

Unix 및 인터넷 개론

2008년 3월 26일

목 차

제 1 장 Overview

제 1 절 Object	9
제 2 절 교재 및 평가	10

제 2 장 Unix

제 1 절 Unix 소개	11
1.1 운영체제란?	11
1.2 Unix 란?	12
1.3 Unix 기초	13
제 2 절 Unix 파일 시스템 기본 사항	20
2.1 File 구조	20
2.2 directory	22
제 3 절 파일 명령어	25
3.1 기본명령	25
3.2 파일의 권한	34
제 4 절 vi 편집기	36
4.1 시작 및 종료	36
4.2 문서작성	37
제 5 절 파일 관련 명령어	44
5.1 파일 및 내용 검색 명령어	44
5.2 파일 정렬, 합병, 비교, 축출 명령어	46
제 6 절 X 윈도우	49
6.1 X-window 소개	49
6.2 작업하기	50
제 7 절 Shell	53
7.1 Shell basic	53
7.2 Shell 이 제공하는 기능	54
7.3 시스템 자원 활용 명령어	57

제 3 장 Internet

제 1 절 Internet basic	61
1.1 Network	61
1.2 Internet Access	62
제 2 절 Internet Utilities	64
2.1 email	64
2.2 telnet	66

2.3 finger	6
2.4 Anonymous FTP	6
2.5 Archie	6
2.6 talk	6
2.7 WWW basic	6

제 4 장 WWW

제 1 절 Homepage 만들기	7
1.1 HTML 사용하기	7
1.2 CGI 사용하기	9
1.3 C 언어를 사용한 CGI 프로그래밍	10
1.4 Java script 사용하기	10
1.5 Java 사용하기	12

그림 목차

4.1 CGI 동작 방식	98
-------------------------	----

표 목차

1.1 강의 주별 내용	9
1.2 참고서적	10
2.1 OS의 비교	12
2.2 wildcard 요약	25
2.3 파일 사용권한	34
2.4 모드 변경 기호	35
2.5 vi에서 편집 시작, 저장 및 종료	37
2.6 커서 이동 명령들	38
2.7 편집버퍼를 통한 이동 명령	38
2.8 정규표현식에 사용되는 기호들	39
2.9 문자 입력 명령	40
2.10 문자 입력 명령	40
2.11 치환 명령 예	40
2.12 데이터 지우기 명령	41
2.13 find 명령에서 expression들	45
2.14 grep 옵션들	46
2.15 sort option들	47
2.16 Bourne shell 계열	54
3.1 TCP/IP 참조 모델	63
3.2 미국과 캐나다에서 사용되는 최상위 Domain 예	63
3.3 국제 최상위 Domain 일부	64
4.1 li tag의 type 속성	81
4.2 li tag의 type 속성	81
4.3 type 속성의 종류	99
4.4 자료 형식	104
4.5 CGI 환경 변수	105
4.6 Java script 와 HTML 의 차이점	107
4.7 Java script 와 Java 의 차이점	108
4.8 특수문자 사용예	109
4.9 Java script에서 사용되는 예약어	109
4.10 대입연산자	112
4.11 연산자 우선순위	114
4.12 event의 종류	122
4.13 Java Applet과 Application의 차이점	126

제 1 장

Overview

제 1 절 Object

- 강의 목적

- Internet 상의 많은 호스트에서 사용되고 있는 운영체제인 Unix 사용법을 학습
- → 실습에서는 Unix 계열인 Linux로 실습
- Internet 개념 및 여러 가지 유틸리티 사용법 소개
- 인터넷 언어 기초 소개

- 강의 내용

표 1.1: 강의 주별 내용

주	강의 내용
1주	강의 소개 및 Unix 소개
2주	Unix 파일 시스템 기본 사항
3주	Unix 파일 명령어
4주	vi 편집기
5주	파일 관련 명령어
6주	X 윈도우
7주	Shell
8주	중간고사
9주	Internet 의 기초
10주	Internet Utilities
11주	Web browser
12주	HTML 을 이용하여 Homepage 만들기
13주	CGI 를 이용하여 Homepage 만들기
14주	Java 및 Java script 을 이용하여 Homepage 만들기
15주	ASP
16주	기말고사

제 2 절 교재 및 평가

- 강의 교재는 따로 없고 다음의 참고서적을 참조
- 참고 서적

표 1.2: 참고서적

번호	책 제목	저자	출판사
1	Unix 및 인터넷 개론	지준/정성훈/김일민 공저	한성대학교 출판사
2	Unix 그리고 Internet	전병실/조준익 공저	복두출판사
3	알기 쉽게 설명한 Unix	유황빈/이재광 공역	이한출판사
4	파워유저를 위한 홈페이지 작성법	이승호 저	복두출판사
5	Internet 완벽해설서	이재광역	이한출판사

- 책의 특성

- 1번 책: 강의노트 중 Unix 부분은 주로 이 책을 참조하여 작성됨
- 2번 책: 잘 정리된 책이나 최신 내용이 빈약
- 3번 책: Unix 전반적인 내용 설명, Unix에 관심 많은 학생은 이 책을 참조
- 4번 책: 강의노트 중 홈페이지 부분은 주로 이 책을 참조하여 작성됨
- 5번 책: Internet에 대한 자세한 설명이 있음

- 참고 internet sites

- best site¹
- Prof. Ji Jun's lecture note²

- 평가

- 출석: 10 혹은 20 %
- 중간: 30 %
- 기말: 40 %
- 과제: 20 혹은 10 % (과제는 여기로³제출하세요)
 - * 모든 과제는 마감일이 지나면 제출이 불가능합니다.
 - * 과제 제출 후 위의 link에서 평가점수를 확인할 수 있습니다.
 - * 과제는 꼭 제출 파일 타입으로 제출하세요.

¹<http://dblab.changwon.ac.kr>

²<http://ice.hansung.ac.kr/jun/UNI/UNI-lec0.htm>

³<http://itsys.hansung.ac.kr/cgi-bin/rpt-grade/uninetmet/rpt-grade.cgi>

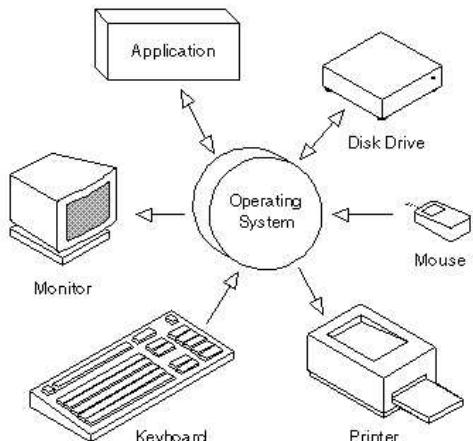
제 2 장

Unix

제 1 절 Unix 소개

1.1 운영체제란?

- 운영체제의 정의



- 사용자와 computer 사이에서 컴퓨터의 자원을 효율적으로 사용할 수 있게 하는 프로그램
- 주요 기능
 1. 사용자 인터페이스
 2. 사용자간 하드웨어 공유
 3. 사용자간 데이터 공유
 4. 사용자간 자원 스케줄링

5. 서로 다른 사용자간의 간접 배제
 6. 입출력에 대한 용이성 제공
 7. 오류의 복구
 8. 자원 사용량 기록
 9. 병렬연산에 대한 용이성 제공
 10. 보안 및 액세스 제어
 11. 네트워크 통신 처리
- 컴퓨터 시스템
 - * 하드웨어 + 소프트웨어
 - * 소프트웨어 ← 시스템 소프트웨어 (운영체제) + 응용 프로그램
 - 운영체제의 예
 - * Unix, Linux, DOS, Windows 95/98/2000 그외에 다수

1.2 Unix 란?

- 특징
 - 1960년대 AT&T Bell lab.에서 시작한 운영체제
 - multi-user, multi-tasking, time-sharing 시스템
 - 대부분 C언어로 개발됨
- Version
 - AT&T 개발 → Unix system V Release 4 (SVR4)
 - Berkeley 개발 → BSD4.4 (Berkeley Software Distributions)

표 2.1: OS의 비교

Computer	OS	비고
PC Workstation	DOS, windows, OS/2 Unix	Linux는 PC에서 수행되는 Unix

- Unix의 변종
 - SunOS, Solaris ← Sun workstation 사용자를 위한 Unix
 - 386-BSD, Linux ← PC 사용자를 위한 Unix
- Unix의 구성요소
 - 커널 + 파일 시스템 + 유ти리티 프로그램
 - 커널
 - * Unix의 핵심부분

- * 시스템 장치간의 입출력 관리
- * 사용자 자원 관리
- 파일 시스템
 - * 여러가지 종류의 파일을 관리
- 유ти리티
 - * Unix 를 사용할때 혹은 여러가지 목적을 위하여 만들어진 프로그램
 - * 예) 쉘 (명령어 해석기), 응용프로그램, 전자우편, 컴파일러 등등
- 쉘
 - * 사용자가 입력한 명령을 해석하고 처리하여 결과를 돌려주는 명령어 해석기
 - * 종류로는 Bourne Shell, C Shell, Korn Shell 등이 있음

1.3 Unix 기초

- 사용자 ID (userid) 와 암호 (password)
 - system manager : system 전반을 관리, 유지, 보수하는 사람
 - userid : Unix system 사용자를 확인 (보통 자기 이름을 이용하여 만듦)
 - password : userid 당 암호를 지정하여 다른 사람의 사용을 방지
 - 계정 (account) : Unix system 을 사용할수 있도록 등록한것
- 작업 들어가기 (로그인)



- login : userid 를 입력 (사용자 확인)
- password : 암호 입력 (사용자 확인) ← 글자가 보이지 않음 (비밀 보호)
- 위와같은 로그인 작업후에는 초기 로그인 메세지가 출력됨
- 주의사항 Unix 는 대소문자를 구별함
- 암호변경 및 선택

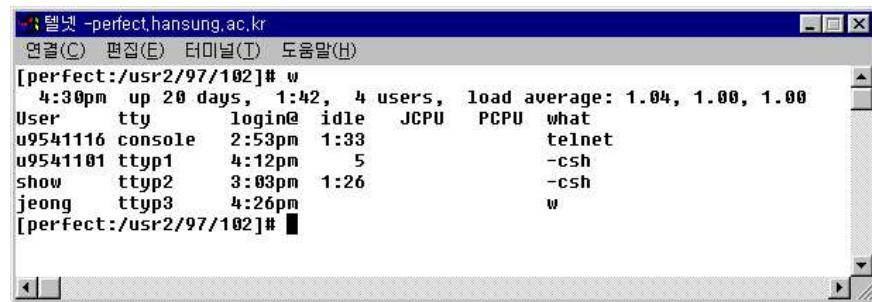
```
[perfect:/usr2/97/102]# passwd
Changing password for jeong on perfect.
Old password:
New password:
Retype new password:
[perfect:/usr2/97/102]#
```

- 암호변경 명령 →passwd
 - * 현재의 암호를 입력 (사용자 확인)
 - * 개신암호
 - * 개신암호 (암호의 정확한 입력을 위하여 두번 입력)
- 암호선택 : 암호는 다른 사람이 추측할수 없는 의미없는 것으로하고 의미있게 바꾼다

- 작업하기
 - shell : 명령을 읽어서 해석, 수행시켜주는 프로그램
 - shell prompt : shell이 대기상태 있을때 보여주는 것
 - * shell prompt 는 재시작 가능하나 default 는 다음과 같음
 - * c-shell : %
 - * Bourne shell, Korn shell : \$
 - * cf.) PC에서 C:> prompt 와 같음
 - 간단한 명령 입력
 - * date: 현재 날짜와 시간을 보여줌

```
[perfect:/usr2/97/102]# date
Fri Jul 23 16:29:31 KDT 1999
[perfect:/usr2/97/102]#
```

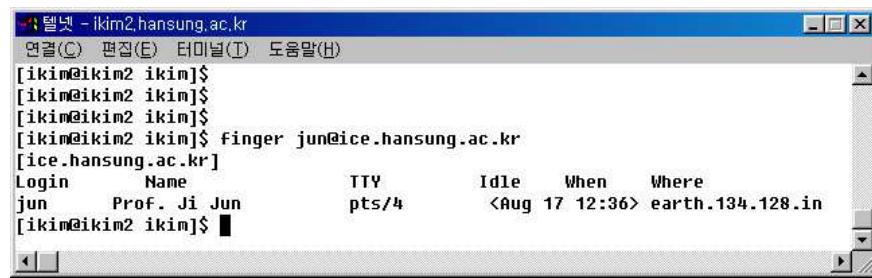
- * cal : 달력을 보여줌
- * w: 현재 로그인 사용자 및 하는일을 보여줌
- * who: 현재 시스템 사용자의 목록을 보여줌
- * whoami: 현재 쉘사용자의 로그인 Id. 를 보여줌
- * finger: 특정 사용자의 정보를 보여줌
- * pwd : 현재 위치한 디렉토리의 경로를 표시해줌
- * cd : 디렉토리를 변경



```
[perfect:/usr2/97/102]# w
4:30pm up 20 days, 1:42, 4 users, load average: 1.04, 1.00, 1.00
User     tty        login@idle   JCPU   PCPU what
u9541116 console  2:53pm 1:33      telnet
u9541101 tttyp1   4:12pm 5          -csh
show    tttyp2   3:03pm 1:26      -csh
jeong   tttyp3   4:26pm          w
[perfect:/usr2/97/102]#
```



```
[perfect:/usr2/97/102]# who
u9541116 console Jul 23 14:53
u9541101 tttyp1 Jul 23 16:12 (128.134.168.92)
show    tttyp2 Jul 23 15:03 (128.134.168.36)
jeong   tttyp3 Jul 23 16:26 (128.134.169.115)
[perfect:/usr2/97/102]#
```



```
[ikim@ikim2 ikim]$
[ikim@ikim2 ikim]$
[ikim@ikim2 ikim]$
[ikim@ikim2 ikim]$ finger jun@ice.hansung.ac.kr
[ice.hansung.ac.kr]
Login      Name           TTY      Idle    When      Where
jun       Prof. Ji Jun    pts/4    <Aug 17 12:36> earth.134.128.in
[ikim@ikim2 ikim]$
```

- * ls : 디렉토리의 파일 목록을 출력해줌
- * cat : 파일을 합치거나 내용을 출력해줌
- * more : 텍스트 파일의 내용을 출력해줌
- * clear: 셸 화면의 내용을 지워줌
- * uptime : 시스템이 부팅된 후 얼마나 지났는지를 표시해줌
- * hostname : 현재 사용하는 있는 컴퓨터의 이름을 표시



```
[perfect:/usr2/97/102]# hostname
perfect
[perfect:/usr2/97/102]#
```

- * ping : 특정 host 가 접속 가능한지를 알려줌
- 1. 예) ping 128.134.168.62 (or ping perfect.hansung.ac.kr)

• 작업종료 (로그아웃)

- 작업을 모두 끝냈을 때 종료하는 것
- 종료를 하지 않으면 다른 사람이 내 계정으로 사용할 수 있음
- 방법
 - * cntl-D, exit, logout

• super user 의 사용자 (root)

- 시스템 관리자의 사용자 ID : root → super user
- 시스템 관리자는 어떤 일도 할 수 있다.

• 실습

1. 로그인을 수행해본다 (계정이 아직 없는 학생은 login 에 guest, password에 guest 를 입력)
2. date, cal, w, who, whoami, finger, pwd, cd, ls, cat, more, clear, uptime, host name, ping 명령을 수행해 본다
3. 암호를 변경해 본다.
 - 반드시 자신의 계정이 있는 학생만 변경할 것
 - guest 로 login 한 학생은 변경하지 말 것
4. logout 을 해본다 (logout 후 다시 login 할 것)

- Unix 온라인 manual

- disk 상에 있는 manual
- ex.) man bc
- 특정문자 검색 → /문자
- 도움말 → h
- 구성
 - * 1절 : 명령어들
 - * 2절 : 시스템 호출
 - * 3절 : 라이브러리 함수
 - * 4절 : 특수파일
 - * 5절 : 파일형식
 - * 6절 : 게임
 - * 7절 : 기타 정보
 - * 8절 : 유지 보수 명령어

- 절 지정 명령

- * man kill → 1절의 kill manual
- * man -s2 kill → 시스템 호출의 kill manual
- * man date

- 내용참고

- * manual 을 보는 중 umask(2) 라고 나오는 것은 man -s2 umask 로 참조하는 듯
- * manual 의 마지막 부분에 SEE ALSO 는 해당 명령과 관련된 명령어를 보여줌

- 간단하게 설명

- * man -f time date
- * whatis time date

- 관련된 명령어 찾기

- * keyword로 명령어 찾기
- * file 관련 명령어를 찾으려면 → man -k file

- 실습

1. man 명령으로 man 의 사용법을 알아본다

- 명령어 일반형식

- option 과 매개변수
 - * ls -lF file1
 - ls 는 명령

- lF 는 option (-l -F 혹은 -Fl 로 해도 같은 의미)

- file1 은 매개변수

- * 모든 명령 option과 매개변수 사이에는 공백이 있어야 함

- 명령어 매뉴얼 상에 [] 표시 → 선택사항

- 기호 [] 이외에 것은 반드시 입력해야 함

- 안되는 명령

- * ex.) ls -lz file1 file2 → z option 이 없다

- * ex.) ls file1 -l file2 → option은 매개변수 전에 와야 함

- 실습

1. 위의 명령어 형식에서 맞는 것과 틀린 것을 모두 수행시켜본다.

- 여러 명령어를 함께 실행시키기

- 여러 명령어를 ; 기호로 분리하여 입력함

- 예) cal 1 1999; date

- * 순차적으로 cal 이 수행되고 date 가 수행됨

- 여러 명령어를 & 기호로 분리하여 입력함

- 예) cal 1 1999 & date &

- * & 는 background 로 수행됨으로 두 개의 프로세스가 동시에 수행됨

- 앞의 명령어의 수행 결과에 따라 뒤 명령어의 수행 여부 결정하기

- * && 기호 사용

- * 예) cal 1 1999 && date

- cal 명령이 성공적으로 끝나야 (즉 return 값이 0 이어야) date 수행

- * 예) cal 1 0 && date

- cal 이 성공적으로 끝나지 않음으로 date 도 실행되지 않음

- * || 기호 사용

- * 예) cal 1 1999 || date

- cal 명령이 성공적으로 끝나지 않아야 (즉 return 값이 0 이 아니어야) date 수행

- * 예) cal 1 0 || date

- cal 명령이 성공적으로 끝나지 않음으로 date 실행됨

- 실습

1. 위의 예를 실행시켜보고 설명한데로 잘 수행되는지 확인해본다

- 터미널 사용

- TTY (teletype) : 모든 terminal 을 TTY 로 인식

- terminal type 에 따라 제어 문자가 다름

- 사용 terminal type setting 하기
 - * setenv TERM terminal_type
 - * ex.) setenv TERM vt100
- 사용중인 terminal type 보기
 - * echo \$TERM or env | grep TERM
- 제어문자의 등록 : /etc/termcap
- <ctrl> 글쇠
 - * 표시 : ^C → <ctrl> 과 C를 동시에 누름 → interrupt 가 걸림
 - * 특수 제어문자 표시 → stty all
- 입력시 사용하는 특수 글쇠들
 - * erase : 마지막 입력 문자를 지움 → 보통 ^H
 - * werase : 마지막 입력 단어를 지움 → 보통 ^W
 - * kill : 한줄 전체를 지움 → 보통 ^U
- 프로그램 중지 (interrupt)
 - * 보통 ^C 를 사용
- 화면 멈추기
 - * stop : 화면 scroll 중에 멈춤 → 보통 ^S
 - * start : 다시 시작 → 보통 ^Q
 - * 다른 방법 : 명령어 | more 혹은 명령어 | less
- 파일 끝 표시
 - * eof (end of file) → 보통 ^D
- shell 과 eof
 - * shell 상에서 ^D → shell 을 빠져나감
 - * 빠져나감 방지 : set ignoreeof 를 실행 → ^D 무시
- terminal 글쇠 새로 설정
 - * stty all : 현재 setting 된 모든 글쇠를 보여줌
 - * 설정 : stty kill ^U → kill 을 ^U로 설정
- linefeed 와 newline
 - * linefeed : ^M → 첫 문자로
 - * newline : ^J → 다음줄로
 - * ^M^J : 다음줄 첫글자로
 - * Unix 에서는 ^J만 사용
 - PC 파일 → Unix 로 : ^M^J 를 ^J 로 바꿔야함
 - Unix 파일 → PC 로 : ^J 를 ^M^J로 바꿔야함
- stty sane
 - * terminal 의 설정을 원래의 것으로 재생시킴

- 실습
 1. 자신이 사용하고 있는 terminal 이 어떤 것인지 확인해본다
 - tty (return)
 2. 현재 설정된 terminal type 이 무엇인지 알아본다
 3. ^C 의 기능을 알아본다
 - sleep 1000 (return) → 1000초 동안 기다림
 - ^C 를 눌러 sleep 프로세스를 종료시킴
 4. 입력시 사용되는 특수 글쇠가 어떻게 지정되어 있는지 알아본다
 - stty all (return) 혹은 stty -a (return)
 - erase 를 ^H 로 바꾸어보고 동작을 확인한다
 5. 화면 멈추기 기능을 테스트해 본다
 - cat /etc/hosts 를 하고 파일이 끝나기 전에 ^S 를 누름 → 화면 scroll 이 멈춰나감
 - ^Q → 다시 scroll 됨
 - more /etc/hosts 로 한 화면씩 본다
 6. ^D 기능을 테스트해 본다
 - terminal 상에서 ^D (return) → terminal 이 사라짐
 - terminal 이 종료되는 것 방지하고 다시 ^D 를 눌러본다

제 2 절 Unix 파일 시스템 기본 사항

2.1 File 구조

- 디스크나 테이프에 저장된 데이터의 집합
- 모든 물리적인 구동장치 → ex.) 자판, 모니터, 프린터
- 세 가지 유형
 - 일반파일 : 일반적인 데이터 파일
 - * 워드 프로세싱 프로그램, 문서편집기, 컴파일러들에 의해 생성됨
 - 디렉토리 : 일반파일을 관리하는 파일
 - 특수파일 : 물리적인 장치파일
- 문서파일과 2진파일
 - 문서 (text) 파일 : ASCII 문자로만 구성된 파일
 - 2진 (binary) 파일 : 문자가 아닌 데이터가 들어있는 일반파일
 - 문서파일은 7bit의 ASCII code 사용
 - 2진 파일은 8bit의 binary code 사용
 - bit (binary digit의 약자) : 2진 데이터 0 또는 1을 저장하는 최소 단위

- byte : 8bit 는 1 byte
- 디렉토리 와 서브디렉토리
 - 디렉토리 (directory) : 파일이나 다른 디렉토리에 대한 정보를 갖고 있는 파일
 - 서브 디렉토리 : 다른 디렉토리 밑에 설정된 directory
- 나무 구조의 파일 시스템
 - root directory : 맨 위의 directory → / 로 표시
 - root 밑의 sub-directory 표시 → /home, /usr, /etc 등
- root directory
 - 유닉의 kernel 이 있음
 - /bin : 유닉스의 많은 명령어가 저장되어 있음
 - /dev : 물리적인 장치 파일들
 - /etc : 시스템을 관리하는데 필요한 프로그램과 파일이 있음
 - /lib : 라이브러리 파일들
 - /lost+found : 이상한 파일이 발견되면 여기에 저장
 - /sys : 시스템 소스파일 보관
 - /tmp : 임시 저장용 → 다음 booting 시 자동 삭제
- /dev 밑에 있는 특수파일들
 - ls /dev
 - tty → 터미널
 - cp file /dev/pts/5 → 해당 터미널을 쓰고 있는 사람에게 파일의 내용이 보임
 - /dev/null → 없는 장치 (불필요한 출력을 보낼때 사용)
- /usr directory
 - /usr/bin : 실행파일 보관
 - /usr/dict : 유닉스 사전 보관
 - /usr/games : game 프로그램 보관
 - /usr/include : include 파일들 보관
 - /usr/lib : library 파일들 보관
 - /usr/local : 시스템 관리자 사용
 - /usr/man : 온라인 매뉴얼 보관
 - /usr/spool : 장기 대기하고 있는 파일들
 - /usr/src : 소스 프로그램 보관
 - /usr/uucb : 베클리 대학에서 만든 프로그램들 보관

- home directory
 - 특정 사용자가 사용할수 있게 만든 directory
 - 보통 /usr 밑에 있음, 다른곳도 가능
 - \$HOME 에 자신의 home directory 명이 있음
 - ~→csh 에서 자신의 home directory 를 가리키는 약어
 - ex.)
 - * cd ~→home directory 로 이동
 - * cp ~/test/file1 /tmp →home directory 밑의 test 밑에 file 1 을 /tmp 로 copy
- 실습
 1. root directory 로 이동해 본다
 2. root directory 밑에 어떤 directory 가 있는지 살펴본다
 3. root directory 밑에 usr directory 로 이동해본다
 4. 자신의 home directory 로 이동해본다

2.2 directory

- 경로명과 작업 directory
 - 경로명 : / 에서 시작 특정 directory 및 file 을 가리킴
 - ex.) /usr/dict/words
 - 특정파일 access : 프로그램명 경로명/파일명
 - ex.) vi /usr/dict/words
- 절대 경로명과 상대경로명
 - 절대 경로명 : /로 시작해서 가리키는 경로명
 - 상대 경로명 : 현재의 directory 와 상관해서 가리키는 경로명
- 세가지 경로명 약어
 - .. : 바로 위 directory
 - . : 현재의 directory
 - ~: 현재 사용자의 home directory
 - ~user-id : user-id 의 홈디렉토리
- directory 사이의 이동
 - cd (change directory)
 - 형식 : cd directory
 - pwd (print working directory) → 현재의 directory 표시

- ex.)
 - * cd /bin → /bin directory로 이동
 - * cd ~→ home directory로 이동
 - * cd → home directory로 이동
 - * cd .. → 웃 directory로 이동

● 실습

1. 자신의 home directory에서 상대 경로명으로 root directory까지 가본다
2. 자신의 home directory에서 절대 경로명으로 root directory까지 가본다
3. 자신의 home directory가 어디에 있는지 확인해 본다
4. root 밑에 tmp directory로 절대 경로명으로 가본다
5. root 밑에 tmp directory로 상대 경로명으로 가본다

● 새로운 directory 만들기

```
[perfect:/usr2/97/102]# mkdir kk
[perfect:/usr2/97/102]# ls
./ .Xdefaults .netscape/ hw/
../.cshrc .wastebasket/ hw1/
.Xauthority .login Makefile hw3/
[perfect:/usr2/97/102]#
```

```
[perfect:/usr2/97/102]# ls -l
total 17
drwxr-xr-x 8 jeong 512 Jul 23 19:43 .
drwxr-xr-x 271 root 3584 Apr 11 10:41 ..
-rw-r--r-- 1 jeong 202 Jun 25 14:44 .Xauthority
-rw-r--r-- 1 jeong 167 Apr 8 14:07 .Xdefaults
-rw-r--r-- 1 jeong 1026 May 19 1998 .cshrc
-rw-r--r-- 1 jeong 122 May 19 1998 .login
drwxr-xr-x 2 jeong 512 Jul 16 11:59 .netscape/
drwxr-xr-x 2 jeong 512 Jun 25 14:30 .wastebasket/
-rw-r--r-- 1 jeong 173 May 6 1998 Makefile
drwxr-xr-x 2 jeong 512 Jul 23 19:34 hw/
drwxr-xr-x 2 jeong 512 Jul 23 19:19 hw1/
drwxr-xr-x 2 jeong 512 Jul 23 19:38 hw3/
drwxr-xr-x 2 jeong 512 Jul 23 19:43 kk/
[perfect:/usr2/97/102]#
```

- 형식 : mkdir directory...
- ex.) mkdir test → test라는 이름의 directory를 현재의 directory 밑에 만듦
- 주의사항
 - * 같은 directory 밑에는 똑같은 이름의 directory를 만드는 것은 불가
 - * 다른 directory에서는 가능

● directory 삭제

```
[perfect:/usr2/97/102/kk]# ls
./ .. stack.o
[perfect:/usr2/97/102/kk]# cd..
[perfect:/usr2/97/102]# rmdir kk
rmdir: kk: Directory not empty
[perfect:/usr2/97/102]# cd kk
[perfect:/usr2/97/102/kk]# rm stack.o
rm: remove stack.o? y
[perfect:/usr2/97/102/kk]# cd..
[perfect:/usr2/97/102]# rmdir kk
[perfect:/usr2/97/102]# ls
./ .Xdefaults .netscape/ hw/
../.cshrc .wastebasket/ hw1/
.Xauthority .login Makefile hw3/
[perfect:/usr2/97/102]#
```

- 형식 : rmdir directory...
- 비어있지 않은 directory는 지울 수 없다 → 안전을 위해서
- 비어있지 않은 directory를 지울 때 → rm -r directory

● directory의 이동 또는 이름 바꾸기

- 형식 : mv directory target
- target directory가 있으면 directory를 target 밑으로 옮김
- target directory가 없으면 directory의 이름을 target으로 변경
- mv는 sub-directory도 모두 옮김

● 실습

1. 자신의 home directory 밑에 test라는 directory를 만든다
2. /etc/hosts 파일을 현재 디렉토리로 copy 한다.
3. hosts 파일의 이름을 unix로 바꾼다.
4. unix 파일을 test 디렉토리로 옮긴다.
5. test2 디렉토리 생성 한다.

6. test2 디렉토리 이름을 mydir 로 변경한다.

7. mydir 을 test 로 옮긴다.

- 디렉토리의 내용 살펴보기

- 행으로 표시 →ls -c
- 종으로 표시 →ls -l
- 이름 역순으로 표시 →ls -r
- 파일형식 →ls -F (directory 는 이름 끝에 / 를 붙임, 실행프로그램은 *를 붙임)
- 모든 sub directory 표시 →ls -R

- 숨겨진 파일

- . 으로 이름이 시작되는 파일
- ls -a 로만 보임

- 실습

1. 위의 예들을 수행시켜본다

제 3 절 파일 명령어

3.1 기본명령

- 와일드 카드, 파일이름 치환

표 2.2: wildcard 요약

*	0 또는 그 이상의 연속적인 모든 문자와 대응
?	모든 문자하나와 대응
[]	묶여진 문자중의 하나와 대응

```
텔넷 -perfect,hansung.ac.kr
연결(C) 편집(E) 터미널(I) 도움말(H)
[perfect:/usr2/97/102/hw]# ls test.?
test.c test.o
[perfect:/usr2/97/102/hw]#
```

```
텔넷 -perfect,hansung.ac.kr
연결(C) 편집(E) 터미널(I) 도움말(H)
[perfect:/usr2/97/102/hw]# ls t??t.?
test.c test.o
[perfect:/usr2/97/102/hw]#
```

```
텔넷 -perfect,hansung.ac.kr
연결(C) 편집(E) 터미널(I) 도움말(H)
[perfect:/usr2/97/102/hw]# ls te*
test* test.c test.o
[perfect:/usr2/97/102/hw]#
```

```
텔넷 -perfect,hansung.ac.kr
연결(C) 편집(E) 터미널(I) 도움말(H)
[perfect:/usr2/97/102/hw]# ls *.c
stack.c test.c
[perfect:/usr2/97/102/hw]#
```

```
[perfect:/usr2/97/102/hw]# ls *t*
stack.c stack.o test* test.c test.o type.h
[perfect:/usr2/97/102/hw]#
```

- ex.)
 - * ls tm* → tm 으로 시작되는 모든파일들
 - * ls tm? → tm 으로 시작되고 한문자가 더있는 파일들
 - * ls tm[pt] → tmp 나 tmt 파일
 - * ls /bin/[a-z][a-z] → /bin 밑에 소문자 2글자의 파일들
 - * ls ttt.{bak,bat} → ttt.bak 이나 ttt.bat 파일
 - * ls {old,new}data{1,2,3} → olddata1, olddata2, olddata3, newdata1, newdata2, newdata3 파일
- jesc; 와 ^d
 - C 쉘에서 파일이름을 자동완성하는 기능 → jesc;
 - * 예1)
 - 현재 디렉토리에 project.information 이라는 파일이 있다고 가정
 - % cat projjesc; → cat project.information 으로 자동완성
 - * 예2)
 - 현재 디렉토리에 project.information 과 project_contents 라는 파일이 있다고 가정
 - % cat projjesc; → cat project_ 까지만 완성
 - % cat project_jjesc; 하면 → cat project.information 으로 자동 완성
 - * tcsh 에서는 jjtab; 키를 사용
 - ^d → 여러개의 매칭파일을 완성하지 않고 보여줌
 - * 예1)
 - 현재 디렉토리에 project.information 과 project_contents 라는 파일이 있다고 가정
 - % cat projctrl-d; → proj 로 시작하는 파일이 project.information 과 project_contents 가 있음을 보여줌
 - 파일 만들기
 - vi memo
 - ls > listing

- cp data extra
 - 파일 날자변경 : touch filename (touch 한 시점의 날자로 변경됨)
- 파일 이름의 선택
 - 255자까지 가능
 - 공백을 넣고 싶을경우 공백앞에 를 붙임
 - ex.) vi test\ do
- 파일의 복사
 - 형식 : cp [-ip] file1 file2
 - ex.)
 - * cp data extra
 - * cp -i data extra → extra 라는 파일이 있을경우 overwrite 할지 문의
 - * cp -p data extra →원래의 파일과 같은 시각, 사용권으로

```
[perfect:/usr2/97/102/hw]# ls
./ Makefile test* test.o type.h
../ stack.o test.c
[perfect:/usr2/97/102/hw]# cp test.c test3.c
[perfect:/usr2/97/102/hw]# ls
./ Makefile test* test.o type.h
../ stack.o test.c test3.c
[perfect:/usr2/97/102/hw]#
```

```
[perfect:/usr2/97/102/hw]# ls
./ Makefile test* test.o type.h
../ stack.o test.c
[perfect:/usr2/97/102/hw]# cp test3.c test.c ../../hw3
cp: t/../.hw3: No such file or directory
[perfect:/usr2/97/102/hw]# cp test3.c /usr2/97/102/hw3
[perfect:/usr2/97/102/hw]# cd ..
[perfect:/usr2/97/102/hw]# cd hw3
[perfect:/usr2/97/102/hw3]# ls
./ Makefile link.o test* test.o type.h
../ link.c
[perfect:/usr2/97/102/hw3]#
```

- 파일의 앞부분 보기
 - 형식 : head [-count] [file_name ...]
 - default 로 10줄 보여줌

```
[perfect:/usr2/97/102/hw]# head -4 test.c
#include      <stdlib.h>
#include      <stdio.h>
#include      <string.h>
#include      "type.h"
[perfect:/usr2/97/102/hw]#
```

```
[perfect:/usr2/97/102/hw]# tail -10 test.c
}
    Stack_Pop(stk2, &top2);
}
if(!Stack_IsEmpty(top2)){
    printf("Error: Stack is not empty\n");
    return 0;
}
printf("You win!!!!!!\n");
return 0;
[perfect:/usr2/97/102/hw]#
```

- 20줄을 보려고 할때 head -20 file_name

- 파일의 끝부분보기

- 형식 : tail [+ startfr] [file_name]
- default로 10줄 보여줌
- 파일 시작 35줄부터 끝까지 →tail +35 file_name
- 파일의 끝에서 20줄 →tail -20 file_name
- 역순으로 보고 싶을때 →tail -20r file_name
- 파일의 끝에서 기다림, 파일 내용이 추가되면 보여줌, 끝낼때 ^C
→tail -f file_name
 - * running 중인 프로그램 결과 file 을 file_name에 저장중일때 편리

- 페이지ing 프로그램

- more

```
[perfect:/usr2/97/102/hw]# more -10 test.c
#include      <stdlib.h>
#include      <stdio.h>
#include      <string.h>
#include      "type.h"

main()
{
    int          i;
    token_t      x;
    token_t      stk1[MAX_STACK];
--More--(17%)
```

- * 형식 : more [-cs] [+startline] [+pattern] [file...]

- * space : 다음 페이지로

- * -h : 사용법 보여줌

- * -c : 화면 출력을 맨위부터

- * -s 공백여러줄을 하나의 공백줄로

- * +/pattern : pattern 이 있는줄 두 줄위부터 출력

- pg

- * 형식 : pg [-cn] [+startline] [+pattern] [file...]

- * return : 다음 페이지로

- * -h : 사용법 보여줌

- less
 - * 형식 : less [-cmsCM] [-xtab][+command] [file...]
 - * space : 다음 페이지로
 - * 종료 : q 누름
 - * -h : 사용법 보여줌
 - * -m : more 처럼 보여준 line 을 %로 표시
 - * -M : 많은 정보를 보여줌
- 페이징 환경변수 설정
 - 특정 paging 프로그램으로 설정하기 → setenv PAGER less
 - .login 에 넣어놓으면 login 할 때 자동으로 설정됨
 - option 설정 (csh 에서)
 - * setenv MORE '-cs'
 - * setenv LESS '-seM'
 - option 설정 (Bourne shell 에서)
 - * PAGER=more; export PAGER
 - * MORE='-cs'; export MORE
- 파일을 다른 directory에 복사
 - 형식 : cp [-ip] file... directory
 - ex.)
 - * cp data dir1
 - * cp data1 data2 data2 dir2
- directory 복사
 - 형식 : cp -r [-ip] directory1 directory2
- 파일의 이동
 - 형식 : mv [-if] file... directory
 - -i : overwite 물어봄
 - -f : 강제로 덮어씀
 - ex.) mv data1 data2 dir1
- 파일의 이름변경
 - 형식 : mv [-if] oldname newname
- 파일의 삭제
 - 형식 : rm [-fir] file

- 파일의 잘못 삭제 방지
 - rm -i file → 지우기 전에 사용자에게 지울지 묻는다.
- 파일의 크기 보기
 - 파일의 라인수, 단어수, 문자수를 보여줌
 - 명령 : wc [-cmlw] files...
 - * -c: count bytes
 - * -m: count characters
 - * -l: count lines
 - * -w: count words
 - * 두 개 이상의 파일이면 total 을 보여줌

```
텔넷 -perfect.hansung.ac.kr
연결(C) 편집(E) 터미널(T) 도움말(H)
[perfect:/usr2/97/102/hw3]# wc -l test.c
41 test.c
[perfect:/usr2/97/102/hw3]#
```

```
텔넷 -perfect.hansung.ac.kr
연결(C) 편집(E) 터미널(T) 도움말(H)
[perfect:/usr2/97/102/hw3]# wc -w test.c
80 test.c
[perfect:/usr2/97/102/hw3]#
```

- directory 지우기
 - rm -r dir...
 - ex.)
 - * rm -r dir
 - * rm -ir dir → 물어보면서 지움
 - * rm -rf dir → 무조건 지움

```
[perfect:/usr2/97/102/hw3]# wc -c test.c
757 test.c
[perfect:/usr2/97/102/hw3]#
```

```
[perfect:/usr2/97/102/hw3]# wc test.c
41 80 757 test.c
[perfect:/usr2/97/102/hw3]#
```

```
[perfect:/usr2/97/102/hw3]# wc test.c
41 80 757 test.c
[perfect:/usr2/97/102/hw3]#
```

- 주의사항: 삭제된 파일은 절대로 되살릴수 없다.

- 실습

- test라는 directory를 만든다
- ls를 이용하여 새로운 이름의 파일을 test라는 directory에 만든다
- /etc/hosts라는 파일을 test라는 directory에 copy 한다
- cp -r 명령으로 test라는 directory 전체를 test_unix라는 이름의 directory로 copy 한다
- mv 명령으로 test 밑에 있는 hosts 파일을 test_unix directory 밑으로 옮긴다
- rm -r 명령으로 test라는 directory를 지운다

3.2 파일의 권한

- 파일의 사용권한

- 다른 사용자의 접근으로부터 보호
- 자신의 실수로 부터 보호
- 파일의 사용권한의 요약

표 2.3: 파일 사용권한

일반 파일	읽기 : 파일을 읽을수 있다 쓰기 : 파일에 쓸수 있다 실행 : 파일을 실행할수 있다
디렉토리	읽기 : 디렉토리를 읽을수 있다 쓰기 : 디렉토리에 파일을 만들고, 이동하고, 복사하고 삭제할수 있다 실행 : 디렉토리를 탐색 할수 있다

- Unix의 파일 관리 방법

- 사용권한
 - * 자신(rwx), 그룹(rwx), 모든사람(rwx)
 - * 그룹 id 목록 : /etc/group에 있음
 - * id 보기 →id (return)

- 파일 권한 보기

- ls -l → 파일인 경우
- ls -ld → directory인 경우

- 파일 모드

- 읽기권한 → 4

- 쓰기권한 → 2
- 실행권한 → 1
- 권한없음 → 0
- ex.)
 - * 사용자 : 읽기 + 쓰기 + 실행 → 7
 - * 그룹 : 읽기 + 쓰기 → 6
 - * 전체 : 읽기 → 4

- 파일 권한의 변경

- 형식 : chmod mode file...
- ex.)
 - * chmod 764 filename

- 기호에 의한 변경

- 변경기호

표 2.4: 모드 변경 기호

기호	의미
u	소유자
g	소유자가 속한 그룹의 사용자들 (/etc/group에 그룹이 있음)
o	그밖의 사용자들
a	모든 사용자들
+	추가
-	삭제
=	지정
r	읽기
w	쓰기
x	실행 및 접근

- 예)
 - * chmod o+wx file_name → 그밖의 사용자들에게 쓰기 실행권한 추가
 - * chmod u+r file_name → 소유자에게 읽기 권한 추가
 - * chmod a-w file_name → 모든 사용자들에게 쓰기권한 해제
 - * chmod a=w file_name → 모든 사용자들에게 쓰기권한 지정

- 실습

1. 자신의 home directory 밑에 있는 파일들의 permission 을 본다
2. /etc/hosts 를 home directory에 test_hosts라는 이름으로 copy 한후 permission 을 본다
3. test_hosts라는 파일의 permission 을 나만읽고 쓰기 가능하게 한다

4. test_hosts라는 파일의 permission 을 위의 예) 에서와 같은 방법으로 조정한 노트에 정리해본다

- Unix의 사용권한 부여

- 새로 만들어지는 파일에 대한 자동권한 부여방법
- 형식 : umask [mode]
- ex.) 모든 그룹과 전체 사용자에게 쓰기권한 배제 → umask 022

- 사용자 및 그룹 바꾸기

- 사용자 바꾸기: chown 명령 (super user 만이 할수 있음)
 - 예)
 - * chown login_id file_name
 - * chown -R login_id dir_name (모든 subdirectory 포함)
- 그룹 바꾸기: chgrp 명령 (super user 만이 할수 있음)
 - 예)
 - * chgrp group_id file_name
 - * chgrp -R login_id dir_name (모든 subdirectory 포함)

제 4 절 vi 편집기

4.1 시작 및 종료

- Unix 표준 편집기
- vi 의 실행
 - 형식 : vi [filename]
- 명령모드와 입력모드
 - 명령모드 : 작업명령
 - 입력모드 : 글자 입력
 - 입력모드에서 명령모드로 가는 방법은 <Esc> 입력
 - 명령모드에서 입력모드로 가는 방법은 입력명령 입력
- 읽기 전용 vi 편집기
 - vi -R [filename] 혹은 view [filename]
 - 읽을수는 있으나 쓸수는 없다
- 시스템 다운된후 되살리기
 - vi 사용도중 시스템 다운되면 다음 booting 후 vi -r 하면 되살림

- vi 는 email 로 되살릴 파일이 있음을 알림

- vi 종료하기
 - 명령모드에서
 - * ZZ 를 누름 →저장후 종료
 - * :q! →저장하지 않고 종료

- 화면 사용
 - 화면을 다시 보여줄때 →^L
 - 입력모드에서 ^C 를 입력하고 싶을때 →^V^C 를 누름

- vi 의 사용과 ex 명령어
 - vi 에서 ex 명령 사용시
 - ex.)
 - * :1,5 d →1줄에서 5줄까지를 지움
 - * :s/harley/Harley/g →모든 harley 글자를 Harley 로 바꿈

- 명령어 요약

표 2.5: vi 에서 편집 시작, 저장 및 종료

vi filename	vi 를 시작하고 filename 을 편집
vi	파일명 칭을 명시하지 않은 채 vi 시작
:e filename	편집 중 새로운 파일 filename 을 편집
:e! filename	편집 중 파일이 저장되지 않았어도 강제로 새로운 파일을 편집
ZZ	편집한 파일을 저장하며 vi 를 종료
:q!	변경내용 저장없이 vi 를 종료
:w	vi 에 있으면서 편집한 파일을 저장
:w filename	filename 의 이름으로 파일을 저장
:w && filename	filename 에 편집버퍼의 내용을 추가
:1,20 w filename	1행부터 20행까지를 filename 의 이름으로 저장
:wq	저장 후 종료, ZZ 와 같은 기능
:r filename	filename 을 읽어서 삽입
vi -R filename	읽기 전용 모드로 시작
vi -r filename	비정상적인 종료 후 저장되지 않은 내용 복구

4.2 문서작성

- vi 에서 글자 입력

1. 데이터를 쓰고 싶은곳으로 커서 이동

2. 입력모드 명령

3. 데이터 입력

4. 명령모드로 바꾸려면 <Esc> 누름

- 커서의 이동

표 2.6: 커서 이동 명령들

명령	의미
h	한 칸 왼쪽
j	한 칸 아래쪽
k	한 칸 위쪽
l	한 칸 오른쪽
-	앞줄의 처음
+	다음줄의 처음
return	다음줄의 처음
0	커서를 맨 앞으로
\$	커서를 맨 끝으로
^	커서를 현줄의 첫글자로 (tab 이나 공백이 아닌)
w	커서를 다음단어의 첫글자로
e	커서를 다음단어의 끝글자로
b	커서를 이전단어의 첫글자로
W, E, B	문장부호 무시 (그외 w,e,b와 같은 동작)
)	다음 문장의 처음 (문장은 . 으로 구분)
(이전 문장의 처음
}	다음 문단의 처음 (문단은 공백 줄로 구분)
{	이전 문단의 처음
H	커서를 화면 맨 위로
M	커서를 화면 중간으로
L	커서를 화면 맨 아래로

* number 명령 →해당 명령을 number 만큼 반복 실행

- ex.) 10w →10단어 뒤

- 편집 버퍼를 통한 이동

표 2.7: 편집버퍼를 통한 이동 명령

명령	의미
^F	한 화면 아래로 이동
^B	한 화면 위로 이동
^D	반 화면 아래로 이동
^U	반 화면 위로 이동

- 패턴 (pattern) 의 검색

- /검색어 → 아래로 검색어를 찾음
- ? 검색어 → 위로 검색어를 찾음
- 똑같은 패턴을 반복 찾을 경우 → / 나 ? 만 입력
- 같은 방향 같은 검색어 재검색 → n
- 역 방향 같은 검색어 재검색 → N

- 정규 표현식

표 2.8: 정규표현식에 사용되는 기호들

기호	의미
.	newline 을 제외한 모든 단일 문자와 대응
0	0 또는 그이상의 문자와 대응
^	줄의 시작과 대응
\$	줄의 끝과 대응
\<	단어의 시작과 대응
\>	단어의 끝과 대응
\[]	묶여진 문자의 하나와 대응
[^]	묶여진 문자를 제외한 이외의 것과 대응
\	이어지는 기호를 문자그대로 해석

- /H.. → H 다음에 임의의 두단어가 나오는것 검색
- /H[a-z][a-z] → H 다음에 두개의 소문자가 나오는것 검색
- /^Harley → Harley 로 시작하는 줄 검색

- 줄 번호 사용하기

- :set number → 줄 설정
- :set nonumber → 줄 설정 없앰
- 특정 줄로 가기 → 줄번호G
- ex.)
 - * 100G → 100번째 줄로가기
 - * G → 파일의 맨 끝으로 가기

- 데이터 추가

- 내용 고치기

- ex.)
 - * c4b → 현 커서부터 4단어 뒤로 변경
 - * c6} → 현 커서부터 6개의 문단 변경

- 치환

표 2.9: 문자 입력 명령

명령	모드	의미
i	입력모드로 전환됨	커서위치 앞에서 삽입
a	입력모드로 전환됨	커서위치 뒤에서 삽입
I	입력모드로 전환됨	현재줄의 맨앞에서 삽입
A	입력모드로 전환됨	현재줄의 맨끝에서 삽입
o	입력모드로 전환됨	현재줄 아래에서 삽입
O	입력모드로 전환됨	현재줄 위에서 삽입

표 2.10: 문자 입력 명령

명령	의미	예
r	한문자 대치	ra → 커서의 단어를 a로 치환
R	커서부터 원하는곳까지 대치	R하고 계속입력
s	하나의 문자를 여러문자로 대치	s하면 커서단어에 입력문자들로 대치
C	커서부터 줄끝까지 대치	C하면 줄끝까지 입력문자들로 대치
S, cc	한줄대치	커서의 줄을 지우고 입력문자들로 대치

표 2.11: 치환 명령 예

명령	의미
:s/UNIX/Unix/g	커서가 있는 줄의 UNIX라는 문자를 Unix로 모두 대치
:s/UNIX/Unix/c	대치전 대치할것인지 물어봄
:s/UNIX//g	UNIX를 모두 없앰
:57s/UNIX/Unix	57번째 line 만 대치
:57,60s/UNIX/Unix	57에서 60번째 줄까지에서 UNIX를 Unix로 대치
::,\$s/UNIX/Unix/g	현재 커서부터 끝까지를 모두 대치
:1,s/UNIX/Unix/g	첫째줄부터 현재커서까지 모두 대치
:%s/UNIX/Unix/g	모든 줄에서 모두 대치

- 형식 →:s/pattern/replace/
 - ex.)
- 고치기 취소 또는 반복
 - 바로 전 명령 취소 →u →u 명령 취소 →u
 - 해당 줄에서 행한 모든 명령 취소 →U
 - . →마지막 명령 반복
- 대소문자 바꾸기
 - ~→해당 글자로 옮기고 ~를 치면 대 ↔ 소로 변환됨
- 줄 길이의 조정
 - 줄을 나눌때 : 나누고 싶은곳으로 커서 이동 후 r return
 - 줄을 합칠때 : J 입력 → 아래줄이 커서줄로 붙음
- 데이터 지우기

명령	의미
x	커서 문자 지우기
X	커서 왼쪽 문자 지우기
D	커서부터 줄끝까지 지우기
dmove	커서부터 move 까지 지우기 (move는 위치 지정 명령)
dd	한 줄 지우기
:lined	지정줄 삭제
:line,lined	지정 줄 범위 삭제
dw	한단어 삭제
d10w	10단어 삭제
db	한단어 뒤로 삭제
d2)	두문장 삭제
d5}	5문단 삭제
dG	커서부터 맨끝까지 삭제
d1G	커서부터 맨앞까지 삭제

- 마지막으로 지운것 복사하기
 - p : 마지막으로 지운것을 커서 뒤/아래 삽입 (단어/줄)
 - P : 마지막으로 지운것을 커서 앞/위 삽입 (단어/줄)
- 여러줄 복사와 이동
 - co (copy), m (move) 이용

- ex.)
 - * :5co10 → 5번줄 복사 10번줄 밑에 삽입
 - * :4,8co20 → 4 ~ 8 줄 복사 20번줄 밑에 삽입
 - * :5m10 → 5번줄 10번줄 밑으로 이동
 - * :4,8m20 → 4 ~ 8 줄을 20번줄 밑으로 이동
- y (yank) 명령 → 원하는곳에 p나 P로 삽입
 - * yy → 커서줄 복사
 - * yw → 커서 word 복사
 - * 3yy → 커서에서 3줄 복사
- shell 명령의 사용
 - !: shell 명령
 - ex.)
 - * ! date → date 명령 실행 결과를 보여줌
 - * ! sh → sh로 구동 (되돌아가기 ^D 혹은 exit 입력)
 - * ! csh → csh로 구동 (되돌아가기 ^D 혹은 exit 입력)

- 편집 버퍼로 data 읽어오기
 - 다른 file 읽어오기
 - 형식 →:r filename → 현 커서 위치에 삽입
 - ex.)
 - * :10r info → info라는 파일의 내용을 10번째 줄부터 삽입
 - * :r !ls → ls 실행 결과를 커서위치에 삽입
 - * :r ! date → date 실행 결과를 커서 밑줄로 입력
 - * :r ! ls -l → ls -l 실행결과를 커서 밑줄로 입력
- 데이터를 처리하기위한 shell 명령의 사용
 - 데이터 파일의 특정 영역에 대한 명령수행
 - 형식 → 줄수!!명령
 - ex.)
 - * 5!!sort → 현재줄부터 5개의 줄을 sort하고 대체함
 - * :1,10lsort → 1 ~ 10 줄을 sort하고 대체함
- 임의의 줄에 marking 하고 가기
 - 커서가 있는줄에서 ma 누름 → 그 줄이 a라는 이름으로 marking됨
 - marking 된 줄로 가기 → 'a
 - ex.)
 - * :, 'a d → 커서줄 부터 marking a 줄까지 지우기

- * ..,'a s/test/TEST/g →커서줄부터 marking a 줄까지 test를 TEST로 모두 바꾸기
- 데이터 파일로 저장하기
 - :w →원래의 파일이름으로 저장
 - :w filename →filename의 이름으로 저장
 - :w >> filename →filename에 추가
 - :1,20 w filename →1~20줄을 filename으로 저장
- 편집중 파일 바꾸기
 - :e filename →저장하지 않았을 경우 경고문구 나옴
 - :e! filename →현 file 수정 저장않고 바꾸기
 - 원래의 파일로 돌아가기 →^^
- vi의 초기화
 - home directory에 .exrc file에서 읽어서 setting
 - ex.) ~/.exrc에 다음과 같은것을 넣어놓음
 - * set ts=4
 - * set autoindent
- 문서의 인쇄
 - 개요
 - * 여러 사람이 공유 →모든 작업에 작업번호로 식별
 - * 작업번호 보기 →lpq
 - 스팔링 (spooling)
 - * print queue로 인쇄작업 목록 관리
 - * spooling : 바로 인쇄하지 않고 차례차례 인쇄하는 방식
 - * spool (simultaneous peripheral operations offline)의 약자
 - daemon과 dragon
 - * daemon : 항상 대기하고 있다가 해당 작업의 요청이 있으면 수행하는 프로그램
 - * ex.) lpd (printer daemon)
 - * dragon : 백그라운드에서 어떤 임무를 수행하면서 대기중인 daemon
 - 파일 형식 맞추기
 - * 형식 : pr [-h title] [-l pagelength] [file...]
 - * ex.) pr memo | lpr
 - * nl : 줄번호 생성기
 - * ex.) nl memo | pr | lpr

- 파일의 인쇄
 - * 형식 : lpr [-mprh] [-#num] [-J name] [-P printer] [-T title] [file...]
 - * ex.) lpr memo
 - * 보기 : lpq
 - * 지우기 : lprm
 - * 끝낸후 mail 얻기 : lpr -m memo
 - * 끝낸후 file 지우기 : lpr -r memo
 - * 특정 printer 인쇄 : lpr -P laser2 momo
 - * 특정 printer로 향시 지정 : setenv PRINTER laser1
 - * pr 실행후 찍기 : lpr -p memo
- 인쇄작업 점검
 - * lpq [-l] [job...] [-P printer]
- 인쇄작업 취소
 - * lprm [-] [job...] [-P printer]
 - * ex.)
 - lprm jobnumber →특정 작업만 지우기
 - lprm - →자신이 요구한 모든것 지우기
- 실습
 1. 위의 vi 명령어들을 사용하여 임의의 내용의 파일을 작성해본다.

제 5 절 파일 관련 명령어

5.1 파일 및 내용 검색 명령어

- find 명령어
 - 동작: 특정한 파일의 경로명을 찾아주거나 찾은 파일에 대하여 명령어 수행 사용
 - 형식: find path expression
 - expression 들
 - 예)
 - * find . -name "test.txt" -print
 - * →현재 directory 포함 모든 sub-directory에서 test.txt라는 이름의 파일을 찾아 path를 print 하라
 - * find / -type f -mtime -1 ! -name '*.o' -print
 - * →루트 디렉토리 이하의 모든 파일중에서 정규파일이면서 수정된지 하루가 지나지 않은 파일중 확장자가 o가 아닌 파일을 찾아서 경로명 출력
 - * find . \(\ -name test -o -name test2 \) -print

표 2.13: find 명령에서 expression들

-name filename	filename에 해당하는 파일
-type filetype	찾을 파일의 유형을 지정 b: 블록 특수 파일 c: 문자 특수 파일 d: 디렉토리 파일 f: 정규 파일 p: fifo (파이프) l: 심볼릭 링크
-user name	소유자가 name인 파일
-group name	그룹 이름이 name인 파일
-size n[c]	파일의 크기가 n 블록인 파일 (1블록은 512bytes) 파일의 크기가 n bytes인 파일을 찾을 경우 c option 추가
-atime n	파일이 마지막 접근된 시간이 n일 이전인 파일
-mtime n	파일이 마지막 수정된 시간이 n일 이전인 파일
-print	찾은 파일의 경로명을 출력
-exec command {} \;	찾은 파일에 대하여 command에 해당하는 명령을 수행 {}은 찾은 패스명
-ok command {} \;	해당 명령을 수행할지를 각각의 찾은 파일에 대하여 수행할지를 물어봄
-follow	심볼릭 링크도 검사함
-perm oct	파일의 액세스 권한이 oct와 일치하는 파일
!expression	expression의 논리부정
expr1 -a expr2	expr1과 expr2가 모두 참이 되는 파일
expr1 -o expr2	expr1이나 expr2 중 하나가 참이 되는 파일

- * → 현재 디렉토리 이하의 파일에서 이름이 test 이거나 test2인 파일을 찾아 경로명 출력
- * find ~shjung -name "host*" -print -exec cat {} \;
- * → shjung의 home directory 밑으로 모든 sub-directory에서 host라는 이름으로 시작되는 파일을 찾아 path를 print하고 내용을 출력하라

• grep 명령어

- 동작: 파일 내용에서 특정 단어의 패턴을 검색
- 형식: grep [options] pattern [filelist]
- option 들

표 2.14: grep 옵션들

-c	count option으로 일치하는 패턴의 갯수만 출력
-i	대소문자를 구분하지 않음
-l	일치되는 패턴이 있는 파일명을 한번만 표시
-n	패턴 발생 파일명과 행번호를 같이 표시
-v	패턴을 포함하지 않는 모든 행 출력

– 예)

- * grep UNIX test → test라는 파일에서 UNIX라는 패턴 검색
- * grep -n UNIX test → 패턴과 발생 행 표시
- * grep -i UNIX test test2 → 대소문자를 구별하지 않고 test와 test2 파일에서 UNIX 검색
- * grep -l main *.c → 모든 .c 파일에서 main이라는 패턴의 발생을 찾음
- * grep 'kr\$' * → 모든 파일에서 행의 끝이 kr로 끝나는 줄을 표시
- * grep 'profits for' * → profits for가 발생되는 행 표시 (패턴에 공백이 들어가면 반드시 "으로 묶어줌)
- * egrep 'this—that' * → egrep은 모든 정규표현식 사용 가능
- * fgrep 'this—that' * → fgrep은 정규표현식 사용 못함. 그러므로 this—that 패턴을 검색

5.2 파일 정렬, 합병, 비교, 축출 명령어

• sort 명령어

- 동작: 여러 개의 파일의 내용을 정렬하거나 합병할 때 사용
- 형식: sort [options] [field-specifier-list] [file-list]
- options 들
- 예)
 - * 다음의 내용으로 list라는 파일을 작성 (수행 결과는 직접 확인)

표 2.15: sort option들	
-b	여러개의 공백을 하나의 분리자로 인식
-c	파일이 정렬되어 있는지 검사
-d	사전순서로 정렬
-f	모든 소문자를 대문자로 취급
-i	ASCII 문자가 아닌 문자를 무시
-m	파일별로 정렬 후 파일의 합병을 수행
-n	산술적인 수치로 정렬
-o filename	정렬된 결과를 filename으로 저장
-r	역순으로 정렬
-u	동일행을 삭제
-tx	x를 분리자로 사용

```

Tom Winstrom 94201
janet Dempsey 94111
Alice MacLeod 94114
David Mack 94114
Toni Barnett 95020
jack Cooper 94072
Richard MacDonald 95510
* sort list → 첫번째 필드부터 정렬
* sort +1 list → 두번째 필드부터 정렬
* sort -f +1 list → 대소문자 구별 않고 정렬
* sort -b +2 list → 세번째 필드로 정렬하고 다중 공백은 무시
* sort -fb +2.3 → 공백무시 및 대소문자 구별 않고 세번째 필드의 3문자 건너뛰고 정렬
* 다음의 내용으로 names라는 파일을 작성 (수행 결과는 직접 확인)
10:1234:mike
-1:2345:mark
8:1990:julie
4:1493:louise
* sort -t: +2 -3 names → 구분자 :를 사용하여 세번째 필드로만 정렬
* sort -t: -n +0 -1 → 첫번째 필드를 숫자로 인식하여 정렬

```

• 파일 비교 명령

- comm 명령
 - * 동작: 두개의 파일을 라인단위로 비교후 결과 출력
 - * 형식: comm [options] file1 file2
 - * 예)
 - comm file1 file2 → 첫번째만 있는 라인 출력, 두번째만 있는 라인 출력, 공통라인 출력

- 위에서 세그룹의 출력정보는 tab으로 구분
 - comm -12 file1 file2 → 공통으로 있는 내용만 출력
- cmp 명령
 - * 동작: 두개의 파일 내용조사, 같으면 그냥 끝, 다르면 처음으로 다른곳 위치 출력
 - * 형식: cmp [options] file1 file2
 - * 예)
 - cmp -l file1 file2 → 일치하지 않는 바이트의 offset과 값을 보여줌
- diff 명령
 - * 동작: 두 파일의 차이점을 표시
 - * 형식: 4가지가 있음
 1. diff [options] file1 file2
 2. diff [options] file1 directory
 3. diff [options] directory file2
 4. diff [options] directory directory2
 - * 예)
 - diff file1 file2 → 두개의 파일의 내용을 비교
 - file1에만 있는 내용 ;로 표시후 출력
 - file2에만 있는 내용 ;로 표시후 출력
 - 두 파일을 같은 만드는데 필요한 명령 및 위치 출력: a(추가), d(삭제), c(변경)
- cut 명령
 - 동작: 파일에서 지정한 필드의 내용을 추출
 - 형식: cut [options] filename
 - 예)
 - cut -c12-30 file1 → file1 파일의 모든 행에서 12번째 문자부터 30행 문자까지를 추출
 - cut -c12-30,40-45 file1 → 두개의 범위의 문자를 추출
 - cut -d: -f2-4 file1 ; file2 → 구분자 :를 사용하여 필드 2번째 부터 4번째까지를 추출 file2에 저장
- paste 명령
 - 동작: 데이터를 열로 결합
 - 형식: paste [-d char] files
 - 예)
 - paste file1 file2 ; file3 → file1과 file2의 내용을 열로 결합 file3에 저장
 - paste -d ' ' file1 file2 ; file3 → 구 파일내용 사이에 공백을 삽입 (default ; tab ;)

- strings 명령
 - 동작: 2진파일안에 있는 프린트 가능한 ASCII 문자열을 찾아 보여줌
 - 예)
 - * strings /bin/cp → /bin/cp 파일에서 프린트 가능한 문자열을 찾아 보여줌
- tr 명령어
 - 동작: 표준입력으로 들어온 문자열을 다른 문자열로 대체, 문자열단위가 아니라 문자열내의 문자단위로 대체
 - 형식: tr [-ds] string1 [string2]
 - * -d option 은 해당 문자들을 삭제
 - * -s option 은 중복되는 문자 단일문자로 대체
 - 예)
 - * cat file1 → tr '[a-z]' '[A-Z]' → file1의 모든소문자를 대문자로 대체
 - * cat file1 → tr -d '[0-9]' & file2 → file1의 모든 숫자를 삭제하여 file2로 저장
 - * cat file1 → tr -s 'aef' → file1의 내용에서 a, e, f 가 중복발생하면 하나로 대체
 - * echo tennessee → tr -s 'tnse' 'srne' → 각각의 문자 대체 (출력은 serene)

제 6 절 X 윈도우

6.1 X-window 소개

- GUI (Graphic User Interface)
 - 문자뿐만 아니라 그림을 통하여 사용자와 정보를 교환하는 방법 → mouse 사용
 - 여러개의 window 사용 : 동시에 여러작업을 할때 편리
 - 대부분 mouse 를 click 함으로서 실행
- X-window 란?
 - Unix 상에서 GUI를 지원하기 위한 소프트웨어
 - PC 에서 windows 와 유사하나 OS 와 통합되어있지 않음
- 윈도우 관리자
 - window 들의 특성 제어및 관리
 - mwm : motif window manager
 - olwm : open look window manager
 - twm : tap window manager
- X를 이용한 원격 프로그램 실행

- network 상의 다른 컴퓨터의 X-client 를 실행
- 자신의 컴퓨터 상의 X-client 와 동시에 사용가능
- 가장 중요한 X-client
 - xterm : terminal 처럼 동작되게 하는 X-client
 - hanterm : 한글이 되는 xterm
- 그외의 X 프로그램들
 - xclock : 시계 프로그램
 - xcalc : 과학용 계산기 프로그램
 - xload : 시스템 load 를 측정
 - xedit : 간단한 텍스트 편집기
 - xcolors : 표현가능한 색을 보여줌
 - xbiff : 전자우편이 도착했음을 알림
 - xv : 여러가지 형식의 그림파일의 출력 및 조작
 - ghostview : postscript 파일을 보여줌
 - tgif : 그림 그리는 프로그램
- 실습
 1. X-client 를 띄어본다
 - xterm (return)
 - xterm window 의 size 와 position 을 지정하여 띄우려면 다음과 같이 한다
 - xterm -geometry 150x50+0+0 (return)
 - xterm 의 제목과 글자색과 배경색을 지정하고 스크롤라인수를 지정하면 다음과 같이 한다
 - xterm -fg white -bg black -sl 500 -title xterm1 (return)
 - 2. X 상에서 계산기 기능을 하는 xcalc 를 띄워본다
 - xcalc (return)
 - 위에서와 같이 크기및 위치를 지정하여 수행시켜본다
- 6.2 작업하기
- X-window 에서 작업하기
 - xdm (X display manager) 가 있을때는 login 만으로 시작
 - xdm 이 없을때는 xinit 한후 window manager 를 실행함
- 원격지 프로그램 실행하기
 - 자신의 컴퓨터의 xterm 상에서

- * 원격지 컴퓨터 (이름 remotecom)에서의 접속을 허용
- * →xhost remotecom 혹은 xhost + (모든 host에서의 접속을 허용)
- 원격지 컴퓨터로 접속
 - * →telnet remotecom
 - * 로그인 함
- X-client의 실행결과를 표시할 컴퓨터 지정
 - * setenv DISPLAY mycom:0
- X-client의 실행
 - * xcalc & 혹은 xclock 등
- 원격지 컴퓨터에 접속하지 않고 프로그램 실행하기
 - 접속하려는 원격 컴퓨터의 /etc/hosts.equiv 파일에 현재 사용하고 있는 컴퓨터의 이름이 있어야함
 - 혹은 원격지 자신의 계정의 HOME directory 밑에 .rhosts라는 파일에 현재 사용하고 있는 컴퓨터의 이름이 있어야함
 - 예) mycom라는 host를 사용중 remotecom이라는 host의 xcalc 프로그램을 수행하고 싶다
 - * remotecom 컴퓨터의 /etc/hosts.equiv에 mycom.hansung.ac.kr이 있어야 함
 - * 혹은 remotecom 컴퓨터의 자신의 계정 밑에 .rhosts 파일에 mycom.hansung.ac.kr이 있어야함
 - * 만약 특정 사용자만이 access하게 하려면 mycom.hansung.ac.kr 뒤에 특정 사용자 loginid를 삽입
 - 위와같은 환경이 갖춰어졌을 경우
 - * mycom에서 remotecom의 특정 X program을 수행하여 mycom 화면에 출력가능
 - * 또한 file copy를 수행할수 있음
 - * 예) rcp remotecom:~login.id/test .
 1. remotecom의 자신의 HOME directory 밑에 test라는 디렉토리를 mycom의 현재의 directory로 copy 함
 2. rcp -r remotecom:test . 해도 됨
 3. rcp -r remotecom:/tmp/test . 하면 remotecom의 /tmp/test 디렉토리가 copy됨
 - 예) rsh remotecom xcalc -display mycom:0 -geometry 150x150+0+0 &
 - * remotecom을 접속하여 xcalc를 실행하는데 그 결과 화면은 mycom에 나타나게 하는데
 - * 크기는 가로 50, 세로 50으로 위치는 왼쪽 위에 나타나게 하라는 명령
 - X-window 종료하기

- menu 상에서 종료
- 첫 윈도우 (xinit으로 생성된 window)에서 logout이나 exit
- 실습
 1. 원격지 컴퓨터를 접속하여 X-client를 수행시켜본다
 - hostname 명령으로 자신의 컴퓨터의 이름을 확인한다.
 - telnet perfect (return)
 - login
 - setenv DISPLAY 자신의 컴퓨터 이름:0 (return)
 - calctool (return)
 - calctool 죽이기
 - * 화면에 Quit를 누르거나
 - * calctool를 수행한 window를 click하고 ^C를 누름
 2. 원격지 컴퓨터를 접속하지 않고 rsh로 X-client를 수행시켜 본다
- 다중 작업 (background로 실행시키기)
 - 임의의 프로그램을 background로 실행시키기
 - * 명령어 뒤에 & 기호를 붙인다
 - * 예) xcalc &
 - background로 실행중인 프로그램 목록 보기
 - * jobs 명령어 사용
 - * 예) jobs (return)
 - 특정한 프로그램을 foreground로 복귀시키기
 - * fg %job_id
 - * 예) fg %1
 - foreground로 수행중인 프로그램을 background로 수행하기
 - * ^Z를 누름 → 해당 프로세스가 suspended 된다
 - * suspended 된 프로세스는 수행중이 아니다
 - * suspended 된 프로세스를 background로 수행하려면 bg라는 명령어을 써
 - * 예) bg (return)
 - background로 수행중인 프로세스 죽이기
 - * kill -9 %job_id
- 실습
 1. xcalc을 background로 실행시켜본다
 2. jobs로 해당 프로세스를 본다
 3. fg로 foreground로 실행시켜본다

4. ^Z 로 suspended 시킨다
5. bg 명령으로 수행되게 한다
6. 해당 프로세스를 죽이는 방법
 - kill 명령을 사용해서 해볼 것
 - fg 시키고 ^C 를 이용하여 해볼 것

- 프로그램 개발

- Unix에서 C 컴파일러의 사용 (GNU C 컴파일러 사용)
 - * 형식: gcc [options] cfiles
 - * 예: gcc -o code code.c → code.c 를 컴파일하여 실행파일 code 를 만들 (-o 없으면 실행 파일 a.out 생성)
 - * 아래의 내용을 hello.c 파일로 작성 (vi 이용)


```
#include <stdio.h>
main(){
    printf("hello world \n");
}
```
 - * 실행


```
$ gcc hello.c
$ a.out
hello world
$ gcc -o hello hello.c
$ hello
hello world
```

제 7 절 Shell

7.1 Shell basic

- 명령어 해석기 ← DOS 에 command.com 과 기능이 유사
- 프로그래밍 언어 ← DOS 에 batch file 과 기능이 유사
- shell 의 종류
 - Bourne shell 계열: Bourne shell, korn shell, Bash, zsh
 - C shell 계열 : c-shell, tcsh
 - * csh → 이름 csh
 - * tcsh : csh 의 확장 → 이름 tcsh
- 어떤 shell 을 사용해야 좋은가?
 - 명령어 해석기 → csh 이 적당

표 2.16: Bourne shell 계열		
Shell	이름	비고
Bourne shell	sh	초기의 shell
Korn shell	ksh	sh의 확장
Bash	bash	ksh의 확장
Zsh	zsh	bash의 확장

- shell script 작성 → Bourne shell 이 적당
- 사용하는 shell 변경하기
 - csh 사용중에 korn shell 사용을 원할 때
 - * ksh (return)
 - * 다사용후 ^D 혹은 exit 입력
 - * → 이전 shell 이 구동됨
- 기본 shell 변경하기
 - 기본 shell 은 /etc/passwd 에 로그인 및 홈디렉토리 정보와 함께 있음
 - 기본 shell 은 chsh 이라는 명령어로 바꿀 수 있음
 - chsh userid name-of-shell
 - ex.) chsh shjung /bin/ksh
- ### 7.2 Shell 이 제공하는 기능

 - shell 이 제공하는 대표적인 기능
 - 사용자가 원하는 환경을 만들
 - 명령어 alias 를 정의
 - 명령어 history 활용
 - 명령어 자동완성 기능
 - shell script 를 수행
 - 사용자가 원하는 환경을 만들
 - 사용자 로그인 시 환경을 자동설정하게 하는 파일들
 - * 특정 shell 을 사용하는 모든 사용자를 위한 스크립트 OS 나 버전별로 특정 디렉토리 밑에 있음
 - * 사용자별로 다른 환경을 설정하기 위한 스크립트 sh, ksh, bash 은 ~./profile
 - csh 은 ~./login 과 ~./cshrc
 - 대표적인 환경 변수들

- * TERM : 터미널 타입 지정
- * HOME : 사용자의 홈디렉토리
- * LOGNAME : 사용자의 로그인 이름
- * SHELL : 사용자의 기본쉘
- * PATH : 명령어를 찾기위해 살펴볼 디렉토리 목록
- 설정된 환경변수 보기
 - * 명령: echo \$환경변수이름
 - * 예) echo \$HOME
- 환경변수 설정 및 취소
 - * bash 에서


```
PS1='my prompt'; export PS1
```
 - * csh 에서


```
set name=value 나 setenv name value 형태로 지정
unset name 나 unsetenv name 형태로 취소
set prompt="[ `hostname -s`:$cwd:t \!]"
set history=150
```
- 로그인시 환경변수 설정
 - * 로그인시마다 특정한 환경변수를 설정하려면 로그인 스크립트에 등록시켜놓아야함
 - * 로그인 스크립트를 수정한후 새로 로그인하지 않고 적용 방법 →source .cshrc

- alias 정의, 보기 및 삭제
 - 정의 형식: alias aliasname command
 - 예)
 - * alias dir 'ls -aF'
 - * alias h 'history — more'
 - 보기: alias (enter)
 - 삭제: unalias aliasname
- 명령어 history 활용
 - 최근에 사용한 명령어는 저장되어있음
 - 버퍼에 저장될 명령어의 갯수 지정 (csh 에서) →set history=n (n개의 명령어 저장)
 - 다음 로그인에서 사용할 저장될 명령어 갯수 지정 (csh 에새) →set savehist=100 (100개의 명령어 저장)
 - 최근에 사용된 명령어 보기 →history (enter)
 - history 에 저장된 명령어 다시 사용하기

- * !! : 바로 전 명령어를 다시 사용
- * ^^ : 명령 입력을 바꾸어줌
 - datxq (return)
 - ^xq^e (return) →date 가 되어서 실행됨
- * !number : number 에 해당하는 명령을 실행
 - !21 temp*
- * !문자: 옛 명령중 문자로 시작하는 명령 재실행 (가장 최근것)
 - date (return)
 - !da (return) →date 재실행
- * !?문자?: 명령중 문자가 포함된 명령 재실행 (가장 최근것)
 - history (return)
 - !?sto? (return) →history 재 실행
- * \!* : 바로 전 사용한 명령의 모든 인수로 치환
- * \\$!: 바로 전 사용한 명령의 마지막 인수로 치환
- * 주의사항 : shell script 에 넣을때는 \ 를 넣어야하지만 shell 상에서는 락해야함

- Shell script 생성과 실행
 - 명령어들을 파일에 저장해놓고 (shell script 라 함) 실행
 - shell script 를 실행하려면 해당파일의 실행권한이 있어야함
 - 예)


```
#!/bin/csh
#lss: sort and list
ls -l | sort -n +4
```

 - * #!/bin/csh →해당파일을 수행시킬 shell 지정
 - * #lss: sort and list →해당 script 가 수행할 일 설명
 - * ls -l — sort -n +4 →수행 명령
 - C-shell script 에 대하여 자세히 알고싶은 학생은 여기를¹ 클릭

- 입출력 재지정 (redirection)
 - 특정 명령어의 입력 및 출력을 재지정
 - 예)
 - * date 명령을 실행하면 화면에 출력됨 이를 파일로 출력시 →date > date.txt
 - * more 명령을 실행할 입력을 재지정 →more < mylist.txt
 - * 파일에 추가 →time >> date.txt
 - * ps 수행후 결과를 sort 로 넘기고 결과를 저장 →ps — sort > pssort.txt
 - shell 에 대하여 자세히 알고싶은 학생은 여기를² 클릭

¹<http://itsys.hansung.ac.kr/lec/uninet/mylec/csh/>

²<http://itsys.hansung.ac.kr/lec/uninet/mylec/shell/>

7.3 시스템 자원 활용 명령어

- 디스크 사용량에 관한 정보

- ls -s : 각 파일의 size 를 보여줌
- du (disk usage)
 - * 형식 : du [-as] [name...]
 - * ex.) du -s name →총 계만 보여줌
 - * ex.) du -a name →모든 directory 와 파일의 크기를 보여줌
- df (disk free blocks)
 - * file system 별로 disk 사용량 및 사용할 수 있는 양을 보여줌
 - * 예) df 혹은 df -k (k bytes 단위로 표시)
- quota : disk 사용량 및 한계량 표시
 - * 형식 : quota -v

- which 명령어

- 실제로 사용되는 명령어의 경로 및 alias 된 경우 alias 를 보여줌
- 예) which ls, which date 등

- 파일의 종류 결정

```
[perfect:/usr2/97/102/hw]# file test.c stack.o
test.c: c program text
stack.o: sparc executable not stripped
[perfect:/usr2/97/102/hw]#
```

- 파일의 내용을 분석하여 종류를 결정해줌
- 형식: file filenames

- 연결관계

- 하나의 파일에 하나 이상의 이름이 있음
- 연결만들기
 - * ln file newname
 - * ln file... directory
 - * ex.) ln .cshrc .mycshrc

- symbolic 연결

- ln 의 제한점을 해결
- ln 의 제한점
 - * directory 들 사이에는 불가능
 - * 다른 파일 시스템 사이에는 불가능
- 전파일의 경로명을 copy
- ex.) ln -s /usr/bin ./bin

- 실습

1. ln_test 라는 directory 를 만든다
2. ln 으로 ttt 라는 파일을 ln_test 밑에 만든다
3. ttt 라는 파일을 지운다
4. ln_test 라는 directory 밑에 어떤 파일이 있는지 확인한다
5. /usr/games 를 자신의 home directory 밑에 symbolic link 한다
6. cd games 해본다 (pwd 로 현재의 directory 를 확인한다)
7. 자신의 home directory 로 간다
8. games 라는 symbolic link 를 지운다

- 파일 압축 및 복구

- 압축프로그램 : compress, gzip
- 압축복구 프로그램: uncompress, gunzip
- 예)
 - * compress file_name →file_name.Z 가 생성됨
 - * uncompress file_name.Z →file_name 이 생성됨
 - * gzip file_name →file_name.gz 가 생성됨
 - * gunzip file_name.gz →file_name 이 생성됨

- tar 명령어

- 여러 파일들을 하나의 파일로 결합시 사용
- 형식: tar [options] [directory]
- options
 - c : 새로운 archive 를 만들
 - t : archive 된 파일의 목차를 만들
 - v : archive 를 하거나 풀 때 목록을 출력
 - c : 새로운 archive 를 만들
 - x : archive 파일로부터 파일을 추출
 - 예)

- * tar cvf report.tar ./report →현재디렉토리 밑에 report 라는 디렉토리를 archive 하여 report.tar 라는 하나의 파일을 만듦
- * tar tvf report.tar →report.tar 의 archive 된 파일 목록을 보여줌
- * tar xvf report.tar →report.tar 의 archive 된 파일에서 파일을 추출하여 저장함

- 실습

1. tar_test 라는 디렉토리를 만든다.
2. /etc/hosts 파일을 tar_test 밑으로 copy 한다.
3. 자신의 홈디렉토리 밑에 있는 .cshrc 파일을 tar_test 밑에 cshrc 라는 이름으로 copy 한다.
4. tar cvf 를 이용하여 tar_test.tar 라는 archive 파일을 만든다.
5. tar_test 디렉토리 이름을 tar_test.org 로 이름을 변경한다.
6. tar_test.tar 파일을 compress 를 이용하여 압축한다.
7. tar_test.tar.Z 파일의 압축을 해제한다.
8. tar_test.tar 의 파일목록을 확인한다.
9. tar_test.tar 의 파일에서 파일을 추출한다.
10. tar_test.org 와 tar_test 라는 두개의 디렉토리 내용이 같은지 확인한다 (명령으로)

제 3 장

Internet

제 1 절 Internet basic

- Internet 이란?
 - 거대한 세계적인 정보자원의 그룹
 - 1970년대 Arpanet 로부터 유래
 - 네트워크를 서로 연결한 거대한 네트워크 (Inter Network에서 유래)
 - 이용 : 전자우편 (email), 다른 컴퓨터 접속, 대화, 정보이용 및 공유
- Internet 이 지원하는 네 가지 중요한 서비스
 - 전자우편 : Internet 상에 연결된 모든 컴퓨터의 모든 사용자에게 우편을 보내고 받음
 - telnet : Internet 상에 모든 컴퓨터를 이용할 수 있게 해줌 (단 account가 있어야 접속 가능)
 - FTP (file transfer protocol) : 컴퓨터 사이에 파일을 보내거나 받음
 - * download : 다른 컴퓨터에서 파일을 내 컴퓨터로 가져옮
 - * upload : 내 컴퓨터에서 다른 컴퓨터로 파일을 보냄
 - WWW (world wide web) : hyperlink로 만들어진 hypertext 기반 멀티미디어 정보서비스
- 그외에 자주 사용되는 서비스
 - rlogin (remote login) : 원격의 컴퓨터로 login하게 함
 - finger 서비스 : 인터넷상의 특정 사용자에 대한 정보를 얻을 때 사용
 - * finger userid@hostname
 - anonymous ftp : 공개된 자료를 접속하게 하는 ftp
 - archie : anonymous 상에 있는 자료를 찾을 때 사용
 - talk : internet 상의 다른 사용자와 대화할 때 사용
 - IRC (internet Relay Chat) : 여러 사람들이 주제별로 대화할 때 사용

- gopher : internet 상에 자원 공유를 쉽게 함
- wais (wide area information service) : internet 상의 여러 database를 찾아 정색 검색

1.1 Network

- Network 접속
 - Network
 - * 컴퓨터들을 연결하기 위한 통신망
 - * 자원공유
 - LAN (Local Area Network)
 - * 근거리 통신망
 - * 비교적 근거리에서 컴퓨터들의 통신을 지원하는 망
 - WAN (Wide Area Network)
 - * 광역 통신망
 - * 다수의 LAN을 연결한 통신망
- Client-server 환경
 - client → 자원을 사용하는 프로그램
 - server → 자원을 제공하는 프로그램
 - ex) print 기능 제공 → printer server
- 대규모 network 접속
 - gateway : network과 network을 연결
 - internet : 전 세계적으로 연결한 network
- 전화선을 사용한 접속
 - modem : modulation + demodulation의 약자
 - 컴퓨터 ↔(digital) modem (analog) ↔전화선 ↔(analog) modem (digital) ↔컴퓨터

1.2 Internet Access

- 접속 유형
 - 인터넷에 직접 연결된 컴퓨터 사용 (digital 전용선 사용)
 - 인터넷에 연결된 컴퓨터의 터미널 사용
 - 전화 접속
 - * modem을 통한 접속
 - * terminal emulator 필요

- * PPP (Point-to-Point Protocol) 나 SLIP (Serial Line Internet Protocol) 를
요
- * PPP나 SLIP 은 전화선을 통하여 전달되는 자료를 TCP/IP 에 연결시켜
줌
- TCP/IP 참조 모델

표 3.1: TCP/IP 참조 모델

계층	이름	설명
4	Application Layer	Telnet, ftp 등 서비스를 사용자에게 제공하는 계층
3	Transport Layer	TCP 와 UDP 로 이루어져 있음
2	Internet Layer	IP 와 ICMP 로 구성됨
1	Network Access Layer	X.25, Ethernet, SLIP/PPP 등처럼 접속카드와의 인터페이스를 위한

표 3.3: 국제 최상위 Domain 일부

domain	의미
ca	캐나다
de	독일
dk	덴마크
fr	프랑스
gr	그리스
jp	일본
nz	뉴질랜드
uk	영국
us	미국
kr	한국

● 주소 지정법

- 인터넷 주소 형식 : userid@domain.name
- ex.) shjung@ice.hansung.ac.kr
- email 보낼 때 : mail userid@domain.name
- ex.) mail shjung@ice.hansung.ac.kr
- sub domain
 - * kr : 한국에 있는
 - * ac : 교육 기관 중에
 - * hansung : 한성대학교 내에
 - * ice : ice 라는 컴퓨터
- domain 예

표 3.2: 미국과 캐나다에서 사용되는 최상위 Domain 예

domain	의미
com	상업 기관
edu	교육 기관
gov	정부
int	국제 기구
mil	군대
net	네트워크 기구
org	비 영리 기관

- IP address

- * 컴퓨터 주소 처리를 위한 숫자
- * ex.) ice.hansung.ac.kr → 128.134.168.131

- * DNS (domain name system) 에서 인터넷 주소를 IP 주소로 바꾸어줌
 - 특정 host 의 IP address 알기: nslookup 명령을 사용 DNS server 에 의
 - 한성대학교 DNS server 는 www.hansung.ac.kr 입
 - perfect.hansung.ac.kr 의 IP address 알려면 → nslookup perfect.hansung www.hansung.ac.kr (return)
 - mail 보낼 때 IP 주소 사용: mail 보낼 곳에 IP address 만 알고 있을 경우
mail shjung@ice.hansung.ac.kr 과
mail shjung@\[128.134.168.131\] 모두 동작

제 2 절 Internet Utilities

2.1 email

- 전자우편 방식
 - SMTP (simple Mail Transfer Protocol)
 - sendmail daemon
- 인터페이스
 - mail (Unix 표준)
 - elm (사용하기 쉬움)
 - netscape messenger
 - MS outlook
- 우편물 자동으로 모으기
 - 여러 컴퓨터에 계정이 있을 때 한 컴퓨터로 모든 mail 을 모으기
 - home directory 밑에 .forward 라는 파일에 모을 컴퓨터 이름을 적어놓음

- ex.) shjung@ice.hansung.ac.kr ←.forward 라는 파일의 내용
- text 와 2진 데이터
 - 2진 데이터를 보내기 위해 uudecode 와 uuencode 사용
 - 한글도 2진 데이터로 처리됨으로 위의 tool로 고쳐야함
 - 그동안 한글 code 를 7bit code 로 상호 바꾸는 필터를 이용
 - PC 의 netscape messenger 나 MS outlook 에서는 이 문제를 해결
- 보내는 방법
 - mail 상대방주소
 - ex.)
 - * mail shjung@ice.hansung.ac.kr (return)
 - * Subject: test →메일 제목 입력
 - * Cc : jun@ice.hansung.ac.kr →여러사람에게 보낼때
 - * 내용 입력
 - * ^D
 - 작성취소 :^C^C 입력
 - 빠르게 보내는 방법
 - * mail -s 제목 주소 (return)
 - * mail -s 제목 주소 < memo →memo 라는 파일의 내용이 보내짐
- 메세지를 읽었는지 알기
 - finger shjung@ice.hansung.ac.kr
- biff, xbiff
 - email 이 왔을때 이를 즉각 사용자에게 알림
- 메일 읽기
 - mail
- 실습
 1. netscape 상에서 email 을 처리해본다
 2. 옆 친구에게 email 을 서로 보내본다
 3. 받은 메일을 확인해 본다

2.2 telnet

- 개요
 - local computer ↔ remote computer 를 상호 연결시켜줌
- 시작하기
 - telnet 컴퓨터주소 혹은 IP address
 - ex.) telnet com2.hansung.ac.kr
 - ex.) telnet com2 (같은 domain 에서는 이렇게만 하여도 접속됨)
 - 접속후 login 과정을 거침
 - 다른 방법
 - * telnet (return)
 - * open 컴퓨터주소
 - * 사용가능한 명령을 보려면 ? 나 h 입력
- telnet 중 명령들
 - telnet 중 작업을 멈추고 다른 명령 수행 →^]
 - 잠시 shell 로 나올때 →z 혹은 ^Z
 - 다시 telnet 으로 복구 →fg
- 2.3 finger
 - finger란?
 - internet 사용자중 특정인에 대한 정보얻는 방법
 - ex.) finger shjung@ice.hansung.ac.kr
 - fingerd (finger daemon) →항상수행됨
 - .plan 파일과 .project 파일
 - home directory 밑에 .plan 과 .project 를 만들어 놓으면 다른 사람이 finger 했을때 이 내용이 보임
 - 호스트 사용자 보기
 - 특정 컴퓨터의 현재 사용자 보기
 - ex.) finger @ice.hansung.ac.kr 혹은 rusers 를 이용
 - users 는 현재 사용하는 컴퓨터의 사용자 보기
 - 실습
 1. 자신의 home directory 에 .plan 파일과 .project 파일을 vi 로 만든다.
 2. finger 로 자신의 정보를 본다.

2.4 Anonymous FTP

- anonymous FTP 란?
 - 인터넷상에 있는 정보를 교류하기 위한 방법
- FTP 란?
 - file transfer protocol 의 약자
 - 호스트와 호스트 사이에 파일 전달
 - remote host 로 파일을 보냄 →upload 라고 함
 - remote host 에서 파일을 가져옴 →download 라고 함
 - 일반 ftp 는 remote host 에 계정이 있어야함
 - anonymous FTP 는 계정이 없어도 가능
- anonymous FTP 접속
 - ftp anonymous.ftp.site (return)
 - login : anonymous
 - password : 자신의 email address
- FTP 사용법
 - ftp 상에서 ? 나 help 명령어 사용
 - 접속 컴퓨터의 directory 표시 →pwd
 - 접속 컴퓨터의 directory 이동 →cd dir_name
 - 접속 컴퓨터의 directory 생성 →mkdir
 - 접속 컴퓨터의 file 지우기 →delete file_name
 - 접속 컴퓨터의 파일 목록 보기 →dir
 - 사용 컴퓨터의 directory 변경 →lcd
 - 파일 가져오기 →get 이나 mget 이용
 - mget 은 여러파일을 가져올수 있음
 - mget 을 사용했을때 묻지 않게 하기 →prompt (return)
 - 파일 올리기 →put 이나 mput 을 이용
 - 파일 모드 선택
 - * ascii : 문서파일로 전송
 - * binary : 2진 파일로 전송
 - 접속 끝내기 →disconnect
 - 전송 byte 표시하기 →hash
 - ftp 끝내기 →quit
- 실습
 1. ftp.kaist.ac.kr 을 anonymous 로 login 한후 임의의 파일을 가져와본다.

2.5 Archie

- archie 란?
 - 인터넷상에 anonymous FTP 제공처와 자료목록을 찾아주는 tool
- archie 작업
 - 규칙적으로 anonymous FTP 에 접속 공개 directory 및 파일에 대한 정보를 가져와서 저장
 - internet archives database 에 저장
 - 시작 하기
 - * telnet archie.server
 - * login: archie
 - * password : 필요없음
 - xarchie : X-window 용 archie ←사용의 용이성
 - server 예) archie.sogang.ac.kr
- 실습
 1. xarchie를 이용하여 PC 에서 사용할수 있는 vi.exe 를 찾아서 자신의 PC에 download 받아 사용하시오.
- talk
 - talk 란?
 - 다른 컴퓨터의 사용자와 대화를 하게 함
 - 접속
 - talk userid@domain
 - ex.) talk jjk@com20
 - 접속 거부: mesg n (return)
 - 접속 허용: mesg y (return)
 - 실습
 1. 옆에 있는 친구와 talk 을 수행하여 메세지를 주고 받아 본다.
- WWW basic
 - WWW 란?
 - World Wide Web 의 약자로 인터넷의 방대한 자원에 대한 간단하며 일관성 있는 interface 제공

- 단순한 텍스트 형식의 정보뿐 아니라 그래픽 형식을 지원하며 오디오, 이미지, 동영상 등의 멀티미디어 서비스를 제공
- 기존의 ftp, gopher, news, mail 등도 서비스됨
- WWW 혹은 Web 혹은 W3 라고 불린다

- 용어들

- protocol: 다른 컴퓨터간의 정보를 주고받기 위하여 약속된 일련의 통신규약
 - * TCP/IP (Transmission Control Protocol/Internet Protocol) : 인터넷에서 사용되는 protocol
 - * 이외에 여러가지 protocol 이 있다
- hyperlink: 다른 자료와의 연결관계를 형성
- hypertext: hyperlink 개념을 사용하여 만들어진 text
- hypermedia: 단순한 문자뿐만 아니라 그림, 음성, 동화상 등의 다양한 형태의 자료를 포함한 hypertext
- browser: hypertext 를 읽을때 사용하는 프로그램
 - * ex.) netscape, explorer, Mosaic 등
- http (HyperText Transfer Protocol) : hypertext 를 전송하는 규약
- URL (Uniform Resource Locator): 인터넷상에 존재하는 web site 의 위치를 지정하는 방법
 - * 형식 : protocol://internet_address/파일경로
 - * 예)
 1. <http://ice.hansung.ac.kr/~shjung/lec.html>
 2. <http://www.kaist.ac.kr>
 3. <telnet://perfect.hansung.ac.kr> (외부 프로그램 필요)
 4. <ftp://shjung@ice.hansung.ac.kr/home/shjung/public.html/lec/uninternet/index.htm>
 5. <news://news.kaist.ac.kr>
 6. <gopher://gopher.dacom.co.kr>
 7. <file:///C1/Program%20Files/Netscape/Communicator/Program/nareadme.htm>
 8. <mailto:shjung@ice.hansung.ac.kr>
- Home page: 특정 site 의 첫번째 web page 를 말함
- HTML (HyperText Markup Language)
 - * hypertext로 기술되는 web 문서를 만들때 사용되는 언어
 - * 확장자는 html (PC 에서는 htm)
- CGI (Common Gateway Interface)
 - * web 서버와 처리프로그램 사이의 interface 규약
 - * web browser 에서 입력된 자료를 서버가 받아서 처리 프로그램에게 넘겨 줌
 - * 처리프로그램을 CGI 프로그램이라고 함

- * 처리프로그램의 결과를 서버가 받아서 다시 web browser 에게 보냄
- JavaScript
 - * Sun Microsystems 사에서 개발된 객체지향형 script 언어 (Object Oriented Script Language)
 - * script
 - shell script 처럼 script 란 명령문이 번역되어 실행되는 인터프리터 형식의 언어
 - compile 방식의 언어와 대비됨
 - * 자료가 서버에 전송되어 처리되는 CGI 프로그램과 달리 client 에서 바로 처리됨
- Java
 - * Sun Microsystems 사에서 개발된 객체지향형 언어
 - * 초기에는 다른 목적으로 만들어졌으나 후에 web에 적용하였음
 - * 플랫폼에 독립적인 byte code 형태로 compile 됨으로 어느 플랫폼에서도 실행됨
 - * 단 Java interpreter 가 있어야함
 - * web browser 에서 실행되는 Java Applet 과 독립적으로 실행되는 Java Application 이 있음

- browser 상에서 다른 프로그램 사용 가능

- telnet 의 사용 →telnet://perfect.hansung.ac.kr
- ftp 의 사용 →ftp://sim.kaist.ac.kr
 - * 특정 userid 로 ftp 할경우 →ftp://userid@domain_name
 - * 특정 userid 의 password 를 알경우 →ftp://userid:password@domain_name
- gopher 의 사용 →gopher://gopher.ieee.org
- news 의 사용 →news://news.kaist.ac.kr
- email 의 사용 →mailto:shjung@ice.hansung.ac.kr

제 4 장

WWW

제 1 절 Homepage 만들기

1.1 HTML 사용하기

1.1.1 Basic tag 사용법

- 문서 작성법

- 문서편집기 (DOS 에서 edit, Unix 에서 vi 등) 를 사용
- HotDog, Web Wizard 같은 HTML 전용 편집기를 사용
- 이미 작성되어 있는 문서를 소프트웨어를 이용 HTML 문서로 변환

- 태그(tag) 의 정의

 - 태그란?

 - * HTML 문서를 구성하기 위하여 사용되는 명령

 - 태그의 3가지 형식

- `<태그> text </태그>` → 예) `<center> text </center>`
- `<태그 속성=인자>` → 예) `` test
- `<태그>` → 예) `<hr>`

- 태그의 특징

 - 시작 태그 `<...>` 와 종료태그 `</...>`로 나타내어 다른 문자열과 구분
 - 속성은 태그의 환경을 설정해줌
 - 대소문자 구별이 없음
 - 여러 태그가 겹치지 않아야함

 - * `<center> ...</center>`
← 잘못된 예
 - * `<center> ...</center>`
← 잘된 예

- 공백은 무시된다

- HTML 문서의 전체구조

```
<html>
<head>
<title> .....</title>
</head>

<body>
.....
.....
</body>

</html>
```

 - `<html>` 태그: html 언어를 사용하여 작성됨을 알림

 - `<head>` 태그: 시작을 알리는 태그, 머리말을 표시

 - `<title>` 태그: web browser 의 제목표시바에 나는나는 제목

 - `<body>` 태그: web browser 화면에 나타나는 부분, `<head>` 와 `<title>` 태그 외에 모든 태그 사용 가능

- HTML 문서를 구성하는 태그

 - `<hn>` 태그

 - * 문서의 제목을 나타내는 태그

- * n 은 1에서 6까지 가능하며 1이 가장 큰 폰트로 나타난다
 - * 예) hn 태그 예제 (수행하기)¹
- ```
<HTML>
<HEAD> <TITLE> Hn 태그 예제 </TITLE>
</HEAD>

<BODY>
<H1> HTML 을 사용 안 H1 태그 폰트 입니다. </H1>
<H2> HTML 을 사용 안 H2 태그 폰트 입니다. </H2>
<H3> HTML 을 사용 안 H3 태그 폰트 입니다. </H3>
<H4> HTML 을 사용 안 H4 태그 폰트 입니다. </H4>
<H5> HTML 을 사용 안 H5 태그 폰트 입니다. </H5>
<H6> HTML 을 사용 안 H6 태그 폰트 입니다. </H6>
```
- `</BODY>`
- `</HTML>`
- `<p>` 태그
- \* 줄바꿈시 사용
- \* `</p>` 를 사용하지 않아도됨
- \* 여러번 사용해도 한번의 효과
- \* 예) p 태그 예제 (수행하기)<sup>2</sup>
- <sup>1</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=hn.htm')
- <sup>2</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=p.htm')
- 71

```
<HTML>
<HEAD>
 <TITLE> P 태그 예제 </TITLE>
</HEAD>
<BODY>
 <H1> 안녕하세요 </H1>
 HTML을 사용한 P 태그 텍스트입니다. <P><P><P>
 Happy New Week!
</BODY>
</HTML>
```

#### - 주석 (comment) 태그

- \* HTML의 이해를 돋기위해 사용
- \* 화면에 출력되지 않음
- \* 사용법: <!-- text --!>
- \* 예) comment 태그 예제 (수행하기)<sup>3</sup>

```
<HTML>
<HEAD>
 <TITLE> 주석 태그 예제 </TITLE>
</HEAD>
<BODY>
 <H1> 주석 텍스트입니다. </H1>
 <!-- 주석문입니다. --!>
</BODY>
</HTML>
```

#### - <br> 태그

- \* 빈줄을 만들며 </br> 이 없다
- \* 사용한만큼 빈줄이 생성된다
- \* 예) <br> 태그 예제 (수행하기)<sup>4</sup>

```
<HTML>
<HEAD>
 <TITLE> BR 태그 예제 </TITLE>
</HEAD>
<BODY>
 <H1> 안녕하세요 </H1>
 안녕하세요.
 세 개 복 많이 받으세요.

 Happy New Week!
</BODY>
</HTML>
```

#### - <center> 태그

- \* <center> 와 </center> 사이의 문장을 가운데로 정렬

<sup>3</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=center.htm')

<sup>4</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=br.htm')

\* 예) <center> 태그 예제 (수행하기)<sup>5</sup>

```
<HTML>
<HEAD>
 <TITLE> CENTER 텍스트 예제 </TITLE>
</HEAD>
<BODY>
```

```
<H1> 가운데 정렬이 아닙니다. </H1>
```

```
<CENTER>
 <H1> 가운데 정렬입니다. </H1>
</CENTER>
```

```
</BODY>
```

```
</HTML>
```

#### - <div> 태그

- \* <div> 와 </div> 사이의 문장의 정렬 방식을 지정
- \* 정렬은 left, right, center 가 있다
- \* 예) <div> 태그 예 (수행하기)<sup>6</sup>

```
<HTML>
<HEAD>
 <TITLE> DIV 텍스트 예제 </TITLE>
</HEAD>
<BODY>
 <DIV ALIGN=LEFT>
 <H1> 왼쪽 정렬입니다. </H1>
 <DIV ALIGN=RIGHT>
 <H1> 오른쪽 정렬입니다. </H1>
 <DIV ALIGN=CENTER>
 <H1> 가운데 정렬입니다. </H1>
</BODY>
</HTML>
```

#### - <hr> 태그

- \* 수평직선을 만듬
- \* </hr> 이 없음
- \* 네 가지 속성: size, width, align, noshade
- \* <hr> 태그 예 (수행하기)<sup>7</sup>

```
<HTML>
<HEAD>
 <TITLE> HR 텍스트 예제 </TITLE>
</HEAD>
<BODY>
```

1. SIZE 속성 이용 <HR>

<sup>5</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=center.htm')

<sup>6</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=div.htm')

<sup>7</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=hr.htm')

```

SIZE=3 입니다. <HR SIZE=3>
SIZE=10 입니다. <HR SIZE=10>

2. WIDTH 속성 이용

WIDTH=500 입니다. <HR WIDTH=500>
WIDTH=300 입니다. <HR WIDTH=300>

3. ALIGN 속성 이용

ALIGN=LEFT 입니다. <HR WIDTH=300 ALIGN=LEFT>
ALIGN=RIGHT 입니다. <HR WIDTH=300 ALIGN=RIGHT>
ALIGN=CENTER 입니다. <HR WIDTH=300 ALIGN=CENTER>

4. NOSHADe 속성 이용

NOSHADe 입니다. <HR NOSHADe>

</BODY>
</HTML>

```

#### - <font> 태그

\* 폰트의 크기 색상을 변화시킴

\* <font> 태그 예 (폰트 크기 조정) (수행하기)<sup>8</sup>

```

<HTML>
<HEAD>
<TITLE> FONT SIZE=n 태그 예제 </TITLE>
</HEAD>
<BODY>

<CENTER> 폰트의 절대크기 </CENTER>

 HTML을 이용한 폰트 크기는 1입니다.

 HTML을 이용한 폰트 크기는 2입니다.

 HTML을 이용한 폰트 크기는 3입니다.

 HTML을 이용한 폰트 크기는 4입니다.

 HTML을 이용한 폰트 크기는 5입니다.

 HTML을 이용한 폰트 크기는 6입니다.

 HTML을 이용한 폰트 크기는 7입니다.

<CENTER> 폰트의 상대크기 </CENTER>

2> HTML을 이용한 폰트 크기는 +2입니다.

<FONT SIZE=<-2> HTML을 이용한 폰트 크기는 -2입니다.

</BODY>
</HTML>

```

\* <font> 태그 예 (color 조정) (수행하기)<sup>9</sup>

```

<HTML>
<HEAD>
<TITLE> FONT COLOR=n 태그 예제 </TITLE>
</HEAD>
<BODY>

<CENTER>
폰트의 RGB 지정값에 의한 색상

 HTML을 이용한 RGB 지정값의 색상은 Red입니다.

 HTML을 이용한 RGB 지정값의 색상은 Green입니다.

 HTML을 이용한 RGB 지정값의 색상은 Blue입니다.

</CENTER>

</BODY>
</HTML>

```

<sup>8</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=font.htm')

<sup>9</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=foncolor.htm')

```
</HTML>
```

#### ● 실습

1. 위의 hn, p, comment, br, center, div, hr, font 태그의 예를 수행시켜본다.
2. vi 를 이용하여 자신의 homepage 에 위의 예를 변형시켜 테스트 해본다.
3. homepage directory 만들기

- 자신의 home directory 밑에 homepage 라는 directory 를 만든다.
- chmod 755 homepage 를 실행한다
- homepage 로 이동하여 index.html 파일을 만든다
- index.html 에 위의 태그 예를 수행하는 html 문서를 만든다
- chmod 755 index.html 파일 모드를 변경한다.

4. 자신의 homepage access 하기

- web browser 상에서 http://perfect.hansung.ac.kr/ 자신의 login\_name 으로 access

#### ● 특수문자를 HTML 문서에 입력하는 방법

- HTML 예약어를 표시하는 방법
- ISO LATIN-1 의 특수문자를 입력하는 방법
- web browser 에 따라 나타나는 형태가 다를 수 있음
- 번호 표시법
  - \* ISO LATIN-1 문자 집합을 침조하여 번호로 표시
  - \* 사용법: &#번호;
  - \* 예) 특수문자를 번호로 표시하는 예 (수행하기)<sup>10</sup>

```

<HTML>
<HEAD>
<TITLE> 특수문자 태그의 번호로 표시하는 예제 </TITLE>
</HEAD>

```

```

<BODY>
<CENTER>
38; 은 ISO LATIN-1 문자 집합에 의해 번호로 표시하는 방법
60; 은 ISO LATIN-1 문자 집합에 의해 번호로 표시하는 방법
62; 은 ISO LATIN-1 문자 집합에 의해 번호로 표시하는 방법
634; 은 ISO LATIN-1 문자 집합에 의해 번호로 표시하는 방법
6198; 은 ISO LATIN-1 문자 집합에 의해 번호로 표시하는 방법
6208; 은 ISO LATIN-1 문자 집합에 의해 번호로 표시하는 방법
6216; 은 ISO LATIN-1 문자 집합에 의해 번호로 표시하는 방법
6222; 은 ISO LATIN-1 문자 집합에 의해 번호로 표시하는 방법
</CENTER>

```

```
</BODY>
```

- \* 예) 특수문자를 문자로 표시하는 예) (수행하기)<sup>11</sup>

<sup>10</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=specnum.htm')

<sup>11</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=specchar.htm')

```
<HTML>
<HEAD>
<TITLE> 특수 문자 테그의 문자로 표시하는 예제 </TITLE>
</HEAD>
<BODY>
<CENTER>
< 은 ISO LATIN-1 문자 집합에 의해 문자로 표시하는 방법의 예제이다.

< 은 ISO LATIN-1 문자 집합에 의해 문자로 표시하는 방법의 예제이다.

> 은 ISO LATIN-1 문자 집합에 의해 문자로 표시하는 방법의 예제이다.

" 은 ISO LATIN-1 문자 집합에 의해 문자로 표시하는 방법의 예제이다.

Æ 은 ISO LATIN-1 문자 집합에 의해 문자로 표시하는 방법의 예제이다.
Ð 은 ISO LATIN-1 문자 집합에 의해 문자로 표시하는 방법의 예제이다.

Ø 은 ISO LATIN-1 문자 집합에 의해 문자로 표시하는 방법의 예제이다.
Þ 은 ISO LATIN-1 문자 집합에 의해 문자로 표시하는 방법의 예제이다.
</CENTER>
</BODY>
</HTML>
```

#### - <address> 태그

- \* 전자우편 주소나 서명, 저작권 등을 나타내기 위해 사용
- \* 이 탤릭체로 나타남
- \* 예) (수행하기)<sup>12</sup>

```
<HTML>
<HEAD>
<TITLE> ADDRESS 테그 예제 </TITLE>
</HEAD>
<BODY>
<H2> 다음의 전자우편 주소로 연락 부탁합니다. </H2>
<HR>
<ADDRESS> shjung@ice.hansung.ac.kr </ADDRESS>
</BODY>
</HTML>
```

#### - <blockquote> 태그

- \* 인용문단을 표시하는데 사용
- \* 예) (수행하기)<sup>13</sup>

```
<HTML>
<HEAD>
<TITLE> BLOCKQUOTE 테그 예제 </TITLE>
</HEAD>
<BODY>
<H2> 다음 가사는 애국가 1절을 인용한 것이다. </H2>
<HR>
<BLOCKQUOTE>
동 애를 과 백두산이 마르고 담도록

아느님의 보우 악사 우리나락 만세

무궁와 삼천리 와여강산 대안 사람 대안으로 길이 보존아세

대안 사람 대안으로 길이 보존아세
</BLOCKQUOTE>
```

<sup>12</sup>javascript:open\_win('testhtml/onlineTest.cgi?file=address.htm')

<sup>13</sup>javascript:open\_win('testhtml/onlineTest.cgi?file=blockque.htm')

```
</BLOCKQUOTE>
</BODY>
</HTML>
```

#### - <xmp> 태그

- \* 예문을 표시할 때 사용 enter, tab, space 등을 인식
- \* 주의사항: 이 태그 안에는 다른 어떤 태그도 사용할 수 없다
- \* 예) (수행하기)<sup>14</sup>

```
<HTML>
<HEAD>
<TITLE> XMP 테그 예제 </TITLE>
</HEAD>
<BODY>
```

```
<H2> 다음 가사는 애국가 1절이다. </H2>
<HR>
<XMP> 동 애 물과 백두산이 마르고 담도록

아느님의 보우 악사 우리나락 만세 <P>

무궁와 삼천리 와여강산 대안 사람 대안으로 길이 보존아세
</XMP>
<HR>
```

```
</BODY>
</HTML>
```

#### - <pre> 태그

- \* enter, tab, space 등을 인식하나 내부에 다른 태그가 삽입될 수 있다
- \* 예) (수행하기)<sup>15</sup>

```
<HTML>
<HEAD>
<TITLE> PRE 테그 예제 </TITLE>
</HEAD>
<BODY>
```

```
<CENTER>
<PRE><H2> 웹 검색기 예시
<PRE> 테그 예제 입니다.
</H2></PRE>
</CENTER>
```

```
</BODY>
</HTML>
```

#### ● HTML 문서의 문자형태를 변경하는 태그

##### - 논리적 스타일 방법: 형태를 바꾸는 것이 아니라 강조하는 방법

- \* <strong> 태그: 문장을 강조하기 위해 사용, 볼드체로 나타남

<sup>14</sup>javascript:open\_win('testhtml/onlineTest.cgi?file=xmp.htm')

<sup>15</sup>javascript:open\_win('testhtml/onlineTest.cgi?file=pre.htm')

- \* <em> 태그: 문장을 강조하기 위해 사용, 이탤릭체로 나타남
- \* <dfn> 태그: 단어나 문장을 정의할 때 사용, 이탤릭체로 나타남
- \* <var> 태그: 변수 이름을 나타낼 때 사용, 이탤릭체로 나타남
- \* <cite> 태그: 참고문헌을 인용할 때 사용, 보통 이탤릭체로 나타남
- \* <samp> 태그: 컴퓨터 메세지를 나타낼 때 사용, 고정폭 글자체로 나타남
- \* <code> 태그: 프로그램 코드를 나타낼 때 사용, 보통 고정폭 글자체로 나타남
- \* <strike> 태그: 문자 중간에 줄긋기를 원할 때 사용
- \* <kbd> 태그: 키보드 입력을 나타내기 위해 사용, 보통 고정폭 글자체로 나타남
- \* 예) (수행하기)<sup>16</sup>

&lt;HTML&gt;

```
<HEAD>
 <TITLE> 논리적 스타일 방법에서 사용하는 태그 예제 </TITLE>
</HEAD>
<BODY>

<CENTER>
 지금 STRONG 를 사용하고 있습니다.

 지금 EM 를 사용하고 있습니다.

 지금 <DFN>DFN </DFN>를 사용하고 있습니다.

 지금 <VAR>VAR </VAR>를 사용하고 있습니다.

 지금 <CITE>CITE </CITE>를 사용하고 있습니다.

 지금 <SAMP>SAMP </SAMP>를 사용하고 있습니다.

 지금 <CODE>CODE </CODE>를 사용하고 있습니다.

 지금 <STRIKE>STRIKE </STRIKE>를 사용하고 있습니다.

 지금 <KBD>KBD </KBD>를 사용하고 있습니다.

</CENTER>

</BODY>
</HTML>
```

#### - 물리적 스타일 방법: 문자의 형태를 바꾸는 것

- \* <u> 태그: 문장에 밑줄을 그을 때 사용
- \* <blink> 태그: 문장을 깜박거리게 할 때 사용
- \* <sup> 태그: 윗첨자로 문장을 나타낼 때 사용
- \* <sub> 태그: 밑첨자로 문장을 나타낼 때 사용
- \* <tt> 태그: 문장을 타자기로 입력한 모양으로 바꿀 때 사용
- \* <i> 태그: 문장을 이탤릭체로 나타낼 때 사용
- \* <b> 태그: 문장을 볼드체로 나타낼 때 사용
- \* <big> 태그: 문장의 크기를 한 단계 크게 만들 때 사용
- \* <small> 태그: 문장의 크기를 한 단계 작게 만들 때 사용
- \* 예) (수행하기)<sup>17</sup>

&lt;HTML&gt;

```
<HEAD>
 <TITLE> 물리적 스타일 방법에서 사용되는 태그 예제 </TITLE>
```

<sup>16</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=logical.htm')

<sup>17</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=physical.htm')

```
</HEAD>
<BODY>

<CENTER>
 지금 <U> 예그 </U>를 사용하고 있습니다.

 지금 <BLINK>BLINK </BLINK>를 사용하고 있습니다.

 지금 ^{SUP}를 사용하고 있습니다. SUP 텍스트를 사용하고 있습니다.

 지금 <TT>TT </TT>를 사용하고 있습니다.

 지금 <I> 예그 </I>를 사용하고 있습니다.

 지금 _{SUB}를 사용하고 있습니다.

 지금 예그 를 사용하고 있습니다.

 지금 <BIG>BIG </BIG>를 사용하고 있습니다.

 지금 <SMALL>SMALL </SMALL>를 사용하고 있습니다.

</CENTER>

</BODY>
</HTML>
```

#### ● 실습

1. 위의 address, blockquote, xmp, pre, 문서 형태를 변경하는 태그의 예를 수행해본다.

2. vi 를 이용하여 자신의 homepage 에 위의 예를 변형시켜 테스트 해본다.

#### 1.1.2 Advanced tag 사용법

##### ● HTML 문서에 목록을 삽입하는 태그

- <ol> 태그: 목록을 나타내며 목록 앞에 여백이 들어감
- 예) (수행하기)<sup>18</sup>

```
<HEAD>
 <TITLE> OL 예제 </TITLE>
</HEAD>
<BODY>

<H2> 이 용 </H2>
<HR>

 Hello!

 Guten Tag!

 안녕!


```

&lt;/BODY&gt;

&lt;/HTML&gt;

##### - <li> 태그

- \* 목록을 나타내며 목록 앞에 숫자가 들어감
- \* 보통 <ol> 이나 <ul> 안에 사용됨
- \* type 속성

<sup>18</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=ol.htm')

표 4.1: li tag의 type 속성

사용법	설명
<li type=A>	알파벳 대문자
<li type=a>	알파벳 소문자
<li type=I>	로마 대문자
<li type=i>	로마 소문자
<li type=1>	아라비아 숫자

\* 예) (수행하기)<sup>19</sup>

```
<HTML>
<HEAD>
 <TITLE> LI 테그 예제 </TITLE>
</HEAD>
<BODY>
 <OL START=4>
 <H2> 테그 예제 </H2>
 <HR>
 안녕!
 <LI TYPE=A> Hello!
 Guten Tag!
 <LI TYPE=a> 안녕!
 Hello!
 Guten Tag!
 <LI TYPE=I> 안녕!
 <LI TYPE=i> Hello!
 <LI TYPE=1> Guten Tag!
 안녕!

 </BODY>
</HTML>
```

#### - <ul> 태그

- \* 순서가 없는 목록을 나타냄
- \* type 속성

표 4.2: li tag의 type 속성

사용법	설명
<li type=disc>	채워진 원
<li type=circle>	빈 원
<li type=square>	네모 표시

\* 예) (수행하기)<sup>20</sup>

```
<HTML>
<HEAD>
 <TITLE> UL 테그 예제 </TITLE>
</HEAD>
```

<sup>19</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=li.htm')

<sup>20</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=ul.htm')

#### <BODY>

```

 <H2> 테그 이용 </H2>
 <HR>
 <H2> Hello! </H2>
 <H2> Guten Tag! </H2>
 <H2><LI TYPE=disc> Hello! </H2>
 <H2> Guten Tag! </H2>
 <H2><LI TYPE=circle> Hello! </H2>
 <H2><LI TYPE=1> Guten Tag! </H2>
 <H2> Hello! </H2>
 <H2><LI TYPE=square> Guten Tag! </H2>

```

#### </BODY>

#### </HTML>

#### - <dd> 태그

\* 용어를 정의할 때 사용, <dd> 를 사용하지 않아도 됨

\* 예) (수행하기)<sup>21</sup>

```
<HTML>
<HEAD>
 <TITLE> 용어를 정의하는 DD 테그 예제 </TITLE>
</HEAD>
<BODY>
 <H2> 안성 대학교 </H2>
 <DD> 서울 시 성북구 삼선동 2가에 위치한
 발전하는 대학이다.
</BODY>
</HTML>
```

#### ● 실습

1. 위의 ol, li, ul, dd 태그의 예를 수행시켜본다.
2. vi 를 이용하여 자신의 homepage 에 위의 예를 변형시켜 테스트 해본다.

#### ● HTML 문서에 표를 삽입하는 태그

#### - <table> 태그

- \* table 을 만드는 기본 태그
- \* 속성으로 width, height, align, border, cellspacing 등을 갖음
- \* 사용법: <table 속성=값 ...> 내용 </table>
- \* width 속성: 표의 가로길이 지정, 미설정 시 내용에 따라 자동 결정
- \* height 속성: 표의 세로길이 지정, 미설정 시 내용에 따라 자동 결정
- \* align 속성: 정렬 방식 지정 (left, right, center 중 하나 기본값 left)
- \* border 속성: 표의 테두리 두께 지정 (숫자로 지정 기본값은 1)
- \* cellspacing 속성: 표의 내부 테두리 두께 지정

<sup>21</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=dd.htm')

## - &lt;th&gt; 태그

- \* 표의 헤더셀에 볼드체의 제목을 입력시 사용
- \* 속성으로 width, height, align, valign, rowspan, colspan 등을 갖음
- \* 사용법: <th 속 성=값 ...> 내용
- \* width 속성: 헤더셀의 수평길이 지정
- \* height 속성: 헤더셀의 수직길이 지정
- \* align 속성: 제목의 수평 정렬방식 지정 (left, right, center 중 하나 기본값 center)
- \* valign 속성: 제목의 수직 정렬방식 지정 (top, middle, baseline, bottom 중 하나 기본값은 middle)
- \* rowspan 속성: 하나의 헤더셀이 몇개의 행을 포함하는지 지정
- \* colspan 속성: 하나의 헤더셀이 몇개의 열을 포함하는지 지정

## - &lt;td&gt; 태그

- \* 헤더셀 밑에 들어가 각 셀의 자료를 입력시 사용
- \* 속성으로 width, height, align, valign, rowspan, colspan, caption 등을 갖음
- \* 사용법: <td 속 성=값 ...> 내용
- \* 속성의 사용법은 위와 유사 단 셀에 적용됨
- \* caption 속성: 표의 제목을 입력시 사용
  - 사용법: <caption 속 성=값> 제목 </caption>
  - align 속성: top, bottom 중 하나 기본값은 top

## - &lt;tr&gt; 태그

- \* 한행의 시작을 알림
- \* 사용법: <tr> 한 행의 내용</tr>

## - 예제

1. (수행하기)<sup>22</sup>

```
<HTML>
<HEAD>
<TITLE> TABLE 테그 예제 1 </TITLE>
</HEAD>
<BODY>

<TABLE BORDER=1 CELSPACING=3 HEIGHT=200 WIDTH=400>
<CAPTION ALIGN=BOTTOM> TABLE 테그 예제 1 </CAPTION>
<TH ALIGN=RIGHT VALIGN=TOP>NAME</TH> <TH ALIGN=MIDDLE>AGE</TH>
<TH ALIGN=BASELINE>그 양</TH> <TH VALIGN=BOTTOM>직업</TH>
<TR> <!-- 다음 행 --!>
<TD>HONG GIL DONG </TD> <TD>30</TD> <TD> <TD>의적</TD>
<TR> <!-- 다음 행 --!>
<TD>성춘 양</TD> <TD VALIGN=TOP>16</TD> <TD ALIGN=CENTER>남원</TD>
<TD><PRE></PRE></TD>
</TABLE>

</BODY>
</HTML>
```

<sup>22</sup>javascript:open\_win('testhtml/onlineTest.cgi?file=table1.htm')

2. (수행하기)<sup>23</sup>

```
<HTML>
<HEAD>
<TITLE> TABLE 테그 예제 2 </TITLE>
</HEAD>
<BODY>

<TABLE BORDER=7 CELLSPACING=10 HEIGHT=300 WIDTH=500>
<CAPTION ALIGN=TOP> TABLE 테그 예제 2 </CAPTION>
<TH ROWSPAN=3 WIDTH=40> 전자 공학과 기록부
<TH ALIGN=LEFT VALIGN= CENTER COLSPAN=2> 이름
<TH WIDTH=100 HEIGHT=20> 전화 번호 <TH> 직위 <TH> 성별
<TR> <!-- 다음 행 --!>
<TD ALIGN= CENTER COLSPAN=2> 안득기 <TD WIDTH=300 HEIGHT=25> 824-9959
<TD VALIGN= BOTTOM ROWSPAN=2> 서울 <TD> 애
<TR> <!-- 다음 행 --!>
<TD WIDTH=30> 김돌석 <TD> 김순돌 <TD> 284-0001 <TD> 남
</TABLE>

</BODY>
</HTML>
```

## ● 실습

1. 위의 태그들을 사용하여 다음과 같은 table 을 만들어본다.

table 예<sup>24</sup>

## ● HTML 문서를 link 하는 태그

- <a> 태그를 사용
  - name이나 href 속성 중 하나와 반드시 같이 사용
  - URL을 사용할경우 href 속성
    - \* 프로토콜(http, telnet, ftp, mailto, gopher, news, file)://인터넷주소 혹은 파일path
    - \* http 사용시: <a href="http://웹사이트 주소"> text </a>
    - \* 예) (수행하기)<sup>25</sup>
- ```
<HTML>
<HEAD>
<TITLE> Http 를 사용한 HREF 속성 예제 </TITLE>
</HEAD>
<BODY>
<A HREF="http://multilab.hansung.ac.kr"> 정보 전산 학부 웹페이지 </A>
</BODY>
</HTML>
```
- * telnet 사용시: text
 - * 예) (수행하기)²⁶

²³javascript:open_win('testhtml/onlineTest.cgi?file=table2.htm')

²⁴http://itsys.hansung.ac.kr/lec/uninet/images/table_ex.gif

²⁵javascript:open_win('testhtml/onlineTest.cgi?file=a_href.htm')

²⁶javascript:open_win('testhtml/onlineTest.cgi?file=a_telnet.htm')

```

<HTML>
<HEAD>
<TITLE> Telnet을 사용한 HREF 속성 예제 </TITLE>
</HEAD>
<BODY>
<A href="telnet://perfect.hansung.ac.kr"> Telnet 서비스를 이용한 perfect 접속 </A>
</BODY>
</HTML>
* ftp 사용시: <a href="ftp://ftp 인터넷주소"> text </a>
* 예) (수행하기)27
<HTML>
<HEAD>
<TITLE> FTP를 사용한 HREF 속성 예제 </TITLE>
</HEAD>
<BODY>
<A href="ftp://ftp.kaist.ac.kr"> KAIST anonymous FTP 접속 </A>
</BODY>
</HTML>
* mailto 사용시: <a href="mailto:전자우편주소"> text </a>
* 예) (수행하기)28
<HTML>
<HEAD>
<TITLE> Mailto를 사용한 HREF 속성 예제 </TITLE>
</HEAD>
<BODY>
<H2><B> 안녕하세요. 과소비를 줄입니다. </B></H2>
<BLOCKQUOTE> 경제를 살립시다. </BLOCKQUOTE>
<HR>
<A href="mailto:shjung@ice.hansung.ac.kr">
<ADDRESS> shjung@ice.hansung.ac.kr </ADDRESS></A>
</BODY>
</HTML>
* gopher 사용시: <a href="gopher:고퍼사이트인터넷주소"> text </a>
* 예) (수행하기)29
<HTML>
<HEAD>
<TITLE> 고퍼를 사용한 HREF 속성 예제 </TITLE>
</HEAD>
<BODY>
<A href="gopher://gopher.dacom.co.kr"> 데이터베이스의 고퍼 사이트 </A>
</BODY>

```

²⁷javascript:open_win('/testhtml/onlineTest.cgi?file=a_ftp.htm')

²⁸javascript:open_win('/testhtml/onlineTest.cgi?file=a_mailto.htm')

²⁹javascript:open_win('/testhtml/onlineTest.cgi?file=a_gopher.htm')

```

</HTML>
* 유즈넷 사용시: <a href="news:유즈넷뉴스그룹"> text </a>
* 예) (수행하기)30
<HTML>
<HEAD>
<TITLE> 유즈넷 뉴스를 사용한 HREF 속성 예제 </TITLE>
</HEAD>
<BODY>
<A href="news:han.test"> 유즈넷 뉴스 앤글 페스트 그룹 </A>
</BODY>
</HTML>
* file 사용시
  · <a href="///절대경로"> text </a>
  · <a href="상대경로"> text </a>
* 예) (수행하기)31
<HTML>
<HEAD>
<TITLE> 파일 경로 및 파일명을 사용하는 경우의 HREF 속성 예제 </TITLE>
</HEAD>
<BODY>
<CENTER>
<A href="a_path1.htm"> 인기 있는 사이트(상대경로) </A><BR><BR>
<A href="a_path2.htm"> 자동차 공부방(상대경로) </A><BR><BR>
<A href="//c:/usr/shlee/html_source/a_path2.htm"> 자동차 공부방(절대경로) </A>
</CENTER>
</BODY>
</HTML>
- name 을 사용할 경우 href 속성
  * link 되는 html 문서가 같은 html 문서에 있는 경우
  * 사용법
    · <a href="#1"> 서른 </a> 으로 접근
    · <a name="1"> 서른 </a> 으로 위치 표시
* 예) (수행하기)32
<HTML>
<HEAD>
<TITLE> 링크되는 HTML 문서가 같은 HTML 문서 내에 있는 경우의 예제 </TITLE>
</HEAD>
<BODY>
<H3> 엔진(Engine) </H3>
<UL>
<A href="#1"> 엔진의 개요 </A><BR><BR>

```

³⁰javascript:open_win('/testhtml/onlineTest.cgi?file=a_usenet.htm')

³¹javascript:open_win('/testhtml/onlineTest.cgi?file=a_path.htm')

³²javascript:open_win('/testhtml/onlineTest.cgi?file=a_name1.htm')

```

<A HREF="#"> 가솔린 엔진의 작동 원리 </A><BR><BR>
<A HREF="#"> 엔진의 성능 선도 </A><BR><BR><BR><BR>
</UL>

<H3><A NAME="1"> 엔진의 개요 </H3><BR>
<UL>
    엔진을 열원으로 하여 그 연소에 의한 열에너지를 작동 요체에 공급하여 열팽창 시켜서 작동
    유체가 팽창하는 일을 기계적 에너지인 측의 외전력이나 유체의 분출력을으로 바꾸어서 동력을
    얻는 장치를 열기관이라고 한다. 즉 열기관이란 열에너지를 기계적 에너지로 바꾸는 장치이며,
    내연기관과 외연기관으로 분류된다.
<LI> 내연기관 <BR>
    연료를 실린더 내부에서 연소, 폭발시켜서 동력을 얻는 기관이다.
<LI> 외연기관 <BR>
    엔진의 외부에 장착된 연소장치에 의하여 연소시켜 열에너지를 실린더 내부로 옮김으로 동력을
    얻는 엔진이며, 증기기관과 증기터빈이 여기에 속한다.
</UL>

<HR>
<H3><A NAME="2"> 가솔린 엔진의 작동 원리 </H3>
<UL>
    실린더 내에 공기와 가솔린의 혼합기를 옮김으로,
    미스톤으로 압축한 다음 정화하고 정화하여 연소시키면 열에너기가 발생되어
    연소 가스는 고온, 고압이 된다.
    이 고온 고압의 가스가 실린더 내를 상아트 십동하는 미스톤 앤드에
    작동 앤미스톤이 움직이게 되어, 이에 따라 열에너기가 기계적 에너지로 바뀌게 된다. <P>
    이때 미스톤은 직선 왕복운동하는 것이므로 이것을 회전 운동으로 바꾸기 위하여 보통 커넥팅
    로드와 크랭크 축을 사용한다.
    미스톤의 운동이 미스톤만으로 미스톤과 연결되어 있는 커넥팅 로드에 전달되고, 다시 이것이
    커넥팅 로드의 대단부에 연결되어 있는 크랭크 축을 회전시켜게 된다. 이와 같이 압여 실린더내
    에 발생된 열에너지를 크랭크 축의 회전력을으로 바꿔 엔진 액. <P>
    미스톤이 주기적인 운동을 하여 일을 계속 하려면 연소 액이 일을 만드는 배출 액과 새로운 혼
    합기자를 실린더에 공급하여야 한다. 이를 위하여 크랭크 축에 의하여 구동되는 펌프를 사용하여
    밸브장치를 작동시켜 물기 배기구 입을 막았는 시기에 개폐한다. <P>
    실린더는 그 안에서 혼합기를 연소시키고 동시에 크랭크 케이스 고정부분과 함께 미스톤에 전달
    하는 연소 가스의 압력을 저지한다. 혼합기 가 실린더내에 옮겨된 후 배기 가스가 되어
    실린더 밖으로 배출될 때까지의 실린더내에서 가스의 주기적인 변화를 엔진의 사이클이라
    한다. 사이클에는 4행정 사이클과 2행정 사이클이 있다. <P>
</UL>

<HR>
<H3><A NAME="3"> 엔진의 성능 선도 </H3>
<UL>
    엔진에 대한 여러 가지 성능을 선도로 표시한 것이다. 출력, 외전력, 연료소비율을 표시한
    것이다. 엔진의 성능이란 실린더내에서 압력의 변화, 엔진의 외전속도, 출력, 연료소비율 등
    엔진의 능률과 관계되는 사항을 말한다.
</UL>

</BODY>
</HTML>

```

* link 되는 html 문서가 다른 html 문서에 있는 경우

* 사용법

- 서론 으로 접근
- 서론 으로 a.name1.htm 파일에 위치 표시

* 예) (수행하기)³³

```

<HTML>
<HEAD>
    <TITLE> 링크되는 HTML 문서가 여러 개의 HTML 문서로 존재하는 경우의 예제 </TITLE>
</HEAD>
<BODY>

```

³³javascript:open_win('/testhtml/onlineTest.cgi?file=a.name2.htm')

```

<H3> 엔진(Engine) </H3>
<UL>
    <A HREF="a_name1.htm#1"> 엔진의 개요 </A><BR><BR>
    <A HREF="a_name1.htm#2"> 가솔린 엔진의 작동 원리 </A><BR><BR>
    <A HREF="a_name1.htm#3"> 엔진의 성능 선도 </A><BR><BR><BR>
</UL>
</BODY>
</HTML>

```

● 실습

1. 위의 태그 예를 실행시켜본다.

2. a 태그를 사용하여 html 문서, telnet, ftp, mailto, gopher, news, file 을 수행하는 html 을 만들고 수행해본다.

1.1.3 Multimedia 삽입하는 방법

● HTML 문서에 이미지를 삽입하는 태그

- 태그

- * gif나 jpg의 이미지를 삽입할 때 사용
 - * src, align, alt, ismap, border, height, width, hspace, vspace, usemap 등 속성이 있음
 - * src 속성: 이미지를 지정할 때 사용
 - * align 속성: 이미지를 기준으로 text의 위치 지정 (top, center, bottom 중 하나 기본값은 bottom)
 - * align 속성 예) (수행하기)³⁴
- ```

<HTML>
<HEAD>
 <TITLE> 태그의 SRC, ALIGN 속성 예제 </TITLE>
</HEAD>
<BODY>
 <CENTER>
 <TABLE BORDER>
 <TR>
 <TD COLSPAN=2> SRC은 이미지 삽입, ALIGN은 기본값
 <TR>
 <TD> SRC은 이미지 삽입, ALIGN은 TOP
 <TD> SRC은 이미지 삽입, ALIGN은 CENTER
 <TR>
 <TD COLSPAN=2> SRC은 이미지 삽입, ALIGN은 BOTTOM
 <TR>
 <TD COLSPAN=2> SRC은 이미지 삽입, ALIGN은 LEFT
 <TR>
 <TD COLSPAN=2> SRC은 이미지 삽입, ALIGN은 RIGHT
 </TABLE>
 </CENTER>
</BODY>
</HTML>

```

<sup>34</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=image1.htm')

- \* alt 속성: 이미지를 설명하는 글자가 나타나게 함
- \* border 속성: 이미지의 테두리 두께를 설정
- \* height, width 속성: 이미지의 가로 세로의 크기지정
- \* hspace, vspace 속성: 이미지의 좌우 상하 여백을 지정
- \* 예) (수행하기)<sup>35</sup>

```
<HTML>
<HEAD>
<TITLE> IMG 예제의 ALT, BORDER, HEIGHT, WIDTH, HSPACE, VSPACE 속성 예제 </TITLE>
</HEAD>

<BODY>
<CENTER>
<TABLE BORDER>
<TR><TD> ALT는 인영
<TR><TD> ALT는 인영, BORDER는 5 픽셀
<TR><TD> ALT는 인영, HEIGHT는 20 픽셀, WIDTH는 20 픽셀
<TR><TD> ALT는 인영, HSPACE 20 픽셀
<TR><TD COLSPAN=2> ALT는 인영, VSPACE 20 픽셀
</TABLE>
</CENTER>
</BODY>
</HTML>
```

- 이미지를 사용하여 HTML 문서를 link 하는 태그

- \* 마우스로 이미지를 클릭하였을때 특정 문서로 연결함
- \* 사용법: <a href="html 파일 명">  text </a>
- \* 예) (수행하기)<sup>36</sup>

```
<HTML>
<HEAD>
<TITLE> 이미지를 사용하여 HTML 문서를 링크하는 태그 예제 </TITLE>
</HEAD>

<BODY>
 웨이브 사의

</BODY>
</HTML>
```

- 이미지의 맵을 사용하여 HTML 문서를 link 하는 태그

- \* 이미지의 특정 영역을 여러개로 나누어 각각의 영역을 html 파일로 link
- \* <map> 태그와 <area> 태그를 사용
- \* <map> 태그 사용예)

<sup>35</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=image2.htm')

<sup>36</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=image3.htm')

- \* 
 <map name="imagemap">
 ...
 ...
 </map>
- \* <area> 태그: 영역을 지정할때 사용, href, shape, coords, nohref, alt 등의 속성이 있음
  - href 속성: 영역에 link 될 html 문서의 URL
  - shape 속성: 이미지를 나눈형태 지정
  - coords 속성: 분할된 이미지 영역을 지정
  - nohref 속성: 이미지의 특정영역을 link 할수 없도록 할때 사용
- \* Map Edit 프로그램
  - 이미지 맵을 만드는데 사용
  - Map Edit 프로그램을 얻으려면 여기를<sup>37</sup> 클릭
- \* 예) (수행하기)<sup>38</sup>

```
<HTML>
<HEAD>
<TITLE> AREA 예제의 HREF, SHAPE, COORDS, NOHREF, ALT 속성 예제 </TITLE>
</HEAD>
```

```
<BODY>
<CENTER>

<MAP NAME="map">
<AREA HREF="imgmap1.htm" ALT="멍" SHAPE=POLY COORDS=" 56,39 161,113 161,5 56
<AREA HREF="imgmap2.htm" ALT="짖" SHAPE=POLY COORDS=" 34,27 148,111 148,139
</MAP>
</CENTER>
</BODY>
</HTML>
```

● HTML 문서에 GIF animation 을 구현

- animation 을 구현하려면 일반적으로 Java 나 CGI 프로그램을 사용해야함
- GIF89a 형식으로 animation 구현 가능
- GIF Construction Set (프로그램을 얻으려면 여기를<sup>39</sup> 클릭)
- 예) (수행하기)<sup>40</sup>

```
<HTML>
<HEAD>
<TITLE> GIF 애니메이션 예제 </TITLE>
</HEAD>

<BODY>
<CENTER>

<H2> GIF 애니메이션 동작 예제 <H2>


```

<sup>37</sup>http://itsys.hansung.ac.kr/tool/mapedit/map32dst.exe

<sup>38</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=map.htm')

<sup>39</sup>http://itsys.hansung.ac.kr/tool/gifcon/gifcon32.exe

<sup>40</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=animation.htm')

```

</CENTER>
</BODY>
</HTML>
```

- <body> 태그

- html 문서의 배경이나 글자색을 변경하기 위해 사용
- 속성으로 bgcolor, background, text, link, vlink, alink 등이 있음
- bgcolor 속성: 문서의 배경색을 지정
- background 속성: 지정한 이미지로 파일의 배경을 나타냄
- text 속성: 글자색을 변경시 사용
- link, vlink, alink 속성: hypertext 의 색깔을 변경
  - \* link: 한번도 접속하지 않은 link의 글자색
  - \* vlink: 한번이라도 접속한 link의 글자색
  - \* alink: 클릭후 놓지 않은 상태의 글자색

- 예) (수행하기)<sup>41</sup>

```
<HTML>
<HEAD>
 <TITLE> BODY 테그 예제 </TITLE>
</HEAD>

<BODY BACKGROUND="sky.jpg" TEXT="#0000FF" LINK="#8B0000" VLINK="#483D8B" ALINK="#A52A2A">

<CENTER><H2> BODY 테그 예제 </H2></CENTER>

 예이 품 '아'의 그 락 낯 음 예이지

 안성 대학교 음 예이지

 디즈 니 텐드 음 예이지

 Excite 음 예이지

</BODY>
</HTML>
```

- HTML 문서에 멀티미디어를 삽입하는 방법

- sound 삽입: au, aif, wav, ra, ram 등의 파일을 삽입 가능

- \* <embed> 태그
  - .wav 파일을 삽입
  - 속성으로 src, autostart, loop, hidden, width, height 등이 있음
  - src 속성: .wav 파일 지정
  - autostart 속성: 자동 시작 (true, false 로 지정됨)

<sup>41</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=body.htm')

- loop 속성: 반복재생 지정 (true, false 로 지정됨)
- hidden 속성: 플러그인 프로그램을 나타나게 할지를 지정 (true, false 로 지정됨)
- height, width 속성: plugin 프로그램의 크기 지정

- 예) (수행하기)<sup>42</sup>

```
<HTML>
<HEAD>
 <TITLE> EMBED 테그의 SRC, AUTOSTART, LOOP, HIDDEN, HEIGHT, WIDTH 속성 예제 </TITLE>
</HEAD>

<BODY>

<CENTER>
<EMBED SRC="wav.wav" AUTOSTART=true LOOP=false HIDDEN=false WIDTH=140 HEIGHT=60>
</CENTER>

</BODY>
</HTML>
```

- 예) <a> 태그를 사용한 예제 (수행하기)<sup>43</sup>

```
<HTML>
<HEAD>
 <TITLE> HTML 문서에 사용 드 파일 삽입 예제 </TITLE>
</HEAD>

<BODY BACKGROUND="sky.jpg" LINK="blue">

 그리그의 [메트 조크] 등 산속 마왕의 궁전에서...

 트비시의 [메트 가마스 쿠] 중 달빛...

 바아의 브란덴부르크 영주국 제 3번...

 베토벤의 고양국 제 5번...

</BODY>
</HTML>
```

- 동영상 삽입: mpg, mov, avi 등을 삽입 가능

- 예) (수행하기)<sup>44</sup>

```
<HTML>
<HEAD>
 <TITLE> HTML 문서에 동영상 파일 삽입 예제 </TITLE>
</HEAD>
```

<sup>42</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=embed.htm')

<sup>43</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=sound.htm')

<sup>44</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=avi.htm')

```
</HEAD>
<BODY BACKGROUND="sky.jpg" TEXT="#FFFFFF">

<H2> westy.avi </H2>

<H2> isvrclip.avi </H2>

</BODY>
</HTML>
```

### ● 실습

1. 위의 태그 예를 실행시켜본다.
2. img, body, embed 태그를 삽입하는 예를 작성하여 실행시켜본다.

#### 1.1.4 Frame을 구성하는 방법

##### ● HTML 문서에 frame 을 구성하는 태그

- 프레임이란 한화면을 여러개로 분할하여 독립적으로 때로는 상호 의존적으로 동작
- <frameset> 태그
  - \* 몇개의 frame 을 어떤 간격으로 만들지 지정
  - \* <body> 태그내에 삽입되지 않음
  - \* 속성으로 rows, cols 가 있음
  - \* rows 속성: 화면을 상하로 분할하며 다음과 같이 값을 지정
    - 숫자 %: 백분율로 화면을 분할 → 예) <frameset rows="40%, 60%">
    - 숫자 : 상대크기로 분할 → 예) <frameset rows="\*, 3\*">
    - 숫자: 픽셀수로 분할 → 예) <frameset rows="50,60">
  - \* cols 속성: 좌우로 분할, 값은 rows 와 동일
- <frame> 태그
  - \* 분할된 프레임에 들어갈 정보 표시
  - \* 속성으로 name, marginheight, marginwidth, src, noresize, scrolling 등이 있음
  - \* name 속성: 이 프레임을 가리킬때 사용
  - \* marginheight 속성: 프레임의 상하 여백 지정
  - \* marginwidth 속성: 프레임의 좌우 여백 지정
  - \* src 속성: 프레임에 표시될 URL이나 HTML 문서 지정
  - \* noresize 속성: 프레임을 나누는 경계선을 고정
  - \* scrolling 속성: 프레임의 경계선에 scroll bar 를 보여줄지 결정 (yes, no ,auto 를 갖음)
  - \* 예) (수행하기)<sup>45</sup>

<sup>45</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=frame.htm')

```
</HTML>
<HEAD>
<TITLE> FRAME 테그 예제 </TITLE>
</HEAD>
<FRAMESET COLS="200, 40%, * " > <!-- 브라우저를 좌우로 분할 --!>
<FRAME SRC="table1.htm" SCROLLING="YES">
<FRAME SRC="avi.htm" NAME="동영상">
<FRAMESET ROWS="3*, *" > <!-- 모든 브라우저를 상아트 분할 --!>
<FRAME SRC="http://www.hansung.ac.kr" NAME="hansung" MARGINWIDTH="20">
<FRAME SRC="http://www.bora.net" NAME="bora" MARGINHEIGHT="20" NORESIZE>
</FRAMESET></FRAMESET>
</HTML>
```

##### - <noframe> 태그

- \* frame 을 지원하지 못하는 web browser를 위한 태그
- \* frame 을 지원하지 못하는 web browser에서만 나타남
- \* 사용법: <noframe> ... </noframe>
- \* 예) (수행하기)<sup>46</sup>

```
<HTML>
<HEAD>
<TITLE> NOFRAME 테그 예제 </TITLE>
</HEAD>
<FRAMESET COLS="40%, * " > <!-- 브라우저를 좌우로 분할 --!>
<FRAME SRC="noframe2.htm" NAME="NOFRAME">
<FRAME SRC="avi.htm" NAME="동영상">
</FRAMESET>
```

```
<!-- 브라우저 기능을 지원하지 않는 웹 브라우저는 아래의 문서를 볼 수 있습니다. --!
<NOFRAME>
<CENTER>
 안성대학교 웹 페이지

 대구공대 웹 페이지
</CENTER>
</NOFRAME>
```

</HTML>

<HTML>

```
<HEAD>
<TITLE> NOFRAME 테그 예제 </TITLE>
</HEAD>
<FRAMESET ROWS="*, *" > <!-- 원쪽 브라우저를 상아트 분할 --!>
<FRAME SRC="http://www.hansung.ac.kr" NAME="hansung" MARGINHEIGHT="20" NORESIZE>
<FRAME SRC="http://www.bora.net" NAME="bora" MARGINWIDTH="20">
</FRAMESET>
```

</HTML>

##### - <target> 속성

- \* link 되어있는 HTML 문서를 다른 프레임이나 윈도우등으로 출력되게 함
- \* <a> 태그내에서 지정
- \* 사용법: <a target="프레임, HTML 키워드 이름, 윈도우" href 속성>

<sup>46</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=noframe1.htm')

\* 다른 프레임으로 출력하는 예) (수행하기)<sup>47</sup>

```
<HTML>
<HEAD>
 <TITLE> TARGET 속성 예제 </TITLE>
</HEAD>

<FRAMESET COLS="*, 4*>
 <FRAME SRC="target.htm">
 <FRAME SRC="about:blank" NAME="OUTPUT">
</FRAMESET>

</HTML>
<HTML>

<HEAD>
 <TITLE> TARGET 속성 예제 </TITLE>
</HEAD>

<BODY BACKGROUND="sky.jpg" LINK="#RRGGBB" TEXT="#RGBRGB">
 <CENTER>
 <H3> TARGET 속성 </H3>

 안성대학교 웹 페이지

 음 텍스트 모음

 자동차 강의

 </CENTER>

</BODY>
</HTML>
```

\* HTML 키워드로 출력: \_self 키워드는 hypertext 가 있는 프레임, default 입

\* 예) (수행하기)<sup>48</sup>

```
<HTML>
<HEAD>
 <TITLE> _self 키워드로 출력하는 예제 </TITLE>
</HEAD>

<FRAMESET ROWS="*, 3*> <!-- 프레임을 상아트 분할 --!>
 <FRAME SRC="fr_self.htm">
 <FRAME SRC="http://www.hansung.ac.kr">
</FRAMESET>

</HTML>
<HTML>

<HEAD>
 <TITLE> _self 키워드로 출력하는 예제 </TITLE>
</HEAD>

<BODY BGCOLOR="#FFFAFA" TEXT="#FFFFFF" LINK="#000000" VLINK="#RGBRGB" ALINK="#FRFGBH">

 <!-- 이미지 전송한 웹페이지 -->

 <!-- 이미지 전송한 웹페이지 -->
</BODY>
```

\* \_top 키워드로 출력: 프레임에 상관없이 전체윈도우에 출력

\* 예) (수행하기)<sup>49</sup>

```
<HTML>
<HEAD>
 <TITLE> _top 키워드로 출력하는 예제 </TITLE>
</HEAD>

<FRAMESET ROWS="*, 3*> <!-- 프레임을 상아트 분할 --!>
 <FRAME SRC="fr_top.htm">
 <FRAME SRC="http://www.hansung.ac.kr">
</FRAMESET>

</HTML>
<HTML>

<HEAD>
 <TITLE> _top 키워드로 출력하는 예제 </TITLE>
</HEAD>

<BODY BGCOLOR="#FFFAFA" TEXT="#FFFFFF" LINK="#000000" VLINK="#RGBRGB" ALINK="#FRFGBH">

 <!-- 이미지 전송한 웹페이지 -->

 <!-- 이미지 전송한 웹페이지 -->
</BODY>
```

\* \_blank 키워드로 출력: 프레임에 상관없이 새로운 윈도우를 만들어 출력

\* 예) (수행하기)<sup>50</sup>

```
<HTML>
<HEAD>
 <TITLE> _blank 키워드로 출력하는 예제 </TITLE>
</HEAD>

<FRAMESET ROWS="*, 3*> <!-- 프레임을 상아트 분할 --!>
 <FRAME SRC="fr_blank.htm">
 <FRAME SRC="http://www.hansung.ac.kr">
</FRAMESET>

</HTML>
<HTML>

<HEAD>
 <TITLE> _blank 키워드로 출력하는 예제 </TITLE>
</HEAD>

<BODY BGCOLOR="#FFFAFA" TEXT="#FFFFFF" LINK="#000000" VLINK="#RGBRGB" ALINK="#FRFGBH">

 <!-- 이미지 전송한 웹페이지 -->

 <!-- 이미지 전송한 웹페이지 -->
</BODY>
```

<sup>47</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=abblank.htm')

<sup>48</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=\_self.htm')

<sup>49</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=\_top.htm')

<sup>50</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=\_blank.htm')

```

</BODY>
</HTML>
* 특정 윈도우로 출력하는 방법
* 예) (수행하기)51
<HTML>

<HEAD>
<TITLE> 특정 윈도우로 출력하는 예제 </TITLE>
</HEAD>

<FRAMESET ROWS="*, 3*> <!— 프레임을 상아트 분할 --!>
<FRAME SRC="fr_window.htm">
<FRAME SRC="http://www.hansung.ac.kr">
</FRAMESET>

</HTML>
<HTML>

<HEAD>
<TITLE> 특정 윈도우로 출력하는 예제 </TITLE>
</HEAD>

<BODY BGCOLOR="#FFFFFF" TEXT="#FFFFFF" LINK="#000000" VLINK="#RGBRGB" ALINK="#FRFGFB">

페이지 상의 전자상거래 링크

페이지 상의 보라넷 링크

</BODY>
</HTML>

```

#### ● 실습

1. 위의 frame 예를 실행시켜본다.
2. 자신만의 frame 을 구성하는 html을 작성하고 여러가지 target 속성을 테스트 해본다.

## 1.2 CGI 사용하기

### ● CGI (Common Gateway Interface) 프로그램의 정의

#### - 두가지 web 처리 방식

1. client 가 web server 에게 자료를 요청하면 web server 가 자료를 전송하는 방식
2. client 가 자료를 web server 에게 전송하면 web server 는 이를 처리할 프로그램을 이용처리하는 방식

#### - 두번째 방식에서 web server 가 넘겨준 자료를 처리하는 프로그램을 CGI 프로그램이라함

<sup>51</sup>javascript:open\_win('testhtml/onlineTest.cgi?file=window.htm')

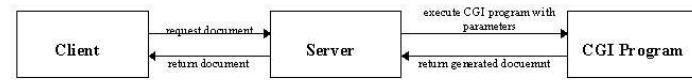


그림 4.1: CGI 동작 방식

#### - CGI 프로그램을 위하여 사용되는 프로그래밍 언어

- \* C, C++, PERL, TCL, Unix shell, Visual Basic, Applescript 등이 있음
- \* Unix OS: C, C++, PERL, TCL, Unix shell 이 유용
- \* Windows NT: C, C++, Visual Basic 등이 유용
- \* Macintosh: Applescript 유용
- \* 실행파일 생성 tool: C, C++
- \* script tool: PERL, TCL, Unix shell, Applescript

#### - 보통 web server directory 밑에 cgi-bin 안에 CGI 프로그램이 있다

### ● CGI 프로그램을 사용하기 위한 HTML 태그

#### - <form> 태그

- \* html 문서내에서 CGI 를 호출하는데 사용
- \* 속성으로 action, method, enctype 이 있음
- \* action 속성: CGI 프로그램의 위치를 URL로 지정
- \* method 속성: server로 전달 방식 지정
  - get 방식: CGI 환경변수인 query string으로 전달
  - post 방식: 환경변수 및 표준입력을 통해 전달
  - get 방식은 전달되는 자료의 양이 제한됨, post는 제한없음
- \* enctype 속성: 전달 타입을 지정
  - 기본값타입은 application/x-www-form-urlencoded

#### - <input> 태그

- \* 입력양식을 지정, 반드시 <form> 태그내에 있어야함
- \* 속성으로 name, type 이 있음
- \* name 속성: name 으로 입력자료를 구분
- \* value 속성: 초기입력자료를 표시
- \* checked 속성: 입력값의 초기화
- \* type 속성: 입력양식의 타입을 지정
- \* checkbox 입력양식의 예) (수행하기)<sup>52</sup>

```

<HTML>
<HEAD>
<TITLE> 체크박스 입력양식 예제 </TITLE>

```

<sup>52</sup>javascript:open\_win('testhtml/onlineTest.cgi?file=checkbox.htm')

표 4.3: type 속성의 종류

type 속성	양식	설명
checkbox	체크박스 입력양식	여러개중 다수개 선택가능
file	파일 upload 입력양식	파일 upload 를 수행
radio	라디오버튼 입력양식	여러개중 하나만 선택가능
text	text 입력양식	한줄의 텍스트를 입력 가능
reset	리셋버튼 입력양식	입력을 지우고 재입력시 사용
submit	자료전송 입력양식	입력된 자료를 전송할때 사용
button	버튼 입력양식	버튼모양을 만들때 사용 name 은 사용되지 않음
password	암호 입력양식	암호입력시 사용, * 로 표시됨
hidden	숨겨진 입력양식	web browser 에는 표시되지 않음

```
</HEAD>
<BODY BACKGROUND="sky.jpg" TEXT="#008000">

<CENTER><H1> 당신의 PC 시스템 상태 </H1></CENTER>

<FORM>

<INPUT TYPE="checkbox" NAME="checkbox_name1"> 모뎀
<INPUT TYPE="checkbox" NAME="checkbox_name2" CHECKED> 사용 드懒得
<INPUT TYPE="checkbox" NAME="checkbox_name3"> 비디오懒得
<INPUT TYPE="checkbox" NAME="checkbox_name4"> LAN懒得
<INPUT TYPE="checkbox" NAME="checkbox_name5"> MPEG懒得

</FORM>
</BODY>
</HTML>
```

- file upload 입력양식의 예) (수행하기)<sup>53</sup>

```
<HTML>
<HEAD>
<TITLE> 파일 업로드 입력양식 예제 </TITLE>
</HEAD>
<BODY BACKGROUND="sky.jpg" TEXT="#008000">

<CENTER>
<H1> 업로드 할 파일의 경로와 이름을 입력하시오. </H1>

<FORM>
<INPUT TYPE="file" NAME="file_name">
</FORM>
</CENTER>
</BODY>
</HTML>
```

- 라디오 버튼 입력양식의 예) (수행하기)<sup>54</sup>

```
<HTML>
<HEAD>
<TITLE> 라디오 버튼 입력양식 예제 </TITLE>
</HEAD>
<BODY BACKGROUND="sky.jpg" TEXT="#008000">


```

<sup>53</sup> javascript:open\_win('/testhtml/onlineTest.cgi?file=upload.htm')

<sup>54</sup> javascript:open\_win('/testhtml/onlineTest.cgi?file=radio.htm')

```
<CENTER><H1> 당신의 직업 </H1></CENTER>

```

```
<FORM>

<INPUT TYPE="radio" NAME="radio_name1"> 학생
<INPUT TYPE="radio" NAME="radio_name2"> 기술자
<INPUT TYPE="radio" NAME="radio_name3" CHECKED> 고수
<INPUT TYPE="radio" NAME="radio_name4"> 디자이너
<INPUT TYPE="radio" NAME="radio_name5"> 연예인

</FORM>
```

```
</BODY>
```

```
</HTML>
```

- 텍스트 입력양식의 예) (수행하기)<sup>55</sup>

```
<HTML>
<HEAD>
<TITLE> 텍스트 입력양식 예제 </TITLE>
</HEAD>
<BODY BACKGROUND="sky.jpg" TEXT="#008000">

<CENTER><H1> 건강 기록부 </H1></CENTER>

<FORM>
<H3>
 성명 <INPUT TYPE="text" NAME="name" SIZE=10>
 나이 <INPUT TYPE="text" NAME="grade">
 주소 <INPUT TYPE="text" NAME="address" SIZE=55 MAXLENGTH=200>
 몸무게(kg) <INPUT TYPE="text" NAME="weight">
 키(cm) <INPUT TYPE="text" NAME="length">
 성별 <INPUT TYPE="text" NAME="sex" VALUE="여자">
</H3>
</FORM>
```

```
</BODY>
```

```
</HTML>
```

- 리셋버튼 입력양식의 예) (수행하기)<sup>56</sup>

```
<HTML>
<HEAD>
<TITLE> 리셋버튼 입력양식 예제 </TITLE>
</HEAD>
<BODY BACKGROUND="sky.jpg" TEXT="#008000">

<CENTER><H1> 건강 기록부 </H1></CENTER>

<FORM>
<H3>
 성명 <INPUT TYPE="text" NAME="name" SIZE=10>
 나이 <INPUT TYPE="text" NAME="grade">
 주소 <INPUT TYPE="text" NAME="address" SIZE=55 MAXLENGTH=200>
 몸무게(kg) <INPUT TYPE="text" NAME="weight">
 키(cm) <INPUT TYPE="text" NAME="length">
 성별 <INPUT TYPE="text" NAME="sex" VALUE="여자">
<CENTER><INPUT TYPE="reset" VALUE="재입력 (Reset)" ></CENTER>
</H3>
</FORM>
</BODY>
```

```
</HTML>
```

<sup>55</sup> javascript:open\_win('/testhtml/onlineTest.cgi?file=text.htm')

<sup>56</sup> javascript:open\_win('/testhtml/onlineTest.cgi?file=reset.htm')

- 자료전송 입력 양식의 예) (수행하기)<sup>57</sup>

```
<HTML>
<HEAD>
<TITLE> 자료 전송 입력 양식 예제 </TITLE>
</HEAD>

<BODY BACKGROUND="sky.jpg" TEXT="#008000">

<CENTER><H1> 긴 문 기록부 </H1></CENTER>

<FORM>
<H3>
 성명 <INPUT TYPE="text" NAME="name" SIZE=10>
 학년 <INPUT TYPE="text" NAME="grade">
 주소 <INPUT TYPE="text" NAME="address" SIZE=55 MAXLENGTH=200>
 몸무게(kg) <INPUT TYPE="text" NAME="weight">
 키(cm) <INPUT TYPE="text" NAME="length">
 성별 <INPUT TYPE="text" NAME="sex" VALUE="여자">

<INPUT TYPE="submit" VALUE="자료 전송(Submit)" >
<INPUT TYPE="button" VALUE="재입력(Reset)" >
</H3>
</FORM>
</BODY>
</HTML>
```

- 버튼 입력양식의 예) (수행하기)<sup>58</sup>

```
<HTML>
<HEAD>
<TITLE> 버튼 입력 양식 예제 </TITLE>
</HEAD>

<BODY BACKGROUND="sky.jpg" TEXT="#008000">
<CENTER>

<FORM>
<INPUT TYPE="button" VALUE="=< 전화번호 >" >
</FORM>
</CENTER>

</BODY>
</HTML>
```

- 암호 입력양식의 예) (수행하기)<sup>59</sup>

```
<HTML>
<HEAD>
<TITLE> 암호 입력 양식 예제 </TITLE>
</HEAD>

<BODY BACKGROUND="sky.jpg" TEXT="#008000">

<CENTER><H1> 이름과 암호를 입력하세요 </H1></CENTER>

<FORM>
<H3>
 이름 <INPUT TYPE="text" NAME="text_name" SIZE=10>
 암호 <INPUT TYPE="password" NAME="password_name">

<INPUT TYPE="submit" VALUE="자료 전송(Submit)" >
<INPUT TYPE="reset" VALUE="재입력(Reset)" >
</H3>
</FORM>
</BODY>
</HTML>
```

<sup>57</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=submit.htm')<sup>58</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=button.htm')<sup>59</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=passwd.htm')

```
</H3>
</FORM>
</BODY>
</HTML>
```

- 숨겨진 입력양식의 예) (수행하기)<sup>60</sup>

```
<HTML>
<HEAD>
<TITLE> 숨겨진 입력 양식 예제 </TITLE>
</HEAD>

<BODY BACKGROUND="sky.jpg" TEXT="#008000">
<CENTER>

<FORM>
<INPUT TYPE="hidden" NAME="hidden_name" VALUE="hidden_value">
<INPUT TYPE="submit" VALUE="자료 전송(Submit)" >
</FORM>
</CENTER>
```

```
</BODY>
</HTML>
```

## - &lt;select&gt; 태그

- \* 여러개의 목록중 하나를 선택하게 함
- \* 반드시 <form> 태그내에 있어야함
- \* 속성으로 name, size, multiple 등이 있음
- \* name 속성: select 태그의 이름 지정
- \* size 속성: 목록의 줄수를 지정 기본값은 1
- \* multiple 속성: 여러개의 목록을 동시에 선택시 사용

## - &lt;option&gt; 태그

- \* 목록을 나열할때 사용
- \* 속성으로 value, selected 를 갖음
- \* value 속성: CGI로 전달되는 값 지정
- \* selected 속성: 초기목록으로 지정됨

- <select> 와 <option> 태그를 사용한 예) (수행하기)<sup>61</sup>

```
<HTML>
<HEAD>
<TITLE> SELECT 태그, OPTION 태그 예제 </TITLE>
</HEAD>

<BODY BACKGROUND="sky.jpg" TEXT="#008000">
<CENTER>

<H2> 선호하는 과자 종류 </H2>

<FORM>
<SELECT NAME="name1">
<OPTION VALUE="potato"> 감자칩
```

<sup>60</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=hidden.htm')<sup>61</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=select.htm')

```

<OPTION VALUE="onion" SELECTED> 양파 향
<OPTION VALUE="ninel"> 청구
<OPTION VALUE="cho.co"> 초코 칩 쿠키
<OPTION VALUE="roast"> 포مي
</SELECT>

<H2> 주요 사용하는 검색 엔진 </H2>

<SELECT NAME="name2" SIZE=5 MULTIPLE>
 <OPTION VALUE="kachi" DISABLED> 까치네
 <OPTION VALUE="mochanni"> 미스 마찰니
 <OPTION VALUE="simmany" SELECTED> 심마니
 <OPTION VALUE="aistar"> 청보 탕정
 <OPTION VALUE="kyahoo"> 야후 코리아
 <OPTION VALUE="altavista"> 알타비스타
</SELECT>
</FORM>
</CENTER>

```

&lt;/BODY&gt;

## - &lt;textarea&gt; 태그

- \* 많은 양의 자료를 전송시 사용
- \* 속성으로 name, rows, cols 를 갖음
- \* name 속성: 태그 이름 지정
- \* rows 속성: 입력영역의 줄수 지정
- \* cols 속성: 입력영역의 칸수 지정

- <textarea> 태그 사용 예 (수행하기)<sup>62</sup>

```

<HTML>
<HEAD>
 <TITLE> TEXTAREA 예제 예제 </TITLE>
</HEAD>

<BODY BACKGROUND="sky.jpg" text="#008000">

 <CENTER><H2> 네시만에 자신의 의견을 올려 주시기 바랍니다..! </H2>
 <FORM>

 <TEXTAREA NAME="memo" ROWS=17 COLS=60>
 </TEXTAREA>

 <INPUT TYPE="submit" VALUE="주요 전송(Submit)">
 <INPUT TYPE="reset" VALUE="재입력(Restart)">
 </FORM>
 </CENTER>

</BODY>
</HTML>

```

## ● 실습

1. 위의 form 예를 실행시켜본다.
2. 자신만의 form 을 구성하는 html을 작성하고 테스트해본다.

<sup>62</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=textarea.htm')

## 1.3 C 언어를 사용한 CGI 프로그래밍

## 1.3.1 Basic concept

- CGI 프로그램에서 web browser로 자료를 보내는 형식
  - 헤더와 자료로 구성됨
  - 헤더
    - \* 헤더는 전송 자료의 종류를 지정
    - \* 헤더 형식: Content-type: 자료 형식
    - \* 형식의 예

표 4.4: 자료 형식

파일 종류	자료형식
HTML 문서	text/html
암호화된 HTML 문서	application/x-www-form-urlencoded
text 파일	text/plain
gif 파일	image/gif
jpeg 파일	image/jpeg
postscript 파일	application/postscript
zip 압축파일	x-zip-compressed
avi 파일	video/x-msvideo
mpeg 파일	video/mpeg

## - 간단한 문자열을 출력할 때

- \* CGI 프로그램 예)
 

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

```

main()
{
 printf("Content-type: text/html\n");
 printf("\n");

 printf("<HTML>");
 printf("<HEAD>");
 printf("<TITLE> 웹 검색기 상에 간단한 문자열을 출력하는 CGI 프로그램 </TITLE>");
 printf("</HEAD>");
 printf("<BODY BGCOLOR=#FFCC00 TEXT=#0000FF>");
 printf("<CENTER>

");
 printf("<H2>이 프로그램은 웹 검색기 상에 간단한 문자열을 출력하는 CGI 프로그램이다.</H2>");
 printf("<CENTER>

");
 printf("</BODY>");
 printf("</HTML>");

}/* main() */
* CGI 프로그램을 호출하는 HTML 예) (수행하기)63
<HTML>

```

<sup>63</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=cgiout.htm')

```

<HEAD>
 <TITLE> 웹 검색기 상에 간단한 문자열을 출력하는 CGI 프로그램을 호출하는 HTML 문서 </TITLE>
</HEAD>

<BODY BACKGROUND="sky.jpg" TEXT="#0000FF">
 <CENTER>

 <H2>
 CGI 프로그램을 호출하면서 여기를 클릭하시오 </H2>
 </CENTER>
 </BODY>
</HTML>

```

- web server 의 CGI 환경 변수

- CGI 프로그램을 수행하는데 도움이 되는 환경 변수들

표 4.5: CGI 환경 변수

환경 변수	의미
SERVER_SOFTWARE	현재 설치되어 있는 웹서버의 종류와 버전을 나타냄
SERVER_NAME	웹서버의 Domain Name을 나타냄
GATEWAY_INTERFACE	웹서버의 프로토콜을 나타냄
SERVER_PORT	웹서버의 포트번호를 나타냄 보통 80
REQUEST_METHOD	CGI 프로그램으로 자료를 보내는 방식 (GET,POST)을 나타냄
HTTP_ACCEPT	사용자의 웹검색기에서 사용될수 있는 MIME 타입을 나타냄
HTTP_USER_AGENT	사용자의 웹검색기에 대한 정보를 나타냄
HTTP_REFERER	현재 HTML 문서의 URL을 나타냄
SCRIPT_NAME	현재 실행되고 있는 CGI 프로그램의 경로와 이름을 나타냄
QUERY_STRING	GET 방식으로 CGI 프로그램에 전달되는 자료, 즉 ? 다음에오는 내용
REMOTE_HOST	사용자가 사용하고있는 클라이언트 이름을 나타냄
REMOTE_ADDR	사용자가 사용하고있는 클라이언트의 IP address를 나타냄

- 환경 변수를 웹검색기 상에 출력하는 예)

```

* CGI 프로그램 예)
#include <stdio.h>

main()
{
 printf("Content-type: text/html\n");
 printf("\n");

 printf("<HTML>");
 printf("<HEAD>");
 printf("<TITLE> 웹 검색기 상에 웹 서버의 환경 변수를 출력하는 CGI 프로그램 </TITLE>");
 printf("</HEAD>");
 printf("<BODY BGCOLOR="#FFCC00" TEXT="#0000FF">");
 printf("
");
 printf("<CENTER><H2> 웹 서버의 환경 변수 </H2></CENTER>
");
 printf("<H3> SERVER_SOFTWARE : %s
", getenv("SERVER_SOFTWARE"));
 printf("<H3> SERVER_NAME : %s
", getenv("SERVER_NAME"));
 printf("<H3> GATEWAY_INTERFACE : %s
", getenv("GATEWAY_INTERFACE"));
 printf("<H3> SERVER_PROTOCOL : %s
", getenv("SERVER_PROTOCOL"));
 printf("<H3> SERVER_PORT : %s
", getenv("SERVER_PORT"));
 printf("<H3> REQUEST_METHOD : %s
", getenv("REQUEST_METHOD"));
}

```

```

printf("HTTP_ACCEPT : %s
", getenv("HTTP_ACCEPT"));
printf("HTTP_USER_AGENT : %s
", getenv("HTTP_USER_AGENT"));
printf("HTTP_REFERER : %s
", getenv("HTTP_REFERER"));
printf("SCRIPT_NAME : %s
", getenv("SCRIPT_NAME"));
printf("QUERY_STRING : %s
", getenv("QUERY_STRING"));
printf("REMOTE_HOST : %s
", getenv("REMOTE_HOST"));
printf("REMOTE_ADDR : %s
<H3>", getenv("REMOTE_ADDR"));
printf("</BODY>");
printf("</HTML>");

}/* main() */
* CGI 프로그램을 호출하는 HTML 예) (수행하기)64
<HTML>
<HEAD>
 <TITLE> 웹 검색기 상에 웹 서버의 환경 변수를 출력하는 CGI 프로그램을 호출하는 HTML 예) (수행하기)</TITLE>
</HEAD>

<BODY BACKGROUND="sky.jpg" TEXT="#0000FF">
 <CENTER>

 <H2>
 웹 서버의 환경 변수를 출력하여 여기를 클릭하시오 </H2>
 </CENTER>
 </BODY>
</HTML>

```

### 1.3.2 Submit method

- CGI 프로그램으로 자료를 전달하는 방식
  - encoding: CGI 프로그램으로 자료가 전달될때 방식
    - \* 영문자, 숫자 이외에 다른 문자는 %16진수의 ASCII code 형태로 변환
    - \* 공백문자는 + 로 변환
    - \* 입력양식 변수값은 변수=변수값 형태로 변환
    - \* 변수사이는 & 로 구분
  - decoding: CGI 프로그램이 자료정보를 해석할때 사용
  - GET 방식
    - \* CGI 환경변수인 QUERY\_STRING 을 통해 자료 전달
    - \* 자료의 크기제한이 있음
    - \* 보안에 취약함
  - POST 방식
    - \* CGI 프로그램으로 표준 입출력을 통해서 자료전달
    - \* 자료의 크기제한이 없음
    - \* 보안상의 취약점이 GET보다 덜 함
- 보다 구체적인 내용
  - 본 과목에서는 학생들이 C 언어에 익숙하지 않음으로 구체적으로 다루지 않음

<sup>64</sup>javascript:open\_win('/testhtml/onlineTest.cgi?file=cgienv.htm')

- 구체적인 것은 web programming 과목에서 다룰 것임
- 개인적으로 구체적으로 알고 싶은 학생은 여기를<sup>65</sup> 참조할 것

## 1.4 Java script 사용하기

### 1.4.1 Java script basic

- 정의

- Sun Microsystems 사가 개발한 Object Oriented Script Language
- client에서 처리됨
- 객체정의 가능, 클래스 정의 및 상속 불가능

- 특징

- 대소문자의 구별이 엄격
- HTML 태그 사용 가능
- 인터프리터 방식의 언어로서 Web browser에서 해석하여 실행
- 사용 객체: 사용자정의 객체, 자바 스크립트에서 제공하는 객체, web browser에서 제공하는 객체
- 사용자에 의한 event 가 클라이언트에서 직접 처리됨

- Java script 와 HTML 의 차이점

표 4.6: Java script 와 HTML 의 차이점

Java script	HTML
web 문서의 외형 및 event handler로 동작 가능 객체를 사용 클라이언트에서 바로 처리	web 문서의 외형을 작성 태그를 사용 서버에 전송하여 처리

- Java script 와 Java 의 차이점

- Java script 를 HTML 문서에 삽입하기 위한 태그

- <script> 태그: 보통 <head>에 객체정의, <body>에 객체호출
  - \* language 속성: script 언어를 지정 → 예) <script language="JavaScript"> 자바스크립트
  - \* src 속성: 코드 지정 → 예) <script language="JavaScript" src="JavaScript.js"> ..
- comment 태그
  - \* Java script 를 지원하지 않는 web browser에서 코드가 화면에 나타나지 않게 하려면 코드를 주석처리
  - \* 예) (수행하기)<sup>66</sup>

<sup>65</sup><http://itsys.hansung.ac.kr/lec/webprogram/mylec/book>

<sup>66</sup>[javascipt:open\('testjscript/onlineTest.cgi?file=commenttag.js.htm'\)](javascipt:open('testjscript/onlineTest.cgi?file=commenttag.js.htm'))

표 4.7: Java script 와 Java 의 차이점

Java script	Java
web browser가 바로 해석함으로 처리 source가 공개되어 보안성이 없음 HTML 문서내의 javascript 태그에 삽입 변수형이 선언되지 않아도 됨 객체에 대한 참조가 실행시 체크됨 클래스를 정의하거나 상속할 수 없다	컴파일된 byte code를 해석하여 실행 소스가 공개되지 않아 보안적이다 applet 형태로 HTML 문서에 포함 반드시 변수형이 선언되어야 함 객체에 대한 참조가 컴파일 시 체크됨 클래스를 정의하거나 참조할 수 있다

```
<HTML>
<HEAD>
<TITLE> 주석 태그를 사용 안 자바 스크립트 예제 </TITLE>
<SCRIPT LANGUAGE="JavaScript">
document.write("<H2> 주석 태그를 사용 안 자바 스크립트 예제 </H2>")
</SCRIPT>
<SCRIPT LANGUAGE="JavaScript">
<!--
document.write("<H2> 주석 태그를 사용하고 있지 않습니다.")

//-->
</SCRIPT>
</HEAD>
<BODY>
</BODY>
</HTML>
```

- 기본 문법

- 주석문 (comments)

- \* HTML 내에서는 <!-- text --> 가 주석문으로 사용
- \* Java script에서는 /\* text \*/ 나 // 이 주석문으로 사용됨

- 자료형 (data type)

- \* 정수: 8진수 → 0숫자, 16진수 → 0x숫자, 10진수 → 숫자
- \* 부동 소수점:  $5.2 \times 10^6 \rightarrow 5.0E6$  으로 표현
- \* Boolean: 1비트 크기 → Netscape에서는 true, false로 Explorer에서는 1이나 0을 갖음
- \* 문자열: " " 나 , , 를 사용 (특수문자 사용 가능)
- \* null: 아무 값도 정해져 있지 않음을 의미

- 변수 (variables)

- \* 영문이나 \_ 문자로 시작 (예약어는 사용 할 수 없음)

표 4.8: 특수문자 사용예

특수문자	설명
\b	backspace
\n	new line
\r	carriage return
\t	tab
\'	작은 따옴표
\"	큰 따옴표
\\	\ 기호

표 4.9: Java script에서 사용되는 예약어

abstract	const	final	import	null	super	try
boolean	continue	finally	in	package	switch	var
break	default	float	instanceof	private	synchronized	void
byte	do	for	int	protected	this	while
case	double	function	interface	public	throw	with
catch	else	goto	long	return	throws	
char	extends	if	native	short	transient	
class	false	implements	new	static	true	

\* 변수의 할당된 type을 임의로 변경 가능

- 정수형 변수로 선언후 문자열 할당 가능
- 문자열과 부동소수점 연산 가능 → 예) tmp = "test" + 14.5

\* 변수선언 예 (수행하기)<sup>67</sup>

```
<HTML>
<HEAD>

<TITLE> 변수를 사용한 자바 스크립트 예제 </TITLE>
<SCRIPT LANGUAGE = "JavaScript">
<!--

 function sub1(){
 i = 5
 document.write("<H3>sub1() 사용자 정의 함수에서 i 의 값 -> " + i + "</H3><P>")
 sub2()
 document.write("<H3>sub1() 사용자 정의 함수에서 i 의 값 -> " + i + "</H3><P>")
 }/* sub1() */

 function sub2(){
 var i = 6
 document.write("<H3>sub2() 사용자 정의 함수에서 i 의 값 -> " + i + "</H3><P>")
 }/* sub2() */

//-->
</SCRIPT>

</HEAD>
<BODY>
```

<sup>67</sup>javascript:open\_win('/testjscript/onlineTest.cgi?file=var\_js.htm')

- 산술 연산자 (operators)

- \*
- 
- \*
- /
- \* 예) (수행하기)<sup>68</sup>

```
<HTML>
<HEAD>

<TITLE> 산술 연산자를 사용한 자바 스크립트 예제 </TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--

var num = 34.2 + 5.0
document.write("<H4>num = 34.2+5.0 -> " + num +<P>")

num = 34.2 - 5.0
document.write("num = 34.2-5.0 -> " + num +<P>)

num = 34.2 * 5.0
document.write("num = 34.2*5.0 -> " + num +<P>)

num = 34.2 / 5.0
document.write("num = 34.2/5.0 -> " + num +<P>)

num = 34.2 % 5.0
document.write("num=34.2%5.0 = " + num +<P>)

++num
document.write("++num = " + num +<P>)

num++
document.write("num++ = " + num +<P>)

--num
document.write("--num = " + num +<P>)

num--
document.write("num-- = " + num +<P>)

var result = ++num
document.write("result = ++num -> " + result +<P>)

result = num++
document.write("result = num++ -> " + result +<P>)
result = --num
document.write("result = --num -> " + result +<P>)

result = num--
document.write("result = num-- -> " + result +<H4>")

//-->
</SCRIPT>
```

<sup>68</sup>javascript:open\_win('/testjscript/onlineTest.cgi?file=arithmetic\_js.htm')

```

</HEAD>
<BODY>
</BODY>
</HTML>

- 관계 연산자
* 크기 비교: <, <=, >, >=, ==, != 사용
* 예) (수행하기)69
<HTML>

<HEAD>
<TITLE> 관계 연산자를 사용 안 해도 됨을 예제 </TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--
document.write("<H2> 2 < 1 -> " + (2 < 1) + "<P>")
document.write(" 5 <= 5 -> " + (5 <= 5) + "<P>")
document.write(" 2 > 1 -> " + (2 > 1) + "<P>")
document.write(" 2 >= 1 -> " + (2 >= 1) + "<P>")
document.write(" 2 == 1 -> " + (2 == 1) + "<P>")
document.write(" 2 != 1 -> " + (2 != 1) + "</H2>")
//-->
</SCRIPT>
</HEAD>
<BODY>
</BODY>
</HTML>

```

#### - 논리 연산자

- \* AND 연산자 (&&): 두개조건이 모두 참일때만 참
- \* OR 연산자 (||): 두개조건중 하나이상이 참일때 참
- \* XOR 연산자 (^): 두개조건중 참,거짓 이거나 거짓, 참일때만 참
- \* NOT 연산자 (!): 참이면 거짓, 거짓이면 참으로 바뀜
- \* ? 연산자 (?): ?앞의 조건이 참이면 :앞에것 거짓이면 :뒤에것 리턴
- \* 예) (수행하기)<sup>70</sup>

```

<HTML>

<HEAD>
<TITLE> 논리 연산자를 사용 안 해도 됨을 예제 </TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--
document.write("<H2>true && false ->" + (true && false) + "<P>")

```

<sup>69</sup> javascript:open\_win('/testjs/testonlineTest.cgi?file=compare.js.htm')

<sup>70</sup> javascript:open\_win('/testjs/testonlineTest.cgi?file=condition.js.htm')

```

document.write(" 0 || 1 -> " + (0||1) + "<P>")
document.write(" 0 ^ 1 -> " + (0^1) + "<P>")
document.write("<H2> !true => " + !true + "</H2>")
var result = (0!=1) ? 2 : 3
document.write(" (0!=1) ? 2 : 3 -> " + result + "</H2>")

```

```
//-->
```

```
</SCRIPT>
```

```
</HEAD>
```

```
<BODY>
```

```
</BODY>
```

```
</HTML>
```

#### - 대입 연산자

표 4.10: 대입 연산자

종류	기능	예)
=	오른쪽값을 왼쪽변수에 저장	x = 3;
+=	오른쪽값을 왼쪽변수에 합후 왼쪽변수에 저장	x+= y;
-=	오른쪽값을 왼쪽변수에 뺀후 왼쪽변수에 저장	x-= y;
=	오른쪽값을 왼쪽변수에 곱한후 왼쪽변수에 저장	x*= y;
/=	오른쪽값을 왼쪽변수에 나눈후 왼쪽변수에 저장	x/= y;
%=	오른쪽값을 왼쪽변수에 나눈후 나머지를 왼쪽변수에 저장	x% = y;

- \* 예) (수행하기)<sup>71</sup>

```
<HTML>
```

```
</HEAD>
```

```
<TITLE> 대입 연산자를 사용 안 해도 됨을 예제 </TITLE>
```

```
<SCRIPT LANGUAGE="JavaScript">
<!--
```

```
var x = 3
var y = 4
```

```
document.write("<H4>변수 : x=" + x + ", y=" + y + "<P>")
document.write(" 출력 : x = " + x + "<HR>")
```

```
document.write("변수 : x=" + x + ", y=" + y + "<P>")
x+=y
document.write(" 출력 : x+=y -> " + x + "<HR>")
```

```
document.write("변수 : x=" + x + ", y=" + y + "<P>")
x-=y
document.write(" 출력 : x-=y -> " + x + "<HR>")
```

```
document.write("변수 : x=" + x + ", y=" + y + "<P>")
x*=y
document.write(" 출력 : x*=y -> " + x + "<HR>")
```

```
document.write("변수 : x=" + x + ", y=" + y + "<P>")
x/=y
document.write(" 출력 : x/=y -> " + x + "<HR>")
```

<sup>71</sup> javascript:open\_win('/testjs/testonlineTest.cgi?file=assign.js.htm')

```

document.write("변수 : x=" + x + ", y=" + y + "<P>")
x%y
document.write("출 력 : x%y -> " + x + "</H3>")

//-->
</SCRIPT>

</HEAD>

<BODY>

</BODY>

</HTML>

- 비트 연산자
 * AND(&), OR(|), XOR(^), 왼쪽 shift (<<), 오른쪽 shift (>>)
 * 예) (수행하기)72
<HTML>

<HEAD>

<TITLE> |= 연산자를 사용한 자바 스크립트 예제 </TITLE>

<SCRIPT LANGUAGE="JavaScript">
<!--

document.write("<H2>115 & 3 -> " + (115 & 3) + "<P>")
document.write("115 | 3 -> " + (115 | 3) + "<P>")
document.write("115 ^ 3 -> " + (115 ^ 3) + "<P>")
document.write("-115 >> 3 -> " + (-115 >> 3) + "<P>")
document.write("115 << 3 -> " + (115 << 3) + "</H2>")

//-->
</SCRIPT>

</HEAD>

<BODY>

</BODY>

</HTML>

```

#### - 연산자 우선순위

```

* 예) (수행하기)73
<HTML>

<HEAD>

<TITLE> 연산자의 우선 순위를 적용 시켜 연산을 처리하는 자바 스크립트 예제 </TITLE>

<SCRIPT LANGUAGE="JavaScript">
<!--

var a = 3
var b = 4
document.write("<H2>")
document.write("변수 : a=3, b=4 <HR> (a+b)/a = "+ (a+b)/a)
document.write("<P> a++ + --b * b = " + (a++ + --b * b))
-->

```

<sup>72</sup> javascript:open\_win('/testjscript/onlineTest.cgi?file=bit\_js.htm')
<sup>73</sup> javascript:open\_win('/testjscript/onlineTest.cgi?file=priority\_js.htm')

표 4.11: 연산자 우선순위

순서	해당 연산자
1	[ ], ( )
2	+,-,!+
3	*, /, %
4	, -+
5	<<, >>
6	<, <=, >, >=
7	==, !=
8	&
9	^
10	
11	&&
12	
13	?, :
14	=, +=, -=, *=, /=, %=,  =, >>=, <<=, ^=

&lt;/SCRIPT&gt;

&lt;/HEAD&gt;

&lt;BODY&gt;

&lt;/BODY&gt;

&lt;/HTML&gt;

- 실습

1. 위의 예를 실행시켜본다.
2. 위의 예를 근거하여 여러가지 테스트를 수행해본다.

#### 1.4.2 Control statements

- 제어문

  - if 문

```

* 구문
if (조건문) {
 Java script code1
} else{
 Java script code2
}
혹은
if(조건문1) {
 Java script code1
} else if(조건문2) {
 Java script code2
} else {
 Java script code3
}

```

\* 예) (수행하기)<sup>74</sup>

```
<HTML>
<HEAD>
<TITLE> if 문을 사용한 자바 스크립트 예제 </TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--
var question=" 음수와 양수, 0 중 임의의 숫자를 입력하세요. "
var num=""
var result=prompt(question, num)
if(result<0)
 document.write("<H2>입력된 수는 음수입니다.</H2>")
else if(result==0)
 document.write("<H2>입력된 수는 0입니다.</H2>")
else if(result<0)
 document.write("<H2>입력된 수는 양수입니다.</H2>")
else
 result=prompt(question, num)
//-->
</SCRIPT>
</HEAD>
<BODY>
</BODY>
</HTML>
```

#### - while 문

\* 구문

```
while(조건) {
 Java script code
}
```

\* 예) (수행하기)<sup>75</sup>

```
<HTML>
<HEAD>
<TITLE> while 문을 사용한 자바 스크립트 예제 </TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--
var a = 0, b = 0
while(a<100){
 b++
 b += a
 a++
}/* while */
document.write("<H2>" + "while 문을 이용한 1에서 100 까지의 합 : " + b + "</H2>")
//-->
</SCRIPT>
```

<sup>74</sup>javascript:open\_win('/testjs/script/onlineTest.cgi?file=if.js.htm')

<sup>75</sup>javascript:open\_win('/testjs/script/onlineTest.cgi?file=while.js.htm')

</HEAD>

<BODY>

</BODY>

</HTML>

#### - for 문

\* 구문

```
for(초기값; 조건문; 증감분) {
 Java script code
}
```

\* 예) (수행하기)<sup>76</sup>

```
<HTML>
<HEAD>
<TITLE> for 문을 사용한 자바 스크립트 예제 </TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--
for(a=0, b=0;a<100;a++){
 b++
 b+=a
}/* for */
document.write("<H2>" + " for 문을 이용한 1에서 100 까지의 합 : " + b + "</H2>")
//-->
</SCRIPT>
</HEAD>
<BODY>
</BODY>
</HTML>
```

#### - break 문

\* 구문

```
while(조건) {
 Java script code1
 break
 Java script code2
}
```

\* 예) (수행하기)<sup>77</sup>

```
<HTML>
<HEAD>
<TITLE> while 문과 for 문 내에서 break 문을 사용한 자바 스크립트 예제 </TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--
var a=0, b=0
while(a<100){
 b++
}
```

<sup>76</sup>javascript:open\_win('/testjs/script/onlineTest.cgi?file=for.js.htm')

<sup>77</sup>javascript:open\_win('/testjs/script/onlineTest.cgi?file=break.js.htm')

```

b+=a
if(a==15)
 break
a++
}/* while */

document.write("<H2>" + "while 문 내에서 break 문을 사용 안 결과: "+ b)

for(a=0, b=0;a<100;a++){
 b++
 if(a==20)
 break
 b+=a
}/* for */

document.write("<H2><P> for 문을 이용 안 break 문의 결과: " + b + "</H2>")

//-->
</SCRIPT>

</HEAD>
<BODY>
</BODY>
</HTML>

- continue 문
* 구문
while(조건) {
 Java script code1
 continue
 Java script code2
}
* 예) (수행하기)78
<HTML>

<HEAD>
<TITLE> While 문과 for 문 내에서 continue 문을 사용 안 자바 스크립트 예제 </TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--

var a=0, b=0

while(a<100){
 b++
 b+=a
 if(b==3)
 continue
 a++
}/* while */

document.write("<H2>while 문 내에서 continue 문을 사용 안 결과:" + b)

for(a=0, b=0;a<100;a++){
 b++
 if(a==3)
 continue
 b+=a
}/* for */

document.write("<H2><P> for 문 내에서 continue 문을 사용 안 결과:" + b)

```

```

//-->
</SCRIPT>

</HEAD>
<BODY>
</BODY>
</HTML>

- return 문
* 함수의 결과를 return 할 때 사용
* 예) (수행하기)79
<HTML>

<HEAD>
<TITLE> return 문을 사용 안 자바 스크립트 예제 </TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--

function sum(){
 for(a=0, b=0;a<100;a++){
 b++
 b+=a
 }/* for */
 return b
}/* sum */

//-->
</SCRIPT>

</HEAD>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
<!--

document.write("<H1> 1부터 100까지의 합 : "+ sum() +"</H1><P>")

//-->
</SCRIPT>

</BODY>
</HTML>

```

### ● 실습

1. if, while, for, break, continue, return 문의 예를 수행해본다.
2. 위의 예를 근거하여 여러가지 테스트를 수행해본다.

### 1.4.3 Java script function

- Java script 의 사용자 정의 함수
  - 사용자정의 함수 만들기

<sup>78</sup>javascript:open\_win('/testjscript/onlineTest.cgi?file=continue.js.htm')

<sup>79</sup>javascript:open\_win('/testjscript/onlineTest.cgi?file{return.js.htm'})

```
* <head> 태그사이에 삽입
* 선언
 function 함수 이름(매개변수1, 매개변수2, ...) {
 변수
 실행문
 }
```

## - 사용자정의 함수 호출하기

- \* 함수 이름(매개변수1, 매개변수2, ...)

- 예) (수행하기)<sup>80</sup>

```
<HTML>
<HEAD>
 <TITLE> 매개 변수가 있는 사용자 정의 함수를 사용한 자바 스크립트 예제 </TITLE>
 <SCRIPT LANGUAGE="JavaScript">
 <!--
 function funcpa(a, b){
 var c
 c = a * b
 return c
 }/* funcpa() */
 //-->
 </SCRIPT>
</HEAD>
<BODY>
 <H2> 매개 변수가 있는 사용자 정의 함수를 사용한 자바 스크립트 예제입니다. </H2>

 <SCRIPT LANGUAGE="JavaScript">
 <!--
 document.write("매개 변수가 있는 사용자 정의 함수 funcpa(a, b)에 a=3, b=4를 전달한 뒤 반환값은 = " + funcpa(3,
 //-->
 </SCRIPT>
</BODY>
</HTML>
```

## ● Java script 의 내장함수

## - eval() 내장함수

- \* 문자열로된 수식을 입력받아 숫자로 변환하여 결과값 리턴

- \* 사용법: eval("문자열")

- \* 예) (수행하기)<sup>81</sup>

```
<HTML>
```

```
<HEAD>
```

```
<TITLE> eval() 내장 함수를 사용한 자바 스크립트 예제 </TITLE>
```

<sup>80</sup>javascript:open\_win('/testjs/test/onlineTest.cgi?file=funcpa.js.htm')

<sup>81</sup>javascript:open\_win('/testjs/test/onlineTest.cgi?file=eval.js.htm')

```
</HEAD>
<BODY>
 <H2> eval() 내장 함수를 사용한 자바 스크립트 예제입니다. </H2>

 <SCRIPT LANGUAGE="JavaScript">
 <!--
 var string
 string = "3 + 4"
 document.write("<H3>" + "문자열 = " + string + "<H3>")
 document.write("<H3>" + "eval() 내장 함수의 사용 결과 -> " + eval(string))
 //-->
 </SCRIPT>
</BODY>
</HTML>
```

## - parseInt() 내장함수

- \* 숫자로구성된 문자열을 입력받아 10진수 정수형으로 리턴
- \* 두번째 매개변수는 문자열의 진수표시
- \* 사용법: parseInt("문자열", 진수)
- \* 예) (수행하기)<sup>82</sup>

```
<HTML>
```

```
<TITLE> parseInt() 내장 함수를 사용한 자바 스크립트 예제 </TITLE>
```

```
<HEAD>
```

```
<BODY>
```

```
<H2> parseInt() 내장 함수를 사용한 자바 스크립트 예제입니다. </H2>

```

```
<SCRIPT LANGUAGE="JavaScript">
<!--
 var string1, string2, string3, string4, string5
 string1 = "34"
 string2 = "34.23"
 string3 = "34e34"
 string4 = "100"
 string5 = parseInt(string1) + parseInt(string2)

 document.write("<H4>문자열1 = " + string1)
 document.write("
parseInt() 내장 함수의 사용 결과 -> " + parseInt(string1) +
 document.write("<H4>문자열2 = " + string2)
 document.write("
parseInt() 내장 함수의 사용 결과 -> " + parseInt(string2) +
 document.write("<H4>문자열3 = " + string3)
 document.write("
parseInt() 내장 함수의 사용 결과 -> " + parseInt(string3) +
 document.write("<H4>문자열4 = " + string4)
 document.write("
parseInt(문자열4, 2) 내장 함수의 사용 결과 -> " + parseInt(string4) +
 document.write("<H4>문자열5 = " + string5)
 document.write("
parseInt(문자열5) 내장 함수의 사용 결과 -> " + parseInt(string5) +
 //-->
```

<sup>82</sup>javascript:open\_win('/testjs/test/onlineTest.cgi?file=parseInt.js.htm')

```

</SCRIPT>
</BODY>
</HTML>

- parseFloat() 내장함수
 * 부동수수집이나 지수형 숫자로구성된 문자열을 입력받아 리턴
 * 예) (수행하기)83
<HTML>
<HEAD>
 <TITLE> parseFloat() 내장 함수를 사용한 자바 스크립트 예제 </TITLE>
</HEAD>
<BODY>
 <H2> parseFloat() 내장 함수를 사용한 자바 스크립트 예제입니다. </H2>

<SCRIPT LANGUAGE="JavaScript">
<!--
 var string1, string2, string3, string4
 string1 = "34"
 string2 = "-34"
 string3 = "34.23"
 string4 = "5e4"
 document.write("<H3>" + "문자열1 = " + string1)
 document.write("<H3>" + "parseFloat() 함수의 사용 결과 -> " + parseFloat(string1) + "</H3>
")
 document.write("<H3>" + "문자열2 = " + string2)
 document.write("<H3>" + "parseFloat() 함수의 사용 결과 -> " + parseFloat(string2) + "</H3>
")
 document.write("<H3>" + "문자열3 = " + string3)
 document.write("<H3>" + "parseFloat() 함수의 사용 결과 -> " + parseFloat(string3) + "</H3>
")
 document.write("<H3>" + "문자열4 = " + string4)
 document.write("<H3>" + "parseFloat() 함수의 사용 결과 -> " + parseFloat(string4) + "</H3>")
//-->
</SCRIPT>
</BODY>
</HTML>

- escape(), unescape() 내장함수
 * escape(): ISO LATIN-1 문자를 %16진수의 ASCII 코드값으로 변환
 * unescape(): %16진수의 ASCII 코드값을 ISO LATIN-1 문자로 변환
 * 예) (수행하기)84
<HTML>
<HEAD>
 <TITLE> escape(), unescape() 내장 함수를 사용한 자바 스크립트 예제
 </TITLE>
</HEAD>

```

<sup>83</sup> javascript:open\_win('/testjs/test/onlineTest.cgi?file=parseFloat.js.htm')

<sup>84</sup> javascript:open\_win('/testjs/test/onlineTest.cgi?file=escape.js.htm')

```

<BODY>
<H2> escape(), unescape() 내장 함수를 사용한 자바 스크립트 예제입니다.
</H2>

<SCRIPT LANGUAGE="JavaScript">
<!--
 var string1, string2, string3, string4
 str1 = "%"
 str2 = "&"
 str3 = "%25"
 str4 = "%26"
 document.write("<H3>" + "문자1 = " + str1)
 document.write("<H3>" + "escape() 함수의 사용 결과 -> " + escape(str1) + "</H3>
")
 document.write("<H3>" + "문자2 = " + str2)
 document.write("<H3>" + "escape() 함수의 사용 결과 -> " + escape(str2) + "</H3>
")
 document.write("<H3>" + "문자3 = " + str3)
 document.write("<H3>" + "escape() 함수의 사용 결과 -> " + unescape(str3) + "</H3>
")
 document.write("<H3>" + "문자4 = " + str4)
 document.write("<H3>" + "unescape() 함수의 사용 결과 -> " + unescape(str4) + "</H3>
")
//-->
</SCRIPT>
</BODY>
</HTML>
```

#### ● event 와 event handler

- event: 사용자가 행한 동작
- event handler: 각각의 event에 대하여 처리하는 프로그램
- event 종류: event 명은 대소문자의 구분이 없다

표 4.12: event 의 종류

event	발생하는 경우
abort	이미지를 읽다가 중지했을 때
blur	키보드 입력 양식에 커서가 제거될 때
click	hyper text, button, check box 등의 객체를 click 했을 때
change	키보드 입력 양식에 값을 입력한 후 바꿀 때
error	문서나 이미지를 읽다가 에러가 발생했을 때
focus	입력 양식 같은 객체에 입력 커서가 부여될 때
load	문서가 읽혀질 때
mouseout	마우스 pointer 가 hyper text 나 객체를 벗어났을 때
mouseover	마우스 pointer 가 hyper text 나 객체 위에 위치될 때
reset	입력 양식의 reset button 을 눌렀을 때
select	text box 안에 있는 문자나 문자열을 선택했을 때
submit	입력 양식을 서버로 보낼 때
unload	접속한 문서를 벗어날 때

## - event 와 event handler 사용법

- \* <script> 태그를 사용하지 않고도 HTML 속성으로 지정할수 있음
- \* HTML 속성으로 사용시 event 명 앞에 on 을 붙여야함
  - 예)
  - <body onLoad="alert('안녕 야 세요')" onUnload="confirm('안녕 이 가세요')"
  - alert()와 confirm()은 event handler

## - event 사용 예

- \* Error event 예) (수행하기)<sup>85</sup>

```
<HTML>
<HEAD>
<TITLE> Error 이벤트를 사용한 자바 스크립트 예제 </TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--

function access(myImage){
 alert("Error : " + myImage.src + " 파일을 불러올 수 없습니다.")
}/* access() */

// --
</SCRIPT>
</HEAD>
<BODY>

<FORM>

</FORM>
</BODY>
</HTML>
```

- \* Focus와 Blur event 예) (수행하기)<sup>86</sup>

```
<HTML>
<HEAD>
<TITLE> Focus 이벤트와 Blur 이벤트를 사용한 자바 스크립트 예제 </TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--

function show(text){
 window.status = text
}/* show() */

// --
</SCRIPT>
</HEAD>
<BODY>

<FORM>
<CENTER>

 <INPUT TYPE = "text" SIZE=50 onFocus="show('텍스트 입력 양식에 입력 커서가 있습니다.')" onBlur="
```

<sup>85</sup> javascript:open\_win('/testjs/testonlineTest.cgi?file=error.js.htm')

<sup>86</sup> javascript:open\_win('/testjs/testonlineTest.cgi?file=focus.js.htm')

```
</CENTER>
</FORM>
```

```
</BODY>
</HTML>
* click event 예) (수행하기)87
```

```
<HTML>
<HEAD>
<TITLE> Click 이벤트를 사용한 자바 스크립트 예제 </TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--

function show(){
 alert("Click 이벤트가 발생했습니다")
}/* show() */

// --
</SCRIPT>
</HEAD>
<BODY>

<FORM>
<CENTER>

 <INPUT TYPE="button" VALUE="▶" onClick="show()">
</CENTER>
</FORM>
</BODY>
</HTML>
```

- \* MouseOver 와 MouseOut event 예) (수행하기)<sup>88</sup>

```
<HTML>
<HEAD>
<TITLE> MouseOver 와 MouseOut 이벤트를 사용한 자바 스크립트 예제 </TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--

function over(){
 alert("이미지 텍스트 상에 마우스 포인ター가 위치해 있습니다.")
}/* over() */

function out(){
 alert("이미지 텍스트 상에 마우스 포인ター가 위치해 있지 않습니다.")
}/* out() */

// --
</SCRIPT>
</HEAD>
<BODY>

<CENTER>

<A HREF="http://ice.hansung.ac.kr/~shjung" onMouseOver="over()" onMouseOut="out()"
```

<sup>87</sup> javascript:open\_win('/testjs/testonlineTest.cgi?file=click.js.htm')

<sup>88</sup> javascript:open\_win('/testjs/testonlineTest.cgi?file=mouse.js.htm')

```
</BODY>
</HTML>
```

- 실습

1. 위의 예들을 실행시켜본다.
2. 자신의 함수를 만들어 테스트해본다.

- 보다 구체적인 내용

- 사용자정의 객체, Java script에서 제공하는 객체, Web browser에서 제공하는 객체의 사용법 등
- 구체적인 내용을 알고싶은 학생은 여기를<sup>89</sup> 참조할것

## 1.5 Java 사용하기

### 1.5.1 Java overview

- 자바의 정의

- 컴파일거쳐 생성되는 파일이 실행파일이 아니라 플랫폼에 독립적인 byte code임
- 자바 Interpreter 만 있으면 플랫폼에 독립적으로 실행가능
- 종류
  - \* Java Applet: web browser에서 실행됨
  - \* Java Application: web browser 없이 독립적으로 수행됨

- 자바의 특징

- 단순한 코드
  - \* 간결한 구조를 가짐
  - \* C++과 구문이 유사하나 예러를 유발하기 쉬운부분을 제외
  - \* 구조체,공용체, 포인터, 연산자의 다중정의, 클래스의 다중상속등을 지원하지 않음
- 객체지향 (Object Orient) 언어
  - \* 객체지향언어로서 프로그램 재사용을 높임
- 다양한 플랫폼에서 구동
  - \* 특정한 플랫폼의 기계어를 생성하지 않음으로 Java Interpreter 가 설치된 모든 플랫폼에서 실행가능
- Multi Thread 기능 제공
  - \* 하나의 작업을 수행하면서 다른 작업도 수행할수 있게끔 Multi Thread 기능 지원

- Garbage Collection 기능 제공

- 신뢰성있는 프로그램 개발 가능
  - \* 컴파일시 자료검사를 엄격히 함으로 예기치않은 예러 방지
  - \* 포인터가 없어서 바이러스 침입을 방지
  - \* 하드웨어 접근을 허용하지 않아 프로그램 버그를 방지

- Java Applet 과 Application

- Java Applet 과 Application 의 차이점

표 4.13: Java Applet 과 Application 의 차이점

Applet	Application
웹 검색기 상에 포함되어 실행 웹 검색기에서 호출시에만 event에 의해 동작	웹 검색기 없이 독립적으로 실행 main() 함수의 순서에 의하여 순차적

### 1.5.2 JDK (Java Development Kit)

- JDK 란?

- Java applet 과 application 프로그램을 개발하는데 필요한 도구를 모아놓은 package
- byte code 를 만들기 위한 컴파일러와 실행하기위한 interpreter 가 있음
- java source code 의 확장자는 java 생성된 byte code 의 확장자는 class임

- JDK 의 구성

- Java compiler
  - \* Java source code (.java) 를 byte code (.class) 로 만들
  - \* 사용법: javac java\_source\_file (확장자 반드시 포함)

- Java Interpreter

- \* byte code 를 실행시킴
- \* 사용법: java byte\_code (확장자는 생략)

- Appletviewer

- \* Java applet 를 web browser 를 사용하지 않고도 실행시켜주는 프로그램
- \* 사용법: appletviewer java\_applet\_을\_호출하는\_HTML\_문서

- Java disassembler

- \* 컴파일된 byte code 를 이용하여 변수, method 등에 대한 정보를 알려줌
- \* 사용법: javap bytecode1 bytecode2 ...

- 보다 구체적인 내용

<sup>89</sup><http://itsys.hansung.ac.kr/lec/webprogram/mylec/book>

- 본 과목에서는 학생들이 객체지향언어에 익숙하지 않음으로 구체적으로 다루지는 않음
- 구체적인 것은 web programming 과목에서 다룰 것임
- Java programming 에 대해 개인적으로 구체적으로 알고 싶은 학생은 여기를<sup>90</sup> 참조할 것

## 찾아보기

., 22  
 .., 22  
 ~, 22  
 biff, 65  
 cal, 14  
 cat, 16  
 cd, 14, 22  
 CGI, 97  
     C\_language, 104  
         ex1, 104  
     env\_var, 105  
     GET, 106  
     POST, 106  
     tag, 98  
         file\_upload, 99  
         form, 98  
         input, 98  
         option, 102  
         select, 102  
         textarea, 103  
 chmod, 35  
 chown, 36  
 chsh, 54  
 clear, 16  
 compress, 58  
 cp, 28  
 csh, 53  
 date, 14  
 df, 57  
 directory, 21  
 du, 57  
 eof, 19  
 exit, 51  
 fg, 52  
 file, 20, 57  
 find, 44  
 finger, 14  
 grep, 46  
 gunzip, 58  
 gzip, 58  
 head, 28  
 home directory, 22  
 homepage, 71  
 hostname, 16  
 html, 71  
     animation, 90  
     frame, 93  
     multimedia, 92  
     sound, 91  
     special character, 76  
     tag, 71  
         a, 84  
         address, 77  
         area, 89  
         b, 79  
         big, 79  
         blink, 79  
         blockquote, 77  
         body, 91  
         br, 73  
         center, 73  
         cite, 79  
         code, 79  
         comment, 73  
         dd, 82  
         dfn, 79  
         div, 74  
         em, 79  
         embed, 91  
         font, 75  
         frame, 93  
         frameset, 93

---

<sup>90</sup><http://itsys.hansung.ac.kr/lec/webprogram/mylec/book>

hn, 72  
hr, 74  
i, 79  
img, 88  
kbd, 79  
li, 80  
map, 89  
noframe, 94  
ol, 80  
p, 72  
pre, 78  
samp, 79  
small, 79  
strike, 79  
strong, 78  
sub, 79  
sup, 79  
table, 82  
target, 94  
td, 83  
th, 83  
tr, 83  
tt, 79  
u, 79  
ul, 81  
var, 79  
xmp, 78

Internet, 61  
address, 63  
anonymous FTP, 67  
archie, 68  
browser, 69, 70  
CGI, 69  
DNS, 64  
domain, 63  
finger, 66  
HTML, 69  
http, 69  
hyperlink, 69  
hypertext, 69  
IP address, 63  
Java, 70  
JavaScript, 70  
mail, 64  
talk, 68

telnet, 66  
URL, 69  
WWW, 68  
Java, 125  
    Applet, 126  
    Application, 126  
    JDK, 126  
JavaScript, 107  
    assign\_op, 112  
    bit\_op, 113  
    control, 114  
        break, 116  
        continue, 117  
        for, 116  
        if, 114  
        return, 118  
        while, 115  
    event, 122  
        click, 124  
        error, 123  
        focus, 123  
        mouse, 124  
    event\_handler, 122  
function, 118  
    escape, 121  
    eval, 119  
    parseFloat, 121  
        parseInt, 120  
logic\_op, 111  
math\_op, 110  
relative\_op, 111  
tag, 107  
    comment, 107  
    script, 107  
    variable, 108  
jobs, 52  
kill, 52  
LAN, 62  
less, 31  
ln, 57  
login, 13  
logout, 51  
lpq, 44  
lpr, 44

lprm, 44  
ls, 16, 25  
man, 17  
mkdir, 23  
modem, 62  
more, 16, 30  
mv, 24, 31  
OS, 11  
    Linux, 12  
    SunOS, 12  
    Unix, 11  
passwd, 13  
path, 22  
permission, 34  
pg, 30  
ping, 16  
PPP, 62  
pwd, 14, 22  
quota, 57  
rcp, 51  
regular expression, 39  
rm, 32  
rmdir, 24  
rsh, 51  
setenv, 51  
shell, 14, 53  
SLIP, 62  
sort, 46  
stty, 19  
tail, 30  
TCP/IP, 63  
telnet, 51, 52  
umask, 36  
uncompress, 58  
uptime, 16  
vi, 36  
    co, 41  
    cursor move, 38  
    delete, 41  
    input, 39