Chapter.7 변수와 자료형 이해하기

목차

- 변수 개념
- 자료형

변수 개념

- 변수
  - 프로그램 내에서 데이터를 사용하기 위해서는 데이터 저장해 둘 공간 필요
  - 변수는 하나의 값만 저장하거나, 여러 개의 변수에 여러 개의 값을 동시에 저장할 수 있음
  - 데이터를 저장하는 그릇을 구별하기 위해서 그릇의 식별자(변수명)가 필요
- 변수 정의
  - 지정할 변수명을 사용자가 임의로 쓰고, '='이라는 기호를 이용하여 변수 값 지정
  - 변수는 변수값이 지정된 변수라 해도 다시 같은 변수명에 다른 변수값을 대입하면 맨 마지막에 지정한 변수값으로 변경

다양한 변수 타입

- 다양한 변수 타입
  - 숫자의 경우, 변수명에 변수값을 지정
  - 문자의 경우, 문자 양쪽에 작은따옴표(‘ ’) 혹은 큰따옴표(“ ”)를 사용해야 함
  - 파이썬 기본함수 print() 함수 이용, 연산 결과 확인

변수 선언 및 초기화, 지정

- 변수 선언 및 초기화
  - 대입연산자를 이용하여 각 변수명에 변수값을 지정하여 변수 선언
  - 변수 지정하고자 할 경우, 좌우가 바뀌지 않게 주의 → 오류발생
- 변수명 지정
  - 변수명은 사용자의 임의대로 지정, 기본적으로 지켜야할 규칙이 있음
  - 파이썬에서 변수명은 한글도 가능, 일반적으로 영문 및 숫자만 사용하는 것이 좋음
    - 변수명은 영문만으로 지정하거나 영문과 숫자가 혼용되어 정의하는 것이 일반적
    - 영문과 숫자를 혼용할 경우 숫자가 맨 앞에서 시작하면 오류 발생, 시작은 영문으로 해야 함
  - 영문은 대소문자 구분
  - 단어 간 연결을 통하여 변수명을 지정할 경우 공백은 사용할 수 없으며, 언더바(_) 이용
  - 파이썬에 이미 예약된 예약어는 변수명으로 사용할 수 없음

자료형 – 숫자, 문자형

- 자료형(data type)
  - 프로그램에서 사용할 수 있는 자료의 종류
  - 변수에는 정수, 실수, 문자열, 부울형 등 다양한 값들을 대입하여 저장
  - type(변수명) 이라는 함수 이용하여 변수 자료형 확인
- 숫자형 (정수, 실수)
  - 일반적으로 모든 숫자 의미, int, float 형으로 정의
  - 연산 가능
- 문자열
  - 자료형으로 문자열은 단어, 문장 등을 큰따옴표(“ ”), 작은따옴표(‘ ’)로 둘러쌓아 표시
  - str형으로 정의

자료형 – 숫자, 문자형 (cont.)

- 문자열 포맷팅(formatting)
  - 문자열은 print() 함수 사용하여 출력
  - 프로그램을 실행되는 과정에서 달라지는 변수값을 문자열과 함께 확인하고자 할 때 사용되는 것이 문자열 포맷팅
  - 포맷 코드(%) 이용하여 특정 위치에 넣어 문자열을 연결하는 방법
    - 변수를 지정하여 변수의 값으로 포맷코드 대체
    - 포맷코드 외 문자열 사이 값이 들어갈 위치에 숫자 혹은 변수 삽입하여 출력하여 확인하는 방법

자료형 – 부울형

- 부울형
  - 프로그램에서 참(True), 거짓(False)만 저장할 수 있는 데이터 형식 정의
  - 첫 글자는 대문자로 작성, 소문자로 작성하면 오류 발생

기본 자료형

- 자료형은 대부분 대입에 의해 자료형이 지정되는데 같은 변수에 마지막으로 정의된 데이터 값에 따라 자료형이 정해짐

자료유형	자료형	예
정수(integer)	int	-3,-2,-1,0,1,2,3,...
실수(float)	float	0.5, 1.5, 5.999,...
문자열(string)	str	"python", '123'
부울형(boolean)	bool	True, False

자료형 변환

- 변수에 저장된 정수, 실수, 문자열 등의 다양한 자료형들의 값들은 사용자에 의해 강제로 다른 자료형으로 변환
  - int() 함수 : 정수 형태의 문자열/실수 값을 정수로 형 변환함
  - float() 함수 : 실수 형태의 문자열/정수 값을 실수로 형 변환함
  - str() 함수 : 실수/정수 값을 문자열로 형 변환함

사용자로부터 데이터 입력 받기

- 변수의 값을 고정하지 않고 프로그램을 실행할 때마다 다른 값으로 지정해서 활용
  - input()함수
    - 입력한 변수 값을 연산이 가능한 정수형으로 변환하고자 할때, int()함수 이용

리스트(List)

- 리스트 정의
  - 여러 개의 데이터를 한꺼번에 하나의 이름으로 담을 수 있는 자료 구조
  - 리스트에 저장되는 각 데이터를 항목이라 하고, 항목은 여러 데이터 유형으로 구성될 수 있음
  - 리스트는 시작과 끝에 대괄호([ ]) 사용, 항목들은 쉼표(,)로 구분
  - 리스트의 각 항목은 자동으로 순서가 매겨지는데 이 순서를 인덱스(index)

리스트(List)

- 리스트 생성 및 구조

```
score = [80, 99, 75, 55, 86]
```

- 리스트 특정 항목 확인
  - 리스트 특정 항목을 불러오고자 할 경우, '리스트명[인덱스]' 형식으로 기술
  - 리스트 명은 변수와 마찬가지로 사용자가 임의로 지정

리스트(List)

- 리스트 값 접근하기
  - 생성된 리스트에 접근할 때 인덱스를 통하여 접근할 수 도 있지만 범위를 지정하여 접근할 수도 있음
- 리스트 특정 색인 값 변경
  - 리스트에서 특정 항목의 값을 변경하려면 인덱스를 이용하여 변경 가능
- 리스트 항목 추가
  - 생성된 리스트 맨 뒤에 값을 추가는 len()라는 기본 함수를 이용하여 추가
  - 리스트에는 맨 뒤에 항목 하나뿐 아니라 리스트의 형태로도 연결하여 추가할 수 있음
- 리스트 항목 삭제
  - del() 함수 : 리스트에 특정 인덱스에 매칭되는 항목을 삭제하는 함수

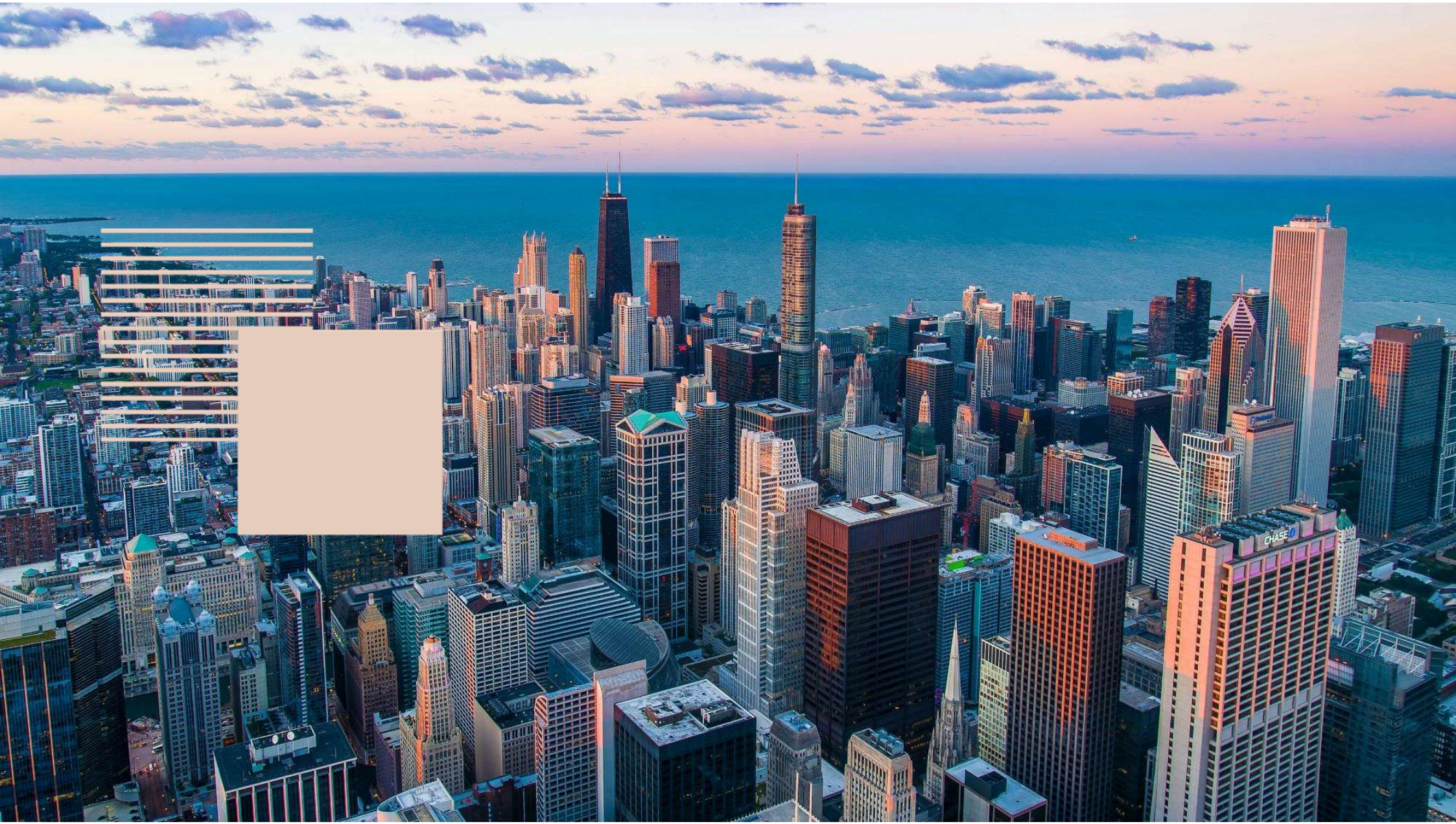
튜플 (Tuple)

- 튜플 정의
  - 튜플은 변경할 수 없는 데이터의 집합
  - 리스트는 읽기, 쓰기가 가능하지만 튜플은 읽기만 가능
  - 튜플은 일반적으로 '서로 다른 종류의 데이터 타입'으로 이루어진 항목들을 변수에 바로 풀어서 할당하거나 인덱스를 매길 때 사용
  - 소괄호 기호(( ))로 감싸거나, 아예 감싸지 않음, 항목을 쉼표(,)로 구분

딕셔너리 (Dictionary)

- 딕셔너리 정의
  - 딕셔너리는 전체 항목이 정렬되지 않은 키와 값의 쌍으로 구성된 데이터 모음으로 정의
  - 딕셔너리는 중괄호({ })로 항목들을 묶어서 표현
  - 각 항목은 키(key)와 값(value)의 쌍으로 이루어짐
  - 키와 값은 콜론(:)으로 연결하여 구성
  - 항목 구분은 쉼표를 사용하여 구분
  - 딕셔너리에 저장된 정보는 키는 수정할 수 없지만, 값은 수정할 수 있음

```
딕셔너리 명 = {key1:value1, key2:value2, key3:value3, ...}
```


CHASE