

Advent of Code 2025 in Typst 🧑🏻

Contents

Advent of Code 2025 in Typst 🧑🏻	1
day 1	1
python	1
typst	3

day 1

python

```
>>> from main import xs
>>>
>>> xs
'\nL68\nL30\nR48\nL5\nR60\nL55\nL1\nL99\nR14\nL82\n'
>>>
>>> xs.split()
['L68', 'L30', 'R48', 'L5', 'R60', 'L55', 'L1', 'L99', 'R14', 'L82']
>>>
>>> xs.replace("L", "+")
'\n+68\n+30\nR48\n+5\nR60\n+55\n+1\n+99\nR14\n+82\n'
>>>
>>> xs.replace("L", "+").replace("R", "-")
'\n+68\n+30\n-48\n+5\n-60\n+55\n+1\n+99\n-14\n+82\n'
>>>
>>> xs.replace("L", "+").replace("R", "-").split()
['+68', '+30', '-48', '+5', '-60', '+55', '+1', '+99', '-14', '+82']
>>>
>>> [int(e) for e in xs.replace("L", "+").replace("R", "-").split()]
[68, 30, -48, 5, -60, 55, 1, 99, -14, 82]
>>>
>>> list( map(int, xs.replace("L", "+").replace("R", "-").split()) )
[68, 30, -48, 5, -60, 55, 1, 99, -14, 82]
>>>
>>> (50 + 68) % 100
18
>>>
>>> (50 - 68) % 100
82
>>>
>>> list( map(int, xs.replace("L", "-").replace("R", "+").split()) )
[-68, -30, 48, -5, 60, -55, -1, -99, 14, -82]
>>>
>>> temp = 50
>>> for r in map(int, xs.replace("L", "-").replace("R", "+").split()):
...     temp
KeyboardInterrupt
>>>
KeyboardInterrupt
>>>
>>>
>>> temp = 50
>>> count = 0
>>>
```

```

52 >>>
53 >>>
54 >>> for r in map(int, xs.replace("L", "-").replace("R", "+").split()):
55 ...     print(temp)
56 ...     temp = (temp + r) % 100
57 ...     print(temp)
58 ...     if temp == 0: count += 1
59 ...     print(count)
60 ...
61 50
62 82
63 0
64 82
65 52
66 0
67 52
68 0
69 1
70 0
71 95
72 1
73 95
74 55
75 1
76 55
77 0
78 2
79 0
80 99
81 2
82 99
83 0
84 3
85 0
86 14
87 3
88 14
89 32
90 3
91 >>>
92 >>>
93 >>> (lambda xs: (
94 ...     lambda pos, c:
95 ...         ([pos := (pos + r) % 100,
96 ...          c := c + (pos == 0)]
97 ...          for r in map(int, xs.replace("L", "-").replace("R", "+").split())),
98 ...         c)[1]
99 ... )(50, 0))(xs)
100 3
101 >>>
102 >>>

```

103 **typst**

```
104 #let count-zeros(xs) = {  
105   let steps = xs.replace("L", "-").replace("R", "+").split().map(int)  
106   let result = steps.fold((50, 0), (acc, r) => {  
107     let pos = calc.rem((acc.at(0) + r), 100)  
108     let cnt = acc.at(1) + int(pos == 0)  
109     (pos, cnt)  
110   })  
111   result.at(1)  
112 }
```

113 #count-zeros("L68 L30 R48 L5 R60 L55 L1 L99 R14 L82")

114 3

Typst \gg Python