

파이썬 개요

인공지능 플랫폼 설계 - 2주차

학습 내용

1. 변수와 연산자
2. 리눅스 개요
3. 리눅스 기본 명령



Python 프로그래밍

_VS Code 에디터 주요 단축키

- 파일 저장 : Ctrl + S
- 실행 : Ctrl + F5
- 소스 정렬 : Shift + Alt + F
- 셋팅 창 : Ctrl + ,
- 설정 검색창 : Ctrl + Shift + P
 - Terminal : Select Default Profile → Command Prompt



_Python 기본

■ print() 함수

- 형식

```
print(자료값1, 자료값2, ... , sep="문자열1", end="문자열2")
```

- 예

```
print("이름:", name, "평균:", avg, sep=" || ", end="\n\n")
```

→ sep : 출력 자료값들을 구분하는 문자열. 기본값은 공백문자임

→ end : 마지막 출력값 출력후 출력하는 문자열, 기본값은 \n 임

```
01 name = "이강인" ; avg = 88.3
02 print("이름:", name, "평균:", avg, sep= " || ", end= "\n\n")
03 name = "손흥민" ; avg = 92.6
04 print("이름:", name, "평균:", avg, sep= " ***** ", end= "\t")
05 name = "차범근" ; avg = 90.3
06 print("이름:", name, "평균:", avg)
07 print("출력을 종료합니다!")
```

실행 결과

이름: || 이강인 || 평균: || 88.3

이름: ***** 손흥민 ***** 평균: ***** 92.6 이름: 차범근 평균: 90.3
출력을 종료합니다!



Python 실습

```
name = "이강인" ; avg = 88.3
print("이름:", name, "평균:", avg, sep= " || ", end= "\n\n")
name = "손흥민" ; avg = 92.6
print("이름:", name, "평균:", avg, sep= " ***** ", end= "\t")
name = "차범근" ; avg = 90.3
print("이름:", name, "평균:", avg)
print("출력을 종료합니다!")
```



_Python 기본

■ input() 함수

- 키보드(표준 입력장치)를 통해 사용자로부터 문자열을 입력받아 반환함

```
변수 = input()
변수 = input("입력 메시지")
```

- input() 함수의 반환값은 항상 문자열임

→ 수치값을 입력하면 바로 수치계산에 사용할 수 없음

+: 문자열 연결

01	num1 = input()
02	num2 = input()
03	num3 = num1 + num2
	print(num3)

실행 결과

45
356
45356

+: 덧셈

01	num1 = int(input())
02	num2 = int(input())
03	num3 = num1 + num2
	print(num3)

실행 결과

45
356
401



Python 실습

```
print("2개의 숫자를 입력하세요.")  
num1 = input()  
num2 = input()  
num3 = num1 + num2  
print(num3)
```

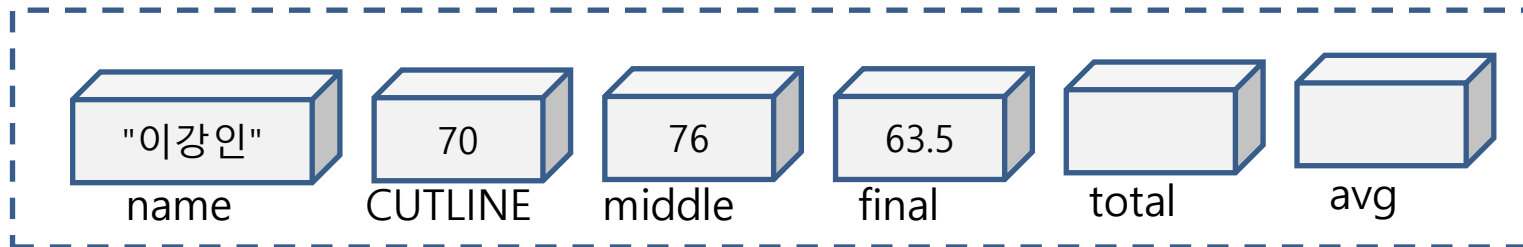
```
print("2개의 숫자를 입력하세요.")  
num1 = int(input())  
num2 = int(input())  
num3 = num1 + num2  
print(num3)
```



_Python 기본

■ 변수

- 프로그래밍에서의 의미
 - 프로그램이 실행되는 동안 사용하기 위해 확보한 메모리의 특정 저장공간
 - 변수라는 명칭은 저장공간의 저장값이 변경될 수 있다는 것을 의미함



_Python 기본

■ 자료형(Data Type)의 개념

- 자료형의 종류
 - 정수, 실수, 문자열 등
- 자료형에 따라 논리적 표현방법과 컴퓨터 내부의 저장방법이 다름
 - 수치형 : 정수(int), 실수(float)
 - 비수치형 : 문자열(str), 부울(boolean)
 - None : '자료값 없음'을 나타냄
 - 복합데이터 (컬렉션형) : 리스트(list), 튜플(tuple), 집합(set), 딕셔너리(dictionary)



_Python 기본

■ 문자열

- 문자들을 순차적으로 연속해서 저장함
- 문자의 인덱스 부여 순서 → _____
 - ✓ 맨 앞쪽부터 0, 1, 2, 3 순서로 인덱스 부여됨

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]
'P'	'y'	't'	'h'	'o'	'n'	' '	'P'	'r'	'o'	'g'	'r'	'a'	'm'	'!'
[-15]	[-14]	[-13]	[-12]	[-11]	[-10]	[-9]	[-8]	[-7]	[-6]	[-5]	[-4]	[-3]	[-2]	[-1]



_Python 실습

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]
'P'	'y'	't'	'h'	'o'	'n'	' '	'P'	'r'	'o'	'g'	'r'	'a'	'm'	'!'
[-15]	[-14]	[-13]	[-12]	[-11]	[-10]	[-9]	[-8]	[-7]	[-6]	[-5]	[-4]	[-3]	[-2]	[-1]

```
str1 = "Python Program!"  
print(str1)  
print(str1[0], str1[7], str1[14])  
print(str1[-1], str1[-8], str1[-15])
```

```
# 인덱스 범위 지정 부분문자열 참조  
print(str1[0:6], str1[10:14])  
print(str1[7:])  
print(str1[:6])
```



_Python 기본

■ 산술 연산자

연산자	의미	사용 예	설명
=	대입 연산자	<code>a = 3</code>	정수 3을 a에 대입
+	더하기	<code>a = 5 + 3</code>	5와 3을 더한 값을 a에 대입
-	빼기	<code>a = 5 - 3</code>	5에서 3을 뺀 값을 a에 대입
*	곱하기	<code>a = 5 * 3</code>	5와 3을 곱한 값을 a에 대입
/	나누기	<code>a = 5 / 3</code>	5를 3으로 나눈 값을 a에 대입
//	나누기(몫)	<code>a = 5 // 3</code>	5를 3으로 나눈 후 소수점을 버리고 값을 a에 대입
%	나머지 값	<code>a = 5 % 3</code>	5를 3으로 나눈 후 나머지 값을 a에 대입
**	제곱	<code>a = 5 ** 3</code>	5의 3제곱을 a에 대입

_Python 기본

■ 관계식과 논리식

관계 연산자와 논리 연산자


구분	연산자	수학연산자	의미
관계연산	>, <	>, <	크다, 작다
	>=, <=	≥, ≤	크거나 같다, 작거나 같다
	==, !=	=, ≠	같다, 같지 않다
논리연산	and, or	∧	논리곱
	or	∨	논리합
	not	~	부정

파이썬 논리연산표

x	y	x or y	x and y	not x
True	True	True	True	False
True	False	True	False	False
False	True	True	False	True
False	False	False	False	True

_Python 기본

■ 연산자 우선순위

구분	연산자	우선순위
괄호	(,)	 높다
지수연산	**	
단항연산	+, -	
산술연산	*, /, //, %	
	+, -	
관계연산	<, <=, >, >=	
	==, !=	
대입연산	=, %=, /=, //=, -=, +=, *=, **=	
논리연산	not	
	and	
	or	낮다

Python 실습

```
a = 10 ; b = 20 ; print(a, b)
```

```
c = a + b
```

```
print("a + b ", c)
```

```
c = b // 3
```

```
print("b // 3 ", c)
```

```
c = b % 3
```

```
print("b % 3 ", c)
```

```
d = a > b
```

```
print("a > b ", d)
```

```
a += b
```

```
print("a += b ", a)
```



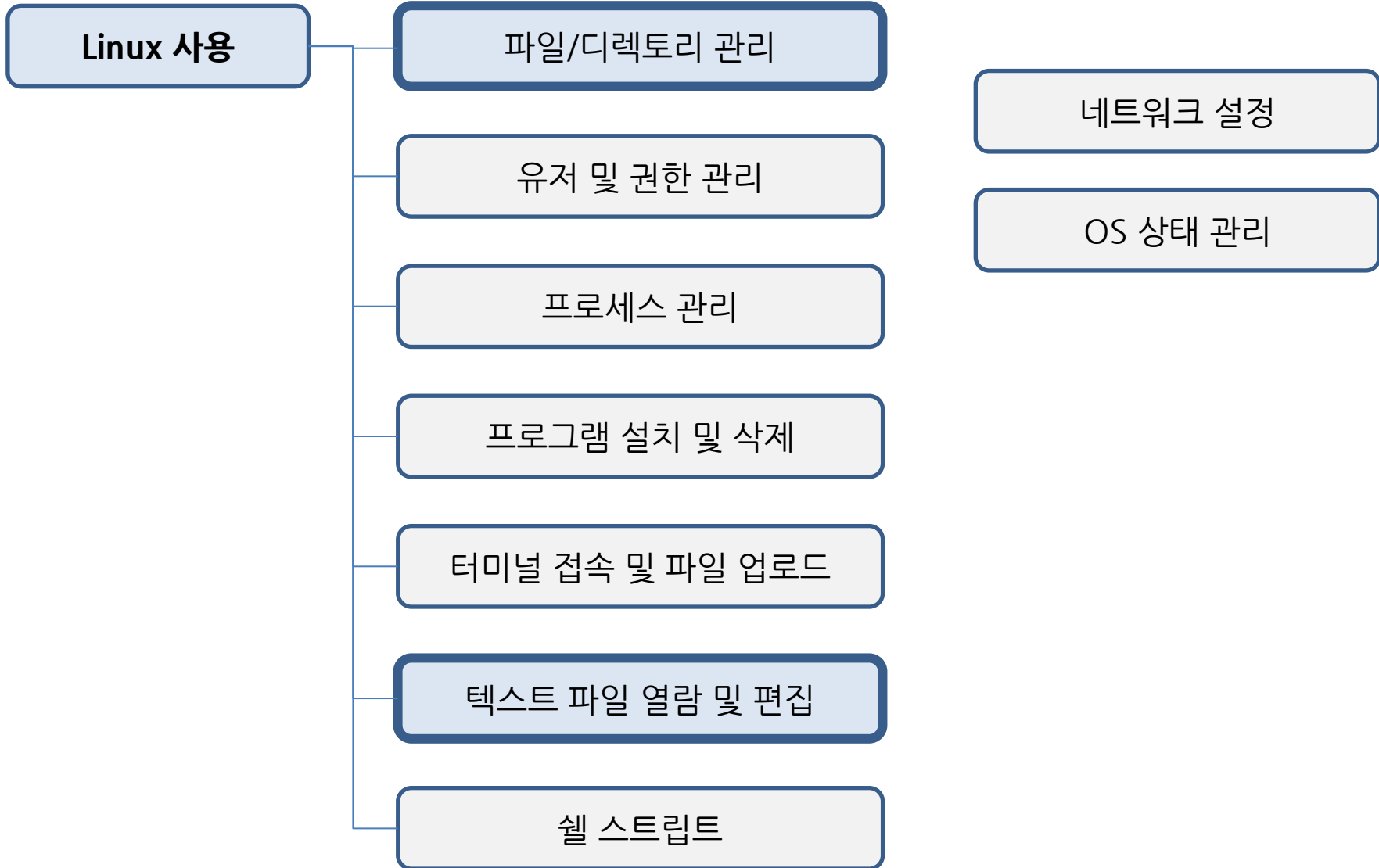
_Python 실습

실습문제 1 : 1개의 정수를 입력받고 그 정수가 짝수이면 '짝수이다 - True'
아니면 '짝수이다 - False'로 출력하는 프로그램
(나머지 연산자 %를 이용하실 것)



IT 플랫폼 실습

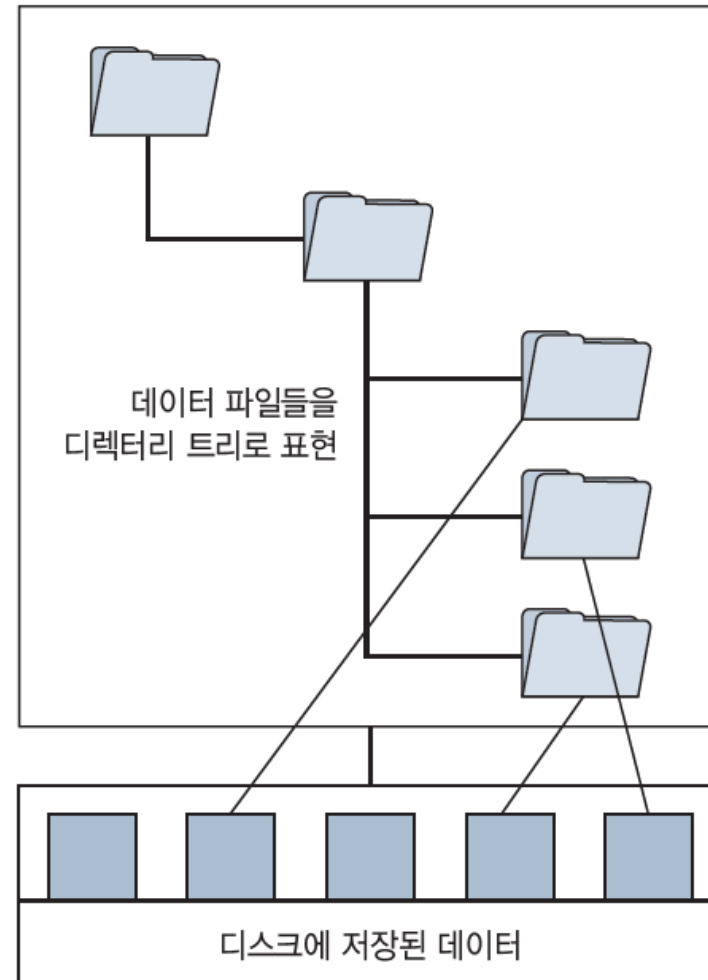
_리눅스 파일 시스템



_리눅스 파일 시스템

파일 시스템

- 물리적 디스크의 식별할 수 있는 고유 위치에 저장된 데이터와 논리적 개념 단위인 파일을 연결하는 일종의 데이터 테이블이나 인덱스



_리눅스 파일 시스템

ls(리스트)

- 현재 디렉터리에 있는 파일과 그 하위 디렉터리들의 이름이 나열
- `$ ls -al` : 파일명뿐만 아니라 접근 권한, 소유자, 그룹, 파일 크기, 최종 수정 날짜도 출력
- `$ ls -al /var` : /var과 같이 특정 디렉터를 지정하면 해당 디렉터리의 내용이 출력

drwxr-xr-x.		20	root	root	4096	9월 26 00:12	var
파일 타입	파일 모드	링크 수	소유자	소유 그룹	크기	타임스탬프	파일 이름 또는 디렉터리 이름
- 보통 파일 d 디렉터리 l 심볼릭 링크							

파일 복사와 이동

- `$ cp` : 파일이나 디렉터리 사본 생성
- `$ mv` : 파일을 한 장소에서 다른 장소로 이동

파일 삭제

- `$ rm` : 파일 삭제. 원래대로 되돌릴 수 없음
- `$ rm -r *` : 이 명령을 지정한 경로 아래에 있는 모든 하위 디렉터리에 반복 적용해 내용 삭제



_리눅스 파일 시스템

pwd(현재 작업 디렉터리)

- 현재 작업 디렉터리를 화면에 출력
\$ pwd

cd(디렉터리 변경)

- \$ cd .. : 상위 디렉터리로 이동
- \$ cd /home/<사용자 계정> : 현재 디렉터리에서 아주 멀리 떨어져 있는 곳으로 가려면 절대 경로를 사용하는 편이 좋다

디렉터리의 생성과 삭제

- \$ mkdir : 디렉터리 생성
- \$ rmdir : 디렉터리 삭제
- \$ rm -r : 디렉터리 안에 파일이 있어도 삭제



_리눅스 파일 시스템

cat(파일 내용 출력)

- 연결한다는 뜻의 concatenate에서 유래, 파일 내용을 터미널에 출력(편집할 수는 없음)
- `$ cat /etc/fstab` : etc 디렉터리에 있는 fstab 파일 출력(이처럼 짧은 파일을 볼 때 좋음)

tail(파일 내용 표시)

- 파일 내용을 뒤에서 부터 텍스트 내용을 출력
- `$ tail -n 100 /etc/services`

텍스트 편집기

- nano 편집기
 - ctrl + o : 저장
 - ctrl+x : 종료
- Vim 편집기



_리눅스 파일 시스템

- 로그인
 - 콘솔 기반에서 로그인할 때 패스워드를 입력하면 화면에서는 다음과 같이 아무것도 표시되지 않음

● 콘솔 기반에서 로그인하기

```
localhost login: ldk  
Password:
```

- 이는 모니터 화면을 통해 암호 길이 같은 정보가 유출되는 것을 방지하기 위해서임



_리눅스 파일 시스템

- 로그아웃

- CLI로 로그인했다면 다음과 같이 `exit` 명령어를 사용하여 로그아웃
- `exit`를 입력

- 로그아웃

```
$ exit
```

- `exit` 명령어는 셸을 종료하는 명령어지만, 셸이 종료되면 자동적으로 로그아웃
- `logout` 명령어는 현재 셸이 로그인 셸이어야만 동작

- `logout` 명령어로는 셸을 종료할 수 없다

```
$ logout
```

```
bash: logout: 로그인 셸이 아님: 'exit'를 사용하세요
```



_리눅스 파일 시스템

- 셧다운
 - 슈퍼 사용자가 되면 프롬프트의 기호가 \$에서 #로 바뀜
 - -h 옵션은 전원을 끄는 것을 의미

```
sudo shutdown -h now
```

- 재부팅
 - -r 옵션을 주면 재부팅(reboot)

```
sudo shutdown -r now
```

