

Reversing a String Done Recursively

Matt Bishop

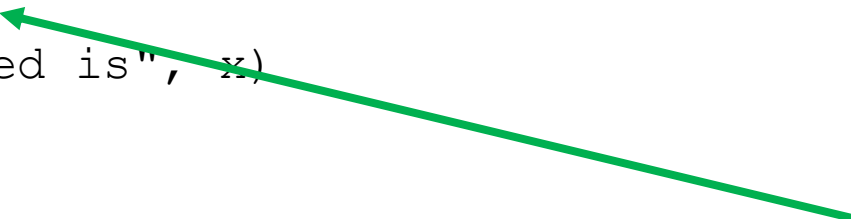
MHI 289I, Fall Quarter 2025

The Recursive String Reverse Program

```
1: def rev(s):
2:     # base case: empty string is its reverse
3:     if s == '':
4:         return ''
5:     # recursion: peel off last char, reverse
6:     # the rest, prepend last char
7:     return s[-1] + revstr(s[:-1])
8:
9: x = revstr("abc")
10: print("abc reversed is", x)
```

Initial call to revstr at line 9: revstr(s ← "abc")

```
1: def rev(s):
2:     # base case: empty string is its reverse
3:     if s == '':
4:         return ''
5:     # recursion: peel off last char, reverse
6:     # the rest, prepend last char
7:     return s[-1] + revstr(s[:-1])
8:
9: x = revstr("abc")
10: print("abc reversed is", x)
```



revstr("abc"): return to line 9
s = "abc"

```
revstr(s ← "abc"):
    3: condition false, so skip
    7: call revstr("ab")
```

```
1: def rev(s):
2:     # base case: empty string is its reverse
3:     if s == '':
4:         return ''
5:     # recursion: peel off last char, reverse
6:     # the rest, prepend last char
7:     return s[-1] + revstr(s[:-1])
8:
9: x = revstr("abc")
10: print("abc reversed is", x)
```

revstr("ab"): return to line 7
s = "ab"

revstr("abc"): return to line 9
s = "abc"

revstr(s ← "ab"):
 3: condition false, so skip
 7: call revstr("a")

```
1: def rev(s):  
2:     # base case: empty string is its reverse  
3:     if s == '':  
4:         return ''  
5:     # recursion: peel off last char, reverse  
6:     # the rest, prepend last char  
7:     return s[-1] + revstr(s[:-1])  
8:  
9: x = revstr("abc")  
10: print("abc reversed is", x)
```

revstr("a"): return to line 7
 s = "a"

revstr("ab"): return to line 7
 s = "ab"

revstr("abc"): return to line 9
 s = "abc"

```
revstr(s ← "a"):
    3: condition false, so skip
    7: call revstr("")
```

```
1: def rev(s):
2:     # base case: empty string is its reverse
3:     if s == '':
4:         return ''
5:     # recursion: peel off last char, reverse
6:     # the rest, prepend last char
7:     return s[-1] + revstr(s[:-1])
8:
9: x = revstr("abc")
10: print("abc reversed is", x)
```

revstr(""): return ""
s = ""

revstr("a"): return to line 7
s = "a"

revstr("ab"): return to line 7
s = "ab"

revstr("abc"): return to line 9
s = "abc"

```
revstr(s ← ""):  
    3: condition true, so return "
```

```
1: def rev(s):  
2:     # base case: empty string is its reverse  
3:     if s == '':  
4:         return ''  
5:     # recursion: peel off last char, reverse  
6:     # the rest, prepend last char  
7:     return s[-1] + revstr(s[:-1])  
8:  
9: x = revstr("abc")  
10: print("abc reversed is", x)
```

revstr(""): return ""
s = ""

revstr("a"): return to line 7
s = "a"

revstr("ab"): return to line 7
s = "ab"

revstr("abc"): return to line 9
s = "abc"

```
revstr(s ← "a"):
    7: return "a" + "" = "a"
```

```
1: def rev(s):
2:     # base case: empty string is its reverse
3:     if s == '':
4:         return ''
5:     # recursion: peel off last char, reverse
6:     # the rest, prepend last char
7:     return s[-1] + revstr(s[:-1])
8:
9: x = revstr("abc")
10: print("abc reversed is", x)
```

revstr(""): return ""

revstr("a"): return "a" to line 7
s = "a"

revstr("ab"): return to line 7
s = "ab"

revstr("abc"): return to line 9
s = "abc"

```
revstr(s ← "ab"):
    7: return "b" + "a" = "ba"
```

```
1: def rev(s):
2:     # base case: empty string is its reverse
3:     if s == '':
4:         return ''
5:     # recursion: peel off last char, reverse
6:     # the rest, prepend last char
7:     return s[-1] + revstr(s[:-1])
8:
9: x = revstr("abc")
10: print("abc reversed is", x)
```

revstr(""): return "" to blue

s = ""

revstr("a"): return "a" to line 7

s = "a"

revstr("ab"): return "ba" to line 7

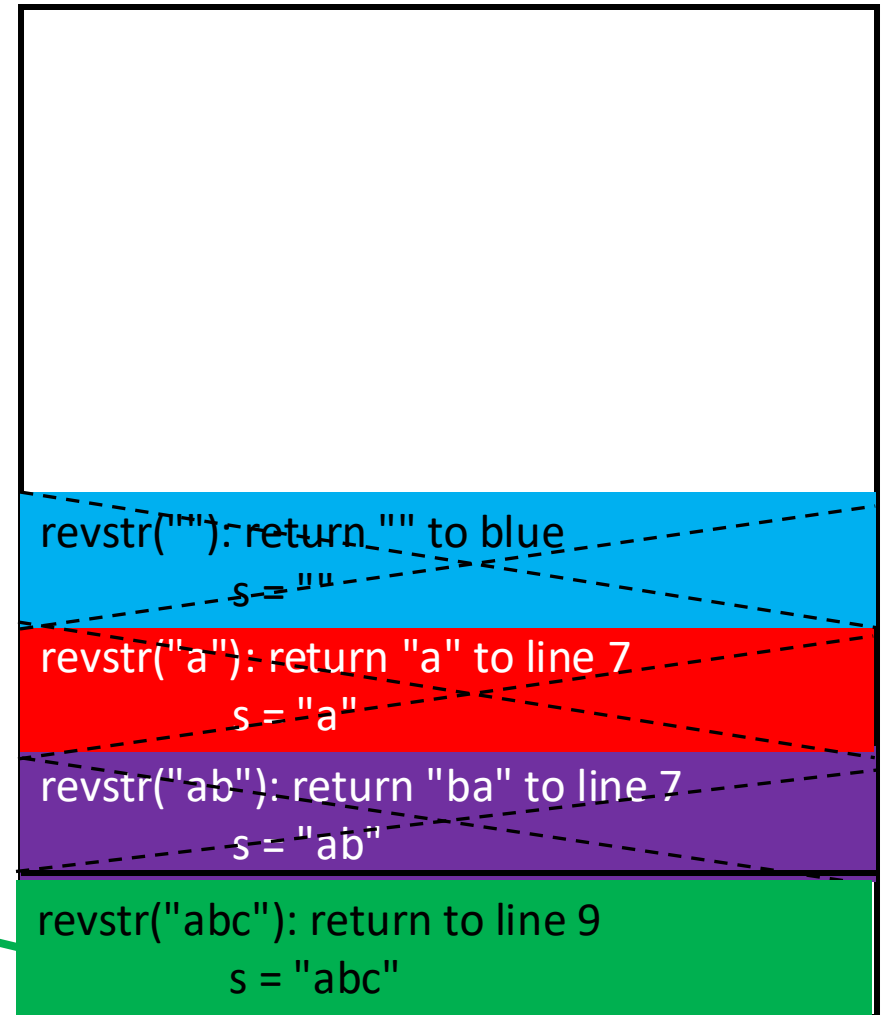
s = "ab"

revstr("abc"): return to line 9

s = "abc"

```
revstr(s ← "abc"):
    7: return "c" + "ba" = "cba"
```

```
1: def rev(s):
2:     # base case: empty string is its reverse
3:     if s == '':
4:         return ''
5:     # recursion: peel off last char, reverse
6:     # the rest, prepend last char
7:     return s[-1] + revstr(s[:-1])
8:
9: x = revstr("abc")
10: print("abc reversed is", x)
```



```

1: def rev(s):
2:     # base case: empty string is its reverse
3:     if s == '':
4:         return ''
5:     # recursion: peel off last char, reverse
6:     # the rest, prepend last char
7:     return s[-1] + revstr(s[:-1])
8:
9: x = revstr("abc")
10: print("abc reversed is", x)

```

Printed:

abc reversed is cba

