**[ 실습 1 ]**

**1. 파일명 : funcLab11.py**

**2. 구현해야 하는 함수 사양**

**함수명 : mydict**

**매개변수 : 가변 키워드형(키=값 형식으로 전달받을 수 있는 아규먼트 개수에 제한이 없다.)**

**리턴값 : 1개**

**기능 : 아규먼트는 키=값 형식으로 전달되며 몇 개가 전달되든 처리해야 한다.**

**아규먼트가 한 개도 전달되지 않으면 비어있는 딕셔너리를 리턴한다.**

**비어있는 딕셔너리를 생성한 다음 아규먼트로 전달된 키=값 쌍을 저장하는데**

**키는 앞에 my 를 붙여서 저장한다.**

**생성된 딕셔너리를 리턴한다.**

**3. 다양한 구성으로 키워드 아규먼트를 전달하면서 mydic() 함수를 호출하고 리턴 결과를**

**화면에 출력한다.**

**[ 실습 2 ]**

**1. 파일명 : funcLab12.py**

**2. 구현해야 하는 함수 사양**

**함수명 : myprint**

**매개변수 : 가변 아규먼트1개, 가변 키워드 아규먼트 1개**

**리턴값 : 없음**

**기능 : 전달되는 아규먼트의 개수에는 제한이 없다.**

**호출시 전달되는 아규먼트의 데이터 타입에도 제한이 없다.**

**아규먼트가 전달되지 않으면 “Hello Python”을 출력한다.**

**출력 형식은 다음에 제시된 실행 결과 예시를 보고 처리하는데 한 번의 print() 함수로 처리한다.**

**myprint(10, 20, 30, deco="@", sep="-") 호출시**

**@10-20-30@ 출력**

**myprint("python", "javascript", "R", deco="$") 호출시**

**$python,javascript,R$ 출력**

**myprint("가", "나", "다") 호출시**

**\*\*가,나,다\*\* 출력**

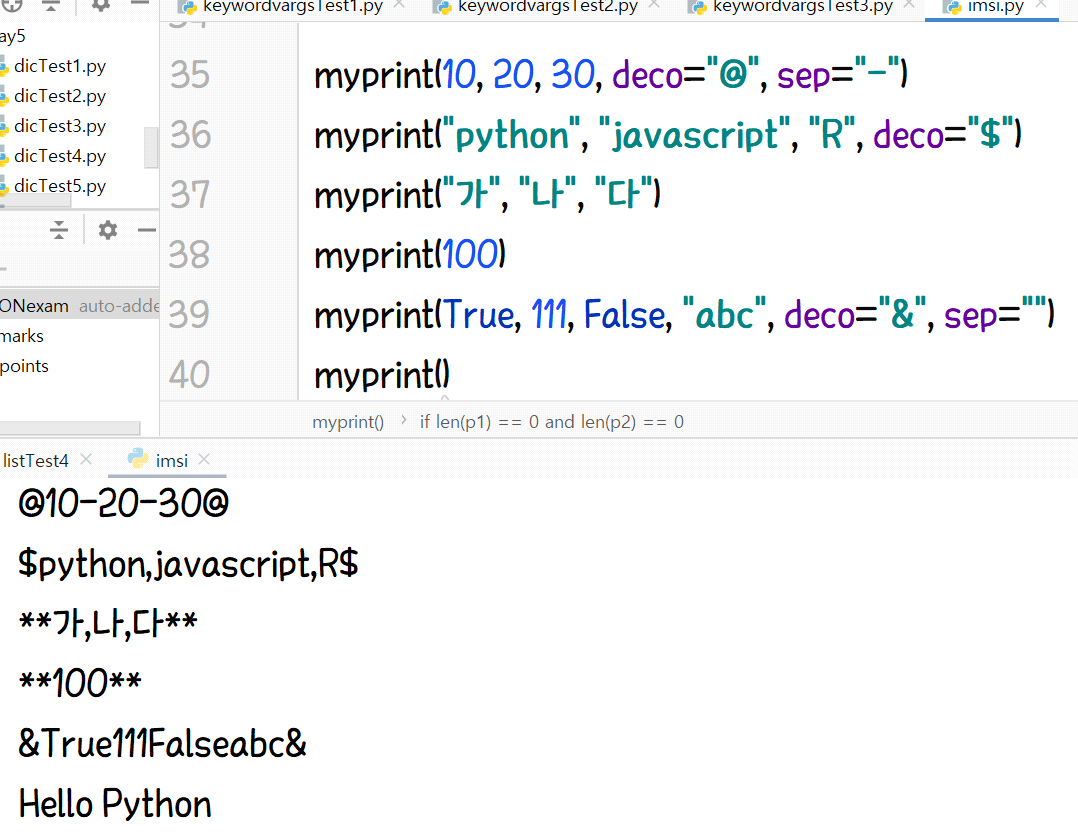
**myprint(100) 호출시**

**\*\*100\*\* 출력**

**myprint(True, 111, False, "abc", deco="&", sep="") 호출시**

**&True111Falseabc& 출력**

**3. 위에 제시된 호출식들을 가지고 호출했을 때 제시된 결과가 출력되면 완성이다.**



**[ 실습 3 ]**

**1. 파일명 : compreLab1.py**

**2. 구현해야 하는 함수 사양**

**함수명 : createList**

**매개변수 : 0개 이상의 위치 아규먼트를 받는 매개변수 1개 --> 가변인수**

**기본값이 있는 매개변수 1개(매개변수 명은 type이며 기본값은 1이다.)**

**리턴값 : 1개**

**기능 :**

**0개 이상의 위치 아규먼트를 가지고 아래에 제시된 타입에 따른 리스트를 생성하여**

**리턴한다.**

**위치 아규먼트가 전달되지 않은 경우에는 1부터 30의 값을 가지고 type에**

**따른 리스트를 생성하여 리턴한다.**

**type 이 2이면**

**데이터 값들에서 짝수에 해당하는 데이터들만을 가지고 리스트 생성**

**type 이 3이면**

**데이터 값들에서 홀수에 해당하는 데이터들만을 가지고 리스트 생성**

**type 이 4이면**

**데이터 값들에서 10보다 큰 데이터들만을 가지고 리스트 생성**

**type 이 1이면**

**데이터 값들을 모두 가지고 리스트 생성**

**리스트 생성은 리스트 컴프리핸션(지능형 리스트) 구문을 사용한다.**

**3. 다양한 구성으로 아규먼트를 전달하면서 createList() 함수를 호출하고 리턴 결과를**

**화면에 출력한다.**

**[ 실습4 ]**

**1. 파일명 : compreLab2.py**

**2. 구현해야 하는 함수 사양**

**함수명 : mycompredict**

**매개변수 : 가변 키워드형(0 개 이상의 키=값 형식의 아규먼트를 받아서 처리한다.)**

**리턴값 : 1개**

**기능 : funcLab11.py 에서 구현한 mydict() 라는 함수의 기능과 동일하게 구현하는데**

**이번에는 딕셔너리 컴프리핸션(지능형 딕셔너리) 구문을 사용해서 생성한다.**

**3. 다양한 구성으로 가변 키워드 아규먼트를 전달하면서 mycompredict() 함수를 호출하고 리턴**

**결과를 화면에 출력한다.**

**[ 실습 5 ]**

**1. 파일명 : strLab1.py**

**2. 구현해야 하는 함수 사양**

**함수명 : myemail\_info**

**매개변수 : 이메일 주소 문자열 1개**

**리턴값 : 2개의 원소를 갖는 튜플**

**기능 : 전달된 이메일 주소 문자열에서 @ 를 기준으로 쪼개서 튜플을 만들어 리턴한다.**

**단, 전달된 문자열에 @가 없으면 None을 리턴한다.**

**3. 아규먼트를 전달하면서 myemail\_info() 함수를 여러 번 호출하고 리턴 결과를**

**화면에 출력한다.**

**[ 실습 6 ] 문자열 관련 실습**

**소스명 : strLab2.py**

**(1) s1 = "pythonjavascript" 에서 의 길이를 출력한다.**

**(2) s2 = "010-7777-9999" 에서 01077779999 을 출력한다.**

**(3) s3 = "PYTHON" 에서 NOHTYP 를 출력한다.**

**(4) s4 = "Python" 에서 python 을 출력한다.**

**(5) s5 = "**https://www.python.**org/" 에서 www.python.org 만을 뽑아서 출력한다.**

**(6) 주민등록 번호에서 7자리 숫자를 사용해서 남성, 여성을 판별한다. (1,3 : 남성, 2,4 : 여성)**

**s6 = '891022-2473837'**

**(7) s7 = '100' 에서 s7 값을 정수형 숫자로 그리고 실수형 숫자로 변환하여 출력한다.**

**(8) s8 = '** **The Zen of Python' 에서 '** **n'** **문자가 몇 개인지 출력한다.**

**(9) s8 에서 '** **Z'**  **의 위치를 출력한다.**

**(10) s8 에서 모두 대문자로 변환한 후 공백단위로 분리해서 리스트로 만들어 출력한다.**

**[ 실습 7 ]**

**파일명 : compreLab3.py**

**아래 리스트 항목 중에서 소문자만 추출해서 리스트를 생성하여 listv2에 저장하고 출력한다.(리스트 컴프리헨션 사용)**

**listv1 = ["A", "b", "c", "D", "e", "F", "G", "h"]**

**[ 실습 8 ]**

**파일명 : packunpacLab.py**

**다음 리스트를 생성하고**

**listv3 = [ 'p', 'y', 't', 'h', 'o', 'n' ]**

**(1) v1, v2, v3, v4, v5,v6 에 언 패킹해서 저장한 후에 각 변수의 값을 행 단위로 화면에 출력한**

**다.**

**(2) listv3 를 언패킹하여 print() 함수에 전달하여 출력한다. **

**(3) listv3 를 그냥 print() 함수에 전달하여 출력한다. **

**[ 실습 9 ]**

**파일명 : compreLab4.py**

**컴프리핸션 구문을 사용해서 다음에 제시된 데이터들로 구성되는 자료구조를 생성한다.**

**(1) 난수 추출 함수를 사용하여 0 부터 100 사이의 값 10개로 구성되는 리스트를 하나 생성한다.**

****

**(2) 위에서 생성된 리스트를 이용하여 다음과 같이 구성되는 딕셔너리를 생성한다.(추출된 점수가 60점 이상이면 pass, 60점 미만이면 nopass 로 처리한다.)**

****

**모든 소스를 첨부파일로 추가하여 메일로 제출한다. 이 때 메일 제목은 “7월14일실습1-XXX” 로 작성하여 전송한다.**