String Methods COMS 261

Here are some of the most commonly used string methods in Python.

Method	What it does	Example
`strip()`	Removes characters from the ends	`" hi ".strip()`
`split()`	Splits a string into a list	`"a,b".split(",")`
`join()`	Uses a string to join strings from an iterable	`",".join(["a","b"])`
`replace()`	Replaces all occurrences of a substring	`"cat".replace("c","b")`

1. The `strip` method removes either whitespace (spaces, newlines, returns) or optionally specified characters from the *beginning and end* of a string. It does not remove those characters from the middle. What does each expression produce?

   (a) `"  spaces ".strip()`

   (b) `"www.example.com".strip("w.com")`

   (c) `"  hello world ".strip().replace(" ", "_")`

2. The `split` method has an optional argument to specify the separator. When set to None (the default value), it will split on any whitespace character (including `\n` (newline), `\r` (return), `\t` (tab), and spaces) and will discard empty strings from the result.

   (a) What is the value of `"one  two   three".split()`?

   (b) What is the value of `"one  two   three".split(" ")`? Why is it different than (a)?

   (c) What does `"2026-11-01".split("-")` produce?

3. The `join` method uses the target string (the string before the `.join()` method call) to glue together strings from an iterable into a single string. Why does the following code produce an error? Rewrite it so that it prints `1-2-3-4`.

```
numbers = [1, 2, 3, 4]
print("-".join(numbers))
```

4. Write code to solve the following problems using *one line* of Python.

   (a) Given a string like `date = "11/01/2026"`, create a list containing the month, day, and year as three separate strings.

   (b) Given a string like `s = "The quick brown fox"`, create a variable called `last_word` containing just the final word in `s`.

   (c) Convert a string like `fake_list = "[1, 2, 3]"` into a valid Python list with integer values. Hint: Use a list comprehension to get a list of integers at the end.

5. Write a `word_count` function that inputs a string and returns how many words are in the string. For example, `word_count("The quick brown fox")` should return 4.

6. Write a function `censor(text, bad_word)` that returns a copy of `text` with every occurrence of `bad_word` replaced by a string of asterisks of the same length. You may assume that `bad_word` is not empty. For example, `censor("I fart in your general direction.", "fart")` should return `"I **** in your general direction."`.

7. Write a function `oxford(item)` that inputs a list of strings and returns a single string with the items formatted using an Oxford comma. For example,

```
oxford([])                    # ""
oxford(["a"])                 # "a"
oxford(["a", "b"])            # "a and b"
oxford(["a", "b", "c"])       # "a, b, and c"
oxford(["a", "b", "c", "d"])  # "a, b, c, and d"
```

   Hint: You'll need to handle the cases when there are less than 2 items, exactly 2 items, and more than 2 items separately.