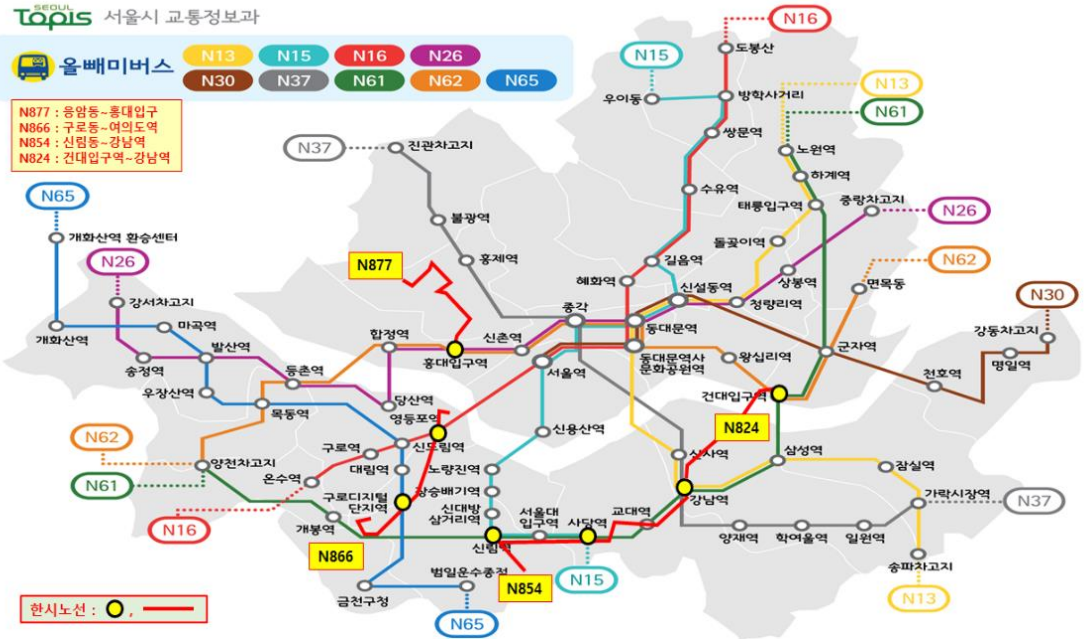
데이터분석 프로그래밍
데이터 과학

임현기

데이터 과학

- Data science is an interdisciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from structured and unstructured data, and apply knowledge and actionable insights from data across a broad range of application domains
- Data science is related to data mining, machine learning and big data.

데이터 과학

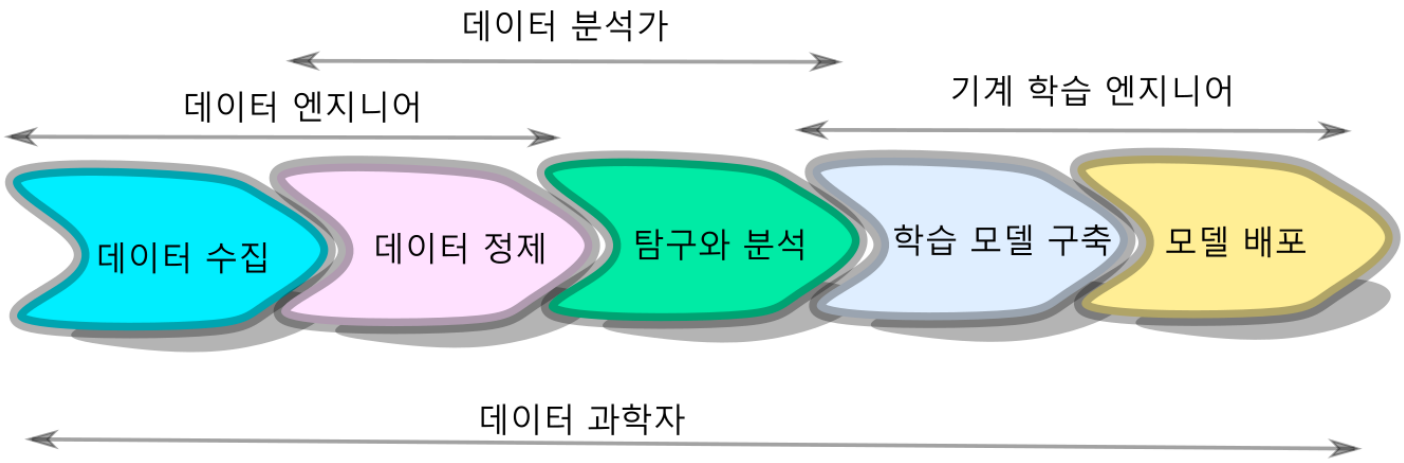


- 서울시는 심야버스 노선을 조정하기 위해 택시 승하차 정보와 통화량 데이터를 사용

데이터 과학

- 데이터 과학은 데이터를 확보하여 응용 분야에 사용할 수 있는 정보로 만들기 위해 다양한 과정을 수행
- 데이터 과학은 컴퓨터를 이용한 계산 과학의 진보를 바탕으로 데이터의 수집과 분석, 활용과 관련된 모든 이론과 기술을 종합적으로 다루는 분야
- 데이터 과학자는 데이터 엔지니어, 데이터 분석가, 기계학습 엔지니어의 역량을 모두 갖추고 다양한 문제를 해결
- 문제에 맞춰 소프트웨어를 개발할 수 있는 프로그래밍 실력은 이제 데이터 과학의 모든 단계에서 필요한 능력이 됨

데이터 과학


데이터 분석가
데이터 엔지니어
기계 학습 엔지니어
데이터 수집
데이터 정제
탐구와 분석
학습 모델 구축
모델 배포
데이터 과학자

파이썬

- 왜 파이썬인가?
  - 높은 생산성
  - 오픈소스 (무료)

파이썬 기본

```
>>> print('Hello Python!!')
Hello Python!!
```

```
>>> 5 + 6
11
>>> 반지름 = 4
>>> 면적 = 3.14 * 반지름 * 반지름
>>> print(면적)
50.24
```

```
# 다음 코드는 반지름을 이용하여 원의 면적을 출력하는 코드이다
반지름 = 4                          # 반지름의 값을 저장한다. 이때 공백이 들어가면 안된다
면적 = 3.14 * 반지름 * 반지름       # 반지름의 값을 이용하여 원의 면적을 구한다
print(면적)                         # 면적을 화면에 출력한다
```

파이썬 기본

- 주석(comment)은 소스 코드에 붙이는 설명 글
- 주석은 프로그램의 실행 결과에 영향을 끼치지 않음
- 파이썬에서는 #로 시작하면 줄의 끝까지 주석으로 취급

```
# 다음 코드는 반지름을 이용하여 원의 면적을 출력하는 코드이다
반지름 = 4                          # 반지름의 값을 저장한다. 이때 공백이 들어가면 안된다
면적 = 3.14 * 반지름 * 반지름       # 반지름의 값을 이용하여 원의 면적을 구한다
print(면적)                         # 면적을 화면에 출력한다
```

파이썬 기본

```
>>> 2 + 3
5
>>> print(2 + 3)
5
>>> print(2 - 3)
-1
>>> print(2 * 3)
6
>>> print(2 / 3)
0.6666666666666666
```

```
>>> print(2345 * 9876 - 5678)
23153542
```

```
>>> print(12345789123456789 * 12345789123456789)
152415787806736785156226207501905 21
```

파이썬 기본

- 따옴표로 시작하여 같은 따옴표로 끝나는 문자열을 프롬프트에 입력하면 그 상태 그대로 나타남
- 데이터가 문자열이라는 것을 알려주는 것
- 하지만 print() 함수 안에 문자열이 있을 경우 따옴표는 나타나지 않음

```
>>> 'Hello'                                  # 문자열 'Hello'
'Hello'
>>> "Hello"                                  # 문자열 "Hello"는 'Hello'와 동일하다
'Hello'
>>> print('Hello')                           # print() 함수안에 문자열이 있을 경우 따옴표는 나타나지 않음
Hello
>>> print("즐거운 " + "파이썬 익히기")        # 두 텍스트 데이터를 연결하여 출력함
즐거운 파이썬 익히기
```

파이썬 기본

- 문자열에 + 연산자를 이용하여 다른 문자열을 덧붙이면, 두 문자열이 연결
- 문자열에는 다음과 같이 곱셈 기호를 사용하는 것도 가능

```
>>> print('반가워요 ' * 20)          # '반가워요'를 20회 반복 출력함
반가워요 반가워요 반가워요 반가워요 반가워요 반가워요 반가워요 반가워요 반가워요 반가워요 반가워요 반가워요 반가
워요 반가워요 반가워요 반가워요 반가워요 반가워요 반가워요
```

파이썬 기본

- 문자열과 숫자를 구별해야 함
- "100"은 문자열이고 100은 숫자
  - "100"+"200"을 실행하면 "100200"이 출력
  - 100 + 200은?

```
>>> print("100" + "200")     # 문자열 '100', '200'을 연결한다
100200
>>> print(100 + 200)         # 숫자 두 개의 합을 구한다
300
```

파이썬 기본

잠깐 – 따옴표의 사용법과 여러 줄에 걸친 문자열

문자열을 표현할 때 작은따옴표와 큰따옴표 중 아무것이나 사용해도 된다. 딱 한 가지 지켜야 할 것은 큰따옴표로 시작한 문자열은 큰따옴표로, 작은 따옴표로 시작한 문자열은 작은따옴표로 끝내야 한다는 약속이다. 줄바꿈을 포함하여 여러 줄에 걸친 문자열을 표현하고 싶을 때도 있다. 이때는 **작은따옴표나 큰따옴표 세 개로 문자열을 시작하고 같은 방식으로 닫으면 된다.**

```
>>> multiline_string = """This is a multiline string
with "newline" characters
within the string"""
>>> print(multiline_string)
This is a multiline string
with "newline" characters
within the string
```

따옴표 3개로 시작해서 끝나는 문자열 안에는
큰따옴표, 작은따옴표를 모두 사용 가능

파이썬 기본

- 모듈(module)
  - 파이썬 함수, 변수, 또는 클래스를 별도의 파일로 저장하여 불러서 사용할 수 있도록 한 것
- 표준 라이브러리(standard library)
  - 파이썬 설치시 함께 제공되는 모듈
  - 기본적인 기능들을 제공
- 외부 라이브러리
  - 흔히 패키지(package)라고 부름
  - pip를 이용해 패키지 설치가 가능

파이썬 기본

- pip를 이용하여 설치할 때는 윈도우 컴퓨터의 명령행에서 다음과 같은 명령을 입력

```
C:\> pip install package-name
```

- 예를 들어 numpy라는 패키지를 설치한다면

파이썬 기본

- 데이터 분석, 기계학습 등을 한다면
  - numpy, matplotlib, pandas, scikit-learn, seaborn, opencv-python

```
C:\> pip install numpy matplotlib pandas scikit-learn seaborn opencv-python
```

파이썬 기본

- 설치된 모듈을 불러오기 위해 'import 모듈 이름'
- 사용할 때 모듈 이름에 점(.)을 찍은 후 모듈 안의 구성요소를 호출

```
import module-name
module-name.func()
```

파이썬 기본

- 오류 발생

```
>>> PRINT("Hello World")
Traceback (most recent call last):
File "<ipython-input-2-ab351b16d57b>", line 1, in <module>
PRINT("Hello World")
NameError: name 'PRINT' is not defined
```

```
>>> print(Good Bye)
File "<ipython-input-4-0389bd3941f5>", line 1
print(Good Bye)
^
SyntaxError: invalid syntax
```

잠깐 – 대표적인 오류 메시지만 알아도 당신은 프로그래머로 한 걸음 더 나갈 수 있다

SyntaxError: invalid syntax - 파이썬 언어의 약속된 문법 규칙을 지키지 않은 표현이 나타남
IndentationError: expected an indented block - 필요한 들여쓰기를 하지 않은 오류
IndentationError: unexpected indent - 들여쓰기를 하지 않아야 할 곳에서 글을 들여쓴 오류
NameError: name x is not defined - 무언가 가리키는 이름이 사용되었는데 뭔지 알 수 없을때
TypeError: Can't convert ... - 데이터의 종류가 다른 것들이 서로 값을 주고 받을 때