

문자열 (String)

- 1. 문자열 슬라이싱
- 2. 문자열 메서드
- 3. 아스키(ASCII) 코드

문자열은 **immutable**(변경 불가능한) 자료형 !

```
word = "apple"  
print(word)  
print(id(word))  
>>> apple  
>>> 1352749370800
```

```
word += " banana"  
print(word)  
print(id(word))  
>>> apple banana  
>>> 1352749417520
```

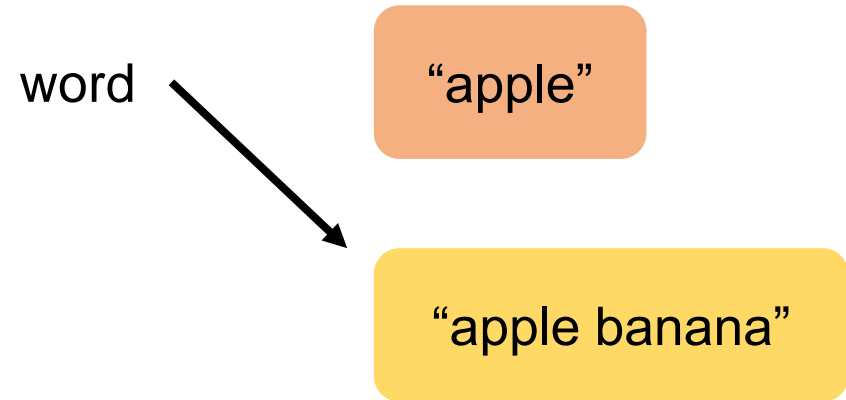
word →

"apple"

문자열은 **immutable**(변경 불가능한) 자료형 !

```
word = "apple"
print(word)
print(id(word))
>>> apple
>>> 1352749370800

word += " banana"
print(word)
print(id(word))
>>> apple banana
>>> 1352749417520
```





1. 문자열 슬라이싱

1. 문자열 슬라이싱

s = 'abcdefghi'

	a	b	c	d	e	f	g	h	i
index	0	1	2	3	4	5	6	7	8
index	-9	-8	-7	-6	-5	-4	-3	-2	-1

1. 문자열 슬라이싱

s = 'abcdefghi'

- s[2:5] □ 'cde'

	a	b	c	d	e	f	g	h	i
index	0	1	2	3	4	5	6	7	8
index	-9	-8	-7	-6	-5	-4	-3	-2	-1

1. 문자열 슬라이싱

s = 'abcdefghi'

- s[2:5] □ 'cde'
- s[-6:-2] □ 'defg'

	a	b	c	d	e	f	g	h	i
index	0	1	2	3	4	5	6	7	8
index	-9	-8	-7	-6	-5	-4	-3	-2	-1

1. 문자열 슬라이싱

s = 'abcdefghi'

- s[2:5] □ 'cde'
- s[-6:-2] □ 'defg'
- s[2:-4] □ 'cde'

	a	b	c	d	e	f	g	h	i
index	0	1	2	3	4	5	6	7	8
index	-9	-8	-7	-6	-5	-4	-3	-2	-1

1. 문자열 슬라이싱

s = 'abcdefghi'

- s[2:5:2] □ 'ce'

	a	b	c	d	e	f	g	h	i
index	0	1	2	3	4	5	6	7	8
index	-9	-8	-7	-6	-5	-4	-3	-2	-1

1. 문자열 슬라이싱

s = 'abcdefghi'

- s[2:5:2] □ 'ce'
- s[-6:-1:3] □ 'dg'

	a	b	c	d	e	f	g	h	i
index	0	1	2	3	4	5	6	7	8
index	-9	-8	-7	-6	-5	-4	-3	-2	-1

1. 문자열 슬라이싱

s = 'abcdefghi'

- s[2:5:2] □ 'ce'
- s[-6:-1:3] □ 'dg'
- s[2:5:-1] □ ''

	a	b	c	d	e	f	g	h	i
index	0	1	2	3	4	5	6	7	8
index	-9	-8	-7	-6	-5	-4	-3	-2	-1

1. 문자열 슬라이싱

s = 'abcdefghi'

- s[2:5:2] □ 'ce'
- s[-6:-1:3] □ 'dg'
- s[2:5:-1] □ ''
- s[5:2:-1] □ 'fed'



	a	b	c	d	e	f	g	h	i
index	0	1	2	3	4	5	6	7	8
index	-9	-8	-7	-6	-5	-4	-3	-2	-1

1. 문자열 슬라이싱

s = 'abcdefghi'

- s[:3] □ 'abc'
- s[5:] □ 'fghi'

	a	b	c	d	e	f	g	h	i
index	0	1	2	3	4	5	6	7	8
index	-9	-8	-7	-6	-5	-4	-3	-2	-1

1. 문자열 슬라이싱

s = 'abcdefghi'

- s[:3] □ 'abc'
- s[5:] □ 'fghi'
- s[:] □ 'abcdefghi'

	a	b	c	d	e	f	g	h	i
index	0	1	2	3	4	5	6	7	8
index	-9	-8	-7	-6	-5	-4	-3	-2	-1

1. 문자열 슬라이싱

s = 'abcdefghi'

- s[:3] □ 'abc'
- s[5:] □ 'fghi'
- s[:] □ 'abcdefghi'
- s[::-1] □ 'ihgfedcba'

	a	b	c	d	e	f	g	h	i
index	0	1	2	3	4	5	6	7	8
index	-9	-8	-7	-6	-5	-4	-3	-2	-1



1. 문자열 슬라이싱

s = 'abcdefghi'

- s[:3] □ 'abc'
 - s[5:] □ 'fghi'
 - s[:] □ 'abcdefghi'
 - s[::-1] □ 'ihgfedcba'
- s[10:20] □ ''

	a	b	c	d	e	f	g	h	i
index	0	1	2	3	4	5	6	7	8
index	-9	-8	-7	-6	-5	-4	-3	-2	-1



문자열 슬라이싱 연습

문제 번호	문제	링크
BOJ 10988	팰린드롬인지 확인하기	https://www.acmicpc.net/problem/10988



2. 문자열 메서드

- 1) `.split()`
- 2) `.strip()`
- 3) `.find()`
- 4) `.index()`
- 5) `.count()`
- 6) `.replace()`
- 7) `.join()`

1) .split(기준 문자)

문자열을 일정 **기준**으로 나누어 **리스트로 반환**

괄호 안에 아무것도 넣지 않으면 자동으로 공백을 기준으로 설정

```
word = "I play the piano"  
print(word.split())
```

['I', 'play', 'the', 'piano']

```
word = "apple,banana,orange,grape"  
print(word.split(","))
```

['apple', 'banana', 'orange', 'grape']

```
word = "This_is_snake_case"  
print(word.split("_"))
```

['This', 'is', 'snake', 'case']

2) .strip(제거할 문자)

문자열의 양쪽 끝에 있는 특정 문자를 모두 제거한 새로운 문자열 반환
괄호 안에 아무것도 넣지 않으면 자동으로 공백을 제거 문자로 설정
제거할 문자를 여러 개 넣으면 해당하는 모든 문자들을 제거

```
word = " Hello World "  
print(word.strip())
```

Hello World

```
word = "aHello Worlda"  
print(word.strip("a"))
```

Hello World

```
word = "Hello World"  
print(word.strip("Hd"))
```

ello Worl

```
word = "Hello Worlddddddd"  
print(word.strip("d"))
```

Hello Worl

3) .find(찾는 문자)

특정 문자가 처음으로 나타나는 **위치(인덱스)**를 반환

찾는 문자가 없다면 **-1**을 반환

```
word = "apple"
print(word.find("p"))
```

1

```
word = "apple"
print(word.find("k"))
```

-1

4) .index(찾는 문자)

특정 문자가 처음으로 나타나는 **위치(인덱스)**를 반환

찾는 문자가 없다면 **오류** 발생

```
word = "apple"
print(word.index("p"))
```

1

```
word = "apple"
print(word.index("k"))
```

```
-----
ValueError
Input In [21], in <cell> 1
      1 word = "apple"
----> 3 print(word.index("k"))
ValueError: substring not
```

5) .count(개수를 셀 문자

)
문자열에서 특정 문자가 **몇 개**인지 반환
문자 뿐만 아니라, 문자열의 개수도 확인 가능

```
word = "banana"  
print(word.count("a"))
```

3

```
word = "banana"  
print(word.count("na"))
```

2

```
word = "banana"  
print(word.count("ana"))
```

1

6) .replace(기존 문자, 새로운 문자)

문자열에서 기존 문자를 새로운 문자로 수정한 새로운 문자열 반환

특정 문자를 빈 문자열("")로 수정하여 마치 해당 문자를 삭제한 것 같은 효과가
능

```
word = "happyhacking"  
print(word.replace("happy", "angry"))
```

angryhacking

```
word = "happyhacking"  
print(word.replace("h", "H"))
```

HappyHacking

```
word = "happyhacking"  
print(word.replace("happy", ""))
```

hacking

7) 삽입할 문자.join(*iterable*)

*iterable*의 각각 원소 사이에 특정 문자를 삽입한 새로운 문자열 반환
공백 출력, 콤마 출력 등 원하는 출력 형태를 위해 사용

```
word = "happyhacking"  
print(" ".join(word))
```

h a p p y h a c k i n g

```
word = "happyhacking"  
print(", ".join(word))
```

h,a,p,p,y,h,a,c,k,i,n,g

```
word = ["edu", "hphk.kr"]  
print("@".join(word))
```

edu@hphk.kr

```
words = ["h", "a", "p", "p", "y"]  
print("").join(words)
```

happy

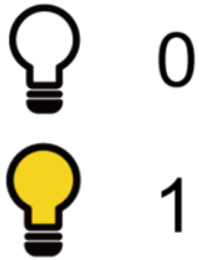
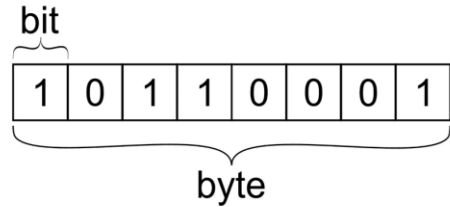
문자열 메서드 연습

문제 번호	문제	링크
BOJ 17249	태보태보 총난타	https://www.acmicpc.net/problem/17249

3. 아스키(ASCII) 코드

3. 아스키(ASCII) 코드

컴퓨터는 숫자만 이해할 수 있다 !

비트(bit)	바이트(byte)
0과 1 두 가지 정보만 표현	데이터를 저장하는 기본 단위 1 byte == 8 bits
	

그렇다면 문자는 어떻게 저장될까?

그렇다면 문자는 어떻게 저장될까?

ASCII

(American Standard Code for Information Interchange)

미국 정보교환 표준부호

3. 아스키(ASCII) 코드

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Source: www.asciitable.com

아스키(ASCII) 코드란?

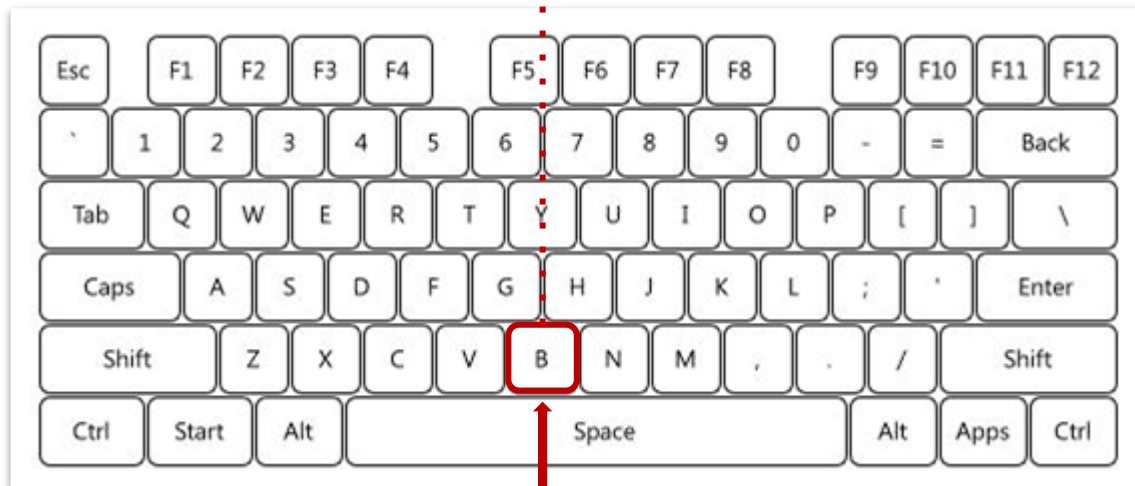
알파벳을 표현하는 대표 인코딩 방식

각 문자를 표현하는데 1byte(8bits) 사용

- 1bit : 통신 에러 검출용
- 7bit : 문자 정보 저장 (총 128개)

3. 아스키(ASCII) 코드

아스키코드의 66번 위치의 문자인 B를 찾아서 보여줌



클릭

Dec	Hx	Oct	Html	Chr
64	40	100	@	@
65	41	101	A	A
66	42	102	B	B
67	43	103	C	C
68	44	104	D	D
69	45	105	E	E
70	46	106	F	F
71	47	107	G	G
72	48	110	H	H
73	49	111	I	I
74	4A	112	J	J
75	4B	113	K	K
76	4C	114	L	L
77	4D	115	M	M
78	4E	116	N	N
79	4F	117	O	O
80	50	120	P	P
81	51	121	Q	Q
82	52	122	R	R
83	53	123	S	S
84	54	124	T	T
85	55	125	U	U
86	56	126	V	V
87	57	127	W	W
88	58	130	X	X
89	59	131	Y	Y
90	5A	132	Z	Z

3. 아스키(ASCII) 코드

1) ord(문자)

문자 □ 아스키코드로 변환하는 내장함수

```
print(ord("A"))
```

65

```
print(ord("a"))
```

97

2) chr(아스키코드)

아스키코드 □ 문자로 변환하는 내장함수

```
print(chr(65))
```

A

```
print(chr(97))
```

a

3. 아스키(ASCII) 코드

아스키(ASCII) 코드 연습

문제 번호	문제	링크
BOJ 10809	알파벳 찾기	https://www.acmicpc.net/problem/10809