

## Web/Python Programming 01

### Mid-term exam

Professor: Sangkeun Park

- Name your Python file, \_StudentNo.py (for example, 2023123456), otherwise -1P.
- Remove or comment all ***print*** functions, otherwise -1P.
- Do not use the ***input*** function, otherwise -1P.
- If I fail to import your submission by an error in the code, you get -1P.
- Be careful of typos.

str Method	Description
<a href="#">isalnum()</a>	Returns True if all characters in the string are alphanumeric
<a href="#">isalpha()</a>	Returns True if all characters in the string are in the alphabet
<a href="#">isdigit()</a>	Returns True if all characters in the string are digits
<a href="#">islower()</a>	Returns True if all characters in the string are lower case
<a href="#">isupper()</a>	Returns True if all characters in the string are upper case
<a href="#">lower()</a>	Converts a string into lower case
<a href="#">upper()</a>	Converts a string into upper case
<a href="#">replace()</a>	Returns a string where a specified value is replaced with a specified value
<a href="#">split()</a>	Splits the string at the specified separator, and returns a list
<a href="#">strip()</a>	Returns a trimmed version of the string

### Define the following 10 functions

#### function1 (1 point)

- The function name: **function1**
- This function takes three positional arguments
  - First argument: **int type**
  - Second argument: **int type**
  - Third argument: **int type**
- The return value: **int type**
  - Sum of the three arguments
- This function will be tested with positional arguments, for example:
  - `print(function1(1, 4, 5))` # 10 (because 1+4+5 is 10)
  - `print(function1(0, 3, -9))` # -6 (because 0+3+(-9) is -6)
  - `print(function1(-3, -2, 5))` # 0 (because (-3)+(-2)+5 is 0)
  - `print(function1(0, 999, 1))` # 1000 (because 0+999+1 is 1000)

### function2 (1 point)

- The function name: **function2**
- This function takes one positional argument: **list type (all elements are int types)**
- The return value: **int type**
  - Sum of elements which are odd numbers, such as -3, -1, 1, 3, 5, 7 ....
  - If the given argument has no odd numbers, return 0.
- This function will be tested with positional arguments, for example:
  - `print(function2([1, 2, 3]))` # 4 (because 1+3 is 4)
  - `print(function2([ ]))` # 0 (because there are no odd numbers to sum)
  - `print(function2([1, 2, 3, 4]))` # 4 (because 1+3 is 4)
  - `print(function2([-3, 1, 3, 2, 100]))` # 1 (because (-3)+1+3 is 1)

### function3 (1 point)

- The function name: **function3**
- This function takes one positional argument: **list type (all elements are int types)**
  - There is no duplicated number.
- The return value: **int type**
  - The second largest number from the given list.
  - If the given argument is an empty list, return 0.
  - If the given argument has only one element, return the element.
- This function will be tested with positional arguments, for example:
  - `print(function3([ ]))` # 0 (because there is no element.)
  - `print(function3([-99]))` # -99 (because there is only 99 in the given list.)
  - `print(function3([3, 1, 2, 0]))` # 2 (because 2 is the second largest number.)
  - `print(function3([99, 2, -1, 100]))` # 99 (because 2 is the second largest number.)
  - `print(function3([-1, -3, -2, 0]))` # -1 (because -1 is the second largest number.)

### function4 (1 points)

- The function name: **function4**
- This function takes one positional argument: **str type**
  - The length of the argument is greater than 2.
  - All the characters are alphabet letters.
- The return value: **str type**
  - A string in which only the last letter is upper-case and the rest is lower-case.
- This function will be tested with positional arguments, for example:
  - `print(function4("Hello"))` # hello
  - `print(function4("PyTHON"))` # pythoN
  - `print(function4("abc"))` # abC
  - `print(function4("ClasS"))` # clasS

### function5 (1 points)

- The function name: **function5**
- This function takes one positional arguments: **str type**
  - The argument includes at least one numeric letter.
  - All letters are alphabetic or numeric.
- The return value: **str type**
  - The average of all digit letters from the given argument **as str type**
- This function will be tested with positional arguments, for example:
  - `print(function5("todayis0425"))` # 2.75 (because  $((0+4+2+5) / 4)$  is 2.75)
  - `print(function5("h2a0p2p3y"))` # 1.75 (because  $((2+0+2+3) / 4)$  is 1.75)
  - `print(function5("A1B2C3"))` # 2.0 (because  $((1+2+3) / 3)$  is 2.0)
  - `print(function5("pyt0hon"))` # 0.0 (because  $0/1$  is 0)

### function6 (1 point)

- The function name: **function6**
- This function takes two positional arguments
  - The first argument: **int type** (greater than 1 and smaller than 1000)
  - The second argument: **int type** (greater than 1 and smaller than 1000)
- The return value: **set type**
  - All common factors that divide the two given arguments leaving no remainder.
  - For example,
    - The factors of 6 are 1, 2, 3, and 6. The factors of 10 are 1, 2, 5, and 10.
    - Then, the common factors of the two numbers are {1, 2}.
- This function will be tested with positional arguments, for example:
  - `print(function6(6, 10))` # {1, 2}
  - `print(function6(2, 3))` # {1}
  - `print(function6(24, 12))` # {1, 2, 3, 4, 6, 12}
  - `print(function6(11, 121))` # {1, 11}

### function7 (1 points)

- The function name: **function7**
- This function takes one positional argument: **dict type**
  - The given dict has at least one key-value pair.
  - All **key** types are **str types** (only lowercase alphametic letters).
  - All **value** types are **int types** that are equal to or greater than 0.
- The return value: **int type**
  - The sum of the values that keys are not "math".
- This function will be tested with positional arguments, for example:
  - `print(function7({"math": 90, "english": 25, "korean": 30}))` # 55 (because  $25+30$ )
  - `print(function7({"math": 80, "ai": 38, "python": 80}))` # 118 (because  $38+80$ )
  - `print(function7({"korean": 90, "python": 73}))` # 163 (because  $90+73$ )
  - `print(function7({"math": 0, "computer": 20, "ai": 99}))` # 119 (because  $20+99$ )
  - `print(function7({"python": 88, "statistics": 12}))` # 100 (because  $88+12$ )

### function8 (1 points)

- The function name: **function8**
- This function takes one positional argument: **list type (all elements are int types)**
  - There is at least one element in the given list.
  - Given the list of integers, only one number appears once, but the other numbers appear twice.
- The return value: **int type**
  - Find the number that appears once. (The number that appears once must exist.)
- This function will be tested with positional arguments, for example:
  - `print(function8([4,1,2,1,2]))` # 4 (because 4 appears once)
  - `print(function8([1]))` # 1 (because 1 appears once)
  - `print(function8([2, 2, -1]))` # -1 (because -1 appears once)
  - `print(function8([-1, 1, -1, 99, 1, 2, 2]))` # 99 (because 99 appears once)

### function9 (1 points)

- The function name: **function9**
- This function takes two or three positional argument: all arguments are **int types**
- The return value: **int type**
  - If two arguments are given, return the sum of the two arguments+10.
  - If three arguments are given, return the sum of the three arguments.
- This function will be tested with positional arguments, for example:
  - `print(function9(1, 2))` # 13 (because 1+2+10 is 13)
  - `print(function9(1, 2, 3))` # 6 (because 1+2+3 is 6)
  - `print(function9(-9, -1))` # 0 (because ((-9)+(-1)+10 is 0)
  - `print(function9(10, 20, -30))` # 0 (because 10+20+(-30) is 0)

### function10 (1 points)

- The function name: **function10**
- This function takes the unknown number of positional or keyword arguments.
  - All arguments' values are **int types**.
- The return value: **int type**
  - (Sum of all keyword arguments' values) - (Sum of all positional arguments)
  - If there are no positional arguments, the sum of all positional arguments is 0.
  - If there are no keyword arguments, the sum of all keyword arguments is 0.
- This function will be tested with positional arguments, for example:
  - `print(function10(1, 2, a=10, b=7))` # 14 (because (10+7) - (1+2))
  - `print(function10(-1, n=10, k=20))` # 31 (because (10+20) - (-1))
  - `print(function10(age=20, height=175))` # 195 (because (20+175) - (0))
  - `print(function10(1, 2, 3, 4))` # -10 (because (0) - (1+2+3+4))
  - `print(function10())` # 0 (because (0) - (0))

\*After submitting the file, you can study another exam using paper-based materials. Please let the professor or TAs know you're done.

\*Instead, you cannot use digital devices in the class anymore.

## 웹/파이썬프로그래밍 01 중간고사

담당 교수: 박상근

- 밑줄학번.py 형식으로 파일을 생성하세요 (for example, 2023123456), otherwise -1P.
- 제출 전에 print 함수 삭제 또는 주석처리하세요, otherwise -1P.
- input 함수 사용하지 마세요, otherwise -1P.
- 해당 파일을 import 할 때 에러가 발생하면, you get -1P.
- 오타 조심!

str Method	Description
<a href="#">isalnum()</a>	Returns True if all characters in the string are alphanumeric
<a href="#">isalpha()</a>	Returns True if all characters in the string are in the alphabet
<a href="#">isdigit()</a>	Returns True if all characters in the string are digits
<a href="#">islower()</a>	Returns True if all characters in the string are lower case
<a href="#">isupper()</a>	Returns True if all characters in the string are upper case
<a href="#">lower()</a>	Converts a string into lower case
<a href="#">upper()</a>	Converts a string into upper case
<a href="#">replace()</a>	Returns a string where a specified value is replaced with a specified value
<a href="#">split()</a>	Splits the string at the specified separator, and returns a list
<a href="#">strip()</a>	Returns a trimmed version of the string

### Define the following 10 functions

#### function1 (1 point)

- The function name: **function1**
- This function takes three positional arguments
  - First argument: **int type**
  - Second argument: **int type**
  - Third argument: **int type**
- The return value: **int type**
  - 세 인자의 합
- This function will be tested with positional arguments, for example:
  - `print(function1(1, 4, 5))` # 10 (because 1+4+5 is 10)
  - `print(function1(0, 3, -9))` # -6 (because 0+3+(-9) is -6)
  - `print(function1(-3, -2, 5))` # 0 (because (-3)+(-2)+5 is 0)
  - `print(function1(0, 999, 1))` # 1000 (because 0+999+1 is 1000)

### function2 (1 point)

- The function name: **function2**
- This function takes one positional argument: **list type (all elements are int types)**
- The return value: **int type**
  - 주어진 list 에서 모든 홀수의 합.
  - 리스트에 홀수가 없다면, return 0.
- This function will be tested with positional arguments, for example:
  - `print(function2([1, 2, 3]))` # 4 (because 1+3 is 4)
  - `print(function2([ ]))` # 0 (because there are no odd numbers to sum)
  - `print(function2([1, 2, 3, 4]))` # 4 (because 1+3 is 4)
  - `print(function2([-3, 1, 3, 2, 100]))` # 1 (because (-3)+1+3 is 1)

### function3 (1 point)

- The function name: **function3**
- This function takes one positional argument: **list type (all elements are int types)**
  - list 안에 중복되는 숫자는 없음
- The return value: **int type**
  - 주어진 list 에서 두 번째로 큰 수.
  - 만약 빈 list 가 주어지면, return 0.
  - 만약 요소가 1개 뿐인 list 가 주어지면, 해당 요소를 return.
- This function will be tested with positional arguments, for example:
  - `print(function3([ ]))` # 0 (because there is no element.)
  - `print(function3([-99]))` # -99 (because there is only 99 in the given list.)
  - `print(function3([3, 1, 2, 0]))` # 2 (because 2 is the second largest number.)
  - `print(function3([99, 2, -1, 100]))` # 99 (because 2 is the second largest number.)
  - `print(function3([-1, -3, -2, 0]))` # -1 (because -1 is the second largest number.)

### function4 (1 points)

- The function name: **function4**
- This function takes one positional argument: **str type**
  - 문자열의 길이는 무조건 3글자 이상
  - 문자열은 무조건 알파벳으로만 구성되어있음
- The return value: **str type**
  - 마지막 글자만 대문자, 나머지는 모두 소문자로 변환된 문자열
- This function will be tested with positional arguments, for example:
  - `print(function4("Hello"))` # hello
  - `print(function4("PyTHON"))` # pythoN
  - `print(function4("abc"))` # abC
  - `print(function4("ClasS"))` # clasS

### function5 (1 points)

- The function name: **function5**
- This function takes one positional arguments: **str type**
  - 해당 문자열에는 최소 1개 이상의 숫자형 문자가 존재.
  - 해당 문자열은 알파벳 또는 숫자형 문자로만 구성됨.
- The return value: **str type**
  - The average of all digit letters from the given argument **as str type**
- This function will be tested with positional arguments, for example:
  - `print(function5("todayis0425"))` # 2.75 (because  $((0+4+2+5) / 4)$  is 2.75)
  - `print(function5("h2a0p2p3y"))` # 1.75 (because  $((2+0+2+3) / 4)$  is 1.75)
  - `print(function5("A1B2C3"))` # 2.0 (because  $((1+2+3) / 3)$  is 2.0)
  - `print(function5("pyt0hon"))` # 0.0 (because  $0/1$  is 0)

### function6 (1 point)

- The function name: **function6**
- This function takes two positional arguments
  - The first argument: **int type** (1보다 크고 1000보다 작음)
  - The second argument: **int type** (1보다 크고 1000보다 작음)
- The return value: **set type**
  - 두 인자(arguments)의 공통 인수를 반환.
  - 예를 들어,
    - 6의 인수는 1, 2, 3, 6. 10의 인수는 1, 2, 5, 10.
    - 그러므로, 두 인자의 공통 인수는 {1, 2}.
- This function will be tested with positional arguments, for example:
  - `print(function6(6, 10))` # {1, 2}
  - `print(function6(2, 3))` # {1}
  - `print(function6(24, 12))` # {1, 2, 3, 4, 6, 12}
  - `print(function6(11, 121))` # {1, 11}

### function7 (1 points)

- The function name: **function7**
- This function takes one positional argument: **dict type**
  - 주어진 사전형 값에는 최소 1개 이상의 키-값 쌍이 들어있음
  - 모든 key는 **str** 타입 (무조건 소문자 알파벳임).
  - 모든 value는 0 이상의 **int** 타입.
- The return value: **int type**
  - key 가 "math"인 키-값 쌍을 제외하고 남은, 키-값 쌍에서 모든 값(value)의 합
- This function will be tested with positional arguments, for example:
  - `print(function7({"math": 90, "english": 25, "korean": 30}))` # 55 (because  $25+30$ )
  - `print(function7({"math": 80, "ai": 38, "python": 80}))` # 118 (because  $38+80$ )
  - `print(function7({"korean": 90, "python": 73}))` # 163 (because  $90+73$ )
  - `print(function7({"math": 0, "computer": 20, "ai": 99}))` # 119 (because  $20+99$ )
  - `print(function7({"python": 88, "statistics": 12}))` # 100 (because  $88+12$ )

### function8 (1 points)

- The function name: **function8**
- This function takes one positional argument: **list type (all elements are int types)**
  - 주어진 리스트에는 무조건 1개 이상의 int 타입 값이 존재.
  - 1개만 존재하는 숫자가 하나 있으며, 나머지 숫자들은 아예 없거나 2개씩 존재.
- The return value: **int type**
  - 딱 한 개만 존재하는 숫자. (딱 한 개만 존재하는 숫자가 무조건 있음.)
- This function will be tested with positional arguments, for example:
  - `print(function8([4,1,2,1,2]))` # 4 (because 4 appears once)
  - `print(function8([1]))` # 1 (because 1 appears once)
  - `print(function8([2, 2, -1]))` # -1 (because -1 appears once)
  - `print(function8([-1, 1, -1, 99, 1, 2, 2]))` # 99 (because 99 appears once)

### function9 (1 points)

- The function name: **function9**
- 이 함수는 2개 또는 3개의 positional 인자를 받음: 모든 인자는 **int** 타입
- The return value: **int type**
  - 인자가 2개만 넘어오면, 그 2개의 인자의 합에 10을 더해서 return.
  - 인자가 3개 넘어오면, 그 3개의 인자의 합을 return.
- This function will be tested with positional arguments, for example:
  - `print(function9(1, 2))` # 13 (because  $1+2+10$  is 13)
  - `print(function9(1, 2, 3))` # 6 (because  $1+2+3$  is 6)
  - `print(function9(-9, -1))` # 0 (because  $((-9)+(-1)+10)$  is 0)
  - `print(function9(10, 20, -30))` # 0 (because  $10+20+(-30)$  is 0)

### function10 (1 points)

- The function name: **function10**
- 이 함수는 positional 또는 keyword 인자를 받지만 그 인자 개수는 정해져있지 않음
  - 모든 positional 인자는 **int** 타입, 모든 keyword 인자의 value 값도 **int** 타입임
- The return value: **int type**
  - (모든 keyword 인자의 values의 합) - (모든 positional 인자의 합)
  - positional 인자가 없다면, positional 인자의 합은 0.
  - keyword 인자가 없다면, keyword 인자의 합은 0.
- This function will be tested with positional arguments, for example:
  - `print(function10(1, 2, a=10, b=7))` # 14 (because  $(10+7) - (1+2)$ )
  - `print(function10(-1, n=10, k=20))` # 31 (because  $(10+20) - (-1)$ )
  - `print(function10(age=20, height=175))` # 195 (because  $(20+175) - (0)$ )
  - `print(function10(1, 2, 3, 4))` # -10 (because  $(0) - (1+2+3+4)$ )
  - `print(function10())` # 0 (because  $(0) - (0)$ )

- 시험 파일을 제출 하고, 본인이 스스로 시험을 더 칠 필요가 없다고 생각되면 조교 및 교수에게 말한 후 종이로 된 시험 자료로 다른 과목 시험을 공부할 수 있습니다.
- 단, 노트북 및 스마트폰은 절대 사용할 수 없으니 가방에 다 넣어야 합니다.