

# 2주차-1

1. 오늘은 무엇을 배우는가?
2. 왜 엑셀이 아니라 Pandas를 배우는가?
3. Pandas
  - 3.1. 파일 불러오기
  - 3.2. 인덱스 설정
  - 3.3. 열 데이터 다루기 : 데이터 접근, 추가, 수정
  - 3.4. 행 데이터 다루기 : 데이터 접근, 추가, 수정
  - 3.5. 합계와 평균
  - 3.6. 파일 저장하기

## 1. 오늘은 무엇을 배우는가?

1. 왜 엑셀이 아니라 Pandas를 배우는가?
2. pandas 라이브러리에 대해 배웁니다.
  - a. 파일 불러오기
  - b. 행 데이터 다루기 : 데이터 접근, 추가, 수정
  - c. 열 데이터 다루기 : 데이터 접근, 추가, 수정
  - d. 파일 저장하기

## 2. 왜 엑셀이 아니라 Pandas를 배우는가?

- 엑셀과 Pandas는 데이터를 다루기 위해 다른 종류의 도구를 제공합니다.
- 엑셀은 일반 목적의 데이터 관리에 적합하며, 이를 위해 자주 쓰는 도구들을 제공하고 있습니다.(필터, VLOOKUP 등)
- Python 등 프로그래밍 언어를 통한 데이터 분석은 훨씬 더 자유도가 높습니다.
  - 특히, **데이터 수집, 문자열 데이터 전처리**를 수행해야 할 때 pandas는 유용하게 사용할 수 있습니다.
  - Pandas와 다른 라이브러리를 연동하여 시각화 자료를 훨씬 더 다채롭게 만들 수 있습니다.
- 즉 엑셀로 다룰 수 없는 데이터를

## 3. Pandas

### 3.1. 파일 불러오기

- read\_excel 함수를 통해 데이터를 불러올 수 있습니다.

```
df = pd.read_excel("성적.xlsx")
```

### 3.2. 인덱스 설정

- 데이터를 빠르게 찾아낼 수 있도록 돕는 데이터 구조입니다.
- 데이터베이스는 아주 많은 데이터를 저장하고, 검색 결과를 빠르게 제공해야 합니다. 그래서 단순히 하나씩 데이터를 검색하지 않고, B-트리 구조와 같은 데이터를 효과적으로 저장하는 구조를 채택합니다. 이런 트리 데이터 구조를 만들기 위해 어떤 기준이 될 값을 지정해 줄 필요가 있으며, 이를 인덱스라고 합니다.
- set\_index를 통해 인덱스를 설정할 수 있습니다.  
drop=True란, 인덱스로 설정한 데이터 부분을 데이터 표에서 없애겠다는 뜻입니다.  
우리가 날짜의 합계를 계산하지 않듯, 인덱스에 대해서도 그런 연산을 하지 않기 때문에 보통 인덱스는 데이터베이스에서 제거합니다.

```
df = df.set_index("번호", drop=True)
```

### 3.3. 열 데이터 다루기 : 데이터 접근, 추가, 수정

- 열 데이터는 대괄호로 간단히 접근할 수 있습니다.

```
df["국어"] # 이런 식으로 한 열을 선택할 수 있습니다.
```

- 데이터에 없는 새로운 열을 추가할 수도 있습니다.

```
# 이렇게 새 열에 들어갈 자료들을 리스트로 주면, 새 열이 하나 생겨납니다.  
df["합계"] = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]  
  
# 또는 기존의 열을 가지고 계산을 할 수도 있습니다.  
df["합계"] = df["국어"] + df["수학"] + df["영어"]  
df["평균"] = df["합계"] / 3  
df["등수"] = df["합계"].rank(ascending=False)
```

### 3.4. 행 데이터 다루기 : 데이터 접근, 추가, 수정

- 열 데이터와 달리 .loc이라는 속성을 통해 접근할 수 있습니다.

```
df.loc[1] # 여기서 숫자 1은 인덱스입니다.
```

- 열 데이터와 마찬가지로 행 데이터도 추가하고 계산할 수 있습니다.

```
# loc 속성으로 접근하는 것만 빼면 사용 방법은 동일합니다.
df.loc[11] = [1, 2, 3, 4, 5, 6, 7, 8]

# 다른 행을 가지고 계산할 수도 있습니다.
df.loc[12] = df.loc[10] + df.loc[11]
```

### 3.5. 합계와 평균

- pandas에서는 합계와 평균을 구할 수 있도록 함수를 제공하고 있습니다.

```
# 형태가 동일하므로 한 번 기억해두면 다음부터는 쉽게 사용할 수 있습니다.
df["합계2"] = df[["국어", "수학", "영어"]].sum(axis=1)
df["평균2"] = df[["국어", "수학", "영어"]].mean(axis=1)
```

- 물론 직접 계산해도 같은 결과를 얻을 수 있습니다.

```
df["합계"] = df["국어"] + df["수학"] + df["영어"]
df["평균"] = df["합계"] / 3
```

### 3.6. 파일 저장하기

- to\_excel 함수를 통해 저장할 수 있습니다.

```
df.to_excel("filename.xlsx")
```