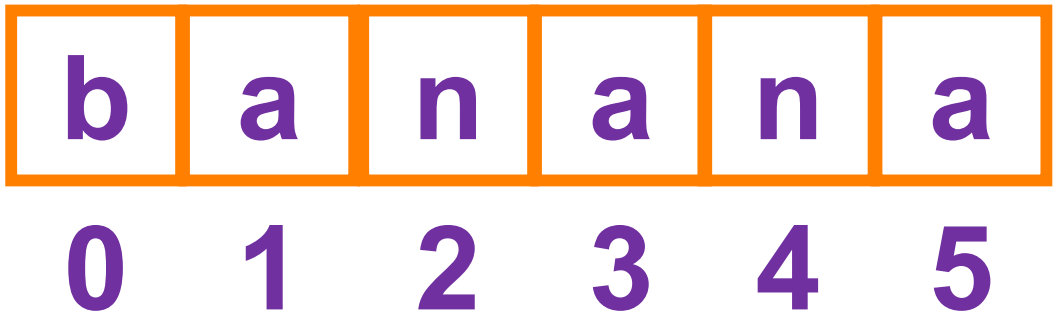
บทที่ 7 ข้อความ (String)

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

What is string?



- Sequence of characters.
- Immutable
- เป็นข้อมูลพื้นฐาน
- ประกอบด้วย ตัวอักษร ตัวเลข สัญลักษณ์ต่าง ๆ
- เก็บข้อมูลในรูปแบบรหัสแอสกี้ หรือยูนิโค้ด (Unicode)
- สามารถมี Escape character
- Docstring

การสร้างสตริง

- Single quote
- Double quote
- Raw string
- F-string
- Docstring
  - Triple-single quote
  - Triple-double quote

Special Characters

- Backslash (\\)
- Single quote (\')
- Double quote (\")
- Newline (\n)
- Carriage return (\r)
- Tab (\t)
- Backspace (\b)
- Null character(\0)
- Unicode character

```
('\u' or '\U')
```

Escape character examples.


```
print("Hello\nworld!")
```

```
Hello
world!
```

```
print("Name\tAge\tCountry")
print("Alice\t25\tUSA")
print("Bob\t30\tCanada")
```

```
Name    Age     Country
Alice   25      USA
Bob     30      Canada
```

```
print('He said, \'I\'ll be back.\'')
```

```
He said, 'I'll be back.'
```

```
print("She said, \"I'm leaving now.\"")
```

```
She said, "I'm leaving now.
```

```
print("This is a backslash: \\")
```

```
This is a backslash: \
```

String operators

- Concatenation (+)
- Repetition (*)
- Indexing (string[index])
- Slicing (string[start:stop:step])
- Comparison
  - Equal (==)
  - Inequality (!=)
  - Greater than (>)
  - Less than (<)
  - Greater than or equal (>=)
  - Less than or equal (<=)
- In and not in
- Formatting (%)
- Format function
- F-string

String operations (+ , *)


```
>>> greeting = "Hello, "
>>> subject = "world!"
>>> message = greeting + subject
>>> print(message)
Hello, world!
>>> message = greeting*2
>>> print(message)
Hello, Hello,
>>> message = subject*3
>>> print(message)
world!world!world!
>>> message = greeting*2 + subject*3
>>> print(message)
Hello, Hello, world!world!world!
```

String operations ([])


```
>>> word = "world"
>>> first_letter = word[0]
>>> last_letter = word[-1]
>>> print(first_letter)
w
>>> print(last_letter)
d
>>> print(word[3])
l
>>> print(word[-3])
r
```

String slicing (string[start:stop:step])


```
>>> #       '0123456789012345678901234567890'
>>> name = "Python creator: Gudio van Rossum"
>>> name[0:6]
'Python'
>>> name[:6]
'Python'
>>> name[7:14]
'creator'
>>> name[16:21]
'Gudio'
>>> name[15:21]
' Gudio'
>>> name[26:]
'Rossum’
>>> name[-6:]
'Rossum'
```

String slicing (string[start:stop:step])


```
>>> #       '0123456789012345678901234567890123'
>>> name = "Python creator: Gudio van Rossum"
>>> name[7:-6]
'creator: Gudio van '
>>> name[::-1]
'mussoR nav oiduG :rotaerc nohtyP'
>>> name[-10:-7]
'van'
>>> name[-32:-7]
'Python creator: Gudio van'
>>> name[-32:25]
'Python creator: Gudio van’
>>> name[7:-6:2]
'cetr ui a '
```

String comparison operators


```
>>> word1 = "Hello"
>>> word2 = "hello"
>>> word3 = "KMITL"
>>> word1 == word2
False
>>> word1 > word2
False
>>> word1 < word2
True
>>> word1 != word3
True
>>> word1 >= word3
False
>>> word1 <= word3
True
```

String f-string


```
>>> name = "Alice"
>>> age = 25
>>> height = 1.65
>>> formatted_string = f"My name is {name}, I am {age} years old, and
I am {height:.2f} meters tall."
>>> print(formatted_string)
My name is Alice, I am 25 years old, and I am 1.65 meters tall.

>>> a = 10
>>> b = 20
>>> formatted_string = f"The sum of {a} and {b} is {a + b}."
>>> print(formatted_string)
The sum of 10 and 20 is 30.
```

String format method


```
# Keyword arguments
>>> template2 = "The {animal} is {adjective} and {adverb}."
>>> formatted_string2 = template2.format(animal="cat",
adjective="cute", adverb="playful")
>>> print(formatted_string2)
The cat is cute and playful.

# Mixed arguments
>>> template3 = "{0} {1} {2} {3} {0}."
>>> formatted_string3 = template3.format("to", "be", "or", "not")
>>> print(formatted_string3)
to be or not to.
# Formatting options
>>> pi = 3.14159269793
>>> formatted_string4 = "Pi is approximately {:.2f}.".format(pi)
>>> print(formatted_string4)
Pi is approximately 3.14.
```

String formatting (%)


```
>>> name = "Alice"
>>> age = 25
>>> formatted_string = "My name is %s, and I am %d years old." % (name, age)
>>> print(formatted_string)
My name is Alice, and I am 25 years old.

>>> num1 = 10
>>> num2 = 3.14159
>>> formatted_string = "The value of num1 is %d, and the value of num2 is
%.2f." % (num1, num2)
>>> print(formatted_string)
The value of num1 is 10, and the value of num2 is 3.14.

>>> name = "Bob"
>>> age = 30
>>> formatted_string = "My name is %s, and I am %d years old." % (name, age)
>>> print(formatted_string)
My name is Bob, and I am 30 years old.
```

String formatting (%)


```
>>> num = 255
>>> formatted_string = "The decimal value of %d is %x in
hexadecimal." % (num, num)
>>> print(formatted_string)
The decimal value of 255 is ff in hexadecimal.

>>> num = 42
>>> formatted_string = "The answer is %04d." % num
>>> print(formatted_string)
The answer is 0042.
```

Common string methods

- lower
- upper
- capitalize
- title
- strip
- rstrip
- lstrip
- join
- startswith
- endswith
- find
- count
- isdigit
- isalpha
- isalnum
- replace

String methods (lower,upper,title,capitalize)


```
>>> word = "KMItL"
>>> print(word.lower())
kmitl
>>> print(word.upper())
KMITL
>>> print(word.capitalize())
Kmitl
>>> print(word.title())
Kmitl
>>> word = 'KMITL'
>>> sep = '-'
>>> k_new = sep.join(word)
>>> k_new
'K-M-I-T-L'
```

String methods (startswith, endswith)


```
>>> email = 'programming@kmitl.ac.th'
>>> email.startswith('pro')
True
>>> email.startswith('com')
False
>>> email.endswith('pro')
False
>>> email.endswith('.ac.th')
True
```

String methods (find, count)


```
>>> email = 'programming@kmitl.ac.th'
>>> email.find('th')
21
>>> email.find('in')
8
>>> email.find('i')
8
>>> email.count('i')
2
>>> email.count('.')
2
>>> email.count('a')
2
>>> email.count('t')
2
```

String methods (isdigit)


```
>>> string = '123'
>>> string.isdigit()
True
>>> string = '123_456'
>>> string.isdigit()
False
>>> string = '0x45'
>>> string.isdigit()
False
>>> string = '1e8'
>>> string.isdigit()
False
>>> string = '3.14'
>>> string.isdigit()
False
```

String methods (isalpha)


```
>>> string = '123'
>>> string.isalpha()
False
>>> string = 'kmitl'
>>> string.isalpha()
True
>>> string = 'hello kmitl'
>>> string.isalpha()
False
>>> string = '1e8'
>>> string.isalpha()
False
>>> string = 'HelloKmitl'
>>> string.isalpha()
True
>>> string = 'HelloKmitl!'
>>> string.isalpha()
False
```

String methods (isalnum)


```
>>> string ='123'
>>> string.isalnum()
True
>>> string ='123abc'
>>> string.isalnum()
True
>>> string ='123 abc'
>>> string.isalnum()
False
>>> string ='3.1415926'
>>> string.isalnum()
False
>>> string ='1e5'
>>> string.isalnum()
True
```

String methods (replace)


```
>>> x ='The quick brown fox jumps over a lazy dog'
>>> x
'The quick brown fox jumps over a lazy dog'
>>> x.replace('a','A')
'The quick brown fox jumps over A lAzy dog'
>>> x.replace('brown','magenta')
'The quick magenta fox jumps over a lazy dog'
>>> x.replace('a',' HOLY ')
'The quick brown fox jumps over  HOLY  l HOLY zy dog'
```

String methods (others)


```
>>> x = [m for m in dir('m') if m[0:2]!='__']
>>> x
['capitalize', 'casefold', 'center', 'count', 'encode', 'endswith',
'expandtabs', 'find', 'format', 'format_map', 'index', 'isalnum',
'isalpha', 'isascii', 'isdecimal', 'isdigit', 'isidentifier',
'islower', 'isnumeric', 'isprintable', 'isspace', 'istitle',
'isupper', 'join', 'ljust', 'lower', 'lstrip', 'maketrans',
'partition', 'removeprefix', 'removesuffix', 'replace', 'rfind',
'rindex', 'rjust', 'rpartition', 'rsplit', 'rstrip', 'split',
'splitlines', 'startswith', 'strip', 'swapcase', 'title',
'translate', 'upper', 'zfill']
```

String summary

- string literal
- String operators (+, *)
- String assignment
- Immutable
- string slicing
- String formatting
  - F-string
  - Format method
  - Formatting (%)
- Comparison operators

  >, <, ==, != , <=, >=
- Operator in, is ,is not

Method

- lower
- upper
- capitalize
- title
- strip
- rstrip
- lstrip
- join
- starstwith
- endswith
- find
- count
- isdigit
- isalpha
- isalnum
- replace