

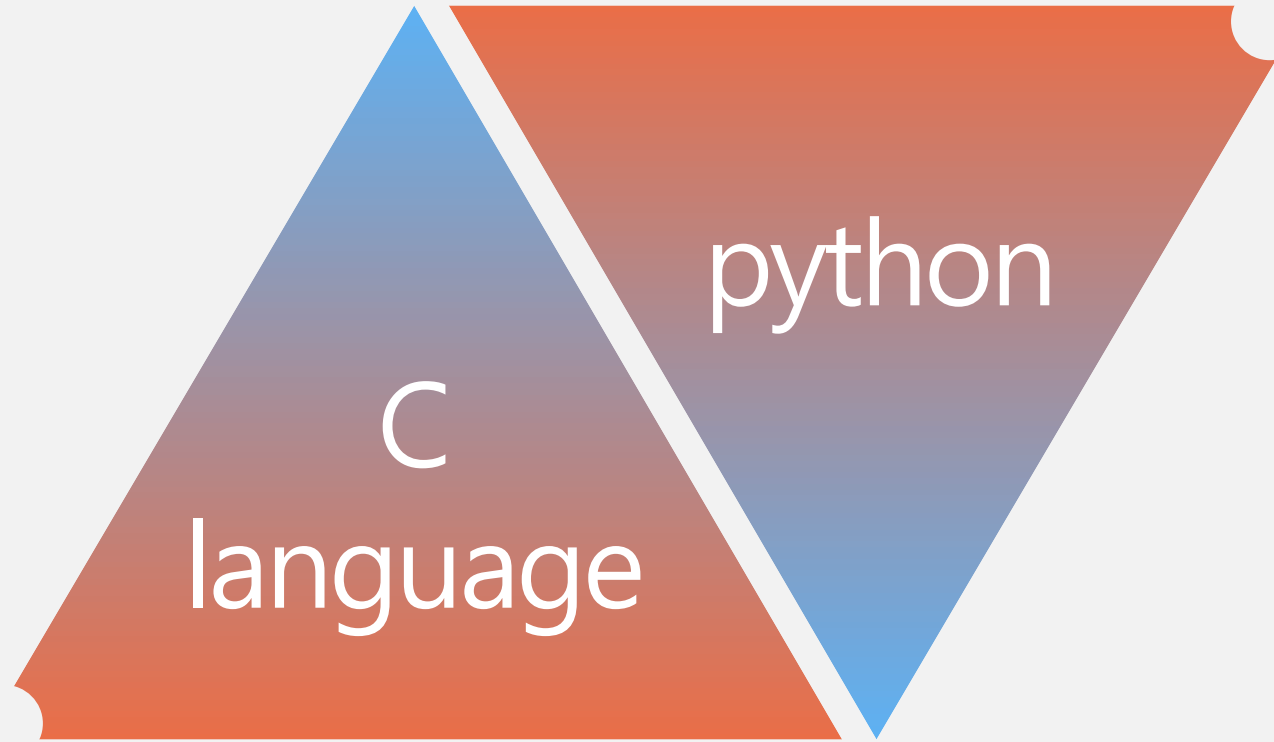


소입설
멘토링 수업
(6주차)

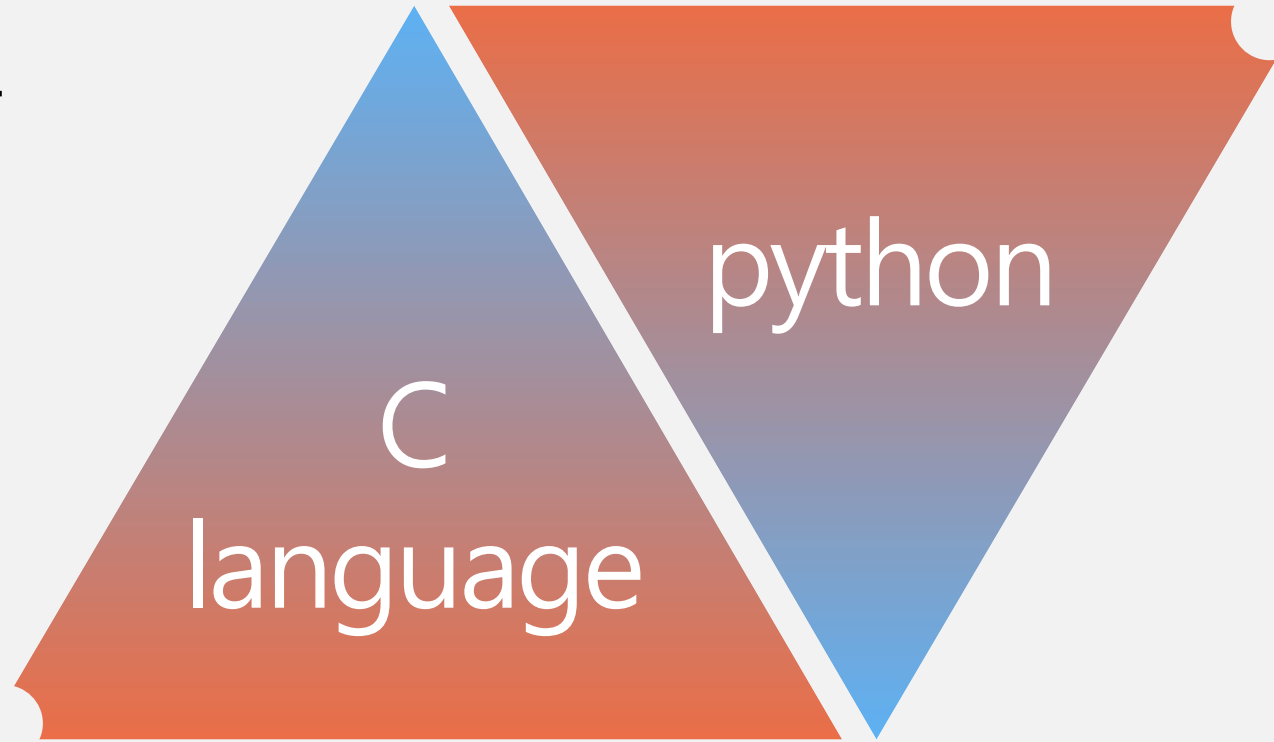


- 1 지난 주 수업 복습
 - 2 소프트웨어 입문 설계 정리
 - 3 Quiz
-

1. 지난주 수업 복습



2. 소프트웨어 입문 설계 정리



중간 복습

1. operators, value, expression, statement
 2. data types (numbers, String, List, Tuple, Dictionary) and variables (local, global), index
 3. print, comma (,), raw_input(), casting function
 4. comment
 5. function (definition, call, parentheses, argument, return)
 6. while statement (brk, cnt), if statement (else), for statement
-

중간 복습

- 7. blocks and colon(:)
 - 8. condition and Boolean (operators, truth table, short circuit)
 - 9. import module (random, time)
 - 10. escape characters (5 things)
 - 11. quotes, double quotes
 - 12. string
 - 13. list and index
-

1. operator, value, expression, statement (연산자, 값, 표현식, -문)

- a. Assignment operator(대입 연산자) # in numbers data type !!!
Ex) = , += , -= , *= , /= , %=
 - value(값)
ex) 7, 0.0 , "hello world" , True
- b. Arithmetic operator(산술 연산자)
Ex) + , - , * , / , %
 - expression(표현식): Any value contained in that statement
ex) 3+2 , "hello world"
- c. Sign operator(부호 연산자)
Ex) + , -
 - statement(-문): Anything that does something
ex) print 'hello' , foo="hello"
- d. Comparison operator(비교 연산자)
Ex) < , > , <= , >= , == , !=
- e. Logical operator(논리 연산자)
Ex) and, or, not

2. data types and variables (자료형, 변수)

- 1) numbers data type
 - boolean(bool,불린): True, False
 - integer(int,정수): 10, 3, -7, 0
 - floating point number(float,실수): 10.5, 3.0, -7.8, 0.0
- 2) string(문자열)
 - "Hello World"
 - immutable, ordered
- 3) list(배열)
 - [1, 2, 3]
 - mutable, ordered
- 4) tuple(튜플)
 - (1, 2, 3)
 - immutable, ordered
- 5) dictionary (사전)
 - Index를 정해줄 수 있는 일종의 list
 - element → key : value 로 구성됨
 - { 1: '1', 2: '2', 3: '3' } → dic[1]=?, dic['1']=?
 - { '1': 1, '2':2, "3":3 } → dic[1]=?, dic['1']=?
 - mutable, unordered

variable

1) global

2) local

→ 코드 3개 비교

3. print, comma (,), raw_input(), casting function (출력함수, 쉼표, 입력함수, 형변환함수)

1) print

- 출력
- print "Hello World"
- print "Hello World",
- print "Name: %s , age: %d" %(string, integer)
- print "Name: %10s" %(string)
- print "Name: %-10s" %(string)

2) raw_input()

- 입력
- string을 반환
- name=raw_input()
- name=raw_input("Type your name: ")

3) casting function

- str(123)
 - int(" 123 ")
-

4. Comment (주석)

Copyright holder: HM

이 함수는 보안이 강화된 표준 입력 함수입니다.

5. Function (함수)

- 1) function definition (함수 정의): 함수 밖에서
 - `def [function name] (parameter 매개변수) :`
 `[block: function]`
 `return [return value]`
- 2) function call (함수 호출): 다른 함수 내에서 또는 main전역범위 내에서
 - `[function name](variable)`
 - `[function name](value)` ➔ `[function name](argument 인자)`
 - `[function name]()`
- 3) 함수의 인자는 여러 개일 수 있지만, 반환 값은 무조건 1개 이하이다.

6. while statement, if statement, for statement (반복문, 조건문)

1) while statement (loop)

- while [condition] :
 [block]
- condition==True → block을 1번 실행하고 다시 condition 으로 돌아옴
- condition==False → 곧바로 while문을 나감: block 아래줄로 이동
- continue: condition 으로 돌아가는 키워드
- break: while문을 나가는 키워드

2) if statement

- if [condition] :
 [block]
- condition==True → block을 1번 실행하고 if문을 나감
- condition==False → 곧바로 if문을 나가거나 else block으로 이동
- else: condition 이 False일 때 else block을 실행
- elif == else if

3) for statement (loop)

- for [variable] in [sequence] :
 [block]
 - variable이 sequence 의 element들에 하나씩 대응하면서 block을 실행 **#sequence가 dictionary일 때!**
 - continue: for문의 맨 위로 이동한 후 다음 element에 대응되어 block을 실행하는 키워드
 - break: for문을 나가는 키워드
-

6. while statement, if statement, for statement (반복문, 조건문)

Q. 프로그램의 출력 결과를 적으시오. 단, 순서는 중요하지 않다.

```
dic = { '0' : 'zero' , "1" : "one" , '2' : "two" }
```

```
for x in dic :  
    print x
```

```
for x in dic :  
    print dic[x]
```

같이!)	'0' in dic	>>> True
	"two" in dic	>>> False

7. blocks and colon(:)

1) block: 같은 크기만큼 이어져서 들여쓰기 된 부분

2) colon:

① : 을 쓴 후 block을 형성

- 함수 정의
- while 문
- if 문
- else
- for 문

② dictionary

- {1: 'hello' , -1: 'bye' }

③ slicing of string, list and tuple

- "hello"[: 3] >>> 'hel'
- (1,2,3,4)[: 3] >>> (1,2,3)
- [1,2,3,4][: 3] >>> [1,2,3]

8. condition and Boolean (조건, 참거짓)

1) condition

- if statement, while statement

2) Boolean

- True, False
 - truth table (logical operator) → 그려보기
 - short circuit: True or X , False and X
 - in operator: 3 in [1,2,3] , "lo" in 'hello world', 1 in { '0': 0 , '1': 1 }
-

9. import module (random, time)

- 1) module(모듈): 여러 함수들을 포함하고 있는 다른 프로그램 또는 코드
 - 2) import [module]
 - 3) import [module] as [other]
 - 4) import random
 - randint(a,b): a, b 사이(a, b포함)의 정수 중 무작위로 하나를 반환 # end 포함 O
 - guessNumber=random.randint(1,30)
 - 5) import time
 - sleep(t): t초만큼 잠시 멈춤
 - time.sleep(2)
-

10. escape characters (확장 문자)

1) 본래 기능이 아닌 다른 기능을 갖는 문자

- \n
- \t
- \\
- \"
- \'

11. quotes, double quotes (작은 따옴표, 큰 따옴표)

- 1) 문자열을 표시할 때 사용
 - 2) 하나의 문자열 값을 나타낼 땐 quotes, double quotes 맞춰주어야 함
 - string = "hello" → ?
 - string = ' hello' → ?
 - 3) string: "hello", 'hello'
 - 4) print 'hello world'
 - 5) escape character를 이용해 "를 일반 문자처럼 사용
 - print "Hm said \" hi, bros\""
 - 6) string = ''' hello
bro
man
hommie'''
 - string >>> ' hello\nbro\nman\nhommie'
 - print string
 - >>> hello
 - >>> bro
 - >>> man
 - >>> hommie
-

12. string(1)

1) "hello world"

2) index

- []: index operator
- "hello world"[3] >>> 'l'
- "hello world"[100] >>> error
- "hello world"[-3] >>> 'r'
- "hello world"[-12] >>> error

3) slicing: 그 string을 잘라서 새로운 string을 반환 # **end 포함 X**

- "hello world"[3:8] >>> 'lo wo'
 - "hello world"[:8] >>> 'hello wo'
 - "hello world"[3:] >>> 'lo world'
 - "hello world"[300:412] >>> ''
 - "hello world"[-3:-1] >>> 'rl'
 - "hello world"[-1:-3] >>> ''
-

12. string(2)

1) + , * operator

- 'HM' + 'mento' >>> 'HMmento' [string] + [string]
- 'HM'*3 >>> 'HMHMHM' [string] * [integer]

2) len() function

- len("hello") >>> 5

3) method

- ① .split(): 그 string을 띄어쓰기 기준으로 잘라 여러 string elements를 갖는 list를 반환
 - ② .upper(): 그 string의 문자를 모두 대문자로 바꾼 string을 반환
 - ③ .lower(): 그 string의 문자를 모두 소문자로 바꾼 string을 반환
 - ④ .count(item): 그 string에서 item(string)과 같은 부분의 수(integer)를 반환
 - ⑤ .center(w): len(newstring)=w(integer), 그 string은 가운데(w==odd인 경우: 한칸오른쪽), 나머지 space인 newstring을 반환
 - ⑥ .ljust(w): len(newstring)=w, 그 string은 왼쪽, 나머지 space인 newstring을 반환
 - ⑦ .rjust(w): len(newstring)=w, 그 string은 오른쪽, 나머지 space인 newstring을 반환
 - ⑧ .index(item): 그 string에서 item과 겹치는 부분의 첫번째 index를 반환,
여러 개 겹치면 첫번째 겹치는 부분의 index를 반환, 없으면 error
 - ⑨ .find(item): 그 string에서 item과 겹치는 부분의 첫번째 index를 반환,
여러 개 겹치면 첫번째 겹치는 부분의 index를 반환, 없으면 -1를 반환
 - ⑩ .replace(old,new): 그 string에서 old(string)와 같은 부분을 new(string)로 대체한 string을 반환
 - ⑪ .startswith(item): 그 string이 item(string)으로 시작하면 True반환, 아니면 False 반환
-

13. list(1)

1) [1,2,3]

2) index

- []: index operator
- [0,1,2,3][2] >>> 2
- [0,1,2,3][5] >>> error
- [0,1,2,3][-1] >>> 3
- [0,1,2,3][-5] >>> error

3) slicing: 그 list를 잘라서 새로운 list를 반환 # **end 포함 X**

- [0,1,2,3][1:2] >>> [1]
 - [0,1,2,3][:2] >>> [0,1]
 - [0,1,2,3][1:] >>> [1,2,3]
 - [0,1,2,3][13:41] >>> []
 - [0,1,2,3][-3:-1] >>> [1,2]
 - [0,1,2,3][-1:-3] >>> []
-

13. list(2)

1) + , * operator

- [0,1,2,3] + [5,4]	>>> [0,1,2,3,5,4]	[list] + [list]
- [0,1,2] * 3	>>> [0,1,2,0,1,2,0,1,2]	[list] * [integer]

2) len() function

- len([0,1,2,3]) >>> 4

3) range() function

end 포함 X

- range(10)	>>> [0,1,2,3,4,5,6,7,8,9]
- range(4,10)	>>> [4,5,6,7,8,9]
- range(0,10,2)	>>> [0,2,4,6,8]

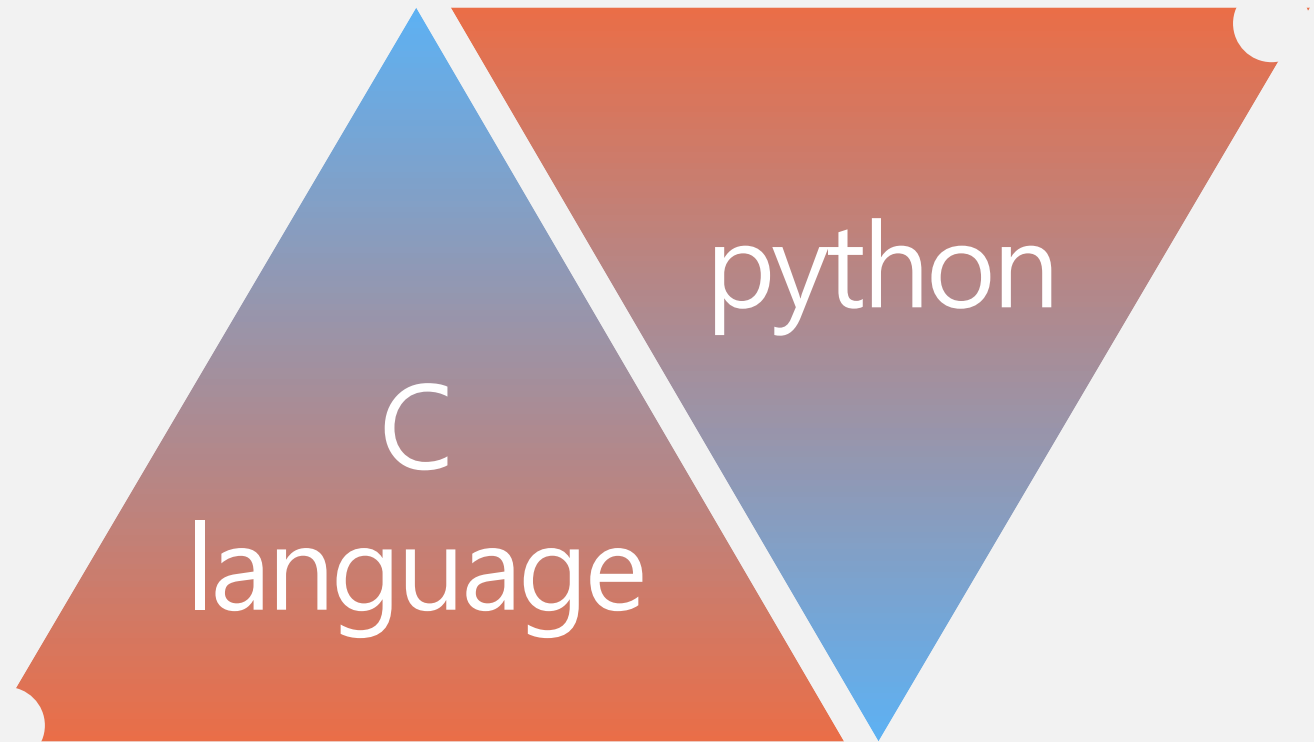
4) method

- ① .reverse(): 그 list의 순서를 뒤바꿈
- ② .append(w): list의 맨 끝에 w를 추가
- ③ .sort(): numbers는 크기순으로, string은 사전식으로 순서를 바꿔줌

5) del

- list의 element를 지움	
- lst=[0,1,2,3,4]	del lst[3] lst >>> [0,1,2,4]

3. Quiz





midterm2015.pdf



이주호_2016기출문제2(Pyth



midterm2016.pdf

복습
