

4주차

파이썬 기초 프로그래밍



문자열

- 문자의 표현
- 문자열이란
- 문자의 연산
- 문자열 비교/ 처리/ 검사

복습 문제 01

- 다음 코드의 출력 결과는?

```
print("hello")  
print('\hello')
```

```
print("I don't know")  
print('I don\t know')
```

```
print(r'c:\usr\bin')
```

```
print("""\  
line1  
line2  
line3\  
""")
```

```
print('aaaaaaaaaaaaaaaaa'  
      'bbbbbbbbbbbbbbbbbb')
```

문자열이란?

- 문자들의 시퀀스
- 변경 불가능
- 문자열 정의
 - 한줄 문자열
 - 'I love Python'
 - "나는 파이썬이 좋아요"
 - 여러줄 문자열
 - `""" |
love
Python...
"""`
 - `""" 나는 파이썬이
좋아요...
"""`

시퀀스 자료형

1. 여러 개의 객체 저장
2. 각 객체는 순서를 가짐
3. 인덱싱, 슬라이싱, 연결, 반복
 멤버 검사, 길이 정보 가능
4. 문자열, 리스트, 튜플

시퀀스 자료형(1)

- `s = '123456'` 일때

구분	연산	설명	예	결과
인덱싱	<code>[k]</code>	k번 위치의 값	<code>s[0]</code>	<code>'1'</code>
슬라이싱	<code>[s:t:p]</code>	s부터 t-1 사이 구간의 값을 p 간격으로 취한다	<code>s[1:5:2]</code>	<code>'24'</code>
연결하기	<code>+</code>	두 시퀀스 연결	<code>'12'+'345'</code>	<code>'12345'</code>
반복하기	<code>*</code>	반복	<code>'=='*3</code>	<code>'==='</code>
멤버 검사	<code>in</code>	어떤 값이 시퀀스 자료형에 속하는지 검사	<code>'3' in s</code>	<code>True</code>
길이 정보	<code>len()</code>	크기	<code>len(s)</code>	<code>6</code>

연산

```
>>> word='abcdefg'   # 문자열 정의
>>> word[0]           # 인덱싱
'a'
>>> word[5]
'f'
>>> word[-1]          # 음수 인덱싱
'g'
>>> word[0] = 'A'      # 수정할 수 없음!!!
Traceback (most recent call last):
  File "<pyshell#263>", line 1, in <module>
    word[0] = 'A'
TypeError: 'str' object does not support item assignment

>>> s = "Hello Python"
>>> s[0:5]             # 슬라이싱. 이 경우 0 생략가능
'Hello'
>>> s[5:]
' Python'
>>> 'J'+s[7:]          # 연결
'Jython'
>>> print ( '=' * 30 )  # 반복
=====
>>> 'Python' in s       # 멤버 검사
True
```

스트링의 문자 접근 : index

G	E	E	K	S	F	O	R	G	E	E	K	S
0	1	2	3	4	5	6	7	8	9	10	11	12
-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

```
String1 = "GeeksForGeeks"
```

```
print(String1)
```

```
# Printing First character
```

```
print("\nFirst character of String is: ")
```

```
print(String1[0])
```

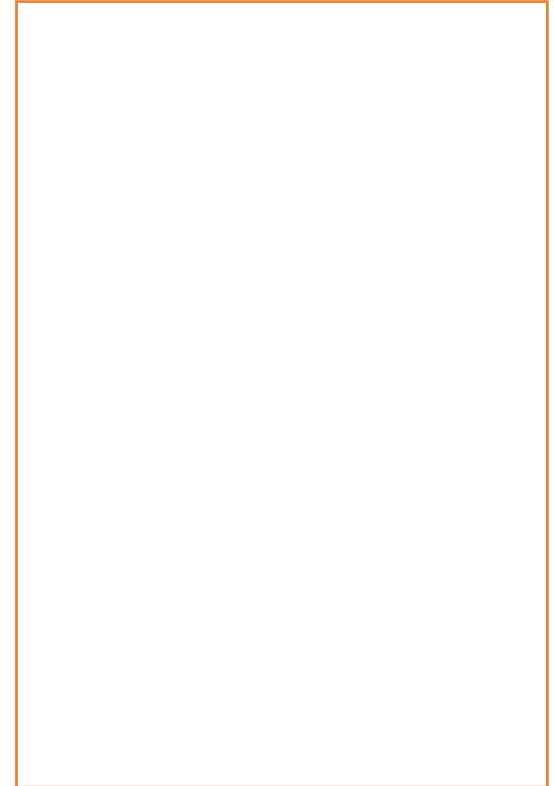
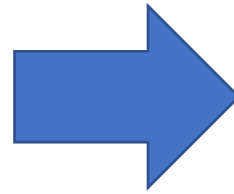
```
# Printing Last character
```

```
print("\nLast character of String is: ")
```

```
print(String1[-1])
```

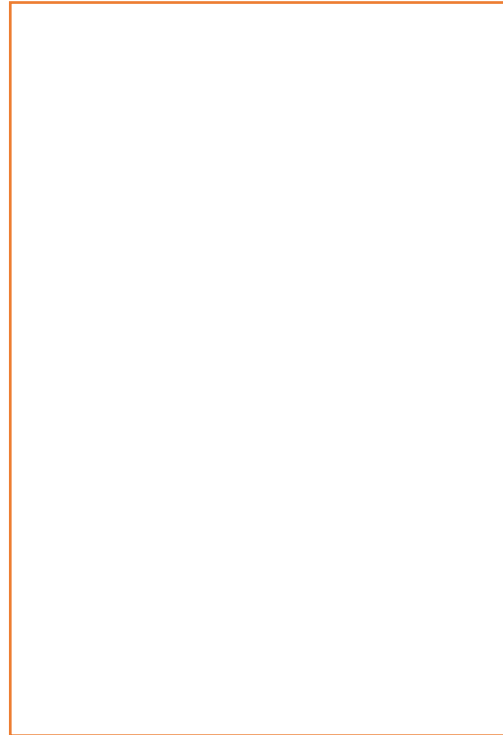
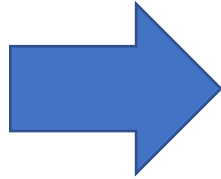
스트링 슬라이싱(Slicing)

```
String1 = "GeeksForGeeks"  
print(String1)  
print(String1[3:12])  
print(String1[3:-2])
```



다음 코드의 결과값은?

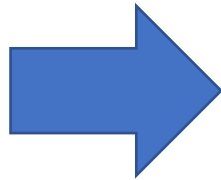
```
word = 'python'  
print(word[0])  
print(word[1])  
print(word[-1])  
print(word[0:2])  
print(word[:2])
```



다음 코드의 결과값은?

```
s = 'My name is Mike. Hi Mike'
print(s)
is_start = s.startswith('My')
print(is_start)

print(s.find('Mike'))
print(s.rfind('Mike'))
print(s.count('Mike'))
print(s.capitalize())
print(s.title())
print(s.upper())
print(s.lower())
print(s.replace('Mike', 'Nancy'))
```






문자열 비교하기

- <, >
 - 사전에 나오는 순서로 비교
- ==
 - 문자열이 동일한지 비교

```
>>> a = 'apple'
>>> b = 'banana'
>>> a < b
True
>>> b < a
False
>>> a == b
False
```

특수 문자

- ₩(백슬래쉬)를 이용하여 키보드로 표현하기 힘든 문자 표시

문자	설명	문자	설명
\ 	다음 줄과 연속임을 표현	\n	줄 바꾸기
\\	\문자 자체	\t	 키
\'	'문자	\uXXXX	유니코드 xxxx
\"	"문자	\e	 키

실습 문제 01

- 주민번호 앞 6자리를 입력 받아 년, 월, 일을 분리하여 출력하시오.

결과

종료를 원하시면 999999를 입력하세요
주민번호 앞 6자리를 입력하세요 : **981011**
1998 년 10 월 11 일

종료를 원하시면 999999를 입력하세요
주민번호 앞 6자리를 입력하세요 : **110101**
2011 년 01 월 01 일

종료를 원하시면 999999를 입력하세요
주민번호 앞 6자리를 입력하세요 : **999999**
종료되었습니다

반복하여 문자 처리

- 문자열에 있는 모든 문자를 하나씩 처리할 때

```
s = input('문자열을 입력하시오 : ')\nfor c in s:\n    print(c)
```

꼭 알아둘것!!!

- 문자열에 있는 문자를 특정한 순서로 방문할 때

```
s = input('문자열을 입력하시오 : ')\nfor i in range(0, len(s), 2):\n    print(s[i])
```

문자열에서 단어 분리

- split()
 - 문자열 안의 단어를 분리하여, 단어들의 리스트 반환
 - 인수가 없으면 공백으로 분리, 있으면 그 문자로 분리

```
>>> s = 'Love never fails'
>>> s.split()                # 공백으로 분리
['Love', 'never', 'fails']
>>> s = 'sjyoun:0:abcdef:/usr/sjyoun:bsh'
>>> s.split(':')            # ':' 기준으로 분리
['sjyoun', '0', 'abcdef', '/usr/sjyoun', 'bsh']
>>>
```

메서드 : 객체에 포함된 함수

부분 문자열 검색

- s = 'I like programming, I like Python.'

메서드	설명	사용 예	결과
count(a)	문자열 중에 a가 반복된 횟수	<code>s.count('like')</code>	
find(a)	문자열 중에 a의 시작 위치	<code>s.find('like')</code>	
rfind(a)	문자열 뒤부터 시작하여 a의 시작 위치	<code>s.rfind('like')</code>	
startswith(a)	문자열이 a 문자열로 시작하는지	<code>s.startswith('like')</code>	False
endswith(a)	문자열이 a 문자열로 끝나는지	<code>s.endswith('like')</code>	

문자열 검사

- 문자열이 알파벳으로 구성되어 있는지, 숫자로 구성되어 있는지 검사

```
>>> a = 'abcdefg'
>>> a.isalpha()
True
>>> b = '1234567'
>>> b.isdigit()
True
>>>
```

```
s = input("주민번호 앞자리를 입력하시오 : ")
if not s.isdigit() :
    print("숫자만 입력하시오")
```

문자열 변환

- `s = 'Python\n'`
- 대소문자 변환
 - `upper()` : `s.upper()` → `'PYTHON\n'`
 - `lower()` : `s.lower()` → `'python\n'`
- 한 문자 치환
 - `replace(s1, s2)` : `s.replace('P', 'J')` → `'Jython\n'`
- 공백 문자 제거
 - `strip()`, `rstrip()` : `s.strip()`, `s.rstrip()` → `'Python'`
 - `lstrip()` : `s.lstrip()` → `'Python\n'`

space, 탭, \n

파이썬 프로그램 소스 분석을 통해 코딩 실력 쌓기

- Memo 만들기
- 카이사르 암호문 만들기
- list 파일 입출력
- GitHub
 - <https://github.com/DIT-Python/list-file-io-example.git>

File Handling



[Python File Handling: 파일 열기, 읽기, 쓰기, 닫기]

`open()`, `read()`, `write()`, `close()`



파일 읽기

- read()
 - 파일의 내용 전체를 한꺼번에 읽음
- readline()
 - 호출할 때마다 파일의 내용을 줄 단위로 읽음
 - 10줄일 경우 전체를 읽기 위해 10번을 실행해야 함
- readlines()
 - 파일의 모든 내용을 읽어 줄 단위로 리스트 항목을 만들어 줌

CR(\r), LF(\n)이란 무엇인가?

- \r이라는 탈출 문자(Escape Character)는 **Carriage Return(CR)**이란 의미를 가짐
- \n은 **Line Feed(LF)**란 의미를 가지며 일반적으로 New Line이라고 읽음
 - 케리지(마차)리턴(되돌림)...? 라인(한 줄) 피드(먹이다)...? 이게 도대체 뭔가?? 이것들은 바로 타자기(Typewriter)에서 온 유물이다.
- <https://m.blog.naver.com/taeil34/221325864981>