Part 1. 시작하기

* + 저장한 코드 실행방법 : Python file1.py
  + 함수란 파이썬이 특정한 기능을 수행하도록 하는 명령
  + print()함수
    - 1. 괄호 안에 있는 내용을 화면에 출력하는 명령
      2. 화면에 표시하고 싶은 글은 따옴표(‘)로 묶어서 표현
      3. 예. print(‘hello, world’)

Part 2. 변수와 계산

1. 변수 사용하기
   * 선언한 변수의 이름으로 저장된 값을 불러 사용할 수 있다
   * 한 번 정해진 값은 언제든지 다른 값으로 바꿀 수 있다
   * 숫자는 따옴표를 사용하지 않는다
2. 주석
   * 코드를 설명하기 위에 코드 위에 적어놓는 문자
   * 코드를 임시로 꺼두기 위해서도 사용
   * 한 줄을 주석으로 처리할 때는 #, 여러 줄을 주석으로 처리할 때는 따옴표 “””로 그 내용을 둘러싸면 됨
3. 숫자와 문자열
   * 변수에 값을 저장할 때, 계산을 위해서는 숫자로 저장하고 그렇지 않을 때에는 문자열로 저장
   * 숫자는 계산이 가능함 – 더하기(+), 곱하기(\*), 나누기(/), 거듭제곱(\*\*), 나머지(%)
   * 텍스트 두 개를 더하면 문자열이 이어 붙여짐 – 텍스트는 더하기만 가능하고 다른 계산은 불가능
     + 1. ex) test = ‘2015’ + ‘1991’ => test에 ‘20151991’라는 값이 저장됨
4. REPL
   * Powershell에서 파일 이름 없이 python이라고만 입력하면 나오는 창
   * 파이썬 코드를 한 줄씩 입력해 가면서 테스트 해 볼 수 있는 입력창
   * REPL창을 종료하려면 exit()라고 입력

Part 3. 조건문

1. if문
   * 구조
     1. if 조건 :

들여쓰기 print(“내용”)

* + - 1. 조건 : 참 / 거짓을 판단할 수 있는 조건. 조건 끝에는 조건이 끝났다는 걸 표현하기 위해 :를 붙여야 함
      2. 조건이 끝나면 조건에 맞추어 실행하고자 하는 코드를 작성한다. 꼭 들여쓰기 후 작성해야 함

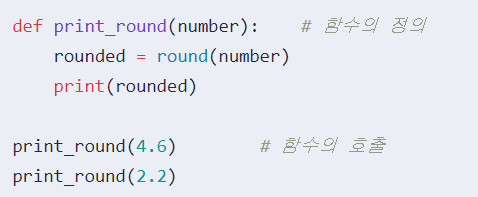
1. 조건식
   * 숫자 비교
     1. 크다
        1. 0 < 10
        2. 10 > 11
     2. 크거나 같다, 작거나 같다 (꼭 부등호 뒤에 등호를 붙여야 함)
        1. 3 <= 10
        2. 15 >= 10
     3. 같다
        1. 5 == 5
     4. 같지않다
        1. 5 != 5

* 비교의 결과는 True 또는 False
  + Boolean 연산
    1. and
       1. 두 조건이 모두 참인지를 체크
    2. or
       1. 두 조건 중 하나라도 참인지 체크
    3. not
       1. True/False를 뒤집기 위해 사용

1. 블록
   * 들여쓰기로 구분 되어 함께 실행되는 하나의 코드 덩어리
   * 콜론 문자 뒤에 들여쓰기로 작성됨
   * 여러 줄로 작성할 때는 모든 줄의 들여쓰기가 일치해야 함
     + 1. 들여쓰기가 어긋나면 문법 오류가 발생하기 때문
   * 블록을 끝내려면 끝마칠 블록의 다음 줄에서 내어 쓰기 하면 됨
   * 블록 안에 다른 블록이 들어갈 수 있음
   * 내부의 블록은 외부 블록에 종속적임
   * 파이썬 코드 전체를 하나의 블록으로 볼 수 있음
2. if else
   * else
     + 1. 조건이 맞지 않을 때 실행하는 코드로 반드시 if 뒤에 작성됨
   * elif
     + 1. else와 if의 결합으로 조건이 맞지 않는 경우 다른 경우를 검사

Part 4. 함수

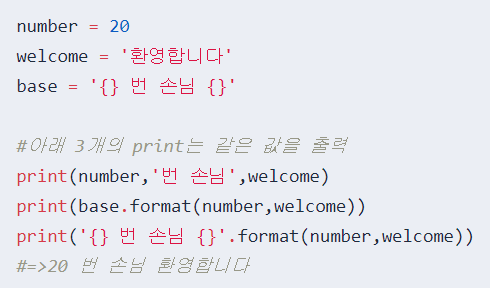
1. 함수 만들기
   * def 함수 이름():
     + 1. def : 새 함수를 정의하는 선언
2. 매개변수
   * 매개변수
     + 1. 함수를 정의할 때 사용하는 이름
   * 실행 인자
     + 1. 함수를 실행할 때 넘기는 변수, 값
   * 매개변수와 실행 인자
     + 1. 매개변수와 실행 인자의 개수는 동일해야 함
       2. 여러 개일 경우 쉼표로 구분



1. 함수의 값
   * Return은 실행된 즉시 함수 실행이 끝나 버림
   * return을 이용해 값을 돌려줄 수 있음
   * 여러 값을 반환할 때에는 return 뒤에 여러 값을 쉼표로 구분해서 값을 보내고, 받을 때도 쉼표로 구분하여 받음

Part 5. 자료형

1. Format
   * 문자열.format()
     1. 문자열의 대괄호 자리에 format 뒤의 괄호안에 들어있는 값을 하나씩 넣는다
     2. 문자열에 포함된 대괄호 개수 보다 format안에 들어 있는 값의 수가 많으면 정상 동작
        1. print('{} 번 손님'.format(number,greeting))
     3. 문자열에 포함된 대괄호 개수 보다 format안에 들어 있는 값의 수가 적으면 에러
        1. print('{} 번 손님 {}'.format(number))



1. 문자열
   * 파이썬에서 따옴표의 역할은 문자열(string)이란 자료형을 만드는 것임
   * 문자열은 파이썬에서 텍스트를 다룰 때 사용하는 자료형(type)
   * ' 또는 "로 문자열 만들기
     + 1. ' 또는 "로 글의 양쪽을 감싸면 문자열로 인식
       2. '로 감싼 문자열 안에는 "를 쓸 수 있다.
       3. "로 감싼 문자열 안에는 '를 쓸 수 있다.
   * 따옴표 세 개(‘’’)를 사용하면 줄 바꿈 문장을 쓸 수 있음
2. 정수와 실수
   * 정수
     1. int라고 표현
     2. 정수끼리 더하거나 곱하면 정수
     3. 정수끼리 나누면 실수가 나올 수 있으나 나눗셈의 몫 만을 구하려면 ‘//’연산자를 이용



* + 1. 실수를 정수로 바꾸려면 int를 이용



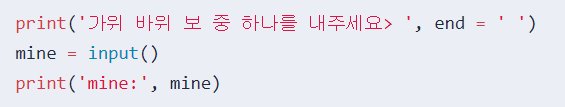
* + 실수
    1. 부동소수점이라는 표현법을 이용해 소수점을 표시할 수 있는 숫자
    2. 어느 정도의 계산 정확도는 가지지만, 계산에 있어서 완벽한 정확성은 가지지 않는다.



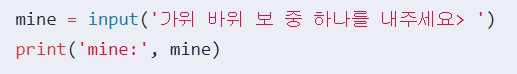
* + 1. 정수를 실수로 바꾸려면 float를 사용



1. 사용자 입력 받기
   * 프로그래밍의 3단계
     1. 사용자 입력
     2. 자료 처리
     3. 결과 출력
   * input()
     + 1. 사용자의 키보드 입력을 return 해 줌



* + - 1. 간단한 print 기능을 내장



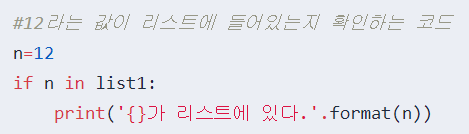
Part 6. 리스트

1. 리스트 사용
   * list : 여러 개의 값을 담을 수 있는 변수



* + 값 읽어 오기
    1. 리스트를 사용할 때는 0번째가 첫번째
       1. 첫번째 값 : list1[0], 두번째 값 : list1[1]
    2. -1은 뒤에서 첫번째 값이라는 뜻
       1. 뒤에서 첫번째 값 : list1[-1], 뒤에서 두번째 값 : list1[-2]
    3. 리스트에 들어있는 값의 개수를 초과하는 값을 불러오면 에러가 남
       1. 위의 list1에서 list1[5] 또는 list1[-6]은 에러
  + 값 쓰기
    - 1. 변수와 같이 list1[0] = 10이라고 하면 list의 첫번째 값이 10으로 변경

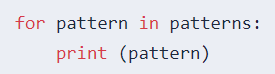
1. 리스트 수정
   * 리스트에 새로운 값을 추가하는 방법
     + 1. list1 = [1, 2, 3]이라고 할 때
     1. append를 이용
        1. list1.append(4)
        2. append를 이용하면 기존 리스트에 새로운 값이 추가됨
     2. 뒤에 새로운 리스트를 더하기
        1. list2 = list1 + [4]
        2. list1은 그대로 두고, 새로운 값을 추가한 새 리스트를 만들어 냄
   * 리스트에 값이 들어있는지 확인하는 방법
     + 1. in 연산을 이용



* + 리스트에서 필요 없는 값을 지우는 방법
    1. del을 이용해서 **특정 위치의 값**을 지우기
       1. del list1[10] => 리스트의10번째 값을 지워라
    2. remove를 이용해서 **특정 값**을 지우기
       1. list1.remove(40) => 리스트에 40이라는 값이 있는 경우 이를 삭제
       2. 여러 개의 값이 있는 경우 가장 앞에 있는 하나만 삭제

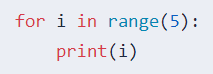
Part 7. for 반복문

1. for in list
   * for in 반복문
     + 1. 코드를 필요한만큼 반복해서 실행하는 함수

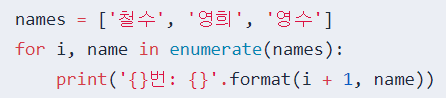


* + - 1. 리스트 patterns의 값을 하나씩 꺼내 pattern으로 전달
      2. 리스트의 길이만큼 print(pattern) 실행

1. for in range
   * range 함수
     + 1. 필요한 만큼의 숫자를 만들어내는 기능



* + enumerate
    - 1. 리스트가 있는 경우 순서와 리스트의 값을 전달하는 기능



* + for in list => 순회할 리스트가 정해져 있을 때
  + for in range() => 순회할 횟수가 정해져 있거나 1씩 증가하는 값이 필요할 때

Part 8. 모듈

1. 모듈 사용하기
   * 모듈
     1. 미리 만들어진 코드를 가져와 쓰는 방법
     2. 모듈은 import를 이용해서 가져올 수 있음 – import 모듈이름
     3. 사용 방법 : <모듈이름>.<모듈 안의 구성요소>
        1. 예) math.pi, random.choice()
   * 모듈의 예
     1. import math
        1. 수학과 관련된 기능
     2. import random
        1. 무작위와 관련된 기능
     3. import urllib.request
        1. 인터넷의 내용을 가져오는 기능
2. 모듈 만들기
   * 사용할 함수, 메소드 코드를 작성한 모듈 파일 생성
   * 사용 방법은 기존 모듈과 동일 - 모듈이 쓰일 파일에 import를 사용하여 모듈 호출
   * 사용자가 만든 모듈과 모듈을 쓸 파일이 같은 폴더에 있어야 함

Part 9. 활용하기

1. 검색하기
   * 검색은 구글로
   * 파이썬에 대한 검색을 할 때는 키워드에 파이썬3 또는 python3를 포함할 것
   * 코드를 포함한 글을 찾아야 함
   * 간단한 예제는 REPL을 사용하여 테스트 해 볼 것
   * 웬만하면 영어로 검색할 것
2. 문서 찾기
   * 공식문서
     + 1. 필요한 내용을 둘러보고 싶을 때
       2. 파이썬 내장 모듈과 함수의 정보가 필요할 때
   * 구글 또는 stackoverflow.com
     + 1. 문제의 구체적인 해결 방법이 알고 싶을 때
       2. 구글 검색 시 사이트 제한 기능 활용 (stackoverflow.com)