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

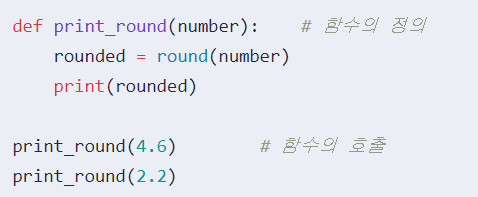
1. 조건식
   * 숫자 비교
     1. 크다
        1. 0 < 10
        2. 10 > 11
     2. 크거나 같다, 작거나 같다 (꼭 부등호 뒤에 등호를 붙여야 함)
        1. 3 <= 10
        2. 15 >= 10
     3. 같다
        1. 5 == 5
     4. 같지않다
        1. 5 != 5

* 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

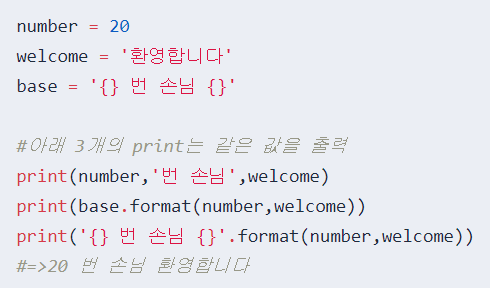
1. 함수 만들기
   * def 함수 이름():
     + 1. def : 새 함수를 정의하는 선언
2. 매개변수
   * 매개변수
     + 1. 함수를 정의할 때 사용하는 이름
   * 실행 인자
     + 1. 함수를 실행할 때 넘기는 변수, 값
   * 매개변수와 실행 인자
     + 1. 매개변수와 실행 인자의 개수는 동일해야 함
       2. 여러 개일 경우 쉼표로 구분



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



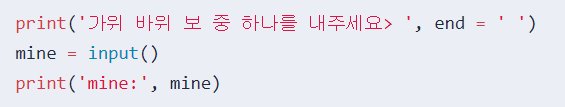
* + 실수
    1. 부동소수점이라는 표현법을 이용해 소수점을 표시할 수 있는 숫자
    2. 어느 정도의 계산 정확도는 가지지만, 계산에 있어서 완벽한 정확성은 가지지 않는다.



* + 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

