

공학 (Engineering)

10.1 로봇 청소기의 이동과 귀환하기

프로그램 p10-01 ... 로봇 청소기의 이동과 귀환하기

관련 학습 ... 리스트의 원소 추가

리스트의 원소 삭제

10.2 ISBN 코드를 통한 도서 번호 검증하기

프로그램 p10-02 ... ISBN 코드 검증하기

관련 학습 ... 문자열 선언, 서식 문자, 문자열의 특정 위치 문자 얻기,

문자열 관련 예약어와 함수

Thinking!

잠깐! Coding

Coding! Programming



1. print() 함수에서 서식 문자열을 다음과 같이 사용하였을 때 출력 결과를 예측하여 작성해보자.


```
i = 20  
print("#123456789#")  
print("#%d#" % i)  
print("#%-d#" % i)  
print("#%9d#" % i)  
print("#%-9d#" % i)
```



서식 문자열 %9d는 9자리의 자릿수가 지정된 상태에서 정숫값이 출력되며, %-9d는 지정한 자릿수의 왼쪽 맞춤을 하여 정숫값이 출력된다.

2. print() 함수에서 서식 문자열을 다음과 같이 사용하였을 때 출력 결과를 예측하여 작성해보자.

```
f = 3.141592
print("#123456789#")
print("#%f#"%f)
print("#%-f#"%f)
print("#%9.3f#"%f)
print("#%9.1f#"%f)
print("#%-9.3f#"%f)
print("#%-9.1f#"%f)
```

 서식 문자열 %9.3f에서 전체 자릿수는 9자리를 지정하고 소수점 이하 자릿수는 3자리 지정한다. 소수점 이하 자릿수 다음의 숫자가 5이상이면 반올림이 이루어진다.

3. print() 함수에서 서식 문자열을 다음과 같이 사용하였을 때 출력 결과를 예측하여 작성해보자.

```
c = 'a'
s = "str"
print("#123456789#")
print("#%c#" % c)
print("#%s#" % s)
print("#%9s#" % s)
print("#%-9s#" % s)
print("#%9c#" % s[0])
print("#%-9c#" % s[1])
```




서식 문자열 %c는 하나의 문자나 기호를 출력하고, %s는 문자열을 출력한다. s="str"에서 s[0]은 s 문자이고 s[1]은 t 문자이다.

4. 문자열 변수 `jn`의 값이 주민등록번호 형식 “980910-1234567”일 때, 년을 뜻하는 0과 1 자리 문자열, 월을 뜻하는 2와 3자리 문자열, 일을 뜻하는 4와 5자리 문자열을 출력해보자. 그리고 남/여를 구분하는 7자리의 문자를 출력하고, 문자열의 전체 위치에서 뒤로부터 마지막 문자, 뒤로부터 마지막에서 세 번째 문자, 뒤로부터 마지막에서 다섯 번째 문자를 각각 출력해보자.

```
98 09 10
```

```
1
```

```
7 5 3
```

 `jn[0:2]` 형식은 0부터 2까지의 위치가 아닌 0부터 1까지를 의미한다. 즉, `jn[n:m]` 형식은 $n \leq \text{위치} < m$ 을 의미한다. `jn[-1]` 형식은 `jn` 문자열에서 뒤로부터 첫 번째 문자를, `jn[-3]`은 마지막에서 세 번째 문자를 의미한다.

5. 문자열 변수 str의 값이 “python”일 때 다음과 같이 출력해보자.

```
p  
py  
pyt  
pyth  
pytho  
python
```



for 문을 이용하여 `range(len(str))` 범위로 반복하면서 `str[0:i+1]` 형식으로 출력한다.


6. ‘Basic Coding 5번 문제’를 참고하여, 문자열 변수 str의 값이 “python”일 때 다음과 같이 출력해보자.

```
p  
py  
pyt  
pyth  
pytho  
python
```

 for 문을 이용하여 range(len(str)) 범위로 반복하면서 print(“%6s”%str[0:i+1])로 출력한다.

7. ‘Basic Coding 5번 문제’를 참고하여, 문자열 변수 str의 값이 “python”일 때 다음과 같이 출력해보자.

```
n  
on  
hon  
thon  
ython  
python
```

 for 문을 이용하여 `range(len(str)-1, -1, -1)` 범위로 반복하면서 `print(str[i:(len(str))])`로 출력한다.

8. 임의의 문자열 두 개를 입력받아 문자열 길이가 긴 문자열만 출력해보자.

```
문자열1 : python  
문자열2 : java  
긴 문자열 = python
```



문자열 길이는 `len(str1)` 형식으로 구한다.

9. 임의의 문자열을 입력받아 숫자만으로 구성되었는지, 문자만으로 구성되었는지, 숫자/문자가 섞여 구성되었는지 판별해보자.

문자열 : 123

숫자로 구성된 문자열

문자열 : python

문자로 구성된 문자열

문자열 : python123

숫자/문자로 구성된 문자열



숫자만으로 구성되었는지, 문자만으로 구성되었는지, 숫자/문자가 섞여 구성되었는지는 각각 `isdigit()`, `isalpha()`, `isalnum()` 함수로 검사한다.

10. while 문을 이용한 무한 반복을 하면서 리스트 data에 입력한 문자열을 추가해보자. 빈 문자열이 입력되면 무한 반복을 종료하고, for 문을 이용하여 리스트의 전체 원소들을 출력해보자.

문자열 : Python

문자열 : C

문자열 : Java

문자열 : C++

문자열 : Swift

문자열 : R

문자열 :

Python C Java C++ Swift R



append() 함수로 리스트의 마지막 위치에 새로운 원소를 추가하고, len() 함수로 리스트 내의 원소 개수를 구한다.

1. 문자열을 입력받은 후, while 문을 이용하여 무한 반복하면서 한 문자를 입력받아 해당 문자가 앞서 입력한 문자열 내에 있으면 “문자 y가 문자열 Python에 존재함” 형식으로 출력하고, 문자열 내에 없으면 “문자 k가 문자열 Python에 존재하지 않음” 형식으로 출력해보자. 빈 문자열을 입력하면 무한 반복을 종료한다.

문자열 : Python

문자 : y

문자 y가 문자열 Python에 존재함

문자 : k

문자 k가 문자열 Python에 존재하지 않음

문자 :



문자열 내에 문자가 존재함을 파악하기 위해서는 in 예약어를 사용한다.


2. 정수 형태의 문자열을 전달받아, 문자열이 숫자만으로 구성되었을 경우 문자열을 숫자로 변경하여 반환하는 `str2int()` 함수를 선언해보자. 문자열이 숫자만으로 구성되지 않았을 경우 `None` 값을 반환해보자. 문자열을 입력받은 후 `str2int()` 함수를 호출하여 실행해보자.

문자열 : 125

125

문자열 : 12k

None

 `s.isdigit()` 함수를 통해 문자열 `s`가 숫자만으로 구성되었는지 여부를 판단하며, 파이썬에서 `None` 값은 '없음'과 '거짓'의 의미이고 타 언어의 `null`과 같은 개념으로 사용된다.

3. 리스트 `list = [1, 2, 3]`을 선언한 후, “append 2” 형태와 “insert 1 3” 형태의 문자열을 입력받아, 입력된 값을 분리하여 리스트 `cmd`에 저장한다. `cmd[0]`의 값이 “append”이면 `cmd[1]`의 값을 리스트 `list`에 저장하고, `cmd[0]`의 값이 “insert”이면 `cmd[1]`의 위치에 `cmd[2]`의 값을 리스트 `list`에 삽입한다. 그리고 리스트 `list`의 값을 출력한다.

```
[1, 2, 3]
```

```
리스트 명령 : append 10
```

```
[1, 2, 3, 10]
```

```
[1, 2, 3]
```

```
리스트 명령 : insert 1 20
```

```
[1, 20, 2, 3]
```



`cmd = indata.split()` 함수로 입력된 값들을 분리하여 리스트 `cmd`에 저장하고, `cmd[0]`의 값에 따라 `list.append(int(cmd[1]))` 또는 `list.insert(int(cmd[1]), int(cmd[2]))`를 실행한다.

4. ‘Enhancement Coding 3번 문제’를 참고하여, 리스트 `list = []`을 선언한 후, 다음 일련의 리스트 명령을 처리하도록 수정해보자. 첫 번째 숫자는 일련의 리스트 명령의 개수이며, “printout”은 리스트를 출력하고, “remove”는 리스트 내의 해당 값을 삭제하고, “sort”는 리스트의 값들을 오름차순으로 정렬한다.

리스트 명령 수 : 9

append 3

insert 0 4

printout

insert 1 5

append 6

remove 6

printout

sort

printout

[4, 3]

[4, 5, 3]

[3, 4, 5]




`list = []`로 선언하고, 반복할 숫자를 입력받아 해당 숫자만큼 for 문으로 반복한다. 숫자 뒤에 리스트 명령을 연속으로 입력하고, `cmd = indata.split()` 함수로 입력된 명령들을 분리하여 리스트 `cmd`에 저장한다. `cmd[0]`의 값에 따라 `list.append(int(cmd[1]))`, `list.insert(int(cmd[1]), int(cmd[2]))`, `list.remove(int(cmd[1]))`, `list.sort()`, `print(list)`를 각각 실행한다.

5. 주민등록번호 형식의 앞 6자리 문자열을 입력받아 년을 뜻하는 0과 1자리 문자열을 추출한 후, 현재 연도와와의 계산을 통해 나이를 출력해보자.

주민등록번호 년월일 : 980910

나이 : 20

 나이는 '현재 연도-(주민번호앞2자리+1900)+1'로 계산한다. 현재 연도는 import time후, now = time.localtime() 함수를 실행하면 now.tm_year의 값에서 구할 수 있다. 참고로 now.tm_mon은 현재 월, now.tm_mday는 현재 일, now.tm_hour, now.tm_min, now.tm_sec는 현재 시/분/초의 값을 의미한다.

6. 주민등록번호 형식의 앞 6자리 문자열을 입력받아 월의 범위가 01~12인지 검사하고, 일의 범위가 01~31인지 검사해보자. 앞 6자리 문자열에 문제가 발생하지 않았다면 남녀 성별을 구분하는 7번째 1자리 숫자를 추가로 입력받아 다음과 같이 출력해보자.

주민등록번호 년월일 : 001334
주민등록번호의 월 형식이 잘못되었습니다.

주민등록번호 년월일 : 001234
주민등록번호의 일 형식이 잘못되었습니다.

주민등록번호 년월일 : 001231
성별 표시 : 1
001231-1



주민등록번호의 월 값은 jnymd[2:4]이고, 일 값은 jnymd[4:6]이다.

7. ‘Enhancement Coding 6번 문제’를 참고하여, 남녀 성별을 구분하는 7번째 1자리 숫자를 추가로 입력받아 9/0이면 1800년대 태어난 남성/여성, 1/2이면 1900년대 태어난 남성/여성, 3/4면 2000년대 태어난 남성/여성, 5/6이면 1900년대 태어난 외국인 남성/여성, 7/8이면 2000년대 태어난 남성/여성으로 판별하여 출력해보자.

주민등록번호 년월일 : 001231

성별 표시 : 1

1900년대 태어난 남성입니다.

주민등록번호 년월일 : 001231

성별 표시 : 3

2000년대 태어난 남성입니다.



성별 표시 연대를 저장하기 위하여 딕셔너리 `mfd = { 9:1800, 0:1800, 1:1900, 2:1900, 3:2000, 4:2000, 5:1900, 6:1900, 7:2000, 8:2000 }`를 선언한다. 성별 표시 숫자를 입력받아 `mfd[mf]`에 의해 태어난 연대를 구하고, 성별은 `mf % 2`로 나누어 나머지가 1이면 남성, 0이면 여성으로 값을 구한다.

8. 리스트 `data = ["Python", "C", "Java", "C++", "Swift", "R"]`인 경우 “Swift”를 찾아 위치를 구하고 해당 위치의 원소(“Swift”)를 “Objective-C”로 변경해보자. 그리고 “Java”를 찾아 위치를 구하여 “C#”을 삽입해보자. 리스트 내에서 첫 번째로 나타나는 “R”을 삭제하고, 리스트의 전체 원소들을 출력해보자.

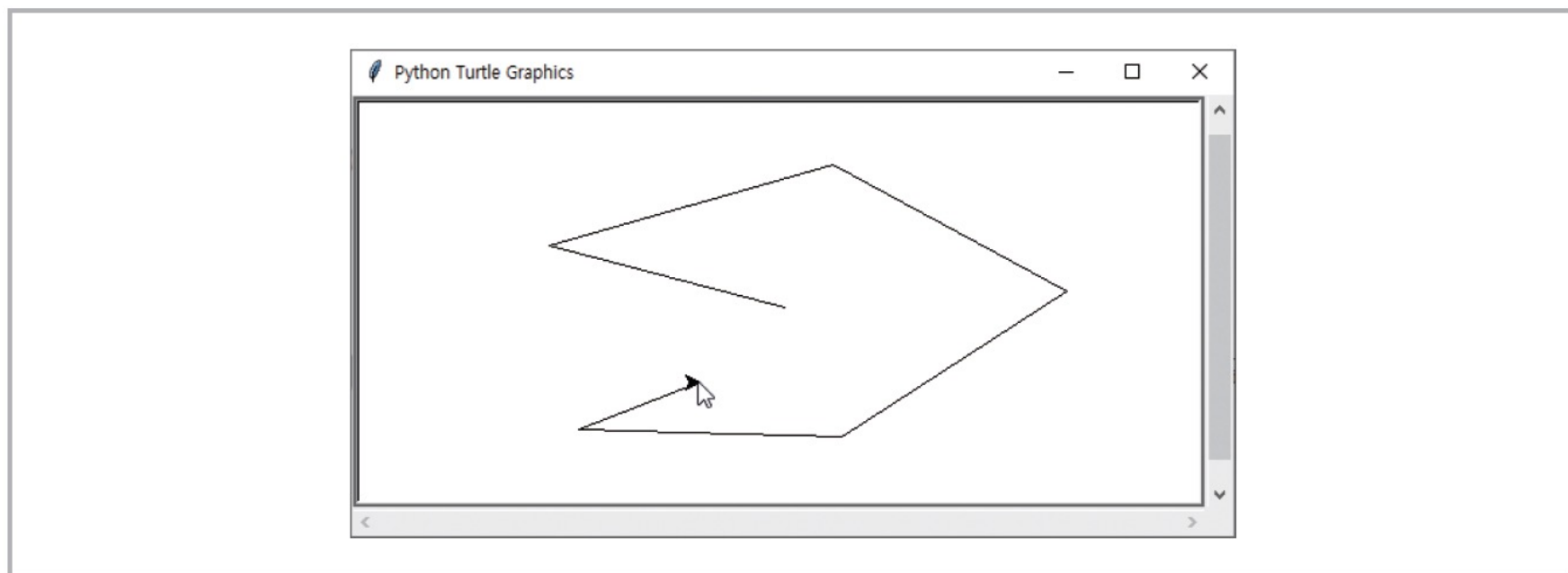
```
['Python', 'C', 'Java', 'C++', 'Swift', 'R']
```


```
Python C C# Java C++ Objective-C
```



`index(x)` 함수를 통해 리스트 내에서 `x` 원소의 위치를 구할 수 있으며, `insert(i, x)` 함수는 지정된 위치 `i`에 `x`를 삽입하고, `remove(x)` 함수는 `x`를 찾아 삭제한다. 리스트 원소의 출력은 `print(data)` 문을 통해 전체 원소를 출력하고, `for` 문을 이용하여 개별 원소들을 출력한다.

9. ‘프로그램 p10-01’을 참고하여, 터틀 스크린에서 마우스를 클릭한 위치(좌표)를 모두 저장한 후, 키보드 d 키가 눌러지면 터틀 스크린의 중앙 지점부터 저장된 순서대로 좌표에 따라 선을 그려보자.



 마우스 클릭을 처리하기 위하여 `clicked()` 콜백 함수를 등록하고, 키보드 d 키가 눌러지면 처리하기 위하여 `key_d()` 콜백 함수를 등록한다. 마우스 클릭이 이루어지면 `clicked()` 함수에서 x, y 좌표를 리스트에 추가하고, d 키가 눌러지면 `key_d()` 함수에서 리스트 크기만큼 반복하면서 `t.goto()` 함수로 좌표 이동하면서 선을 그린다.

10. ISSN 코드는 8자리 숫자로 구성되어 있으며 마지막 자리는 7자리 숫자를 검사하는 체크 기호이다. 8자리의 ISSN 코드를 입력받아 체크 기호를 구하고 검증해보자. ISSN 체크 기호는 7자리 숫자에 8부터 2까지의 숫자들을 순서대로 곱하여 가중치를 구하고 모두의 합계를 구한다. 합계 값을 11로 나누어 나머지를 다시 11에서 뺀 숫자가 체크 기호이다. 이때 체크 기호가 10이면 X로, 11이면 0으로 표시한다.

ISSN							체크기호
0	0	2	7	9	3	5	8
x	x	x	x	x	x	x	
8	7	6	5	4	3	2	
0	0	12	35	36	9	10	
합 = 102							
102 ÷ 11의 나머지 : 3							
11 - 3 = 체크기호							8

ISSN 8자리(- 제외) : 00279358

ISSN 1~7자리의 가중치 반영 합계 : 102

ISSN 1~7자리의 체크 기호 값 : 8

ISSN 코드 : 00279358은(는) 검증되었습니다.



가중치의 변동 값을 갖게되는 변수 weight의 값을 8로 초기화한 후, for문을 `range(len(issn)-1)` 범위만큼 반복하면서 `issn[i]`의 값과 weight 값을 곱하여 합계를 구한다. for문이 반복되는 동안 weight 값은 1씩 감소되어 2까지 줄어든다.