# STRINGS

1. Accessing a character in a string
2. Getting the length of a string
3. Creating a substring from existing string
4. Checking if a string is in another string
5. String methods

# 1. Accessing a character in a string

```
# The string is indexed (numbered). The index starts with 0 for the first character.

# Accessing the first character
breakfast = ‘Spam and Eggs’
breakfast[0]   # returns ‘S’

breakfast[3]   # ‘m’

breakfast[4]   # ‘ ’
breakfast[-1]  # ‘s’
breakfast[-4]  # ‘E’
```

# 2. Getting the length of a string

```
# The len() function returns the length of
# a sequence (i.e. string, bytes, tuple, list or range) or
# a collection (i.e. dictionary, set, frozen set)

breakfast = ‘Spam and Eggs’
len(breakfast)          # 13

full_name = ‘Princess Diana of Themyscira, Daughter of Hippolyta’
len(full_name)          # 51

hero_name = ‘Wonder Woman’
len(hero_name)          # 12

no_name = ‘’
len(no_name)            # 0
```

# 3. Creating a substring from existing string

```
full_name = ‘Princess Diana of Themyscira, Daughter of Hippolyta’
full_name[9:15]       # returns substring starting from index 9 of the string
                      # to 15 exclusive i.e. ‘Diana’

# Substring from the beginning of the string to index 10 exclusive
full_name[:10]        # ‘Princess D’

# Substring from index 25 to the end of the string
full_name[25:]        # ‘ira, Daughter of Hippolyta’

# Substrings are strings, so they can be concatenated like any string
‘full_name[42:] + ‘\’s ’ + full_name[30:39] + ‘is’ + full_name[8:14]

# Experiment: Test the above code on the Python shell to. What is the resultant string?
```

# 4. Checking if a string is in another string

```
full_name = ‘Princess Diana of Themyscira, Daughter of Hippolyta’

‘Diana’ in full_name           # True
‘Hippocrates’ in full_name     # False
‘The’ in full_name             # True
```

# 5. String methods

Strings are ***immutable***. It means that once created, the content of a string can not be modified. Notice that all string methods do not modify the original string. When a modification occurs, it is done on a copy of the string, which is then returned by the method.

The following are common string methods:

| Method | Description | Returns |
|---|---|---|
| capitalize() | Returns a capitalized version of the string. | str |
| casefold() | Returns a version of the string suitable for caseless comparison. | str |
| lower() | Returns a copy of the string converted to lowercase. | str |
| title() | Returns a titlecased version of the string. | str |
| swapcase() | Returns a copy of the string with uppercase characters converted to lowercase and vice versa. | |
| upper() | Returns a copy of the string converted to uppercase. | str |

# 5. String methods

| Method | Description | Returns |
|---|---|---|
| center(width[, fillchar]) | Returns the string centered and padded with optional fillchar. The total length of returned string is of length width. Default fillchar is a space. | str |
| ljust(width[, fillchar]) | Returns left-justified string of length width. Padding is done using the specified fill character (default is a space). | str |
| rjust(width[, fillchar]) | Returns right-justified string of length width. Padding is done using the specified fill character (default is a space). | str |
| zfill (width) | Pads the string with zeros on the left, to fill a field of the specified width. The string is never truncated. | str |

# 5. String methods

| Method | Description | Returns |
|---|---|---|
| count(sub[,start[,end]]) | Returns the number of non-overlapping occurrences of substring sub in string[start:end]. | int |
| find(sub[,start[,end]]) | Returns the lowest index in the string where substring sub is found, such that sub is contained within [start:end]. Returns -1 on failure. | int |
| index(sub[,start[,end]]) | Returns the lowest index in the string where substring sub is found, such that sub is contained within [start:end]. Raises ValueError when the substring is not found. | int |
| rfind(sub[,start[,end]]) | Returns the highest index in the string where substring sub is found, such that sub is contained within [start:end]. Returns -1 on failure. | int |
| rindex(sub[,start[,end]]) | Returns the highest index in the string where substring sub is found, such that sub is contained within [start:end]. Raises ValueError when the substring is not found. | int |

# 5. String methods

| Method | Description | Returns |
|---|---|---|
| endswith(suffix[, start[, end]]) | Returns True if the string ends with specified suffix, False otherwise. start and end are optional positions on the string on which suffix is compared. suffix can also be a tuple of strings to try. | bool |
| startswith(prefix[, start[, end]]) | Returns True if the string starts with specified prefix, False otherwise. start and end are optional positions on the string on which prefix is compared. prefix can also be a tuple of strings to try. | bool |

# 5. String methods

| Method | Description | Returns |
|---|---|---|
| isalnum() | Returns True if all characters in the string are alphanumeric, and there is at least one character in the string. False otherwise. | bool |
| isalpha() | Returns True if all characters in the string are alphabetic and there is at least one character in the string. False otherwise. | bool |
| isdecimal() | Returns True if there are only decimal characters in the string. False otherwise. | bool |
| isdigit() | Returns True if all characters in the string are digits and there is at least one character in the string. False otherwise. | bool |
| isidentifier() | Returns True if the string is a valid identifier according to the language definition. Use keyword.iskeyword() to test for reserved identifiers such as "def" and "class". | bool |
| islower() | Returns True if all cased characters in the string are lowercase and there is at least one cased character in the string. False otherwise. | bool |

# 5. String methods

| Method | Description | Returns |
|---|---|---|
| isnumeric() | Returns True if there are only numeric characters in the string, False otherwise. | bool |
| isprintable() | Returns True if all characters in the string are considered printable in repr() or the string is empty. False otherwise. | bool |
| isspace() | Returns True if all characters in the string are whitespace and there is at least one character in the string. False otherwise. | bool |
| istitle() | Returns True if the string is a titlecased string and there is at least one character in it, i.e. upper- and titlecase characters may only follow uncased characters and lowercase characters only cased ones. Returns False otherwise. | bool |
| isupper() | Returns True if all cased characters in the string are uppercase and there is at least one cased character in it. False otherwise. | bool |

# 5. String methods

| Method | Description | Returns |
|---|---|---|
| strip([chars]) | Returns a copy of the string with leading and trailing whitespace removed. If chars is given and not None, remove characters in chars instead. | str |
| lstrip([chars]) | Returns a copy of the string with leading whitespace removed. If chars is given and not None, remove characters in chars instead. | str |
| rstrip([chars]) | Returns a copy of the string with trailing whitespace removed. If chars is given and not None, remove characters in chars instead. | str |
| expandtabs(tabsize=8) | Returns a copy of the string where all tab characters are expanded using spaces. If tabsize is not given, a tab size of 8 characters is assumed. | str |
| replace(old, new[, count]) | Returns a copy of the string with all occurrences of substring old replaced by new. If the optional argument count is given, only the first count occurrences are replaced. | str |

# 5. String methods

| Method | Description | Returns |
|---|---|---|
| split(sep=None, maxsplit=-1) | Returns a list of words in the string, using sep as the delimiter string. If maxsplit is given, at most maxsplit splits are done. If sep is not specified or is None, any whitespace string is a separator and empty strings are removed from the result. | List of strings |
| rsplit(sep=None, maxsplit=-1) | Returns a list of words in the string, using sep as the delimiter string, starting at the end of the string and working to the front.bIf maxsplit is given, at most maxsplit splits are done. If sep is not specified, any whitespace string is a separator. | List of strings |
| splitlines([keepends]) | Returns a list of lines in the string, breaking at line boundaries. Line breaks are not included in the resulting list unless keepends is given and true. | List of strings |
| join(iterable) | Returns a string which is the concatenation of the strings in the iterable. The separator between elements is the string. | str |

# 5. String methods

| Method | Description | Returns |
|---|---|---|
| format(*args, **kwargs) | Returns a formatted version of the string, using substitutions from args and kwargs. The substitutions are identified by braces ('{' and '}'). | str |
| format_map(mapping) | Return a formatted version of the string, using substitutions from mapping. The substitutions are identified by braces ('{' and '}'). | str |

# 5. String methods

| Method | Description | Returns |
|---|---|---|
| translate(table) | Returns a copy of the string in which each character has been mapped through the given translation table. The table must implement lookup/indexing via __getitem__, for instance a dictionary or list, mapping Unicode ordinals to Unicode ordinals, strings, or None. If this operation raises LookupError, the character is left untouched. Characters mapped to None are deleted. | str |
| partition(sep) | Searches for the separator sep in the string, and returns the part before it, the separator itself, and the part after it. If the separator is not found, returns the string and two empty strings. | (head, sep, tail) |
| rpartition(sep) | Search for the separator sep in the string, starting at the end of S, and returns the part before it, the separator itself, and the part after it. If the separator is not found, returns two empty strings and the string. | (head, sep, tail) |

# 5. String methods

| Method | Description | Returns |
| --- | --- | --- |
| encode(encoding='utf-8', errors='strict') | Encodes the string using the codec registered for encoding. Default encoding is 'utf-8'. errors may be given to set a different error handling scheme. Default is 'strict' meaning that encoding errors raise a UnicodeEncodeError. Other possible values are 'ignore', 'replace' and 'xmlcharrefreplace' as well as any other name registered with codecs.register_error that can handle UnicodeEncodeErrors. | bytes |