return 0;
```

```
}
```

2.9. 호스트 이름 알아내는 프로그램

컴파일 : **gcc -o test test.c**

소스1

```
#include <stdio.h>
#include <fcntl.h>

int main()
{
    char hostname[124];
    FILE *fd;
    fd = fopen("/proc/sys/kernel/hostname", "r");

    fread(hostname, sizeof(hostname), 1, fd);
    printf("%s", hostname);

    return 0;
}
```

소스2

```
#include <unistd.h>
#include <netdb.h>

main ()
{
  struct hostent* hoststruct;
  char hostname[255];

  gethostname(hostname, 100);
  hoststruct = gethostbyname(hostname);
  printf("호스트 이름은 %s 입니다.\n", hoststruct->h_name);
}
```

2.10. 다국어 테스트 프로그램(gettext)

1. hello.po를 만들어 준다.

   xgettext -a -d hello -k_ -s hello.c

2. mo 화일을 만들어 주는 사용법입니다.

   msgfmt hello.po

   messages 라는 화일이 생기는데 이것을 이름을 바꾸어 주면 됩니다.

   mv hello.po hello.mo

3. 로케일 디렉토리에 카피를 합니다.

   cp hello.mo /usr/share/locale/ko/LC_MESSAGES/hello.mo

4. 소스를 컴파일 합니다.

   gcc -o hello hello.c

5. 소스

```
#include <locale.h>
#include <stdio.h>
#include <libintl.h>
#define _(String) gettext (String)

int main(void)
{
    setlocale (LC_ALL, "");
    bindtextdomain ("hello", "/usr/share/locale/");
    textdomain ("hello");

    puts (_("hello"));
}
```

6. hello.po

```
#: hello.c:24
#, fuzzy
msgid ""
msgstr ""
"Project-Id-Version: 1.0\n"
"POT-Creation-Date: 2001-03-08 17:06+0900\n"
"PO-Revision-Date: 2001-03-08 17:06+0900\n"
"Last-Translator: Kim Yong Il <nalabi@formail.org>\n"
"Language-Team: Uniwork Project Team <nalabi@formail.org>\n"
"MIME-Version: 1.0\n"
"Content-Type: text/plain; charset=ecu-kr\n"
```

```
"Content-Transfer-Encoding: 8-bit\n"

#: hello.c:25 hello.c:26 hello.c:28
msgid "hello"
msgstr "안녕하세요
```

7. Makefile

```
# Makefile for locale directory

CWD = locale
MSGFMT = msgfmt
SUB_DIRS =
FILES_PO:=$(wildcard *.po)
FILES_MO:=$(FILES_PO:.po=.mo)

LOCALEDIR ?= /usr/share/locale

ifeq ($(enable_nls),1)
all: mo-files

mo-files: $(FILES_MO)

install:
    $(MAKE) all
    for f in *.mo ; do mkdir -p \
        $(IN-
STALL_PREFIX)$(LOCALEDIR)/`basename $$f .mo`/LC_MESSAGES ; \
        cp $$f $(IN-
STALL_PREFIX)$(LOCALEDIR)/`basename $$f .mo`/LC_MESSAGES/gkrellm.mo ; done

%.mo: %.po
    $(MSGFMT) -f -v -o $@ $<

else

all:

install:

endif

clean:
    $(RM) $(FILES_MO)
```

2.11. define 예제

1. 소스

```
#include "stdio.h"

#define TEST(test)       my_func_##test()
#define test(num)        __asm__ __volatile__("call test_" #num::);

void my_func_hello(void)
{
  printf("HELLO !!\n");
}


void my_func_goodbye(void)
{
  printf("Good Bye !!\n");
}


void test_1(void)
{
  printf("This is test_1 function\n");
}


void test_2(void)
{
  printf("This is test_2 function\n");
}


main()
{
  int i;
  TEST(hello);
  TEST(goodbye);

  test(1);
  test(2);
}
```

2. 컴파일

   gcc -o t test.c

3. 실행

```
HELLO !!
Good Bye !!
This is test_1 function
This is test_2 function
```

2.12. 찾기를 원하는화일

소스

```
#include

int ftw_find();

int main(int argc, char *argv[])
{
  if (argc > 1)
  {
    ftw(argv[1],ftw_find, 0);
  }
  else
  {
    printf("Usage: %s dir\n", argv[0]);
  }
  return 0;
}

int ftw_find(const char *path, struct stat *sb, int flag)
{
  if(strstr(path,"찾고자 하는 파일이나 디렉토리")!=NULL)
  {
    printf("%s:\t%u\t\n", path, sb->st_size);
    exit(0);
  }
  return 0;
}
```

2.13. 커널 시스템 콜 함수

unistd.h 헤더

/usr/src/linux-2.4.18# vi include/asm/unistd.h

```
#define __NR_lremovexattr   236
```

```
#define __NR_fremovexattr   237
#define __NR_mycall         238
```

entry.S

/usr/src/linux-2.4.18# vi arch/i386/kernel/entry.S

```
    .long SYMBOL_NAME(sys_ni_syscall)   /* reserved for lremovexattr */
    .long SYMBOL_NAME(sys_ni_syscall)   /* reserved for fremovexattr */
    .long SYMBOL_NAME(sys_mycall)  /* 238 mycall */
```

mycall.c

/usr/src/linux-2.4.18# vi kernel/mycall.c

```
#include <linux/kernel.h>
#include <linux/errno.h>

asmlinkage void sys_mycall()
{
    printk("Corea Manse!!!!\n");
}
```

krernel/Makefile

/usr/src/linux-2.4.18# vi kernel/Makefile

```
obj-y     = sched.o dma.o fork.o exec_domain.o panic.o printk.o \
        module.o exit.o itimer.o info.o time.o softirq.o resource.o \
        sysctl.o acct.o capability.o ptrace.o timer.o user.o \
        signal.o sys.o kmod.o context.o mycall.o
```

user Makefile

```
COMPILE=gcc

INCLUDEDIR = /usr/src/linux-2.4.18/include

MODCFLAGS := -I$(INCLUDEDIR)

mycall_app : mycall_app.c

    $(COMPILE) $(MODCFLAGS) -o $@ $^
```

소스 (mycall_app.c)

```
#include <linux/unistd.h>

/* system call stub function */
_syscall0(void, mycall);

main()
{
  mycall();
}
```

2.14. 맥어드레스 알아내기

소스

```
#include <netinet/ether.h>
#include <net/ethernet.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <sys/ioctl.h>
#include <net/if.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <stdlib.h>
#include <stdio.h>

int main (int argc, char *argv[])
{
  struct ifreq *iflist;
  struct sockaddr *sa;
  int fd;
  char *out;

  if(argc != 2)
  {
    printf("Usage: progname ifname (ex: progname eth0)\n");
    return -1;
  }

  iflist = malloc (sizeof (struct ifreq));

  fd = socket (PF_INET, SOCK_STREAM, 0);

  strncpy (iflist->ifr_name, argv[1], strlen (argv[1]));
```

```
  if (ioctl (fd, SIOCGIFHWADDR, iflist) == -1)
  {
    perror ("ioctl failed");
    return -1;
  }

  sa = &(iflist->ifr_hwaddr);

  out = ether_ntoa ((struct ether_addr *) sa->sa_data);

  printf ("%s\n", out);

  return 0;
}
```

3. C++

3.1. 클래스를 이용해서 호스트 이름 출력

g++ 을 이용해서 호스트 이름을 출력하는 테스트 프로그램입니다.

컴파일 : **g++ -o test test.cpp**

소스

```
#include <unistd.h>
#include <string>

// 클래스를 선언하나.
class Init
{
private:
public:
  static string get_hostname();
};

// get_hostname 이라는 함수를 만들어 준다.
string Init::get_hostname()
{
  char buf[128]; // 호스트 이름이 들어갈 변수를 선언
  string ret;

  // 호스트 이름을 얻는 함수를 호출한다.
```

```
  if((gethostname ( buf, sizeof(buf) )) < 0)
  {
    ret.assign("Unkown Hostname"); // 호스트 이름이 없을 경우
  } else
  {
    ret.assign(buf); // 호스트 이름을 얻었을 경우
  }

  return ret;  // 결과 값을 리턴한다.
}

int main(void)
{

  cout << Init::get_hostname() << endl; // 화면에 출력한다.

}
```

3.2. 객체 만들어서 출력하기

소스

```
#include <string>

// Hello 객체를 만든다.
class Hello
{
  private:
  public:
    // 함수를 하나 만든다.
    void displayData()
    {
      cout << "I am class" << endl; // 출력을 한다.
    }
};

int main(void)
{
  Hello hello; //hello 객체를 선언한다.

  hello.displayData(); //출력한다.
}
```

결과

```
I am class
```

4. gtk

4.1. 플밍에서 타이틀바 없애는 방법

소스

```
#include <gtk/gtk.h>

int main( int   argc, char *argv[] )
{
    GtkWidget *window;

    gtk_init (&argc, &argv);

    window = gtk_window_new (GTK_WINDOW_TOPLEVEL);
    gtk_widget_set_usize(window, gdk_screen_width(), gdk_screen_height());
    gtk_widget_set_uposition(window, 0, 0);

    gtk_widget_realize(window);
    gdk_window_set_decorations(window->window, 0);

    gtk_widget_show  (window);

    gtk_main ();

    return(0);
}
```

컴파일 : **gcc test.c -o test ‘gtk-config --cflags --libs‘**

4.2. 버튼에 색상 넣기

리소스 화일 작성한다. (testrc)

```
style "button"
{
    fg[PRELIGHT] = { 1.0, 1.0, 1.0 }  # 글자색 마우스를 올려 놓았을때
    bg[PRELIGHT] = { 0, 0.5, 0.5 }    # 배경색 마우스를 올려 놓았을때
```

```
    fg[ACTIVE] = { 0, 0.0, 0.0 }          # 글자색 클릭했을 경우
    bg[ACTIVE] = { 0, 0.6, 0.6 }          # 배경색 클릭했을 경우

    fg[NORMAL] = { 0.0, 0.0, 0.0 }    # 글자색 평상시 색상
    bg[NORMAL] = { 0.0, 0.5, 0.5 }    # 배경색 평상시 색상

    fg[INSENSITIVE] = { 1.0, 0, 1.0 }
    bg[INSENSITIVE] = { 1.0, 1.0, 1.0 }
}

widget "*Button*" style "button"
```

소스

```
#include <gtk/gtk.h>

int main (int argc, char *argv[])
{
  GtkWidget *window;
  GtkWidget *button;

  gtk_rc_parse( "testrc" );
  gtk_set_locale();

  gtk_init (&argc, &argv);

  window = gtk_window_new (GTK_WINDOW_TOPLEVEL);

  button = gtk_button_new_with_label ("버튼");
  gtk_widget_show (button);

  gtk_container_add (GTK_CONTAINER (window), button);

  gtk_widget_show (window);

  gtk_main ();
  return 0;
}
```

컴파일 : **gcc main.c -o main ‘gtk-config --cflags --libs‘**

5. python

5.1. print 명령어

1. 소스

```
def mesg():
        a = 1
        return a

print mesg()
```

결과

```
1
```

2. 소스

```
a = 10
b = 10
def mesg(a, b):
        c = a + b
        return c

print mesg(a, b)
```

결과

```
20
```

3. 소스

```
a = 10
b = 10
def mesg(a, b):
        c = a + b
        return c

if __name__ == '__main__':
        print mesg(a, b)
```

결과

```
20
```

5.2. python 객체 공부하기1

1. 소스

```
#!/usr/bin/env python

from gtk import *

# FirstClass 라는 객체를 만듭니다.
class FirstClass:
    def setdata(self, value):  # 어떤 값을 받는 setdata 함수를 만듭니다.
        self.data = value
    def display(self):         # setdata 에서 받은 값을 출력합니다.
        print self.data

x = FirstClass()
y = FirstClass()

x.setdata("Kim Yong-Il")
x.display()

y.setdata("3.14159")
y.display()

print "오게 바리"
```

2. 결과

python hello1.py

```
Kim Yong-Il
3.14159
오게 바리
```

5.3. python 객체 공부하기2

1. 소스

```
#!/usr/bin/env python
```

```
from gtk import *

# FirstClass 라는 객체를 만듭니다.
class FirstClass:
    def setdata(self, value):  # 어떤 값을 받는 setdata 함수를 만듭니다.
        self.data = value
    def display(self):         # setdata 에서 받은 값을 출력합니다.
        print self.data

# SecondClass 라는 객체를 만듭니다.
# SecondClass 라는 객체는 FirstClass 객체를 포함합니다.
class SecondClass(FirstClass):
    def display(self):         # 화면 출력 함수를 만듭니다.
        print 'Current value = "%s"' % self.data

z = SecondClass()
z.setdata(42)
z.display()

print "오게 바리"
```

2. 결과

python hello2.py

```
Current value = "42"
오게 바리
```

5.4. python 객체 공부하기3

1. 소스

```
#!/usr/bin/env python

from gtk import *

# FirstClass 라는 객체를 만듭니다.
class FirstClass:
    def setdata(self, value):  # 어떤 값을 받는 setdata 함수를 만듭니다.
        self.data = value
    def display(self):         # setdata 에서 받은 값을 출력합니다.
        print self.data

# SecondClass 라는 객체를 만듭니다.
```

```
# SecondClass 라는 객체는 FirstClass 객체를 포함합니다.
class SecondClass(FirstClass):
    def display(self):         # 화면 출력 함수를 만듭니다.
        print 'Current value = "%s"' % self.data

# ThirdClass(SecondClass):
# Third 라는 객체는 Second 객체를 포함합니다.
class ThirdClass(SecondClass):
    def __init__(self, value):
        self.data = value
    def __add__(self, other):
        return ThirdClass(self.data + other)

a = ThirdClass("abc")
a.display()

b = a + 'xyz'
b.display()


print "오게 바리"
```

2. 결과

python hello3.py

```
Current value = "abc"
Current value = "abcxyz"
오게 바리
```

5.5. python 객체 공부하기4

1. 소스

```
#!/usr/bin/env python

from gtk import *

class Subclass:
    data = 'spam'
    def __init__(self, value):
        self.data = value
    def display(self):
        print self.data, Subclass.data
```

```
x = Subclass(1)
y = Subclass(2)

x.display()
y.display()

print "오게 바리"
```

2. 결과

python hello4.py

```
1 spam
2 spam
오게 바리
```

5.6. python 객체 공부하기5

1. 소스

```
#!/usr/bin/env python

from gtk import *

class Super:
    def method(self):
        print 'in Super.method'

class Sub(Super):
    def method(self):
        print 'starting Sub.method'
        Super.method(self)
        print 'ending Sub.method'

x = Super()
x.method()

print "------------------>"

x = Sub()
x.method()
```

2. 결과

python hello5.py

```
in Super.method
------------------>
starting Sub.method
in Super.method
ending Sub.method
```

5.7. python 객체 공부하기6

1. 소스

```
#!/usr/bin/env python

from gtk import *

class Super:
    def method(self):
        print 'in Super.method'
    def delegate(self):
        self.action()

class Inheritor(Super):
    pass

class Replacer(Super):
    def method(self):
        print 'in Replacer.method'

class Extender(Super):
    def method(self):
        print 'starting Extender.method'
        Super.method(self)
        print 'ending Extender.method'

class Provider(Super):
    def action(self):
        print 'in Provider.action'

if __name__ == '__main__':
    for klass in (Inheritor, Replacer, Extender):
        print '
' + klass.__name__ + '...'
        klass().method()
    print '
```

```
Provider...'
    Provider().delegate()
```

2. 결과

python hello6.py

```
ending Extender.method

Inheritor...
in Super.method

Replacer...
in Replacer.method

Extender...
starting Extender.method
in Super.method

Provider...
```

5.8. python 객체공부하기 7

1. 소스

```
#!/usr/bin/env python

from gtk import *

class Number:
    def __init__(self, start):
        self.data = start
    def __sub__(self, other):
        return Number(self.data - other)

x = Number(5)
y = x -2
y.data

print y.data
```

2. 결과

python hello7.py

```
3
```

5.9. python 객체 공부하기8

1. 소스

```
#!/usr/bin/env python

from gtk import *

class indexer:
    def __getitem__(self, index):
        return index ** 2

x = indexer()
for i in range(5):
    print x[i],
```

2. 결과

   python hello8.py

```
0 1 4 9 16
```

5.10. smtp.py 메일발송 프로그램

최희철 소스, ironyjk (at) kldp.org

/var/mail 에 있는 메일을 원하는 메일 주소로 보낸다.

실행 : **cd /var/mail**

python stmp.py ID 멜주소

smtp.py

```
import smtplib , sys , string

def sendmail ( ff , to , text ) :
    if string.find ( ff , "@" ) == -1 :
        print "%s Break" % ff

        return
```

```
#   print " toaddr =  %s" % ff
#   print " text ==  %s" % text
    server = smtplib.SMTP('localhost')
    server.sendmail(ff , to , text)
    server.quit()


fromaddr = "root@localhost"
toaddr = sys.argv[2]

f = open ( sys.argv[1] )

line = f.readline()

text = ""

while line :
    if len ( string.split ( line ) ) > 2 and string.split ( line )[0] == "From" :
        sendmail ( fromaddr , toaddr , text )
        fromaddr = string.split ( line )[1]
        text = ""

    text = text + line
    line = f.readline()

if ( text != "" ) :
    sendmail ( fromaddr , toaddr , text )
```

5.11. bab.py 식당 메뉴 프로그램

최희철씨 소스 , ironyjk (at) kldp.org

점심때 무엇을 먹을까 고민할때 한번 실행해주면 됩니다.

bab.py

```
import random,smtplib,sys


list = ['한국회관:해물된장','한국회관:갈비탕', '한국회관:갈비탕', '한국회관:비빔
밥','한국회관:설렁탕' ,'한국회관:육계장' ,'삼익식당:>만두국','삼익식당:백반' ,'삼익식
당:제육볶음' , '삼익식당:김치찌게' , '삼익식당:김치찌게' , '중국집:복음밥' , '중국
집:짬뽕', '중국집:>짜장면' , '삼익식당:야채볶음밥' ,'분식집:김치복음밥','분식집:부대
찌개']

menu = list [ int ( (random.random () * 100 ) % len (list) ) ]
```

```
fromaddr = "ID@address.org"
toaddr = "ID@address.org"
msg = "To: %s  Subject:오늘의 메뉴!  오늘의 메뉴는 %s 입니
다." % ( toaddr ,  menu )

if  len ( sys.argv ) >  1 :
  msg = msg + "즐거운 저녁식사입니다. 저녁식사 안하실 분은 관리자에게 신고 바랍니
다. "


lastmenu = ""
msg = msg + " -- 현재 메뉴 현황 --  "

for i in list :
  if lastmenu != i :
    msg = msg + " - %s " % i
  lastmenu = i


msg = msg + " 새로운 메뉴의 추가는 언제나 환영합니다.  새로운 메뉴
는 ID@address.org 메일 보내주시기 바랍니다."

server = smtplib.SMTP('address.org')
server.sendmail(fromaddr, toaddr, msg)
server.quit()
```

5.12. mysql 데이타 읽어오기

mysql 데이타를 가져 옵니다.

```
#!/usr/bin/python

import MySQLdb

db = MySQLdb.connect(host="localhost", db="test", user="root" , passwd="")

hash = db.cursor()
hash.execute("select * from test")
result = hash.fetchall()

for i in result:
    print result[0]

db.close()
```

5.13. ncurse 예제

소스

```
#!/usr/bin/python
import curses
import time

stdscr=curses.initscr()
curses.start_color()
curses.init_pair(1,curses.COLOR_CYAN, curses.COLOR_BLACK)

stdscr.addstr(10,10,"test", curses.color_pair(1))
stdscr.addstr(12,10,"aaaaaaaa", curses.A_REVERSE)

stdscr.refresh()
time.sleep(5)
curses.endwin()
```

5.14. CGIHTTPServer.py 사용하기

1. 서버 실행하기

```
/usr/lib/python1.5$ python CGIHTTPServer.py
Serving HTTP on port 8000 ...
```

2. 루트권한 사용하기

vi CGIHTTPServer.py

```
nobody = nobody_uid()
# 여기서 시스템의 nobody 계정 UID를 가져 옵니다.
self.wfile.flush() # Always flush before forking
pid = os.fork()
if pid != 0:
    # Parent
    pid, sts = os.waitpid(pid, 0)
    if sts:
        self.log_error("CGI script exit status %#x", sts)
    return
# Child
try:
    try:
        # 이부분을 nobody로 사용할 경우는 다음과 같이 하고, 루트를 사용할경우
는 주석처리합니다.
```

```
        os.setuid(nobody)
    except os.error:
        pass
    os.dup2(self.rfile.fileno(), 0)
    os.dup2(self.wfile.fileno(), 1)
    os.execve(scriptfile, args, env)
except:
    self.server.handle_error(self.request, self.client_address)
```

3. CGIHTTPServer 모듈에서 CGI 실행하기

모듈을 이용해서 다음과 같이 간단한 서버 프로그램을 할수 있습니다.

```
#!/usr/bin/env python

import CGIHTTPServer
import BaseHTTPServer

Handler = CGIHTTPServer.CGIHTTPRequestHandler

PORT = 8080

httpd = BaseHTTPServer.HTTPServer(("", PORT), Handler)
print "Serving at port ", PORT
httpd.serve_forever()
```

GIHTTPServer.py 에 소스에 보면 cgi 디렉토리가 다음과 같이 설정이 되어 있습니다.

필요시 변경을 할수 있습니다.

cgi_directories = ['/cgi-bin', '/htbin']

4. cgi 프로그램 (test.py)

```
#!/usr/bin/env python

print """Content-type: text/html

성공 했습니다.
"""
```

위와 같은 예제를 이용해서 cgi-bin 밑에 넣어서 다음과 같이 실행을 하면 됩니다.

http://localhost:8080/cgi-bin/test.py

5.15. mysql 테이블 속성 출력하기

소스

```
import MySQLdb

connection = MySQLdb.connect(user='root', passwd='', db='nalabi')
cursor = connection.cursor()  # 커서 객체를 얻어온다.
cursor.execute('select * from test')
res = cursor.description
for record in res:
        print "%-10s %-10s %-10s %-10s %-10s %-10s %-10s" % record
cursor.close()
```

결과

```
no         3          2          6          6          0          0
num        3          2          6          6          0          0
idx        3          2          6          6          0          0
date       3          9          11         11         0          0
host       252        13         255        255        0          1
name       252        6          255        255        0          1
passwd     253        13         13         13         0          1
email      252        18         255        255        0          1
url        252        19         255        255        0          1
title      252        20         255        255        0          1
text       252        3243       16777215   16777215   0          1
refer      3          1          6          6          0          0
reyn       3          1          1          1          0          0
reno       3          1          6          6          0          0
rede       3          1          6          6          0          0
reto       3          1          6          6          0          0
html       3          1          1          1          0          0
moder      3          1          1          1          0          0
bofile     253        18         100        100        0          1
bcfile     253        14         100        100        0          1
bfsize     3          5          4          4          0          1
```

5.16. 디렉토리 권한 바꾸기

소스

```
import commands, string, os

import os
```

```
out=commands.getoutput("ls -l")
a = string.split ( out , '\n' )

for i in range(len(a)):
    b = string.split ( a[i])
    if (b[0] != "\307\325\260\350")  :
        print b[8]
        cc = "chown %s:%s %s" % (b[8], b[8], b[8])
        commands.getoutput(cc)
```

6. perl

6.1. 화일 읽고 split 하기

화일 읽고 split 하기

```
#!/usr/bin/perl

&print_file("/etc/webmin/config");

exit;

sub print_file
{
my ($filename) = @_;

  open(fh,"$filename");

  while(< fh>)
  {
    ($name,$value) = split(/=/,$_);

    if( $name eq "admin_mail" )
    {
      print $value;
    }
  }

  close(fh);
}
```