

# Lecture 1 : 문자열

## 1. 문자열에 사용할 수 있는 내장 함수

- 내장 함수 : 파이썬에서 미리 작성해 두어 바로 사용할 수 있는 함수
- 독립적으로 함수에 문자열을 인자로 전달하여 사용 가능

```
len('python')
```

## 문자열 관련 내장 함수

내장함수	설명	사용 예
<b>str(obj)</b>	obj를 문자열 표현으로 반환	str(3.14) -> '3.14'
<b>len(s)</b>	문자열 s의 길이 반환	len('python') -> 6
<b>max(s)</b>	문자열 s에서 코드 값 기준으로 최대인 문자 반환	max('python') -> 'y'
	2개 이상의 문자열에서 최대 문자열 반환	max('python', 'java') -> 'python'
<b>min(s)</b>	문자열 s에서 코드 값 기준으로 최소인 문자 반환	min('1234') -> '1'
	2개 이상의 숫자에서 최소 수 반환	min(3, 96.3, 13) -> 3
<b>ord(c)</b>	문자 c의 유니코드 값 반환	ord('A') -> 65
<b>chr(n)</b>	유니코드 값 n에 해당하는 문자 반환	chr(44032) -> '가'

## LAB-1 : 문자열 내장 함수 알아보기

```
In [ ]: str(3.14)
```

```
In [ ]: len('python')
```

```
In [ ]: max('python'), min('python')
```

```
In [ ]: max('q', 'python'), min(2, 9.7, 1)
```

```
In [ ]: ord('A'), chr(44032)
```

## 2. 문자열 메서드

- 메서드 : 클래스에 포함되어 있는 함수
- 문자열 메서드 : 문자열 객체(인스턴스)를 통해서만 메서드 호출 가능

```
'python'.capitalize()
```

- 문자열에 사용 가능한 메서드 목록 확인하기

```
dir(str)
```

문자열 메서드

문자열 메서드	설명	사용 예	결과(출력)
s.split()	문자열 s를 (Wn, Wt, Wr, ' ') 기준으로 분리하여 리스트로 반환	"aa bb Wn cc Wt dd".split()	['aa', 'bb', 'cc', 'dd']
	구분자를 지정하면, 구분자로 분리하여 리스트로 반환	'aa bb, cc dd, kk'.split(',')  'aa bb* cc dd, kk'.split('b*')	['aa bb', ' cc dd', ' kk']  ['aa b', ' cc dd, kk']
	2개 까지만 문자열을 구분하고, 나머지는 한 항목으로 반환	'aa bb* cc dd, kk'.split(maxsplit=2)	['aa', 'bb*', 'cc dd, kk']
s.join(k)	문자열 k의 문자와 문자 사이에 문자열 s 삽입	"...".join("1234")	'1::2::3::4'
	리스트 k의 항목과 항목 사이에 문자열 s 삽입	"...".join(['ab', 'cd', 'ef'])	'ab::cd::ef'
s.replace(a, b)	문자열을 s에서 a를 찾아 b로 수정(대체)	"abcdabcd".replace('bc', 'BC')  "1 2 3 4".replace(' ', '')	'aBCdaBCd'  '1234'
s.zfill(n)	문자열을 s에서 n개 폭 만큼의 앞 부분에 0을 채움	"123".zfill(5)  "-123".zfill(5)	'00123'  '-0123'
s.count(p)	문자열을 s에서 문자열 p의 출현 횟수(패턴 p가 몇번 나오나)	"간장 공장 공장장".count('공장')	2
s.find(p)	문자열을 s에서 문자열 p가 나타난 위치 인덱스(0부터 시작)	"간장 공장 공장장".find('공장')  "간장 공장 공장장".find('된장')	3  -1
s.index(p)	문자열을 s에서 문자열 p가 나타난 위치 인덱스(0부터 시작)	"간장 공장 공장장".index('공장')  "간장 공장 공장장".index('된장')	3  value error
s.rfind(p)	문자열을 s를 뒤에서부터 찾아 문자열 p가 나타난 위치 인덱스	"간장 공장 공장장".rfind('공장')	6

문자열 메서드	설명	사용 예	결과(출력)
<b>s.rindex(p)</b>	문자열을 s를 뒤에서부터 찾아 문자열 p가 나타난 위치 인덱스	"간장 공장 공장장".rindex('공장')	6
<b>s.upper()</b>	문자열을 s를 모두 대문자로 변경	"python".upper()	'PYTHON'
<b>s.lower()</b>	문자열을 s를 모두 소문자로 변경	"DONGYANG".lower()	'dongyang'
<b>s.swapcase()</b>	문자열을 s의 대소문자를 서로 변환	"cOmPuTeR".swapcase()	'CoMpUtEr'
<b>s.strip()</b>	문자열을 s에서 맨 앞 뒤의 공백 문자 제거	" python ".strip()	'python'
	제거 문자 지정시 맨 앞 뒤의 해당 문자 제거(중간 문자는 제거 안함)	"*abcd-ef gh---".strip('-*')	'abcd-ef gh'
<b>s.lstrip()</b>	문자열을 s에서 왼쪽 맨 앞의 공백 문자 제거	" python ".lstrip()	'python '
<b>s.rstrip()</b>	문자열을 s에서 오른쪽 맨 뒤의 공백 문자 제거	" python ".rstrip()	' python'

## LAB-2 : 문자열 메서드 알아보기

```
In [ ]: 'aa bb \n cc \t dd kk'.split()
```

```
In [ ]: 'aa bb, cc dd, kk'.split(',')
```

```
In [ ]: 'aa bb* cc dd, kk'.split('b*')
```

```
In [ ]: 'aa bb* cc dd, kk'.split(maxsplit=2)
```

```
In [ ]: "::-".join("1234")
```

```
In [ ]: "::-".join(['ab', 'cd', 'ef'])
```

```
In [ ]: "abcdabcd".replace('bc', 'BC')
```

```
In [ ]: "1 2 3 4".replace(' ', '')
```

```
In [ ]: "123".zfill(5)
```

```
In [ ]: "-123".zfill(5)
```

```
In [ ]: "abcdabcd".count('bc')
```

```
In [ ]: "간장 공장 공장장".count('공장')
```

```
In [ ]: "간장 공장 공장장".find('공장')
```

```
In [ ]: "간장 공장 공장장".index('공장')
```

```
In [ ]: "간장 공장 공장장".find('된장')
```

```
In [ ]: "간장 공장 공장장".index('된장')
```

```
In [ ]: "간장 공장 공장장".rfind('공장')
```

```
In [ ]: "간장 공장 공장장".rindex('공장')
```

```
In [ ]: "python".upper()
```

```
In [ ]: "DONGYANG".lower()
```

```
In [ ]: "cOmPuTeR".swapcase()
```

```
In [ ]: "  python  ".strip()
```

```
In [ ]: "  python  ".lstrip()
```

```
In [ ]: "  python  ".rstrip()
```

```
In [ ]: "**ab*cd-ef gh---".strip('-*')
```

```
In [ ]: '1^2'.isdigit()
```

## LAB 1(과제) : 이메일 유효성 검사 프로그램 만들기

1. 문제 : 사용자로 부터 이메일 주소를 입력 받아 이메일 주소가 유효한지 확인하는 프로그램을 작성하세요.
2. 유효성 조건
  - '@'가 하나 포함되어 있어야 한다.
  - '.'이 '@' 뒤에 하나 이상 있어야 한다.
3. 사용자로 부터 입력 받은 이메일 주소가 유효하면 '유효', 유효하지 않으면 '오류', 'quit'이면 종료한다.

```
str.count(), str.find(), str.rfind()
```

```
In [ ]: # 코드 예제
while True :
    email = input("이메일 주소를 입력하세요: ")

    if email == 'quit' :
        break
    else :
        # 코딩하시오.
        #
```

#  
#

## LAB 2 : URL에서 도메인 이름 추출

1. 문제 : 사용자로 부터 URL을 입력 받아 도메인 이름을 추출하 프로그램을 작성하세요.
2. 입출력
  - 입력(URL) : <https://www.google.com>, <http://www.dongyang.ac.kr>,  
<http://www.dongyang.ac.kr/index.jsp>
  - 출력 : google.com, dongyang.ac.kr
3. 사용자로 부터 입력 받은 이메일 주소가 유효하면 '유효', 유효하지 않으면 '오류', 'quit'이면 종료한다.

```
str.split(), str.replace()
```

```
In [ ]: # 코드 예제
url = input("URL을 입력하세요 (예: https://www.google.com): ")

# 코딩하시오
#
#

print("도메인 이름:", domain)
```

## LAB 3 : 문자열 암호화

1. 문제 : 사용자로 부터 문자열을 입력 받아 다음의 규칙으로 암호화하세요
  - 각 문자의 ASCII 값을 2씩 증가시킨다.
  - 결과 문자열을 반환한다.
2. 입출력
  - 입력) : "abc"
  - 출력 : "cde"

```
ord(), chr()
```

```
In [ ]: # 코드 예제
text = input("문자열을 입력하세요: ")

encrypted = []
for char in text :
    # 코딩하시오.
    #
```

```
''.join(encrypted)  
print("암호화된 문자열:", ''.join(encrypted))
```

In [ ]: