**파이썬 1일**

* 가비지 컬렉션 ?? 으로 메모리 관리해서 좋음

파이썬 코드 스타일 가이드

* 한 라인 79자 입력
* 들여쓰기 공백 4칸
* (), [] 안에는 공백문자 쓰지 않기

Naming Conventions

* 모듈 이름 소문자, snake\_case 사용
* 클래스, 예외는 CapWords(PascalCase) 사용
* 함수, 변수, 속성 소문자와 snake\_case 사용
* \_leading\_underscore : 모듈 내부에서 사용되는 클래스, 함수, 변수, 메서드 선언
* \_\_double\_leading\_underscore : 속성이 맹글링(Mangling) 되어 클래스간 속성명의 충돌이 방지된다.
* \_\_double\_leading\_and\_trailing\_underscore\_\_ : 파이썬이 제공하는 특별메서드 (special, magic method) 가 사용. Built-in method 이며, 충돌 피하기 위해 사용자 선언 피한다.
* 모듈 수준의 상수는 대문자와 SNAKE\_CASE 사용(지역변수??전역변수?)

주석

* 줄단위 #
* 블록 단위 (Block Comments) ‘’’ / “”” - docstring에서 많이 사용

구문 종료와 블록

* 한 행에 여러줄 쓸 경우 ; 사용
* 긴 줄의 경우 내려가기 전에 \ 쓰거나 ( 첫째줄 둘쨰줄 ) 이케 괄호로 묶기

변수 종류

* 가변형(Mutable) 불변형(Immutable) 있음
* Sequence 형(순서있음) , Mapping type(키와 value) , Set type(집합형, 교집합 합집합 차집합 사용 가능)

파이썬 특징 – 숫자 등 최댓값이 없다!!!!! (메모리가 감당가능한 한)

**넘파이**

Ndarray

* 요소별 연산 가능! (a\*3 = 요소별 3 곱함)

차원별

* 0차원 , 하나 : scalar , shape(1,)
* 1차원 : vector , shape(3,)
* 2차원 : matrix , shape(3, 2)
* 3차원 : tensor , shape(3, 2, 3)

인덱싱 / 슬라이싱

* 정수 인덱싱 : a[[row1, row2], [col1, col2]]
* 부울린 인덱싱 : a[[True, True, False …]]
* 팬시 인덱싱 (Fancy indexing)

연산

* 사칙연산자/합(sum)/평균(mean)/표준편차(std)/상관계수(corrcoef)/최대최소(max, min)/분산(var)/누적합(cumsum)/누적곱(cumprod)
* any(a >= 4) = boolean(하나라도 만족하면 True) , all(a >= 4) = boolean(하나라도 틀리면 False)
* 동일 shape / scalar 연산하기
* 요소별 연산 = 벡터화 연산 (vectorized operation) 🡪 속도 빠름
* Shape 같은 두 배열은 요소별 연산 됨
* Shape 다른 경우 Broadcasting 일어남 🡪 ????

넘파이 문제점

* 2차원 배열에서 세로끼리 뭐 하기 잘 안됨 보기 불편함
* Pandas 하기 위한 Numpy

**판다스**

* 넘파이 기반 데이터분석 라이브러리
* 다양한 **인덱스** 기반으로 대용량 다차원 데이터 조작, 검색 빠른속도 지원
* 대용량 파일 처리 🡪 메모리 중요. 8기가 이상 16기가 추천
* CSV, txt, Excel, 데이터베이스 접속 등 다양한 외부정보 입출력 가능
* 다양한 날짜 범위 생성기능 제공, 인덱스로 사용 가능
* 숫자, 라벨(문자) 기반 인덱싱 사용 가능
* 중요코드가 Cython, C여서 최적화된 성능
* 판다스 홈페이지에 기능 설명 (pandas.pydata)

장점

1. 인덱스 - indexing, slicing
   1. 숫자
   2. 문자(label)
   3. 날짜
2. 다양한 source 경로

자료형

* Int64
* Float64
* Bool
* Object : 문자열, 유니코드, 복합 형식
* Datetime64[ns] : 날짜, 시간
* Timedelta64[ns] : 날짜, 시간 간격
* Category : 범주화. 문자🡪숫자 시 사용. Categorize

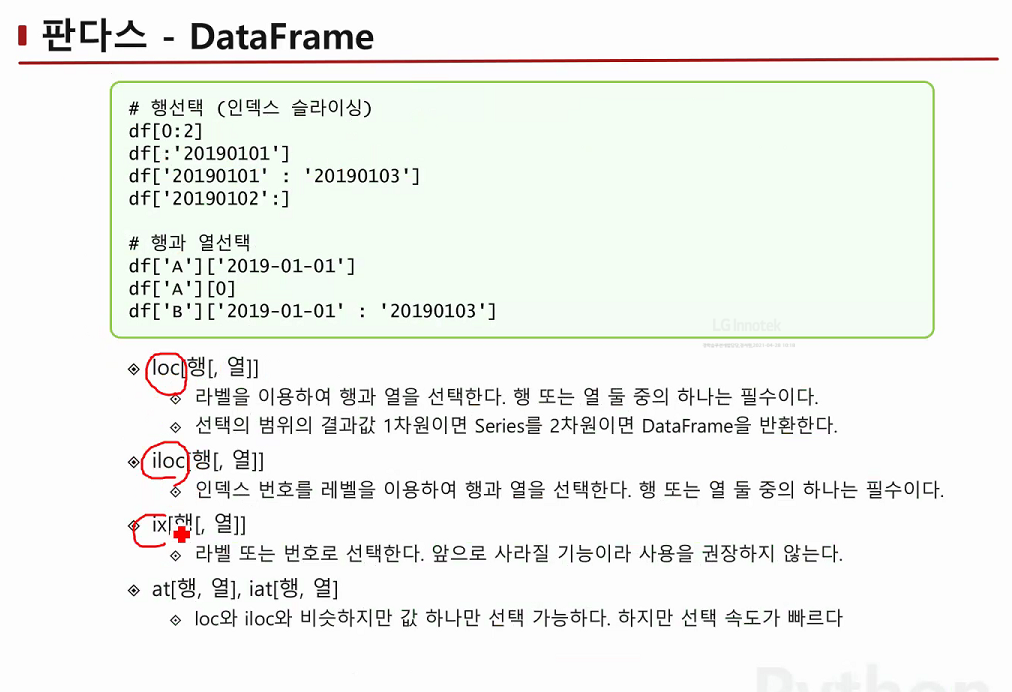
자료구조

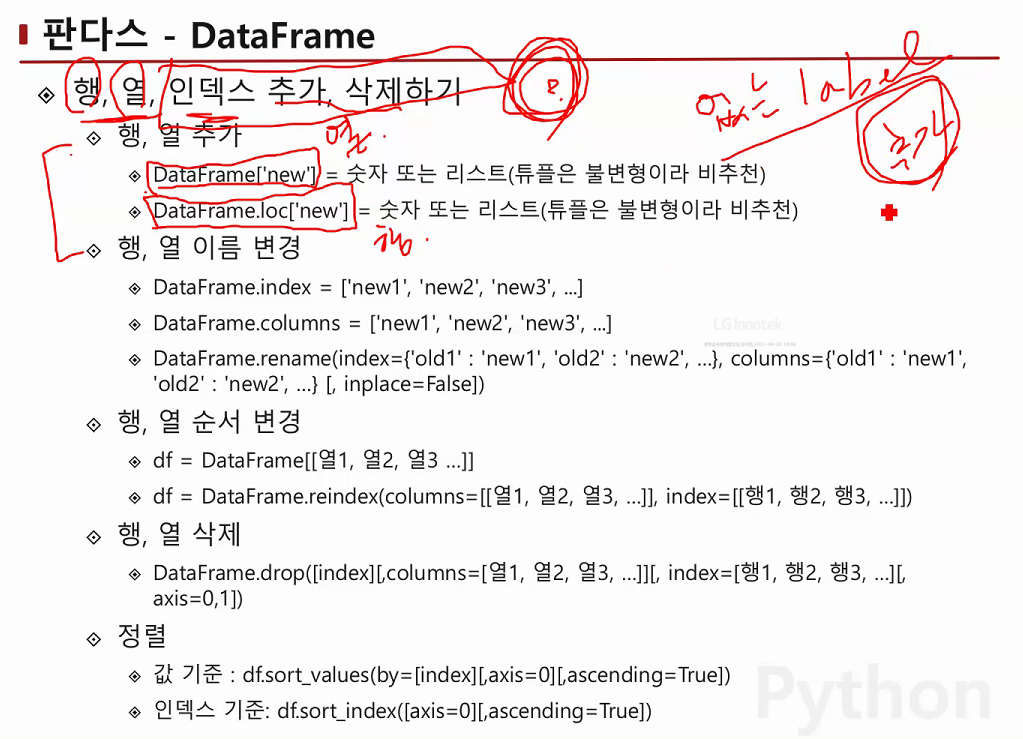
* 1차원 : series
* 2차원 : DataFrame
* 3차원 : Panel

날짜관련

* To\_datetime : 여러 날짜 표현법 하나로 날짜로 인식해 날짜인덱스 만들어줌
* Datetime
* Date\_range : 주어진 조건 만족하는 날짜, 시간 주기의 datetimeindex 반환
  + - * Date\_range(start=’’, end=’’ [, period=…)

DataFrame (데이터프레임)

* 가변형, 2차원 자료구조, 파이썬 Dictionary/관계형 데이터베이스 테이블과 흡사
* 생성: pandas.DataFrame(data [, index=index][,columns=columns][,dtype=dtype][,copy=bool])
* 이용할만한 데이터셋 : 마리아DB 등등
* 권장 인덱싱 : loc, iloc, at, iat
* 
* 주로 열 : df[‘A’] / 행 : loc 많이 씀
* 행, 열 추가 / 삭제 (사용 비권장)



데이터 형 바꾸기

* Apply : series
* Map : series / dataframe
* Applymap : dataframe

판다스 문자열

* 범주형(category) 으로 바꿔 처리하는게 좋음
* Column 지정 후 str 꼭 삽입

연산

* 벡터화 연산 지원, series와 기능 동일, 메소드도 동일
* 결측치 있는 경우 fill\_value 이용하여 기본값 지정 가능

다른 파일 입출력

* CSV
  + To\_csv , read\_csv …
  + 한글 있을 시 encoding 다르므로 지정
* Excel
  + Read\_excel, to\_excel…

Csv 데이터 처리

* Melt : 동일 그룹 데이터가 행이 나뉘어 있어 열로 묶일 때 열로 묶어 표현하면 편함
* 천의자리 , 제거 : thousand=”,”

결측치 (Missing Value)

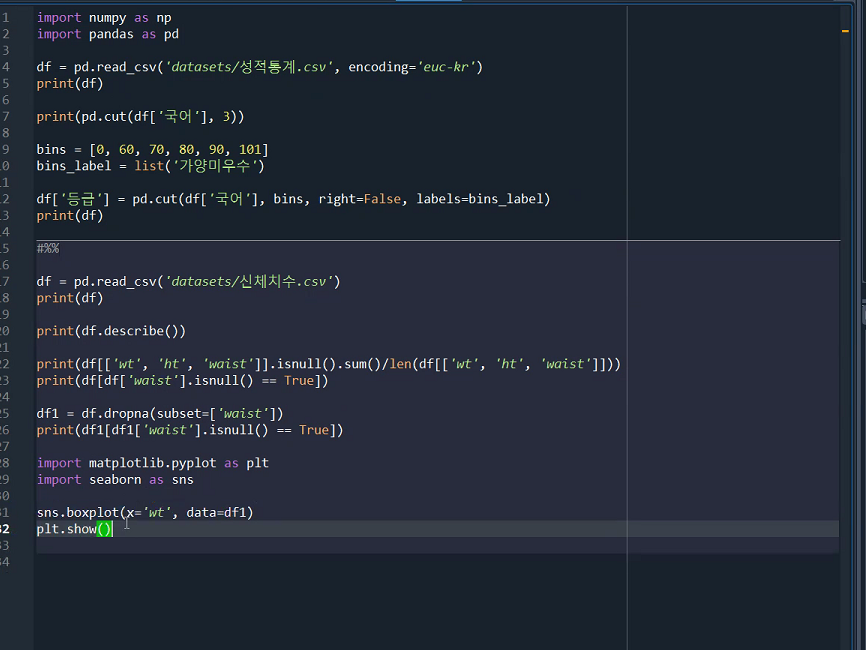
: 데이터 중 값이 없는 것(NaN, None)

: 올바른 데이터 처리에 큰 영향을 주는 값으로 데이터 전처리 과정에서 중요한 작업 중 하나

처리방법

* 결측치 비율 산출
* Df.droopna(axis=0, 1) 1 이 열 삭제

## method 행 – 열 dataframe 열 – 행



Method=’ffill’ : 보간법=결측값 채우기