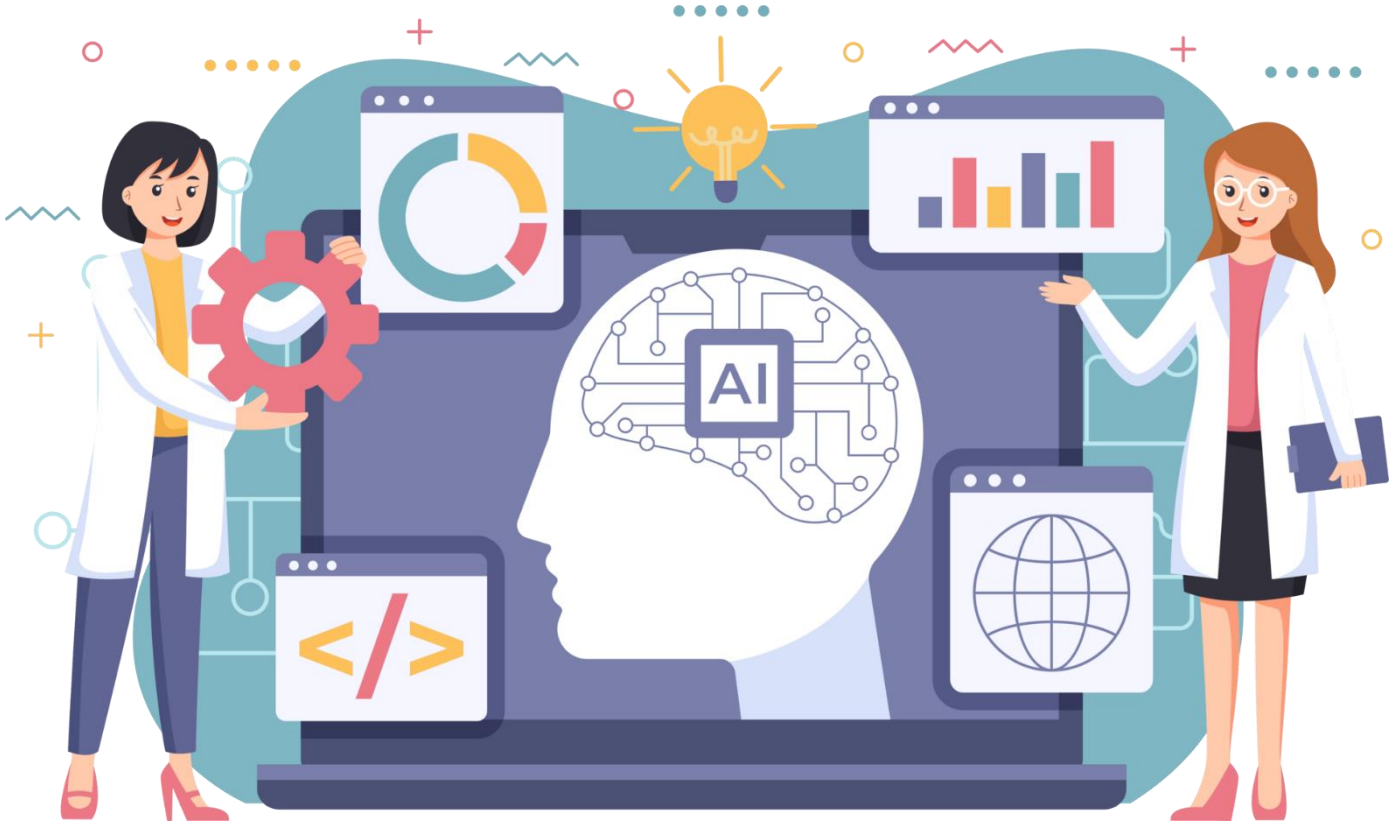


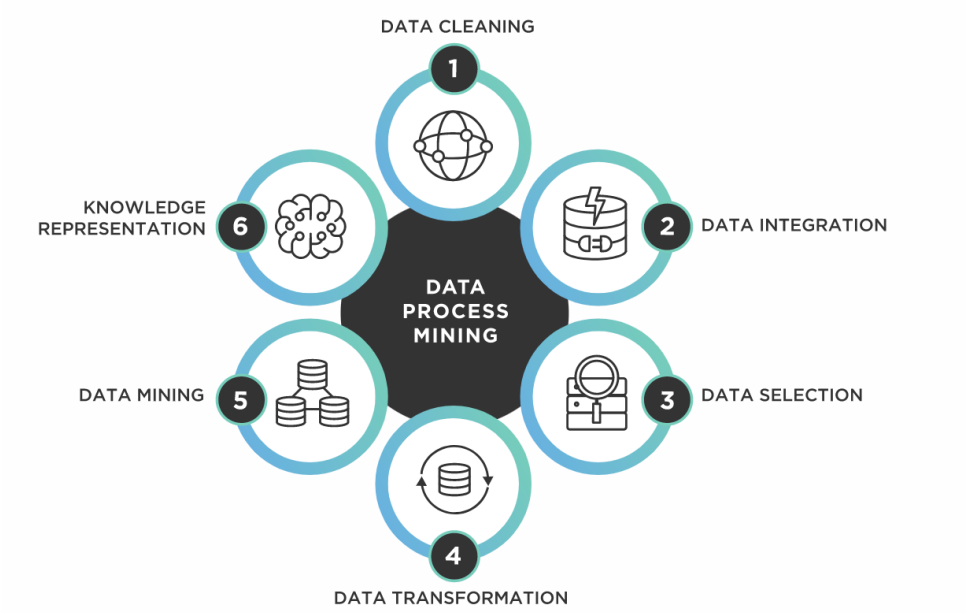
데이터 다루기 (데이터 프레임)

학습 내용

1. 데이터를 다루는 과정
2. 판다스 데이터 프레임
3. 파이썬 – 기본 데이터 형
4. 파이썬 – 기본 연산자

인공지능 이론 및 실습

_데이터의 처리 과정



1. 데이터 정제 (Data Cleaning) : 불필요하거나 일치하지 않는 데이터를 제거
2. 데이터 통합 (Data Integration) : 다수의 데이터 소스들을 결합
3. 데이터 선택 (Data Selection) : 필요한 데이터들을 데이터 저장소로부터 검색
4. 데이터 변환 (Data Transformation) : 데이터 마이닝/모델링을 하기에 적합한 형태로 데이터 가공
5. 데이터 마이닝 / 모델링 (Data Mining, Modeling) : 지능적 방법들을 적용하여 지식(데이터 패턴, 관계 등) 추출
6. 데이터 검증 (Data Evaluation) : 찾아낸 지식(데이터 패턴, 관계 등)를 검증
7. 데이터 시각화 (Data Presentation) : 발견한 지식을 사용자에게 효과적으로 보여주기 위해 시각화

_Pandas 데이터 프레임

데이터프레임(Dataframe) : 판다스의 기본 구조인 자료구조 객체

- 2차원 배열과 비슷한데 엑셀 데이터 형식에 가까움
- 데이터프레임은 행 인덱스, 열 이름(또는 열 인덱스), 값으로 구성.
  . 행 인덱스: 가로줄인 행(Row)을 구분하는 고유한 인덱스.
  . 열 이름: 세로줄인 열(Column)을 열 이름으로 구분
- 데이터프레임 사용 준비
  . pip install pandas
  . import pandas as pd

2차원 배열

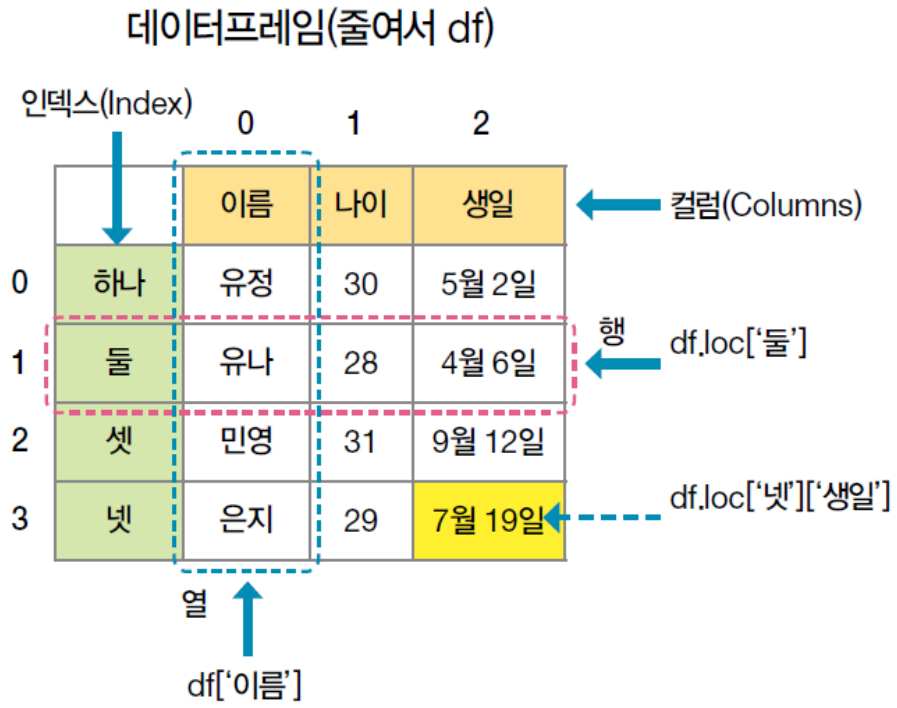
1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

데이터 프레임

	이름	나이	생일
1번	유정	30	5월 2일
2번	유나	28	4월 6일
3번	민영	31	9월 12일
4번	은지	29	7월 19일

_Pandas 데이터 프레임

- 데이터 프레임 생성 : df = pd.DataFrame(list1, columns=col_names)
- 행 : df.loc[인덱스이름]
- 열 : df[열이름]
- 셀 : df.loc[인덱스이름][열이름] df.loc[인덱스이름,열이름]



_Pandas 데이터 프레임

- 데이터 프레임 만들기

행 중심

```
col_names = ['과목번호', '과목명', '강의실', '시간수']
list1 = [['C1', '인공지능개론', 'R1', 3],
    ['C2', '웃음치료', 'R2', 2],
    ['C3', '경영학', 'R3', 3],
    ['C4', '3D디자인', 'R4', 4],
    ['C5', '스포츠경영', 'R2', 2],
    ['C6', '예술의 세계', 'R3', 1]
])

df = pd.DataFrame(list1, columns=col_names)
```

과목번호	과목명	강의실	시간수
C1	인공지능개론	R1	3
C2	웃음치료	R2	2
C3	경영학	R3	3
C4	3D디자인	R4	4
C5	스포츠경영	R2	2
C6	예술의 세계	R3	1

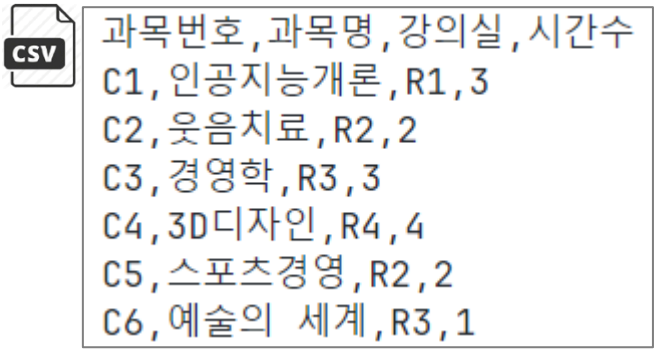
열 중심

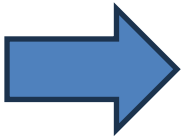
```
data = {
'과목번호' : ['C1', 'C2', 'C3', 'C4', 'C5', 'C6'],
'과목명' : ['인공지능개론', '웃음치료', '경영학', '3D디자인', '스포츠경영', '예술의 세계'],
'강의실' : ['R1', 'R2', 'R3', 'R4', 'R2', 'R3'],
'시간수' : [3, 2, 3, 4, 2, 1]
}

df = pd.DataFrame(data)
```

_Pandas 데이터 프레임

CSV 파일에서 불러 오기

```
df2 = pd.read_csv('file.csv', sep=',')
print(df2)
```





과목번호	과목명	강의실	시간수
C1	인공지능개론	R1	3
C2	웃음치료	R2	2
C3	경영학	R3	3
C4	3D디자인	R4	4
C5	스포츠경영	R2	2
C6	예술의 세계	R3	1

_Pandas 데이터 프레임

- **데이터 프레임 출력하기**

1. 데이터 프레임을 출력하시오
```
print(df)
```

2. 데이터 프레임을 앞에서 3개 행만 출력하시오
```
print(df.head(3))
```

3. '과목명'을 추출해서 출력하시오 (열 추출)
```
sr_name = df['과목명']
print(sr_name, end='₩n₩n')
```

4. 과목번호 'C3'이 있는 행을 출력하시오 (행 추출)
```
sr_no = df.loc[2]
print(sr_no, end='₩n₩n')
```

5. '경영학' 과목의 강의실만 출력하시오 (셀 추출)
```
cell_name = df.loc[2]['강의실']
print(cell_name)
```

6. 데이터프레임을 'lecture.csv'로 파일 저장하시오
```
df.to_csv("lecture.csv", header=True, index=False)
```

	과목번호	과목명	강의실	시간수
0	C1	인공지능개론	R1	3
1	C2	웃음치료	R2	2
2	C3	경영학	R3	3
3	C4	3D디자인	R4	4
4	C5	스포츠경영	R2	2
5	C6	예술의세계	R3	1

_데이터 프레임 : 추가/삭제

- **열 추가**

  - df[새 열이름] = [값1, 값2 …]

- **행 추가**

  - df.loc[new인덱스] = ([값1, 값2 …])

- **1열/행(시리즈) 삭제**

  - 열 삭제: df.drop([열이름], axis=1)
  - 행 삭제: df.drop([인덱스이름], axis=0)

  * axis의 1은 열, 0은 행을 의미

- **여러 열 또는 행을 삭제**

  - 열 삭제: df.drop([열이름1, 열이름2], axis=1)
  - 행 삭제: df.drop([인덱스1, 인덱스2])

_데이터 프레임 : 추가/삭제

```
df['담당교수'] = ['홍길동', '김철수', '이영희', '박영수',
                  '최영희', '김영수']

df.loc[6] = ['C7', '통계학', 'R7', 3, '이철수']
```

	과목번호	과목명	강의실	시간수	
0	C1	인공지능개론	R1	3	
1	C2	웃음치료	R2	2	
2	C3	경영학	R3	3	
3	C4	3D디자인	R4	4	
4	C5	스포츠경영	R2	2	
5	C6	예술의세계	R3	1	

```
df1 = df.drop(['강의실'], axis=1)
df2 = df.drop([5], axis=0)
```

_데이터 프레임 : 필터 (filtering)

- **지정 형식**

- 이름, 리스트 : **[[‘이름1’, ‘이름2’ …]]** → ex) [[‘이름’,’나이’]]
- 범위 : **[시작번호 : 끝번호]** → ex) [0:3]
- 조건

- **범위로 지정 → 필터링 방법**

. df.loc[0:2]

. df[['과목명', '담당교수']]

- **조건으로 지정하는 방법 → df.loc[행조건, 열이름]**

. 행 찾기 : df.loc[df[‘이름’] == ‘은지’]

. 셀 찾기 : df.loc[df[‘이름’] == ‘은지’ , ‘생일’]

df.loc[df[‘이름’] == ‘은지’ , ‘생일’].values[0]

_데이터 프레임 : 필터 (filtering)

```
df.loc[0:2]
df[['과목명', '담당교수']]

df['과목명'] == '경영학'
df.loc[df['과목명'] == '경영학']
df.loc[df['시간수'] > 2]

df.loc[df['과목명'] == '경영학', '담당교수']
df.loc[df['과목명'] == '경영학', '담당교수'].values[0]
```

	과목번호	과목명	강의실	시간수	담당교수
0	C1	인공지능개론	R1	3	홍길동
1	C2	웃음치료	R2	2	김철수
2	C3	경영학	R3	3	이영희
3	C4	3D디자인	R4	4	박영수
4	C5	스포츠경영	R2	2	최영희
5	C6	예술의세계	R3	1	김영수
6	C7	통계학	R7	3	이철수

_데이터 프레임 - 데이터 수정

- 셀 수정 : 셀에 대해 대입연산자 사용

  df.loc[인덱스이름, 컬럼이름] = 새 값

  * 형식에 주의

	과목번호	과목명	강의실	시간수	담당교수
0	C1	인공지능개론	R1	3	홍길동
1	C2	웃음치료	R2	2	김철수
2	C3	경영학	R3	3	이영희
3	C4	3D디자인	R4	4	박영수
4	C5	스포츠경영	R2	2	최영희
5	C6	예술의세계	R3	1	김영수
6	C7	통계학	R7	3	이철수

- 검색으로 셀을 찾아서 수정

  df.loc[df['과목명'] == '경영학' , '담당교수'] = '이경영'

```
df.loc[3, '담당교수'] = '이경영'
print(df, end='₩n₩n')

df.loc[df['과목명'] == '경영학', '담당교수'] = '이경영'
print(df, end='₩n₩n')
```

RPA 실습

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!
```

형식 출력 - 문자열 포매팅

표시와 데이터 매칭에 초점을 둔다

- 형식1 : format 함수 쓰기

```
print("a - {0} b - {1}".format(a, b))      → 함수 형식
print(f"a - {a} b  - {b}")                 → 키워드 형식
```

- 형식2 : % 형식 쓰기

```
print("a:%d b:%f" % (a, b))          → 연산자 형식
print(f"a - {a:05d} b  - {b:.2f}")
```

_Python 실습

```
a=123
b=15.556

print("a:{0} b:{1}".format(a, b))

print(f"a:{a} b:{b}")

print(f"a:{a:05d} b:{b:.2f}")

print("a:%05d b:%.2f" % (a, b))
```

input() 함수

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

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

- input() 함수의 **반환값은 항상 문자열**임

→ 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
```

산술 연산자

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

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 실습

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

num4 = num1 // 3
print("num1 // 3   ", num4)
```

_Python 실습

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

_Python 실습

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

```
num = int(input("정수 입력: "))
prime = (num % 2) == 0
print("짝수이다 - ", prime)
```

_제어문

제어문

- **블록 → 들여쓰기 준수가 중요함**

- **조건/선택문**

  → if, if~else, if~elif~else문

- **반복문**

  → for, while

_제어문-조건/선택

조건/선택문

- if (조건) :

  [Tab] 실행문

```
a = 5

if a == 5:
   print('Right!')
   print('a is 5')

if a == 3:
   print('Right!')
   print('a is 3')

if a != 3:
   print('Right!')
   print('a is not 3')
```

```
Right!
a is 5
Right!
a is not 3
```

_제어문-반복

for-in문의 형식

- 컬렉션 자료의 원소들을 대상으로 동일한 처리를 함
- range함수와 같이 쓰는 경우가 많음
- 컬렉션 자료(리스트, 튜플, 집합, 딕셔너리)에 대해 컬렉션 자료 개수만큼 반복함

```
a = 5

for i in range(1, 10):
    print(f"{a} X {i} = {a*i}")
```

```
5 X 1 = 5
5 X 2 = 10
5 X 3 = 15
5 X 4 = 20
5 X 5 = 25
5 X 6 = 30
5 X 7 = 35
5 X 8 = 40
5 X 9 = 45
```