

2021년 1학기

4

제어지능SW개발

openCV를 활용한 지능형시스템 개발

교수 도경민

컴퓨터전자공학과



4차시 내용

- 3주차 자기평가 질문
- 2주 과제 해결
- 함수
- 지역변수/전역변수
- List / Tuple / Set / Dictionary

- 2주차 질문사항들 중에 '2주차 과제 4번'문제에 대한 질문들에 보충하는 설명입니다.
4번 문제는
"입력된 알파벳 문자열의 첫 번째 알파벳과 마지막 알파벳 정중앙에 위치하는 알파벳을 출력하시오. 정중앙의 알파벳이 두 개면 두 개를 출력하시오." 입니다.
여기서 질문자들이 헷갈리는 것은, 예를 들어 'apple'이라는 단어를 입력 했을 때,
[1] 'apple'에서 문자열의 맨 앞'a' 맨 뒤'e' 이니 아스키코드로 계산했을 때 (a b c d e) 'a'와 'e' 사이의 가운데 값인 'c'를 출력한다.
[2] 'apple'에서 문자열의 가운데 값인 (ap'p'le), 'p'를 출력한다.
이렇게 두 가지로 갈리는 것 같습니다.
제가 문제를 읽고 처음 이해한 것은 [2]이어서 [2]방식으로 제출했고,
질문들을 보니, 저 말고도 2번 방식으로 제출 한 사람이 꽤 있다고 생각합니다.
그리고 3주차 1번째 강의에서 교수님께서 과제를 내주신 의도가 [1]방식이라고 이해했습니다.
하지만, 문제를 다시 봐도 충분히 [2]로 이해 할 만한 과제라고 생각합니다.
[2]방식으로 문제를 풀면 오답인가요?

- 3주차 예제 프로그램 중 for_range_else.py에 관한 질문입니다.

<프로그램 내용>

for_range_else.py 13~17 Line

```
print("bye")
for i in range(1, 5, 2):
    print(i)
else:
    print(i)
```

위의 (1)과 (2)코드의 실행 결과를
저는 다음과 같이 예측 했습니다.

#(1)실행결과예측=====

bye

1

3

5

=====

그러나, 실제 실행결과는 제가 예측한 결과와 달랐습니다.

#(1)실제실행결과=====

bye

1

3

3

- for i in range(1,5,2)의 경우, 저는 i가 1,3일 때까지만 동작하고 다시 for문의 조건 문장으로 올라가, i가 range의 끝값인 5로 증가하면 조건에 맞지 않기 때문에 else문으로 빠져나간 후, 5를 출력할 것이라 예측했습니다. 하지만 실제 실행결과 for문은 i를 끝값 이전까지(*i=3)만 증가시킨 후 반복을 중단했고 그러한 이유 때문에 else문의 print(i)에서 for문에서의 마지막 i값인 3을 출력했습니다.

위의 for_range_else.py 13~17 Line을 while문으로 교체해 보면 아래와 같습니다.
<프로그램 내용>

```
*****
# for_range_else.py 13~17 Line - while문
*****
i=1

print("bye")
while i<5:
    print(i)
    i+=2
else:
    print(i)
```

임을 알 수 있습니다.

- 그런데, 이때 for문을 while문으로 교체했을 때의 실행 결과는 제가 조금 전 예측했던 결과인

#(1)실행결과예측=====

bye

1

3

5

=====

와 동일함을 알 수 있었습니다.

결론적으로 제 질문은

Q. for문을 while문으로 교체했을 뿐인데 이러한 실행결과 차이가 발생한 이유는 무엇입니까?

Q. for문은 range의 조건을 미리 계산한 후에 반복을 실행하기 때문인가요?

입니다.

이에 대한 답변 부탁드립니다. 감사합니다.

- 제공해주신 예제 소스 파일을 실행해보면서 loop()안에서의 break; 는 상속되어있는 전체loop()를 끝내는 것이 아니라 부모loop()만 끝낸다는 것을 알았습니다.
- 노트에 적고 해보는게 진짜 기억에 잘 남는거 같습니다!
- 없습니다. 개인적인 질문은 3명 있는 그룹 스터디? 그건 언제부터 하면 좋을까요?
- 수업시간에 관련해서 질문 없습니다.
+ 실무를 하고 있는 프로그래머들은 어느정도의 역량을 가지고 취업을 하고, 업무를 해결하는지 궁금합니다!

- for else와 while else를 잘 모르겠어서 구글에 검색을 해보았는데 else는 if와 함께 쓰인다고하는데 교수님 처럼 if없이 사용해도 되는건지 잘 모르겠습니다. 그리고 교수님이 올려주신 코드 for list else와 for range else에서 for 동작을 다하고 나서 마지막 순서를 왜 한번더 출력을 하는지 이해가 안됩니다.

```
print("hi")
```

```
for i in range(5):
```

```
    print(i)
```

```
else:
```

```
    print(i)
```

1234를 찍고 i가 4인 상태로 그대로 내려와서 else문을 실행시키는건가요?

- 배열에서
array[x-1]은 실행을 합니다.
하지만 array[x/2]혹은[x*2]오류가 납니다+,-연산자는 실행합니다.
왜 다른 연산자는 오류가 나는지 궁금한다

- 1. 1~2주차에 궁금한 질문했었는데 그당시에는 토요일까지 써야한다는 말이 없어서 다 날아간거 같습니다. '질문에 답하세요' 가 아닌 'X주차 진행 후, 자기 평가미리보기' 에 돋보기를 눌러도 설문지가 나왔거든요.. 이에 대해서 어떻게 해야하나요?
- 2. list 관련되서 질문
list 예제를 보면 초기화를 한번도 진행하지 않고 실행했기에 항상 color 값이 달라집니다.
이를 실행할때마다 초기 color 값으로 계속 초기화하면서 진행하려면 어떤식으로 해야하나요?
- 4. for 문 실행할때 i에 초기값을 주지않으면 i는 무조건 0인가요?
c에서는 초기값이 없으면 오류가 나거나 프로그램에서 아무 값이나 넣어서 실행되는데 파이썬은 다른가요?(for문이야기가 아닌 초기값에 대한 질문)
- 5. for문에서 range(3 ,5) 라면 <=이나 >= 같은 값은 없는건가요?
오로지 <와 >만으로 for을 구성하나요?
그래서 5보다 작은 정수인 4까지만 나오는건가요?

- 6. while 문에서 else의 의미가 어떨때 사용되는건가요?
굳이 else를 안붙여도 while에 해당하는 조건을 벗어나면 알아서 루프를 빠져나가니까 else가 필요 없는거 같은데
while을 쓰고 싶는데 내가 원하는 특정 조건을 굳이 사용 할때 else를 사용하나요?
예를 들면 1~10 증가를 하고싶은데 4에서 탈출하고 싶으면 쓰는건가요?
while a<10 :
 print(a)
 a += 1
else a = 4:
 print("end")
 print('goodbye')
- 이러한거처럼?

- 프로그래밍을 배울때 break문은 가급적 사용하지 말라고 배웠는데 그 이유가 무엇인지 궁금합니다.
- 강의 내용과 별개의 질문인데 대면진행 계획이 있으신지 +대면강의 진행시 어떤 방식으로 진행되는지 궁금함.
- 아직 break문이 익숙하지않아서 조금더 공부해야 할 것 같습니다.
- 과제 워드로 제출했는데 감점인가요 ? pdf로 꼭 바꿔서 내야하나요 ??
- for ~ else문 예제에서
for i in a :
print(i)
반복후 else : print(i)를 하면 리스트의 마지막 요소가 출력되는 이유가 궁금했는데 인터넷을 뒤져보고 간단한 예제를 만들어 풀어보니 쉽게 이해가 됐습니다

- 교수님 for_range_else와 for_list_else에 대해 질문이 있습니다.

```
for_range_else에서  
print("hi")
```

```
for i in range(5):
```

```
print(i)
```

```
else :
```

```
print(i)
```

이면 range의 범위는 0~4이고 그때까지는 for문이 참으로 돌다가 i가 5가 되는 순간 for문이 끝나고 else절로 넘어가면서 마지막 i인 4가 출력이 되는 거라면

```
for_list_else에서
```

```
a = [1, 5, 10]
```

```
for i in a:
```

```
print(i)
```

```
else :
```

```
print(i)
```

일 때, i는 list순서대로 1, 5, 10일 때는 for문을 실행하다가 다음 list값이 없어서 끝나는 것이 아닌, 없다 = null로 인식해서 else절로 넘어가 마지막 i값인 10을 출력하고 끝나는 것인가요?

- 이번 강의에서 새로 배우는 내용에 대해서는 질문이 없습니다.
하지만 연산자 우선순위를 설명하실때 위에서부터 1순위로 취급하신다고 하셨는데
책과는 전혀 정 반대로 나와있습니다. 한번 확인 부탁드립니다.

추가로 저는 수강신청 정정 기간을 통하여 2주차부터 수업을 듣게 되었습니다. 당시 과제 제출 기간을 너무나 당연하게 2주라고 생각을 하여 제출을 하지 못하였습니다. 정정 기간에 수업에 참여하게 된 학생들에게 1주차의 과제 제출을 추가로 받아주실 수 있을까 하여 늦었지만 늦었을 때가 가장 빠르다 라는 생각이 들어 질문을 남깁니다.

| Operator | Description |
|---|--|
| (expressions...), [expressions...], {key: value...}, {expressions...} | Binding or parenthesized expression, list display, dictionary display, set display |
| x[index], x[index:index], x(arguments...), x.attribute | Subscription, slicing, call, attribute reference |
| await x | Await expression |
| ** | Exponentiation 6 |
| +x, -x, ~x | Positive, negative, bitwise NOT |
| *, @, /, //, % | Multiplication, matrix multiplication, division, floor division, remainder 5 |
| +, - | Addition and subtraction |
| <<, >> | Shifts |
| & | Bitwise AND |
| ^ | Bitwise XOR |
| | Bitwise OR |
| in, not in, is, is not, <, <=, >, >=, !=, == | Comparisons, including membership tests and identity tests |
| not x | Boolean NOT |
| and | Boolean AND |
| or | Boolean OR |
| if – else | Conditional expression |
| lambda | Lambda expression |
| := | Assignment expression |

- 1) 입력 수가 2과 7의 배수인지 확인하는 프로그램을 작성하시오.
- 2) 입력된 3개의 수 중에서, 차이가 가장 큰 두 수를 출력하시오.
- 3) 3개의 수를 입력 받아, 첫번째 수가 2의 배수이면 세번째 수를 2배, 2의 배수가 아니면 3배하고, 2번째 입력된 수가 2와 3의 배수이면 다시 세번째 수를 6배 하여 출력하고 그렇지 않으면 5배 하여 출력하는 프로그램을 작성하시오.
- 4) 입력된 알파벳 문자열의 첫번째 알파벳과 마지막 알파벳 정중앙에 위치하는 알파벳을 출력하시오. 정중앙의 알파벳이 두 개면 두개를 출력하시오.

Ex 3. Let's enjoy python!

17

- 1) 입력 문자열의 대소문자를 바꾼 문자열을 만들어 출력하시오.
- 2) 입력 홀수에 대해 아래와 같이 출력하시오.

ex) 3

```
*****  
  *****  
*****
```

5

```
*****  
  *****  
    *****  
  *****  
*****
```



ex 3-1.py
ex 3-2.py

python에서 제공되는 함수를 사용하지 않고 프로그램을 작성하시오.

1) 1부터 첫번째 입력수까지 수에서 두번째 입력수의 배수의 합을 구하시오.

ex) 10 2 → 30

2) 1부터 첫번째 입력수까지 수에서 두번째 입력수로 나눈 나머지가 0이고 세번째 입력수로 나눈 나머지가 1인 수를 모두 출력하시오.

3) 입력된 문자열을 좌우 대칭의 문자열을 만들어 출력하시오.

ex) "hello" → "olleh"

4) 입력 문자열의 각 알파벳을 순서대로 정렬된 문자열을 만들어 출력하시오.

ex) "hello" → "ehllo"

5) 입력 홀수에 대해 아래와 같이 출력하시오.

ex) 3

```
***
 ***
***
```

5

```
*****
 *****
  *****
   *****
    *****
     *****
```

- 프로그램 내의 여러 곳에서 호출되는 일정 단위의 동작 또는 기능은 함수로 작성하여 사용하면 매우 효율적인 프로그램을 작성할 수 있다.
- 함수의 입력, 출력 정보 결정에 따라 프로그램의 효율 및 성능이 크게 달라진다.

- 함수 정의

```
def 함수명(입력 정보1, 입력 정보 2, ...):  
    함수 body
```

- Return 문

- 함수 출력값 반환

```
return 반환값
```



local.py
global.py
global_declare.py
global2.py

- 함수 내에서 정의된 변수는 전역변수와 이름이 동일하더라도 함수 내에서만 유효하다.
- 함수 밖에서 선언된 변수(전역 변수)를 사용하기 위해서 global을 사용하여 선언한다.
 - 변수 사용하기 전에 선언하여야 한다.
 - 함수 내에서 global을 사용하는 경우, 해당 전역변수와 동일 이름의 지역변수는 선언(사용)할 수 없다.



local.py
global.py
global_declare.py
global2.py

Ex 4-1. Let's enjoy python!

21

- 1) 입력 문자열의 대소문자를 바꾼 문자열을 구하는 함수를 작성하시오.
- 2) 입력 홀수에 대해 아래와 같이 출력하는 함수를 작성하시오.

ex) 3

```
*****  
  *****  
*****
```

5

```
*****  
  *****  
    *****  
  *****  
*****
```



ex 3-3.py
ex 3-4.py

python에서 제공되는 함수를 사용하지 않고, 자신만의 함수로 작성하시오.(함수에서는 화면 출력을 하지 않음)

1) 1부터 첫번째 입력수까지 수에서 두번째 입력수의 배수의 합을 구하시오.

ex) 10 2 → 30

2) 1부터 첫번째 입력수까지 수에서 두번째 입력수로 나눈 나머지가 0이고 세번째 입력수로 나눈 나머지가 1인 수를 모두 출력하시오.

3) 입력된 문자열을 좌우 대칭의 문자열을 만들어 출력하시오.

ex) "hello" → "olleh"

4) 입력 문자열의 각 알파벳을 순서대로 정렬된 문자열을 만들어 출력하시오.

ex) "hello" → "ehllo"