|  |  |
| --- | --- |
| 교육 제목 | **Python 마무리 및 Pandas** |
| 교육 일시 | 2021년 09월 28일 |
| 교육 장소 | YGL |
| **교육 내용** | |
| 오전 | 객체: 여러 가지 속성을 가질 수 있는 모든 대상, 자세한건 좀 다르지만 데이터라고 생각하면 편하다.  객체지향 프로그래밍: 객체를 우선하는 프로그래밍으로 C를 제외한 대부분의 프로그래밍 언어들이 그렇다.  클래스: 객체를 효율적으로 생성하기 위해서 만들어진 구문이다. 이름을 지을 때는 대부분 대문자를 사용하여 햇갈리는 걸 방지한다. 기본적인 형태는 class 클래스이름:  메서드 ->함수이다  생성자: 클래스 이름과 같은 함수  class 클래스이름:  def \_\_init\_\_(self, 추가적인 매개변수):  pass  인스턴스: 변수인데 생성자를 사용하여 이러한 클래스 기반으로 만들어진 객체  인스턴스 이름(변수이름) = 클래스이름() --->생성자 함수  메서드: 각 객체에 있는 함수들  상속: 어떤 클래스를 기반으로 그 속성과 기능을 물려받아 새로운 클래스를 만드는 것  isinstance()함수 : 객체가 어떤 클래스로부터 만들어졌는지 확인  srt()함수  클래스 변수와 메소드: 인스턴스가 속성과 기능을 가질 수도 있지만, 클래스가 속성(변수)과 기능(함수)를 가질 수도 있다.  -클래스 변수  class 클래스이름:  클래스 변수 = 값  -클래스 변수에 접근하기  클랫 이름.변수 이름  클래스 함수: 클래스가 가진 함수로서 '클래스가 가진 기능'을 명시적으로 나타낸다.  -클래스 함수 만들기  class 클래스 이름:  @classmethod ->데코레이터로  def 클래스함수(cls, 매개변수(클래스이름)):  pass  소멸자-가비지 컬렉터: 더 사용할 가능성이 없는 데이터를 메모리에서 제거하는 역할을 한다.  프라이빗 변수: 변수를 마음대로 사용하는 것을 방지한다. 인스턴스 변수 이름을 \_\_변수 이름 형태로 선언하면 된다.  -게터와 세터: 프라이빗 변수는 외부에서 접근 할수 없기 때문에 프라이빗 변수를 가진 함수를 선언하고(get/게터라고 부름) 그 함수를 또 호출하고 다른 변수에 넣는 함수를 선언(set/세터라고 부름)한뒤 외부에서 그 프라이빗 변수를 가진 함수를 변수에 넣은 함수를 호출하는 것이다.  다중 상속: 다른 누군가가 만들어 놓은 형태들을 가져와서 내가 원하는 것으로 만드는 것  판다스: 시리즈(1차원 배열)와 데이터프레임(2차원 배열)이라는 구조화된 데이터 형식을 제공한다.  시리즈: 데이터가 순차적으로 나열된 1차원 배열의 형태를 갖는데 인덱스와 데이터 값이 일대일 대응을 한다. 키와 값이 짝을 이루는 파이썬의 딕셔너리와 비슷한 구조를 갖는다고 볼 수 있다. 그래서 파이썬에서 딕셔너리를 시리즈로 대체 가능. |
| 오후 | 데이터프레임: R의 데이터프레임에서 유래 되었고 시리즈가 열벡터(vector)라고 하면, 데이터 프레임은 여러개의 열벡터들이 같은 행 인덱스를 기준으로 줄지어 결합된 2차원 벡터 또는 행렬이다.  -rename 메소드 사용: DataFrame 객체.rename(index={기존인덱스:새인덱스})  행/열 삭제: 행 삭제는 데이터프레임객체.drop(행이름, axis=0) 열 삭제는 데이터프레임객체.drop(행이름, axis=1)  데이터 프레임 객체 복사: 변수 = 데이터프레임객체[:]으로 해도되지만 버전이 바뀌면서 변수=데이터프레임객체.copy()로 하는걸 추천된다.  행 선택: loc는 인덱스 이름을 기준으로 행을 선택하는 것이고, 정수형 인덱스 위치를 기준으로 선택할 때는 iloc를 사용한다.  loc -> ['a' : 'c'] -> 'a','b','c' iloc -> [3:5] -> 3,4,5,6(7제외) |