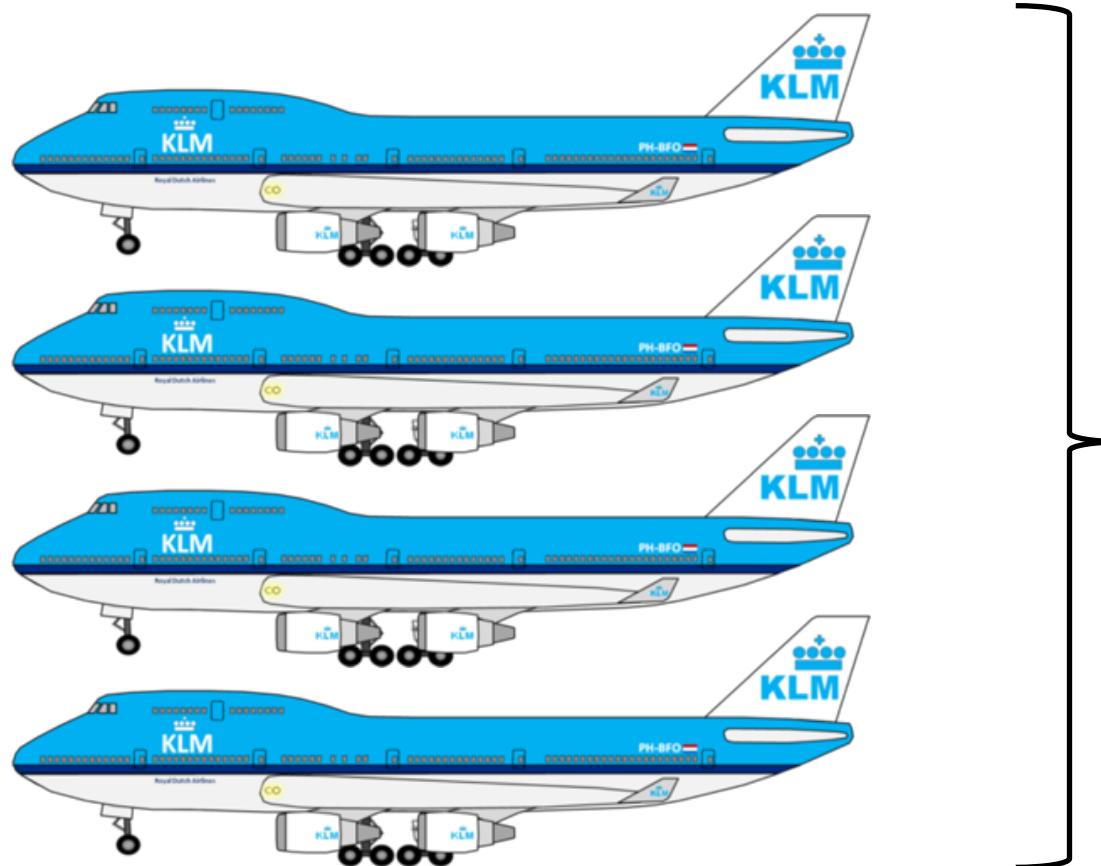


Text Processing

Chris Piech and Mehran Sahami
CS106A, Stanford University

Fake Medicine was a Problem

700,000 deaths a **year** from **fake** malaria and tuberculosis drugs [1]

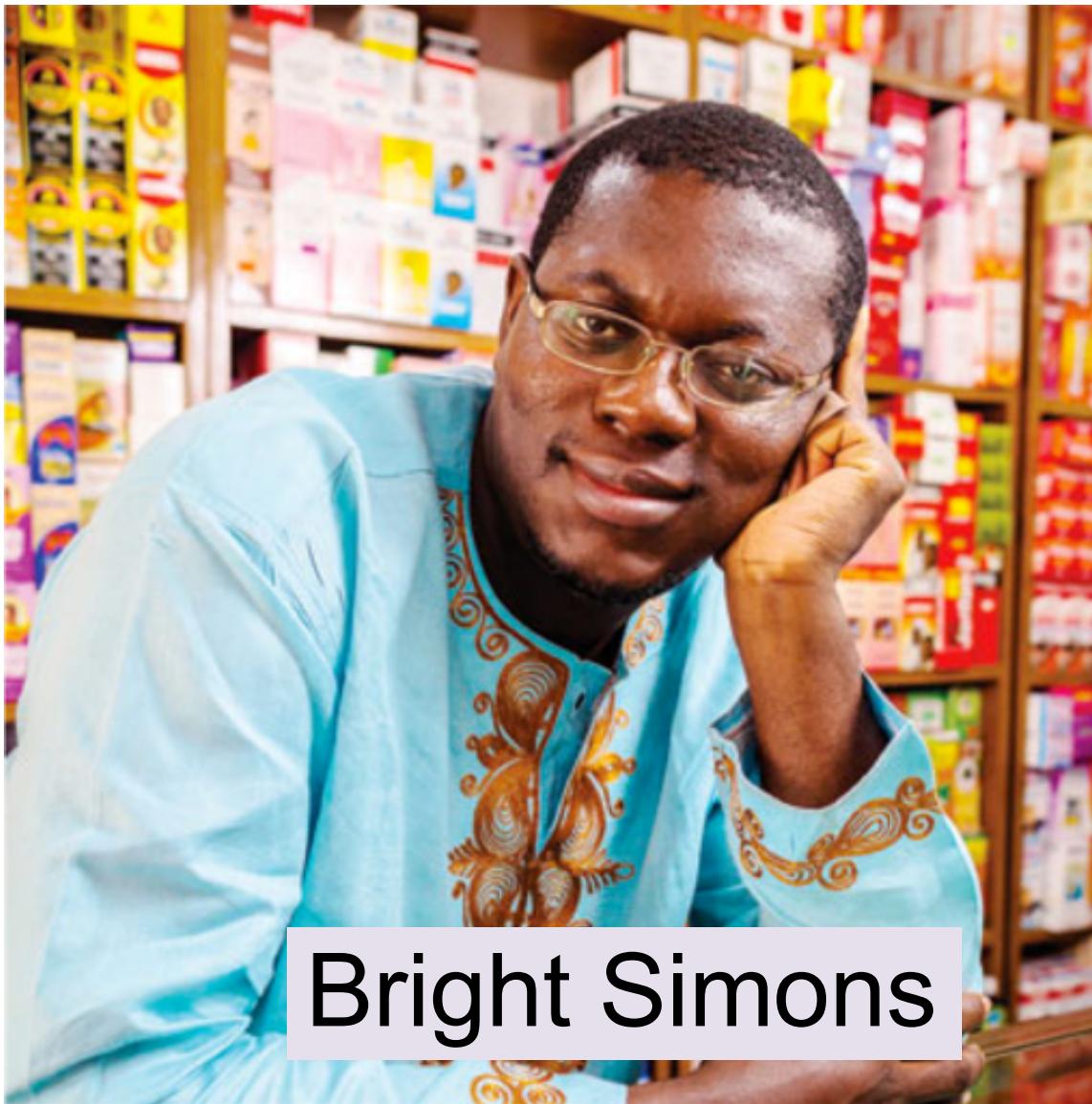


Equivalent of this
many crashes per
day

[1] <http://www.un.org/africarenewal/magazine/may-2013/counterfeit-drugs-raise-africa%E2%80%99s-temperature>



Chris' Favorite Program



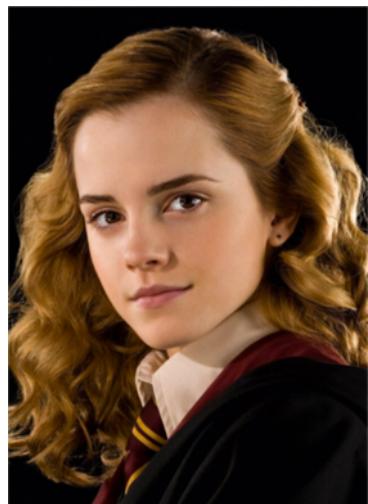
Bright Simons

Piech and Sahami, CS106A, Stanford University



Underlying Puzzle

Counterfeiter



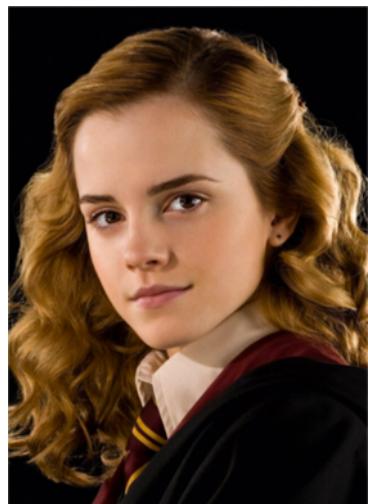
You (Distributor)

User



Underlying Puzzle

Counterfeiter



You (Distributor)



User



Revisit this problem...

Learning Goals

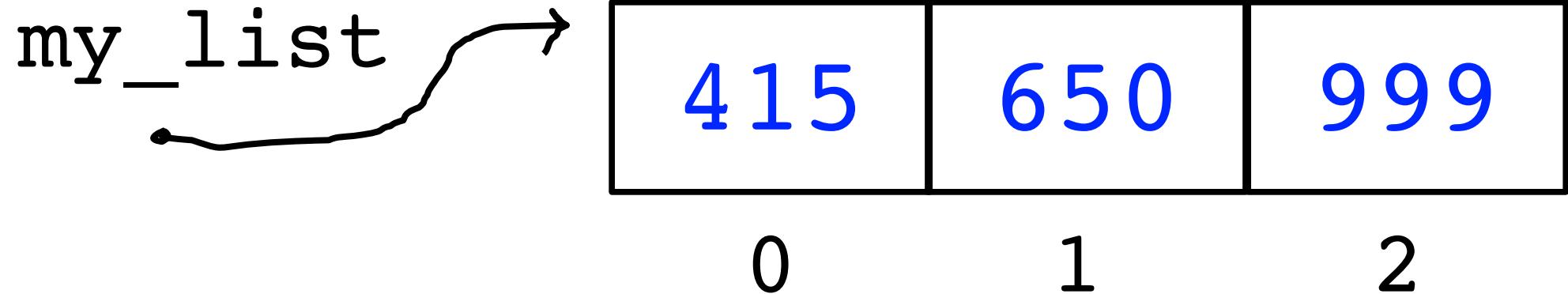
1. Write string algorithms that loops over each character



Review

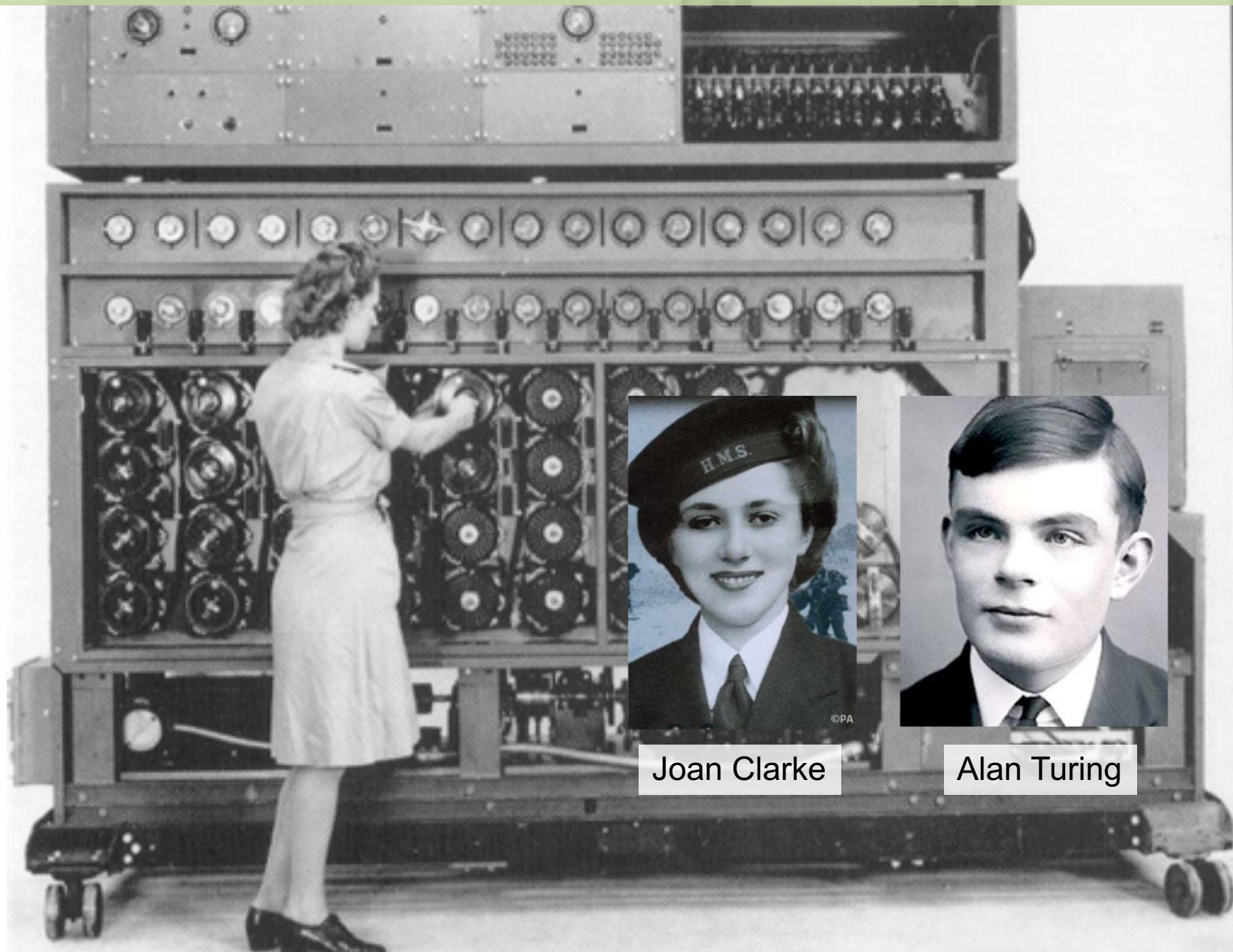
Review of lists

```
def main():
    my_list = [415, 650, 999]
    print(my_list)
```



End Review

Text Problem: Decryption



Joan Clarke

Alan Turing



Text Problem: Translation

The spirit is willing but the flesh is weak.



(Russian)



The vodka is good but the meat is rotten.



*This result cost billions of dollars (adjusted for inflation)



Text Problem: DNA Analysis



Piech and Sahami, CS106A, Stanford University

```
AAGTCAGTCAGATTACCCCTGGCTCACC  
TGTTCGTACAACCAATTAGGTGAGTTCA  
TTCGGAAAGACTCCCTGGTACCATCCCCG  
ACGGGGGTTGGAATTACGGGTAGAACG  
ACCAATCGTAACATATGAGAGGCCACTGC  
ATAATAGGGGAGGGTTCATTCGTCGCT  
CTAACTTGTAAATACCCGACCACAC  
CCACCCCTGGCATTATAGTACCCCGAAC  
CGTAGAGCCAGATGTATCCAATCCCCG  
GTAAGATCTCCAAAAAGTCGACCGATGA  
ACTGGTACTTGGATACCATCATTGGT  
ATCCGCTGATTGCTGGTTAATTGTTATG  
TCCCGTTTCAAGTTCAGACACTAGTT  
CCTAGGGCGTCACTGGCACCATACT  
TCAATAGGGTATCGGGAGGTTTATTAC  
TGGCACCCGTCCCTAACGGTGTGCCA  
GCCGCACCTACCTCGAAAGTCATAGGA  
CCTGATCGTCCATTACCCGGATGTGTGC  
GCGGGAGGGTCAACACACGTAACCTCTC  
CTAACGGTCCCCATCCCATCCGAGATT  
TTTTTAGAAATGTTTGAGAATGGGAT  
TCCAGGGGTCTTCGTTACCCATGGCGA  
AGTTCGCACTATTGACAAACGAGCATGG  
CCAGGAATTCCGATGCCGATCGTCTGAC  
ACACCTTGTCCAACTAACAAAGTAACCG  
TGTGAAAGTTTACCTAGATGGTCGTAG
```



How is text
represented?

The variable type **string**

Text is stored using the variable type **string**.

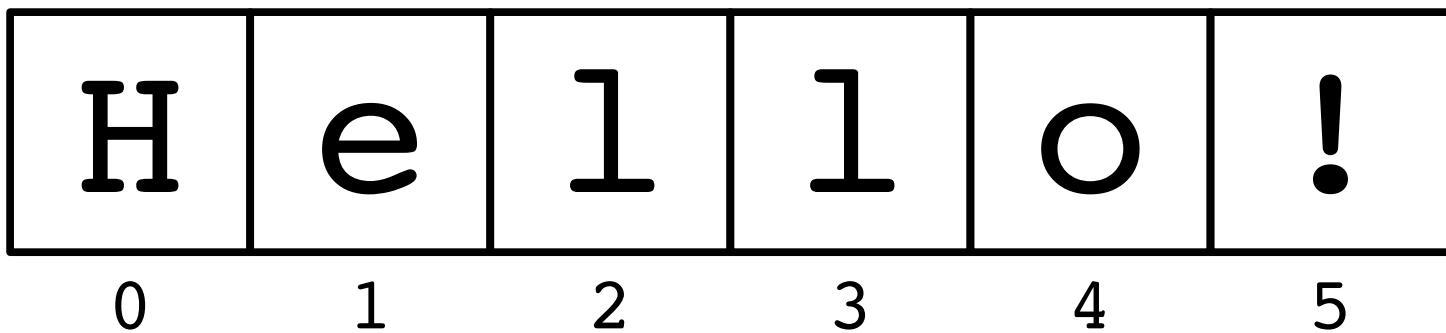
A **string** is a sequence of characters.

```
def main():
    text = "hello!"
    print(text)
```



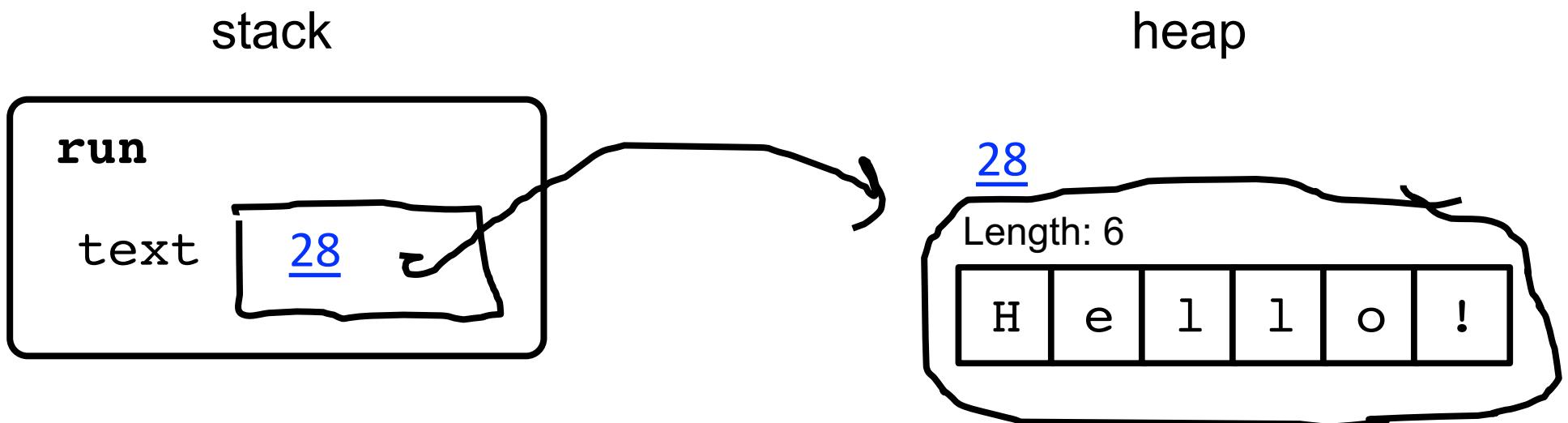
H e l l o !



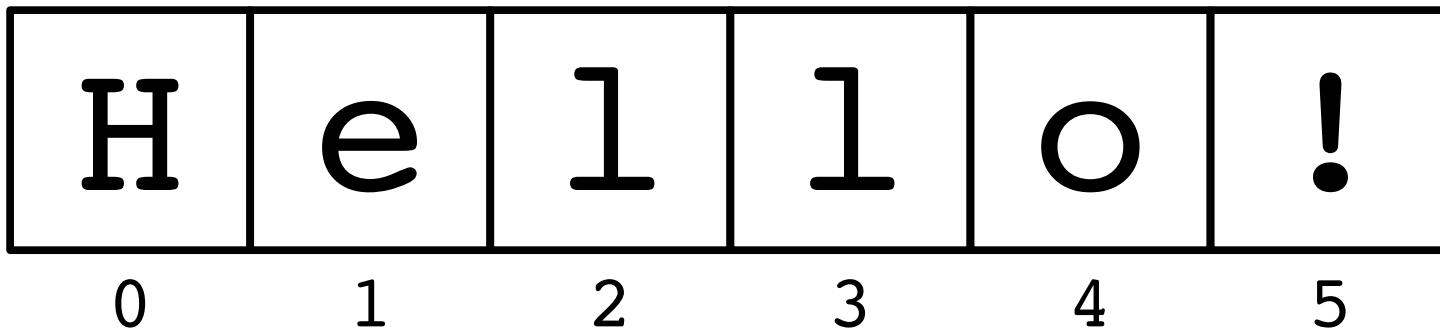


How it is actually stored

```
def main():
    text = "hello!"
```



```
def main():
    text = "hello!"
```



text[index]





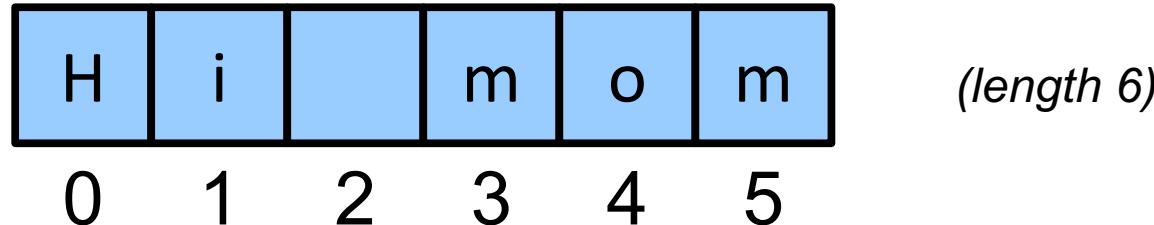
All characters in a string have an index.

You can access a character in the sequence via its *index*



String Functions

- The `len(string)` function returns the number of characters in the string. This is one larger than the last valid index in the string.
- the `string[i]` function returns the character at a given index.





A string is indexed just like
a list! Slices work too.

It is *almost* like it is a list of
characters.



Lets play!

```
Chris@ndoto Desktop % python3
Python 3.8.1 (v3.8.1:1b293b6006, Dec 18 2019, 14:08:53)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> x = 'Hi mom'
>>> x[0]
'H'
>>> x[1]
'i'
>>> x[2]
'm'
>>> x[3]
'i'
>>> x[4]
'o'
>>> x[5]
'm'
>>> x[6]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
IndexError: string index out of range
>>>
```



String Functions

```
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length) # prints 6

    # example of getCharAt
    first = example[0]
    print(first) # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
```



String Functions

```
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length) # prints 6

    # example of getCharAt
    first = example[0]
    print(first) # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
```

Console



String Functions

```
def main():
    example = "Hi mom"

# example of length function
length = len(example)
print(length) # prints 6

# example of getCharAt
first = example[0]
print(first) # prints 'H'

# loop that prints letters one-by-one
for i in range(len(example)):
    ch = example[i]
    print(ch)
```

Console

example

H	i		m	o	m
0	1	2	3	4	5



String Functions

```
def main():
    example = "Hi mom"
```

```
# example of length function
length = len(example)
print(length) # prints 6
```

```
# example of getCharAt
first = example[0]
print(first) # prints 'H'
```

```
# loop that prints letters one-by-one
for i in range(len(example)):
    ch = example[i]
    print(ch)
```

Console

example

H	i		m	o	m
0	1	2	3	4	5

length

6



String Functions

```
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length) # prints 6
```

```
# example of getCharAt
first = example[0]
print(first) # prints 'H'
```

```
# loop that prints letters one-by-one
for i in range(len(example)):
    ch = example[i]
    print(ch)
```

Console

6

example

H	i		m	o	m
0	1	2	3	4	5

length

6



String Functions

```
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length) # prints 6

    # example of getCharAt
    first = example[0]
    print(first) # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
```

Console

6

example



length

6

first

'H'



String Functions

```
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length) # prints 6

    # example of getCharAt
    first = example[0]
    print(first) # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
```

Console

6

H

example

H	i		m	o	m
0	1	2	3	4	5

length

6

first

'H'



String Functions

```
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length) # prints 6

    # example of getCharAt
    first = example[0]
    print(first) # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
```

Console

6
H

example



length

6

first

'H'



String Functions

```
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length) # prints 6

    # example of getCharAt
    first = example[0]
    print(first) # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
```

Console

6
H

example



length

6

first

'H'

i

0



String Functions

```
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length) # prints 6

    # example of getCharAt
    first = example[0]
    print(first) # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
```

Console

6
H

example	length	first	i												
<table><tr><td>H</td><td>i</td><td></td><td>m</td><td>o</td><td>m</td></tr><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr></table>	H	i		m	o	m	0	1	2	3	4	5	6	'H'	0
H	i		m	o	m										
0	1	2	3	4	5										



String Functions

```
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length) # prints 6

    # example of getCharAt
    first = example[0]
    print(first) # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)): [0, 1, 2, 3, 4, 5]
        ch = example[i]
        print(ch)
```

Console

6
H

example



length

6

first

'H'

i

0



String Functions

```
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length) # prints 6

    # example of getCharAt
    first = example[0]
    print(first) # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
```

Console

6
H

example



length

6

first

'H'

i

0



String Functions

```
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length) # prints 6

    # example of getCharAt
    first = example[0]
    print(first) # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
```

Console

6
H

example

H	i		m	o	m
0	1	2	3	4	5

length

6

first

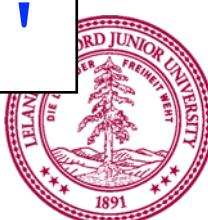
'H'

i

0

ch

'H'



String Functions

```
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length) # prints 6

    # example of getCharAt
    first = example[0]
    print(first) # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
```

Console

6
H
H

example

H	i		m	o	m
0	1	2	3	4	5

length

6

first

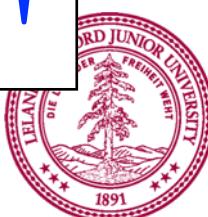
'H'

i

0

ch

'H'



String Functions

```
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length) # prints 6

    # example of getCharAt
    first = example[0]
    print(first) # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
```

Console

```
6
H
H
```

example



length

6

first

'H'

i

0



String Functions

```
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length) # prints 6

    # example of getCharAt
    first = example[0]
    print(first) # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
```

Console

```
6
H
H
```

example



length

6

first

'H'

i

1



String Functions

```
def main():
    example = "Hi mom"
```

```
# example of length function
length = len(example)
print(length) # prints 6
```

```
# example of getCharAt
first = example[0]
print(first) # prints 'H'
```

```
# loop that prints letters one-by-one
for i in range(len(example)):
    ch = example[i]
    print(ch)
```

Console

```
6
H
H
```

example

H	i		m	o	m
0	1	2	3	4	5

length

6

first

'H'

i

1



String Functions

```
def main():
    example = "Hi mom"
```

```
# example of length function
length = len(example)
print(length) # prints 6
```

```
# example of getCharAt
first = example[0]
print(first) # prints 'H'
```

```
# loop that prints letters one-by-one
for i in range(len(example)):
    ch = example[i]
    print(ch)
```

Console

```
6
H
H
```

example

H	i		m	o	m
0	1	2	3	4	5

length

```
6
```

first

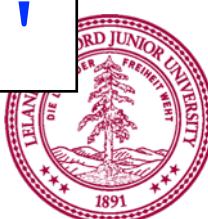
```
'H'
```

i

```
1
```

ch

```
'i'
```



String Functions

```
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length) # prints 6

    # example of getCharAt
    first = example[0]
    print(first) # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
```

Console

```
6
H
H
i
```

example

H	i		m	o	m
0	1	2	3	4	5

length

6

first

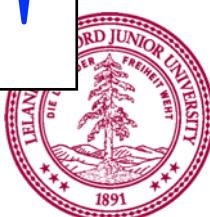
'H'

i

1

ch

'i'



String Functions

```
def main():
    example = "Hi mom"
```

```
# example of length function
length = len(example)
print(length) # prints 6
```

```
# example of getCharAt
first = example[0]
print(first) # prints 'H'
```

```
# loop that prints letters one-by-one
for i in range(len(example)):
    ch = example[i]
    print(ch)
```

Console

```
6
H
H
i
```

example

H	i		m	o	m
0	1	2	3	4	5

length

6

first

'H'

i

1



String Functions

```
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length) # prints 6

    # example of getCharAt
    first = example[0]
    print(first) # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
```

Console

```
6
H
H
i
```

example



length

6

first

'H'

i

2



String Functions

```
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length) # prints 6

    # example of getCharAt
    first = example[0]
    print(first) # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
```

Console

```
6
H
H
i
```

example



length

6

first

'H'

i

2



String Functions

```
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length) # prints 6

    # example of getCharAt
    first = example[0]
    print(first) # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
```

Console

```
6
H
H
i
```

example

H	i		m	o	m
0	1	2	3	4	5

length

6

first

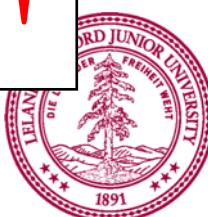
'H'

i

2

ch

! !



String Functions

```
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length) # prints 6

    # example of getCharAt
    first = example[0]
    print(first) # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
```

Console

```
6
H
H
i
```

example

H	i		m	o	m
0	1	2	3	4	5

length

6

first

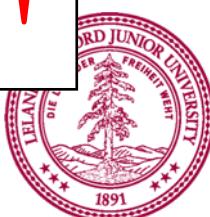
'H'

i

2

ch

' '



String Functions

```
def main():
    example = "Hi mom"
```

```
# example of length function
length = len(example)
print(length) # prints 6
```

```
# example of getCharAt
first = example[0]
print(first) # prints 'H'
```

```
# loop that prints letters one-by-one
for i in range(len(example)):
    ch = example[i]
    print(ch)
```



Console

```
6
H
H
i
```

example

H	i		m	o	m
0	1	2	3	4	5

length

6

first

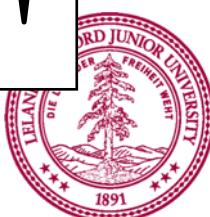
'H'

i

2

ch

' '



String Functions

```
def main():
    example = "Hi mom"
```

```
# example of length function
length = len(example)
print(length) # prints 6
```

```
# example of getCharAt
first = example[0]
print(first) # prints 'H'
```

```
# loop that prints letters one-by-one
for i in range(len(example)):
    ch = example[i]
    print(ch)
```



Console

```
6
H
H
i
m
o
m
```

example

H	i		m	o	m
0	1	2	3	4	5

length

6

first

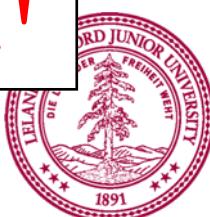
'H'

i

5

ch

'm'



Again there are two for loops



```
for i in range(len(example)):  
    ch = example[i]  
    print(i, ch)
```

```
for ch in example:  
    print(ch)
```



Functions you can call on strings

```
x = 'this is a Test '
```

must know:

split	x.split(' ') #['this', 'is', 'a', 'Test']
upper	x.upper() #'THIS IS A TEST'
lower	x.lower() #'this is a test'
replace	x.replace('is','lol') #'thlol lol a Test'
find	x.find('is') #2
strip	strip() #'this is a test'

good to know:

startswith	x.startswith('th') # True
endswith	x.endswith('end') # False
title	x.title() #'This Is A Test'
isalpha	x.isalpha() #False
isdigit	'521'.isdigit() #True
isspace	' '.isspace() #True



Just the number please

DNA → mRNA

Advanced version

How are characters
represented?

Single characters

- Some examples:

```
letter_A = 'A'
```

\n newline char
\t tab char

```
plus = '+'
```

\\ backlash char
\' single quote

```
zero = '0'
```

\\" double quote

```
space = ''
```

```
korean_ch = '보'
```

```
new_line = '\n' # special
```

```
tab = '\t' # special
```

```
backslash = '\\\' # special
```

```
backslash = '\\\\' # special
```

```
emoji = '😊'
```

```
first = text[0]
```





Advanced course

Chars are just a giant
enumeration!



ASCII

Code	Char	Code	Char	Code	Char	Code	Char	Code	Char	Code	Char
32	[space]	48	0	64	@	80	P	96	~	112	p
33	!	49	1	65	A	81	Q	97	a	113	q
34	"	50	2	66	B	82	R	98	b	114	r
35	#	51	3	67	C	83	S	99	c	115	s
36	\$	52	4	68	D	84	T	100	d	116	t
37	%	53	5	69	E	85	U	101	e	117	u
38	&	54	6	70	F	86	V	102	f	118	v
39	'	55	7	71	G	87	W	103	g	119	w
40	(56	8	72	H	88	X	104	h	120	x
41)	57	9	73	I	89	Y	105	i	121	y
42	:	58	:	74	J	90	Z	106	j	122	z
43	+	59	:	75	K	91	[107	k	123	{
44	,	60	<	76	L	92	\	108	l	124	
45	-	61	=	77	M	93]	109	m	125	}
46	.	62	>	78	N	94	^	110	n	126	~
47	/	63	?	79	O	95	_	111	o	127	[backspace]

* This is only the first half of the table

The letter A, for example, has the ASCII value 65



Unicode (bigger ASCII)

3200

Enclosed CJK Letters and Months

32FF

	320	321	322	323	324	325	326	327	328	329	32A	32B	32C	32D	32E	32F		
0	(ㄱ)	(ㄷ)	(ㄴ)	(ㅂ)	(ㅅ)	(ㅈ)	PTE	(ㄱ)	(ㄷ)	(ㄴ)	(ㅂ)	(ㅅ)	(ㅈ)	1月	(ㄱ)	(ㄷ)	(ㄴ)	
1	(ㄴ)	(ㄹ)	(ㅁ)	(ㅂ)	(ㅅ)	(ㅈ)	(ㅎ)	(ㄴ)	(ㄹ)	(ㅁ)	(ㅂ)	(ㅅ)	(ㅈ)	(ㅎ)	2月	(ㄱ)	(ㄴ)	(ㅁ)
2	(ㄷ)	(ㅌ)	(ㄴ)	(ㅂ)	(ㅅ)	(ㅈ)	(ㅎ)	(ㄷ)	(ㅌ)	(ㄴ)	(ㅂ)	(ㅅ)	(ㅈ)	(ㅎ)	3月	(ㄱ)	(ㄷ)	(ㅌ)
3	(ㄹ)	(ㅂ)	(ㅁ)	(ㅂ)	(ㅅ)	(ㅈ)	(ㅎ)	(ㄹ)	(ㅂ)	(ㅁ)	(ㅂ)	(ㅅ)	(ㅈ)	(ㅎ)	4月	(ㄱ)	(ㄴ)	(ㅁ)
4	(ㅁ)	(ㅅ)	(ㅍ)	(ㅎ)	(ㅌ)	(ㅊ)	(ㅋ)	(ㅁ)	(ㅅ)	(ㅍ)	(ㅎ)	(ㅌ)	(ㅊ)	(ㅋ)	5月	(ㄱ)	(ㄴ)	(ㅋ)
5	(ㅂ)	(ㅏ)	(ㅓ)	(ㅗ)	(ㅓ)	(ㅗ)	(ㅓ)	(ㅂ)	(ㅏ)	(ㅓ)	(ㅗ)	(ㅓ)	(ㅗ)	(ㅓ)	6月	(ㄱ)	(ㅓ)	(ㅕ)
6	(ㅅ)	(ㅈ)	(ㅊ)	(ㅍ)	(ㅌ)	(ㅊ)	(ㅋ)	(ㅅ)	(ㅈ)	(ㅊ)	(ㅍ)	(ㅌ)	(ㅊ)	(ㅋ)	7月	(ㄱ)	(ㄴ)	(ㅋ)
7	(ㅇ)	(ㅊ)	(ㅍ)	(ㅎ)	(ㅌ)	(ㅊ)	(ㅋ)	(ㅇ)	(ㅊ)	(ㅍ)	(ㅎ)	(ㅌ)	(ㅊ)	(ㅋ)	8月	(ㄱ)	(ㅓ)	(ㅕ)
8	(ㅈ)	(ㅊ)	(ㅋ)	(ㅍ)	(ㅌ)	(ㅊ)	(ㅋ)	(ㅈ)	(ㅊ)	(ㅋ)	(ㅍ)	(ㅌ)	(ㅊ)	(ㅋ)	9月	(ㄱ)	(ㅓ)	(ㅕ)
9	(ㅊ)	(ㅌ)	(ㅍ)	(ㅎ)	(ㅌ)	(ㅊ)	(ㅋ)	(ㅊ)	(ㅌ)	(ㅍ)	(ㅎ)	(ㅌ)	(ㅊ)	(ㅋ)	10月	(ㄱ)	(ㄴ)	(ㅋ)
A	(ㅋ)	(ㅍ)	(ㅎ)	(ㅍ)	(ㅌ)	(ㅊ)	(ㅋ)	(ㅋ)	(ㅍ)	(ㅎ)	(ㅍ)	(ㅌ)	(ㅊ)	(ㅋ)	11月	(ㄱ)	(ㄴ)	(ㅍ)
B	(ㅌ)	(ㅎ)	(ㅍ)	(ㅎ)	(ㅌ)	(ㅊ)	(ㅋ)	(ㅌ)	(ㅎ)	(ㅍ)	(ㅎ)	(ㅌ)	(ㅊ)	(ㅋ)	12月	(ㄱ)	(ㅓ)	(ㅕ)
C	(ㄱ)	(ㅈ)	(ㅊ)	(ㅍ)	(ㅌ)	(ㅊ)	(ㅋ)	(ㄱ)	(ㅈ)	(ㅊ)	(ㅍ)	(ㅌ)	(ㅊ)	(ㅋ)	Hg	(ㄱ)	(ㅓ)	(ㅕ)
D	(ㅎ)	(ㅍ)	(ㅎ)	(ㅍ)	(ㅌ)	(ㅊ)	(ㅋ)	(ㅎ)	(ㅍ)	(ㅎ)	(ㅍ)	(ㅌ)	(ㅊ)	(ㅋ)	erg	(ㄱ)	(ㅓ)	(ㅕ)
E	(ㄱ)	(ㅍ)	(ㅎ)	(ㅍ)	(ㅌ)	(ㅊ)	(ㅋ)	(ㄱ)	(ㅍ)	(ㅎ)	(ㅍ)	(ㅌ)	(ㅊ)	(ㅋ)	eV	(ㄱ)	(ㅓ)	(ㅕ)
F	(ㄴ)	(ㅎ)	(ㅍ)	(ㅎ)	(ㅌ)	(ㅊ)	(ㅋ)	(ㄴ)	(ㅎ)	(ㅍ)	(ㅎ)	(ㅌ)	(ㅊ)	(ㅋ)	₩	(ㄱ)	(ㅓ)	(ㅕ)

P





'A' -> 'Z' are sequential.
'a' -> 'z' are sequential.
'0' -> '9' are sequential.

`ord(ch)`



Functions which take strings

```
x = 'this is a Test '
```

```
len           len(x)    # 15
ord           ord('A')   # 65
hash          hash(x)   # 2466759895439727657
<             'abc' < 'zabc' # True
==            x == 'this is a Test ' # True
in             'his' in x # True
```

Won't need these too
much in CS106A, but
they are super clutch
in cryptography and
datastructures



Strings have some unique properties

Strings are Immutable

- Python strings are ***immutable***: once a string has been created **you cannot set characters**.
- To change a string:
 - ***Create a new string*** holding the new value you want it to have via concatenation.
 - Reassigning the String variable (that's allowed).
- ***Important consequence:*** if you pass a String into a function, you are guaranteed your string won't be changed.





Can survive:
-300F to +300F
Massive radiation
The vacuum of space

Strings are Immutable (Take 1)

`x = 'abc'`



`x[1] = 'z'`

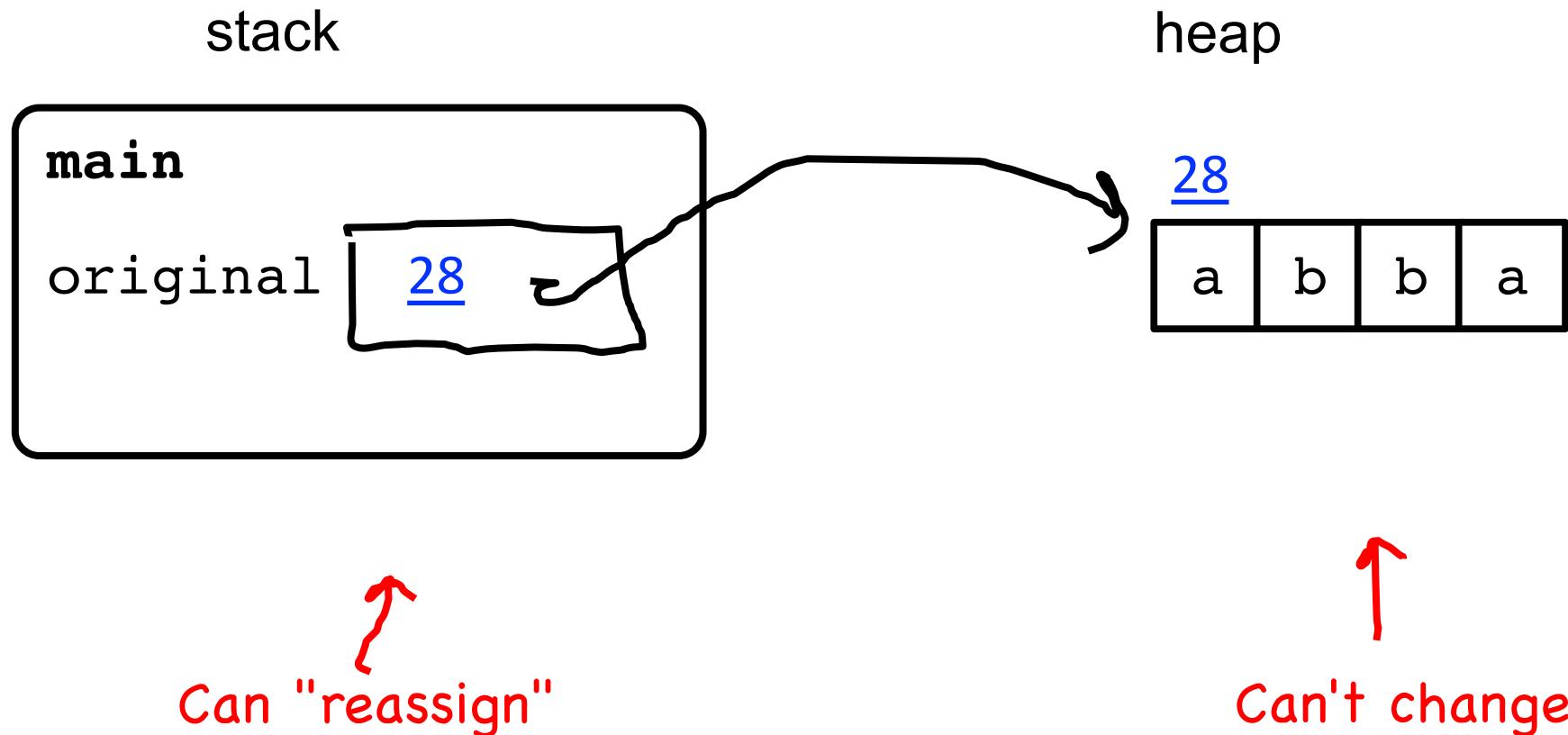
Traceback (most recent call last):
...
TypeError: 'str' object does not support item assignment

`x = 'azc'`



Strings are Immutable (Take 2)

`original = 'abba'`

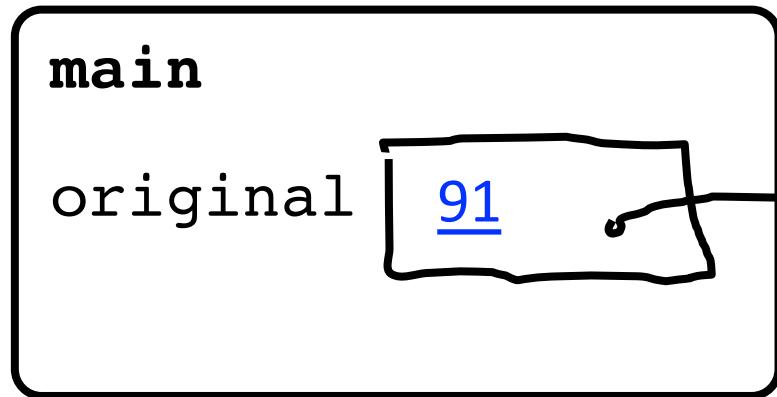


Strings are Immutable (Take 1)

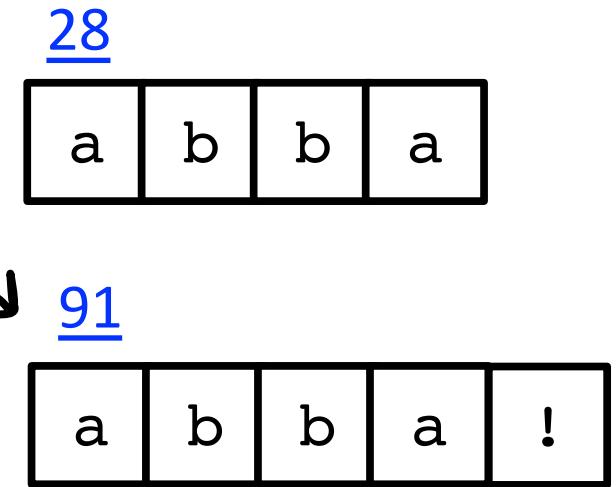
```
original = 'abba'
```

```
original = original + '!'
```

stack



heap



Pro tip: use id function to check a reference!



Strings are often made through concatenation

```
def main():
    s1 = "CS106"
    s2 = "A"
    s3 = "I got an " + s2 + " in " + s1 + s2

print(s3)
```

I got an A in CS106A





Lists are **mutable**

Strings are **immutable**

*Immutable is a guarantee
that a function won't be
cheeky*

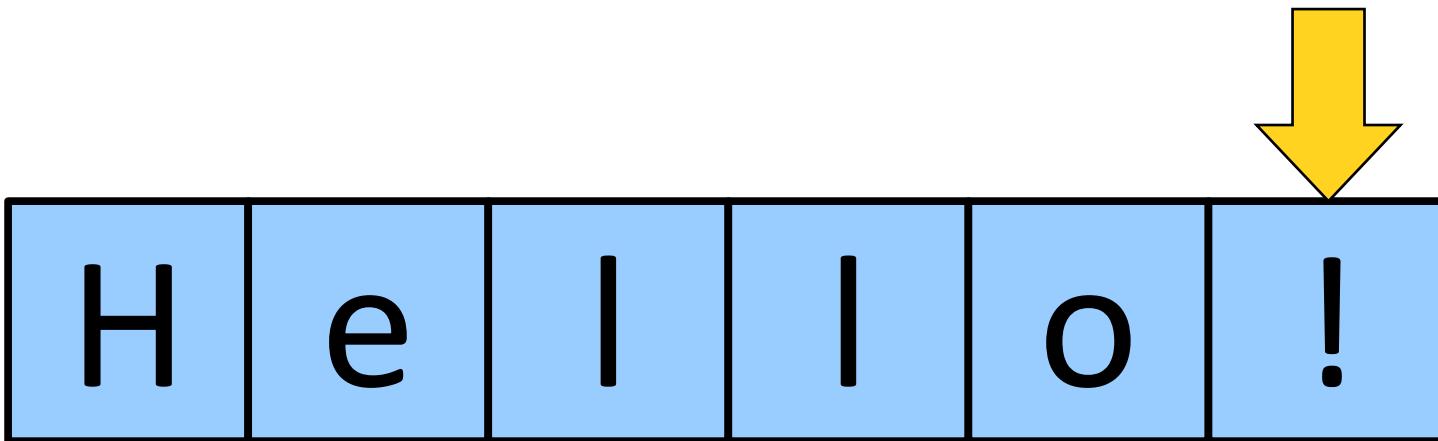




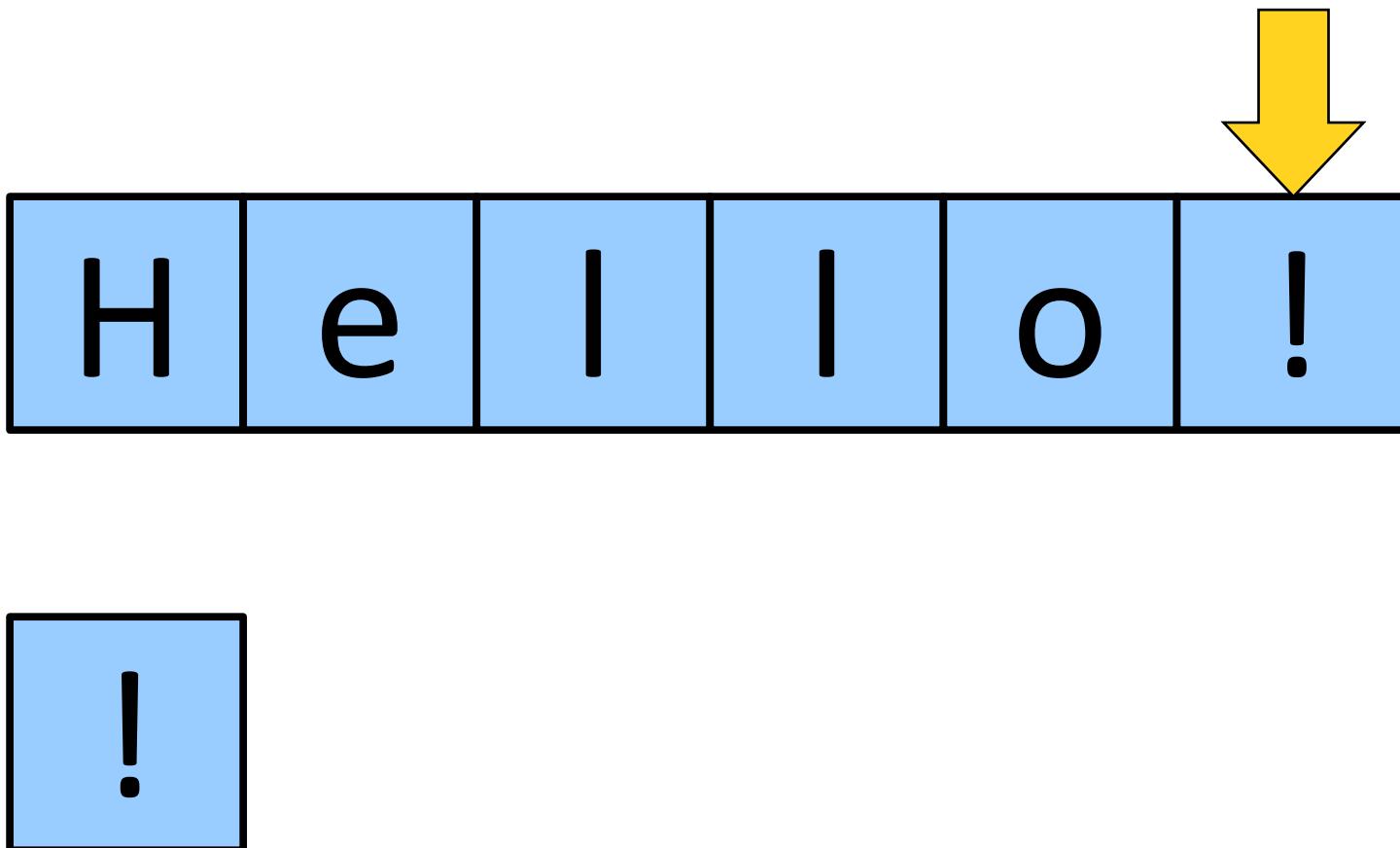
Many string algorithms use
the "loop and construct"
pattern.



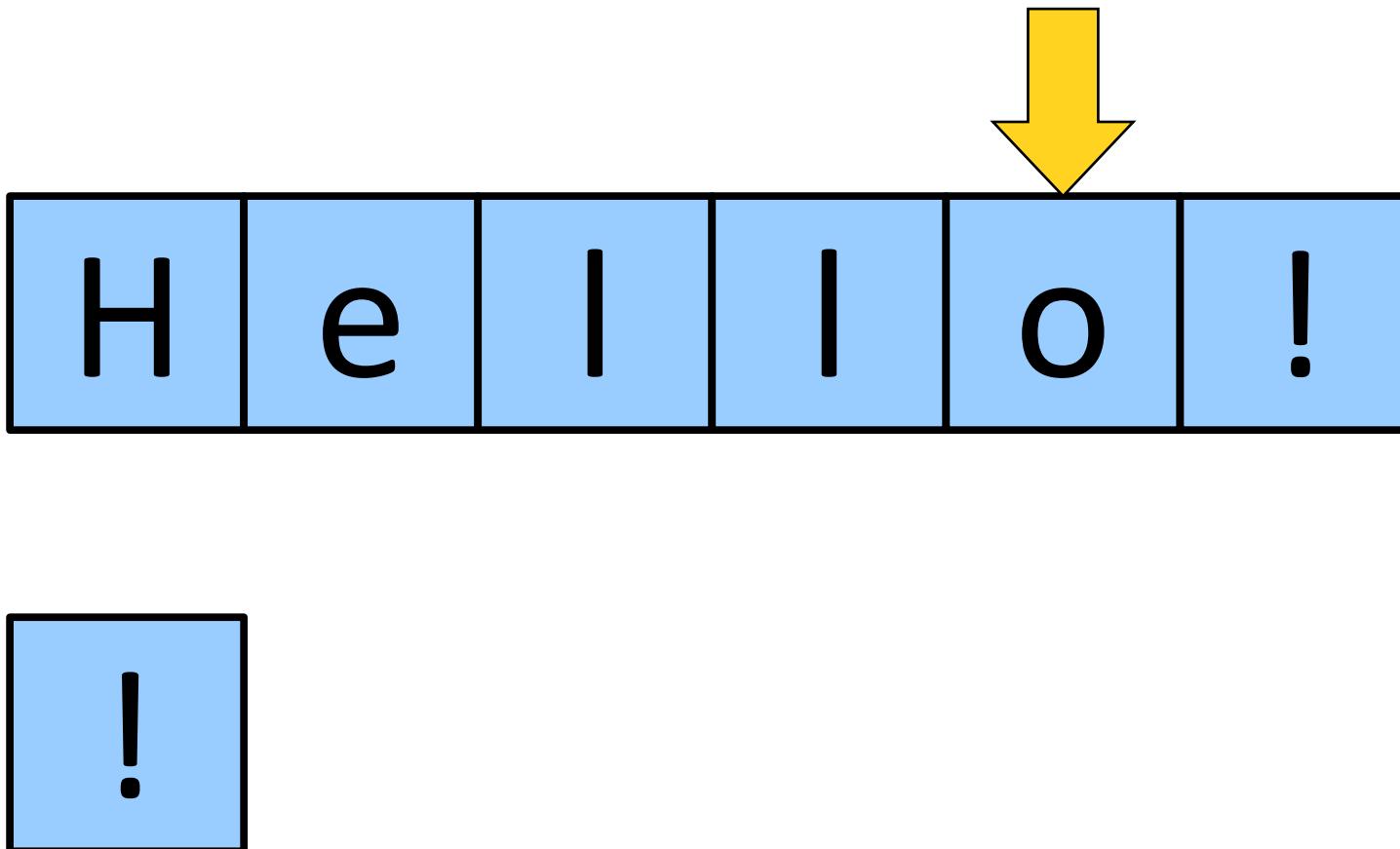
Reversing a String



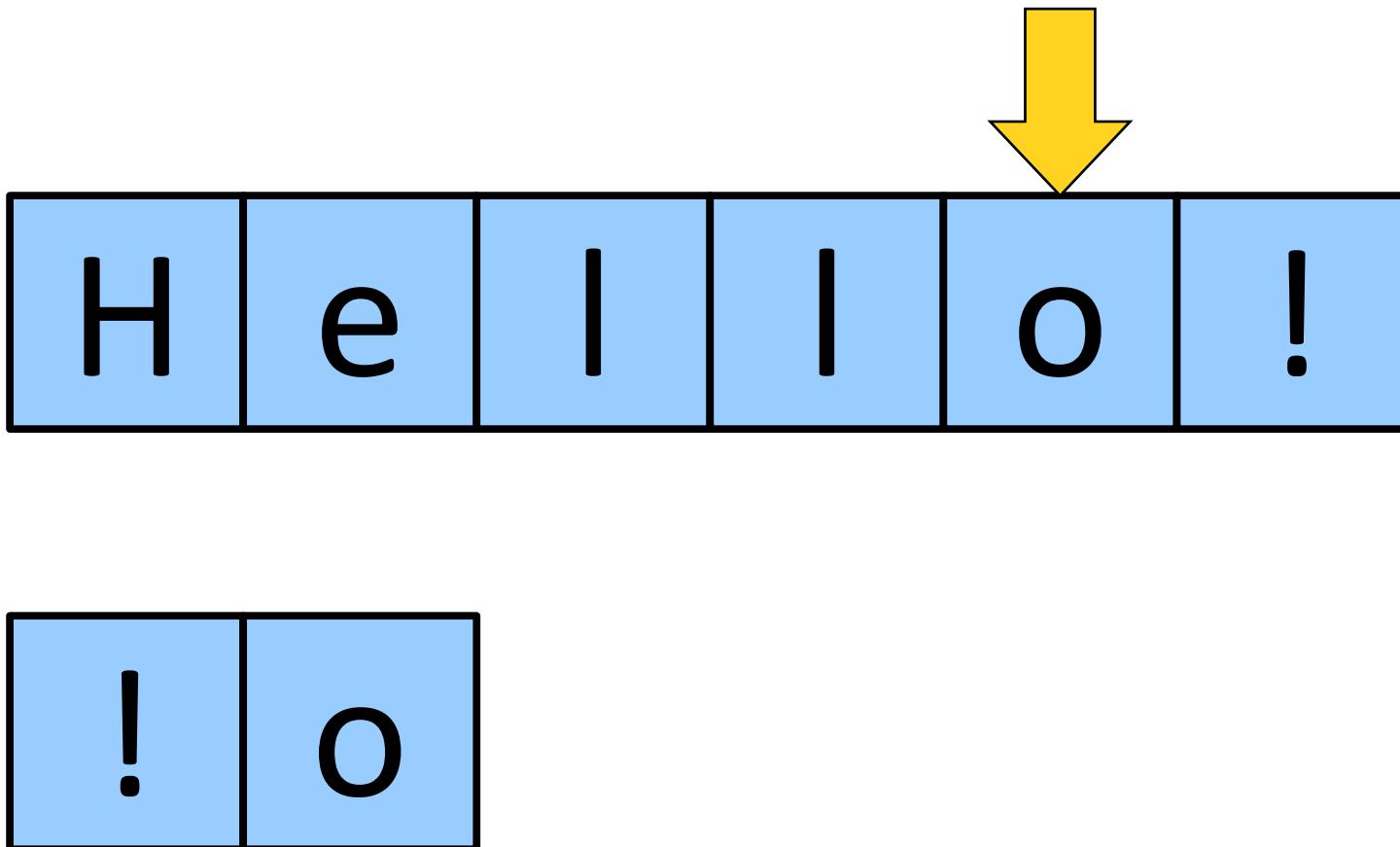
Reversing a String



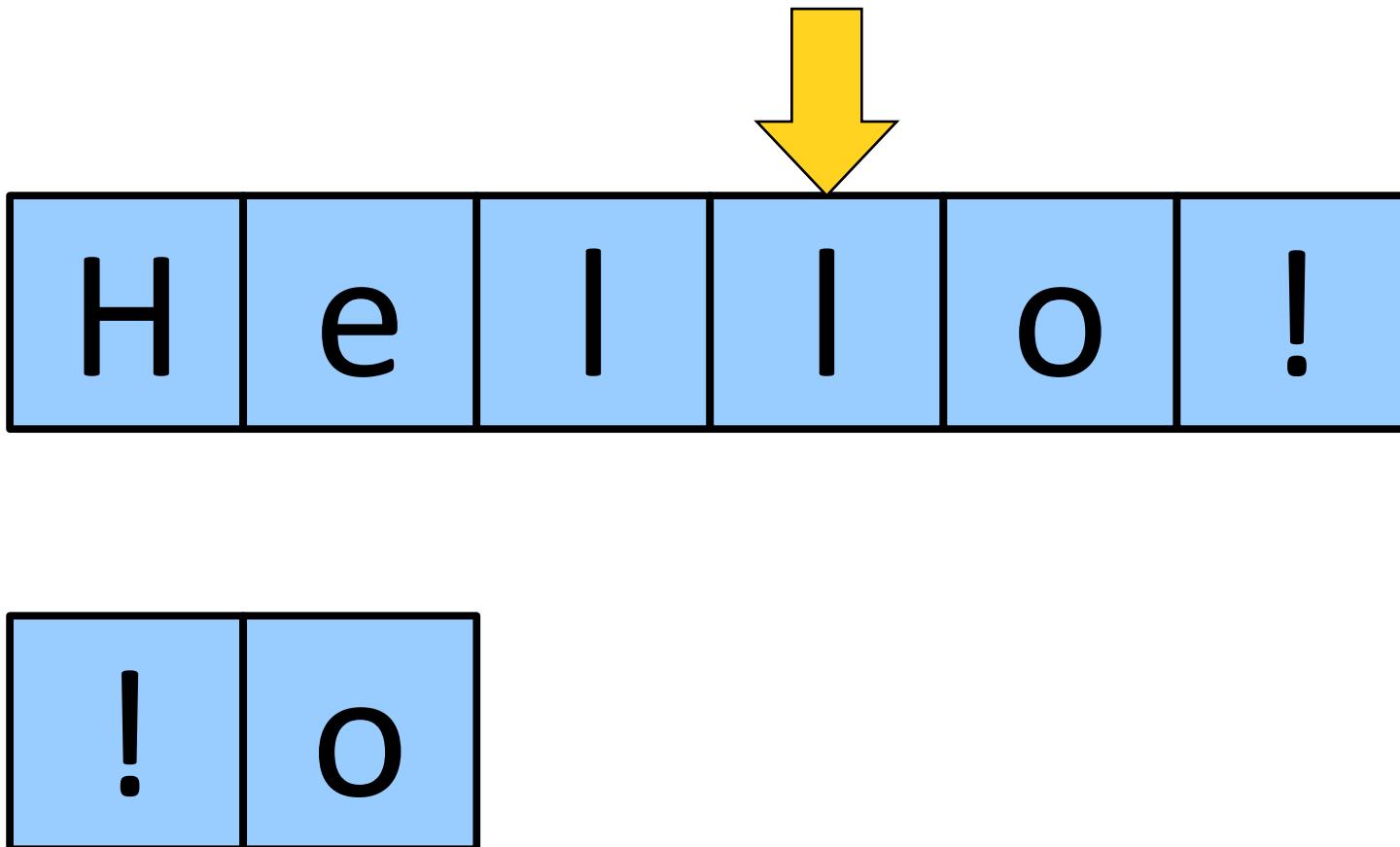
Reversing a String



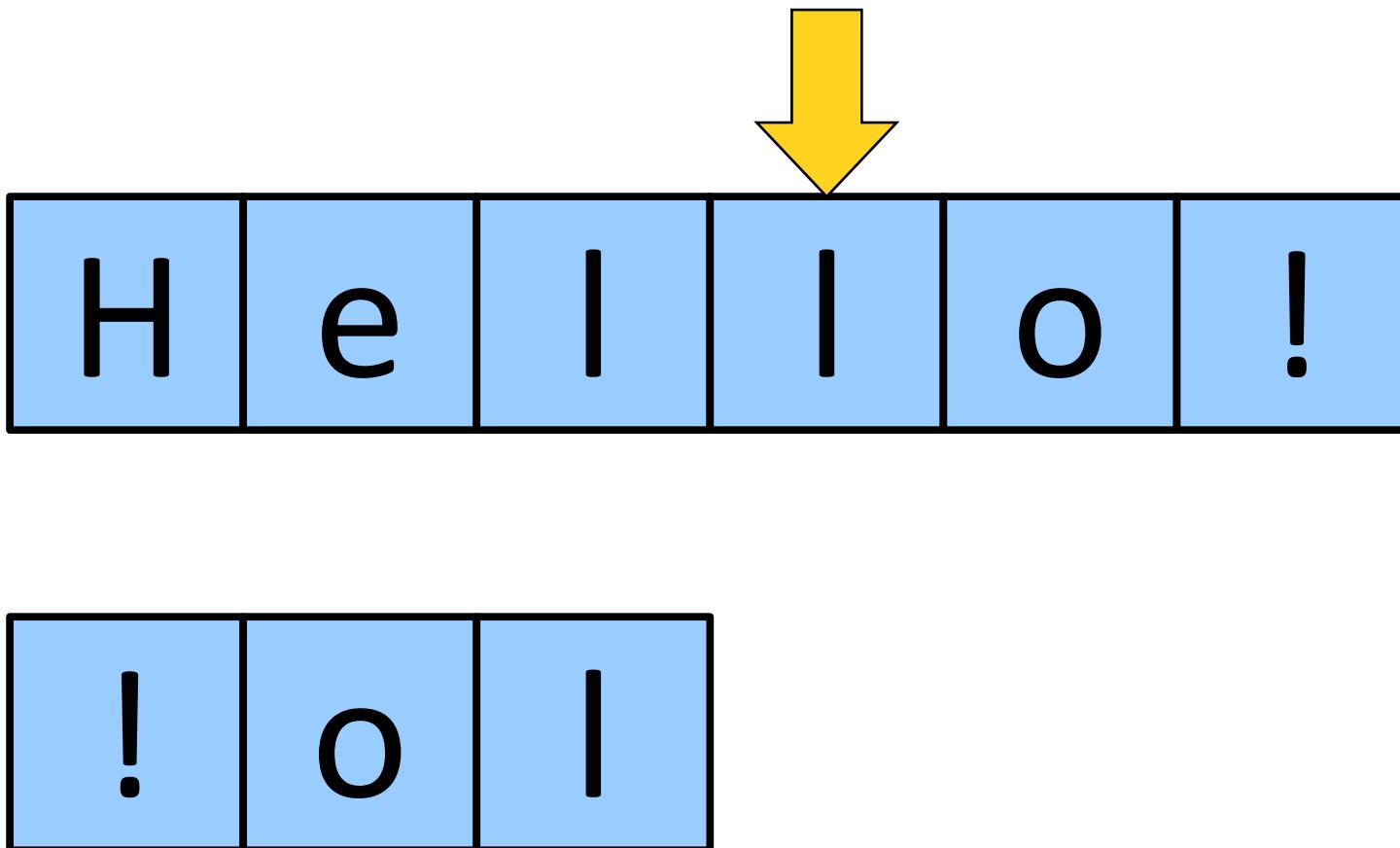
Reversing a String



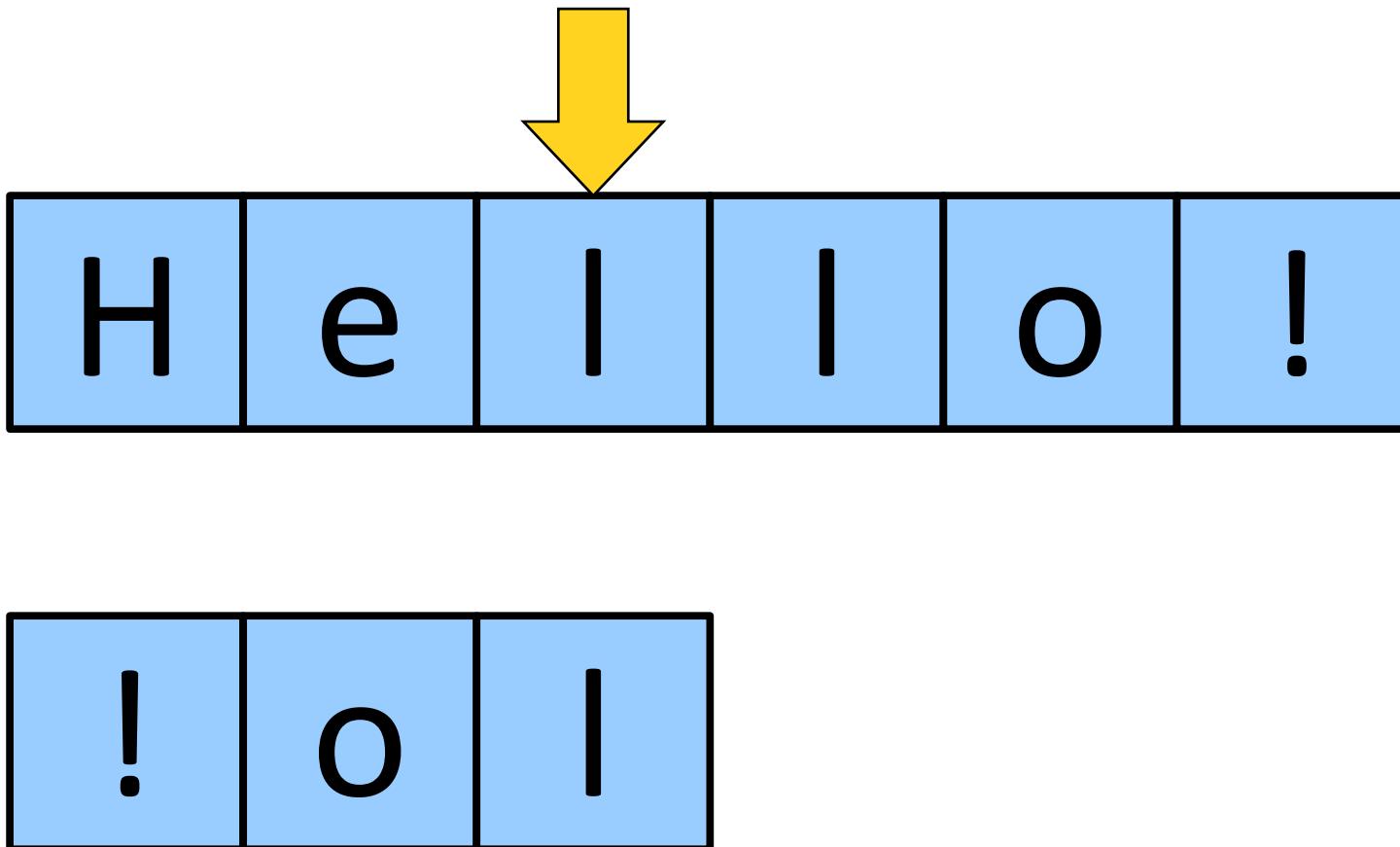
Reversing a String



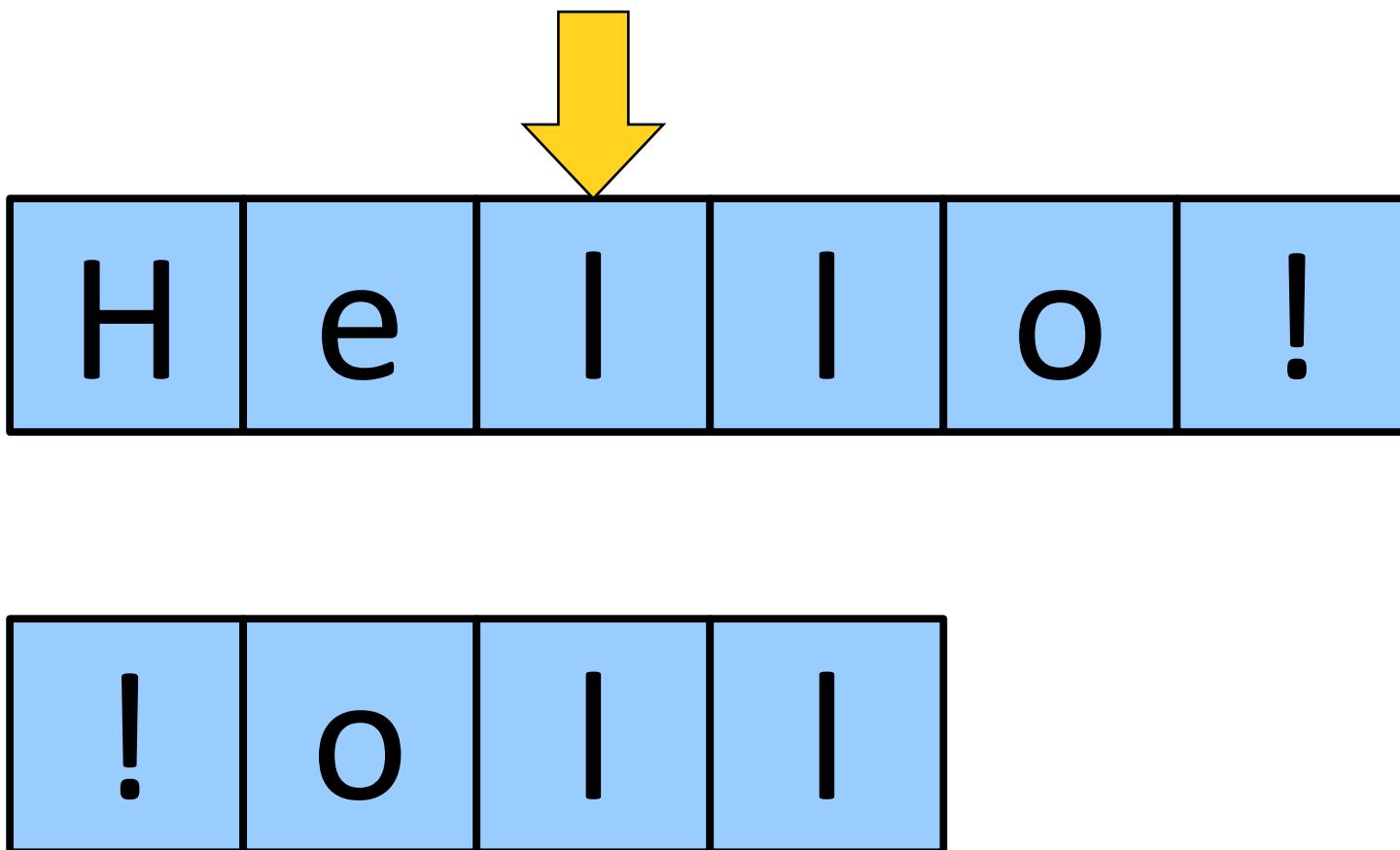
Reversing a String



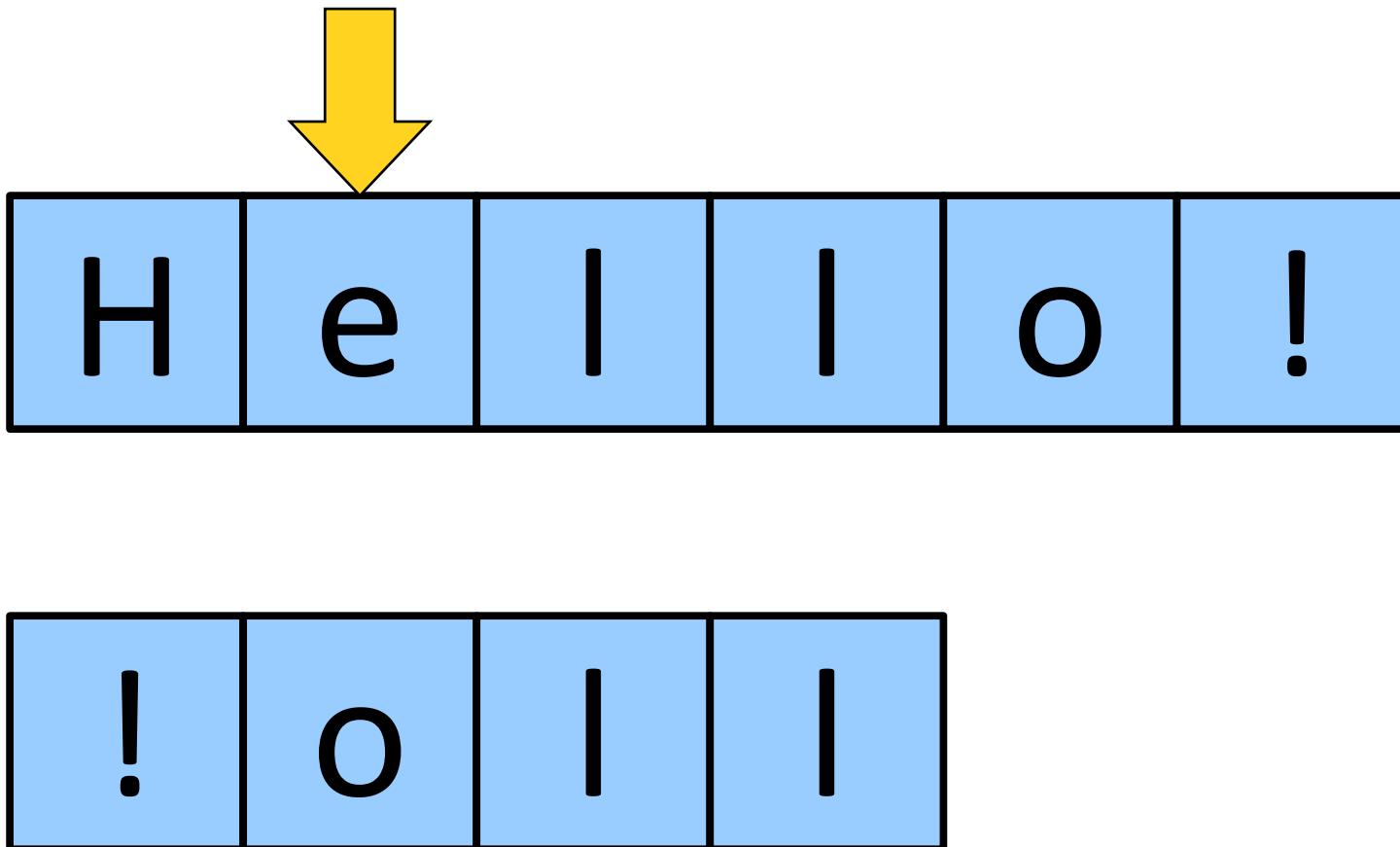
Reversing a String



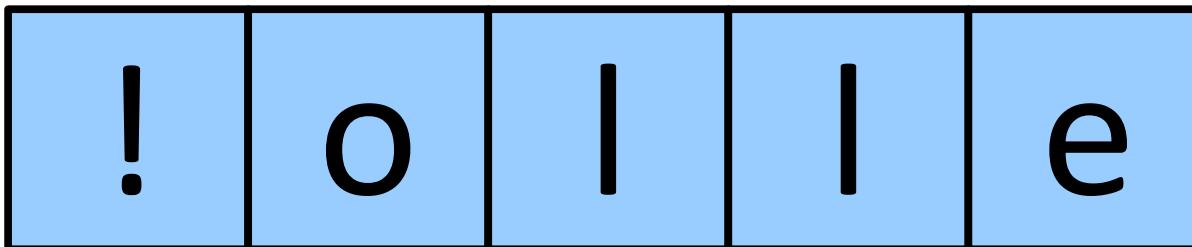
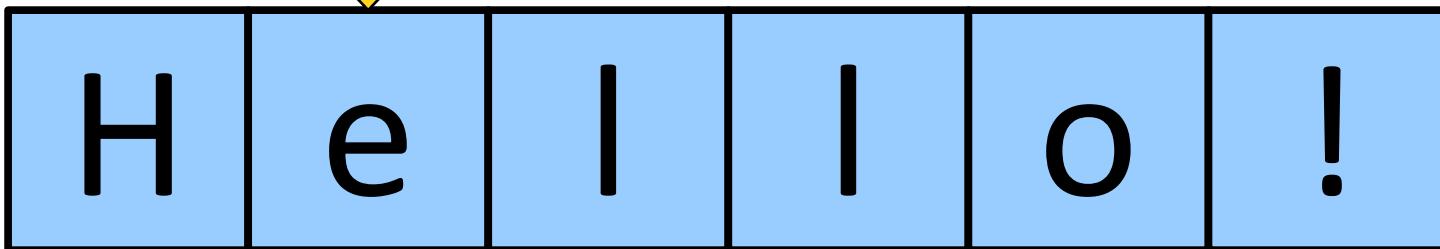
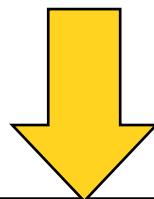
Reversing a String



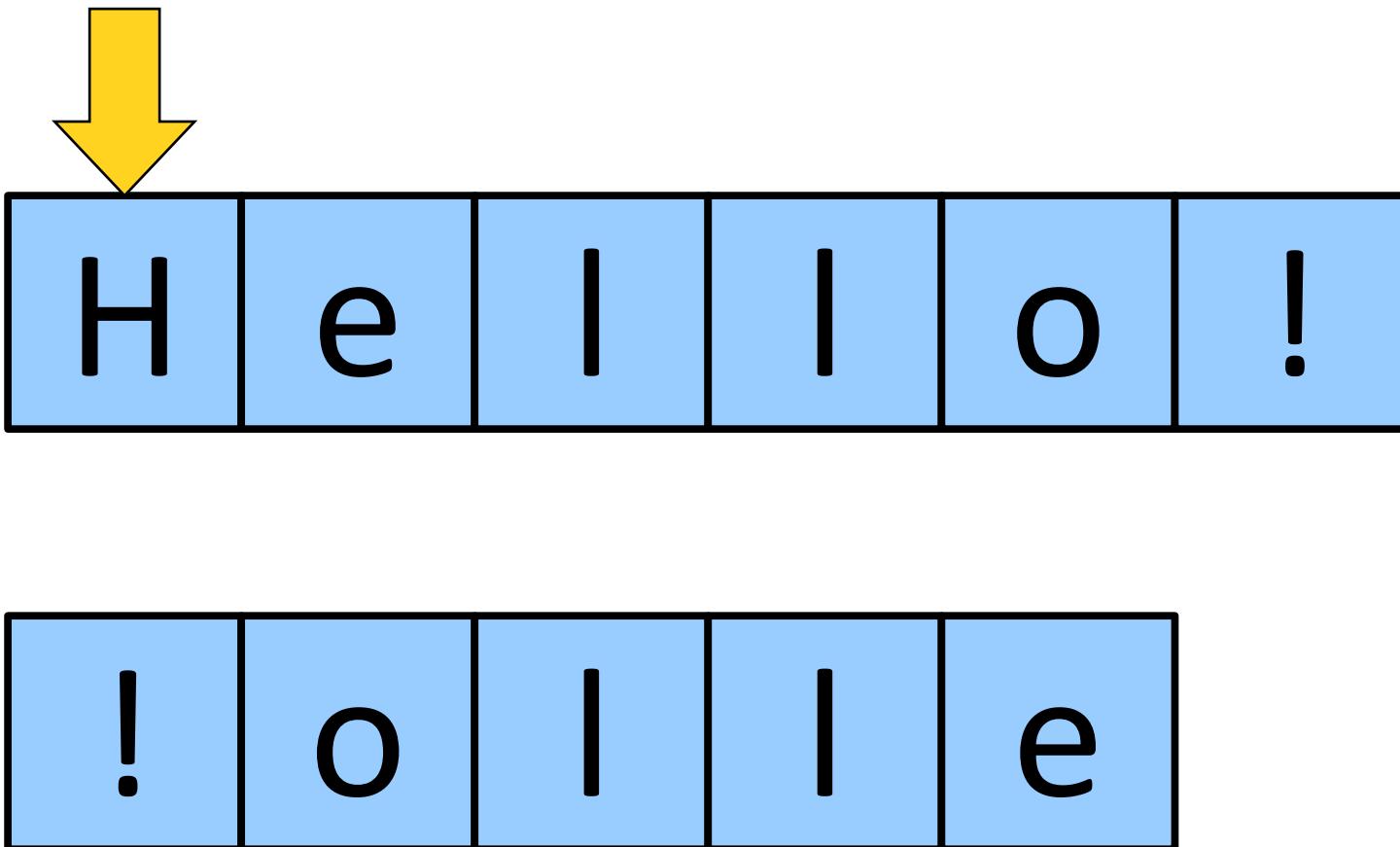
Reversing a String



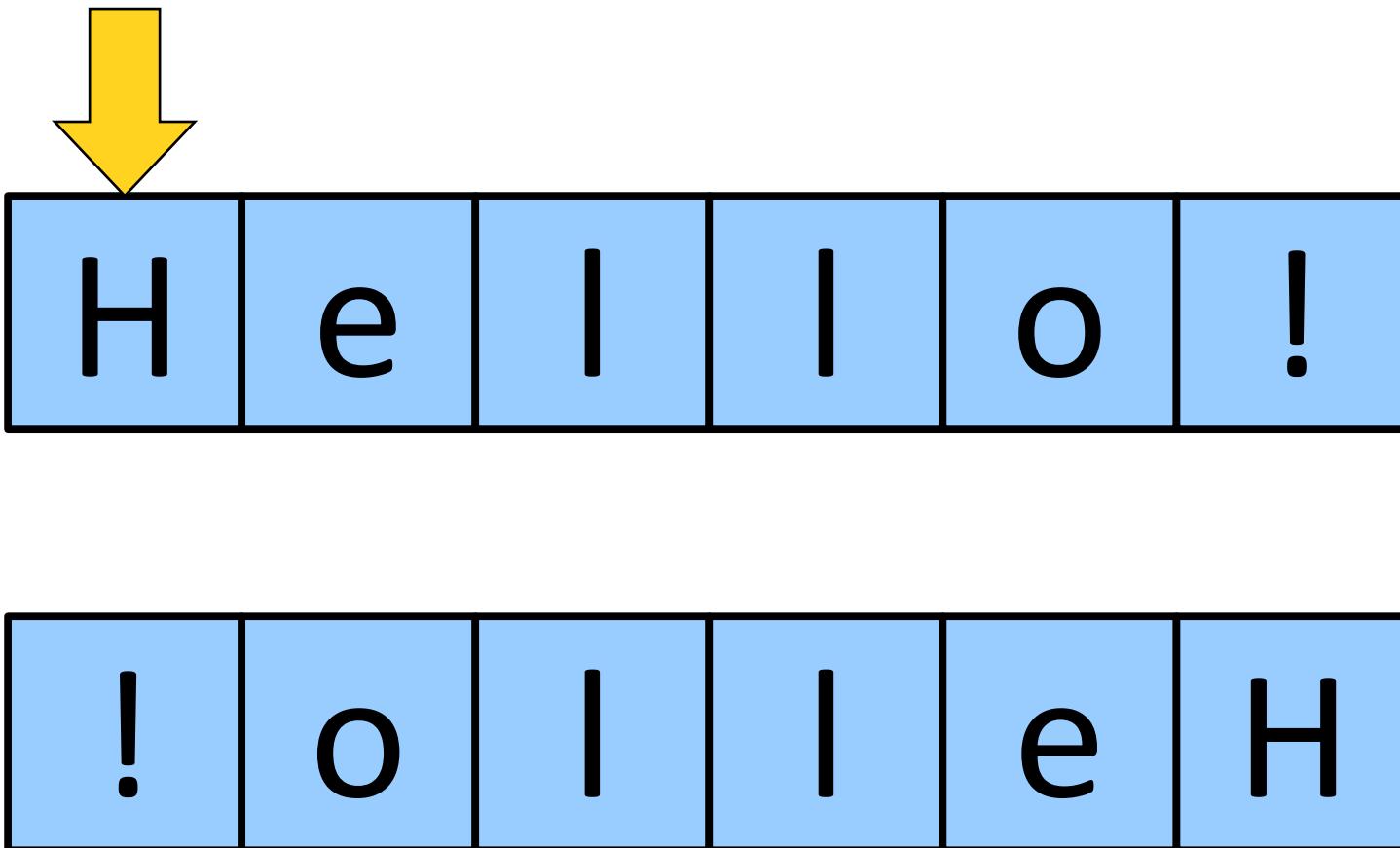
Reversing a String



Reversing a String



Reversing a String



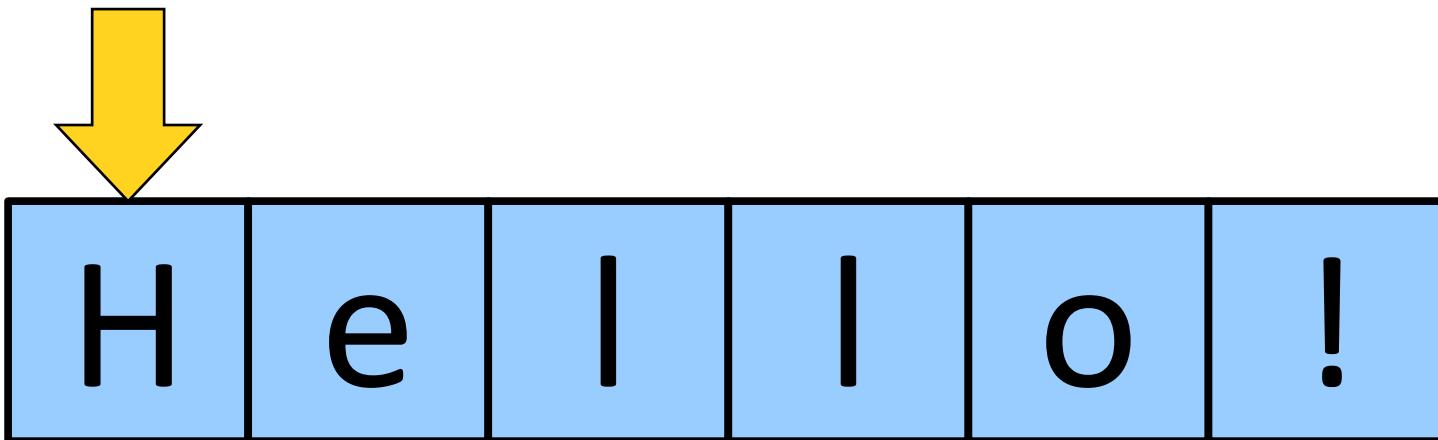
Reversing a String

H	e	I	I	o	!
---	---	---	---	---	---

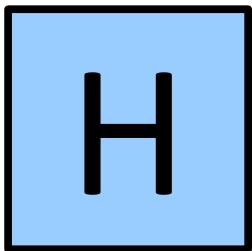
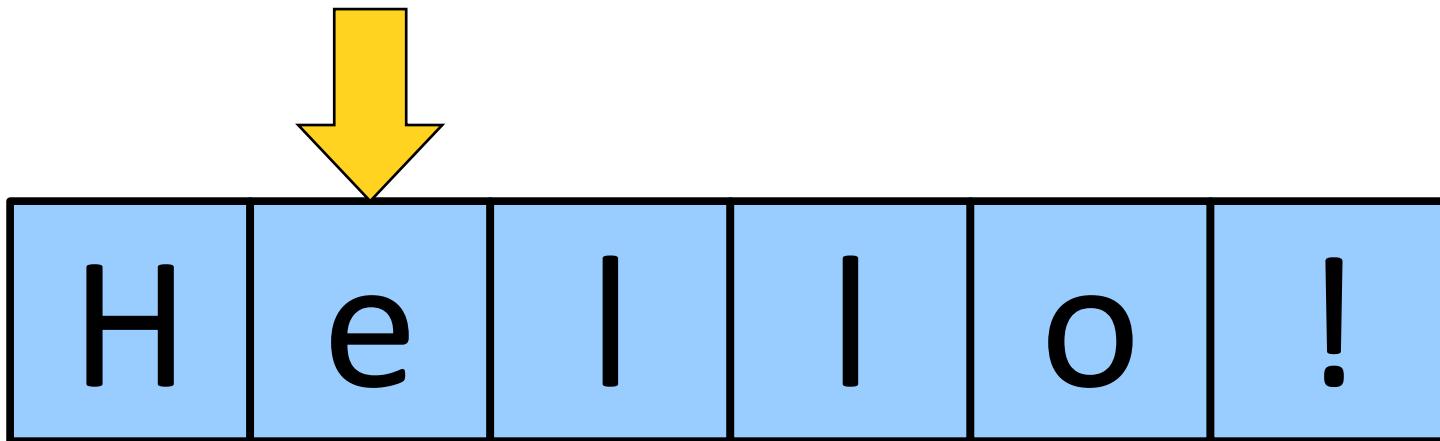
!	o	I	I	e	H
---	---	---	---	---	---



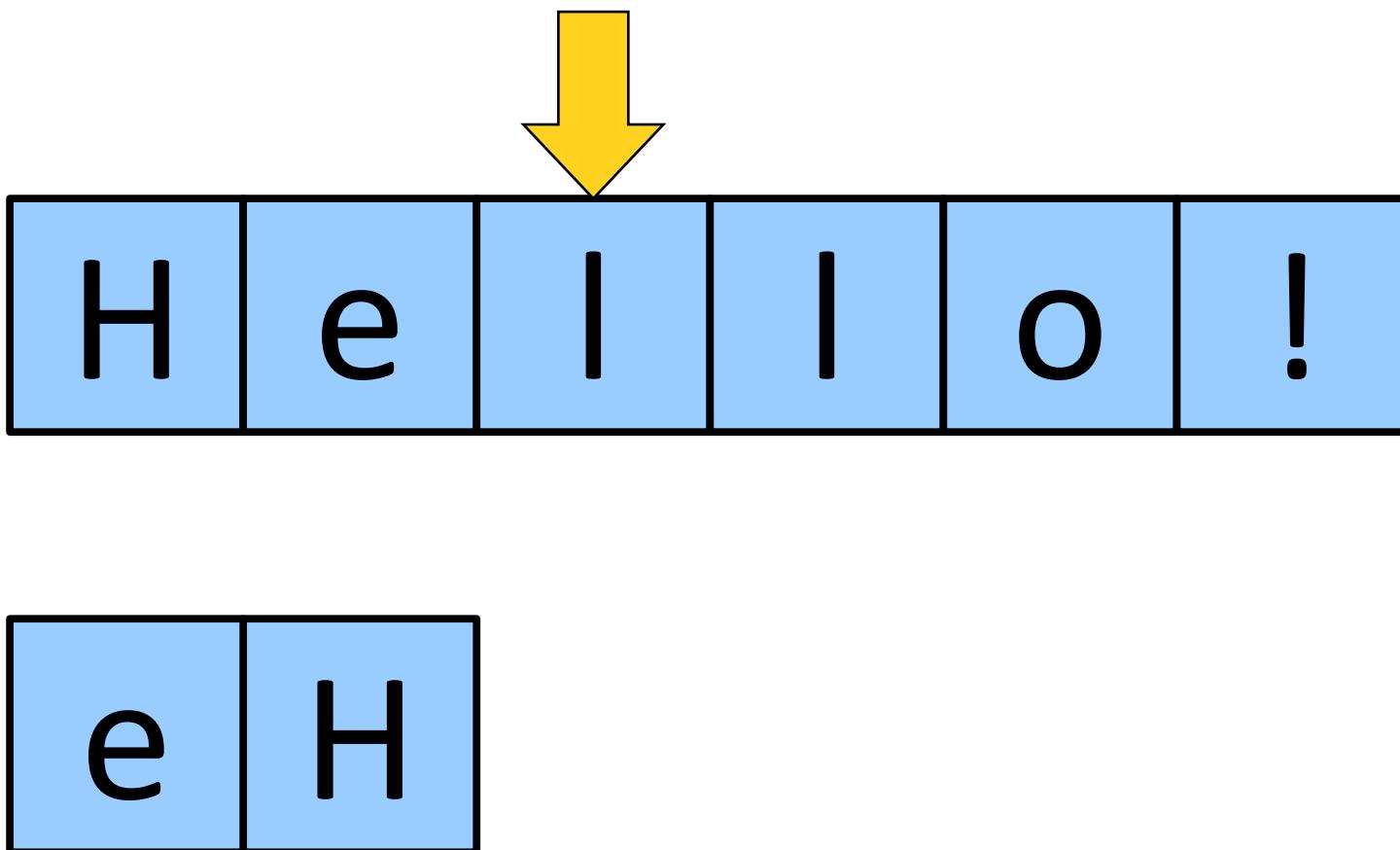
Reversing a String



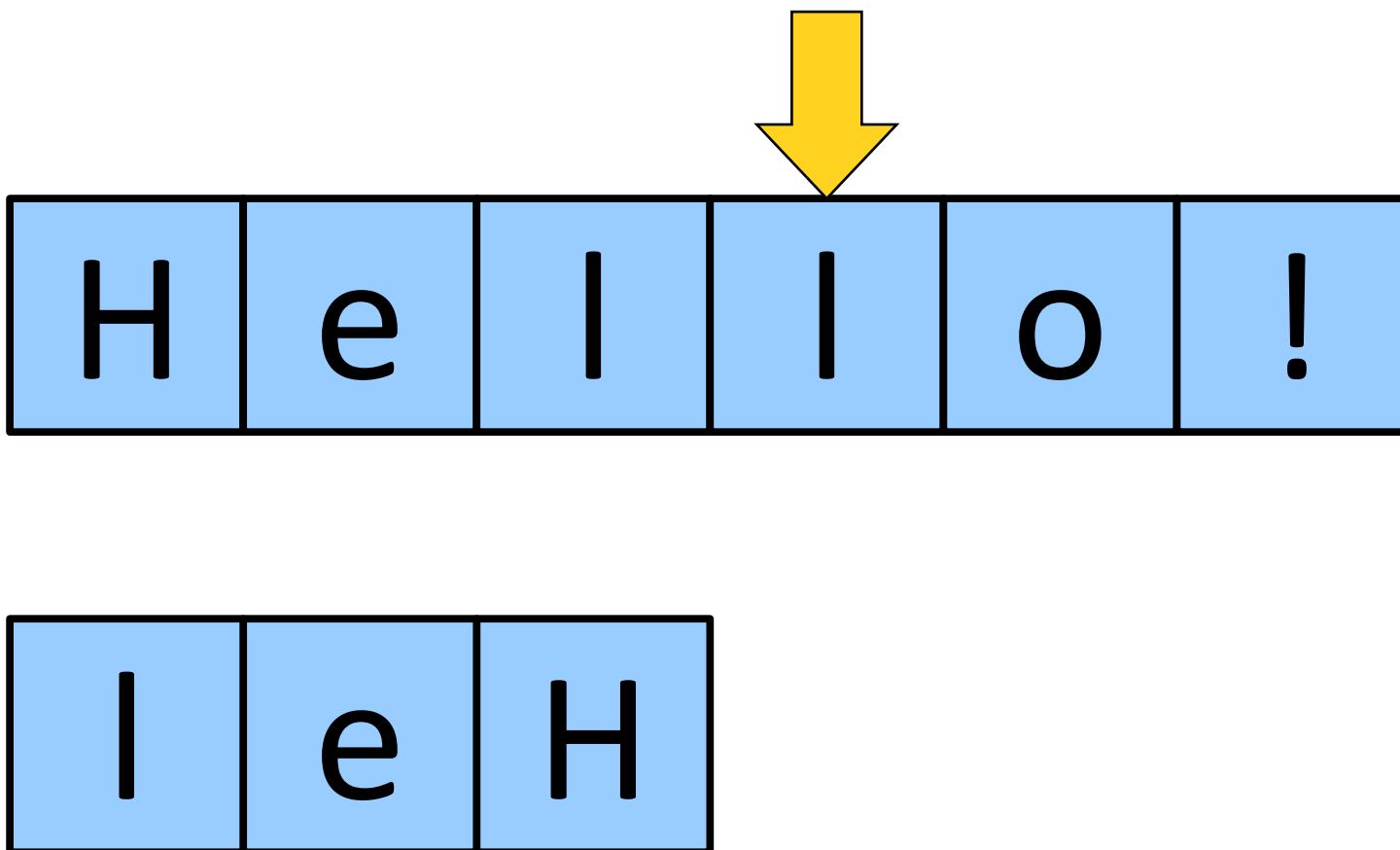
Reversing a String



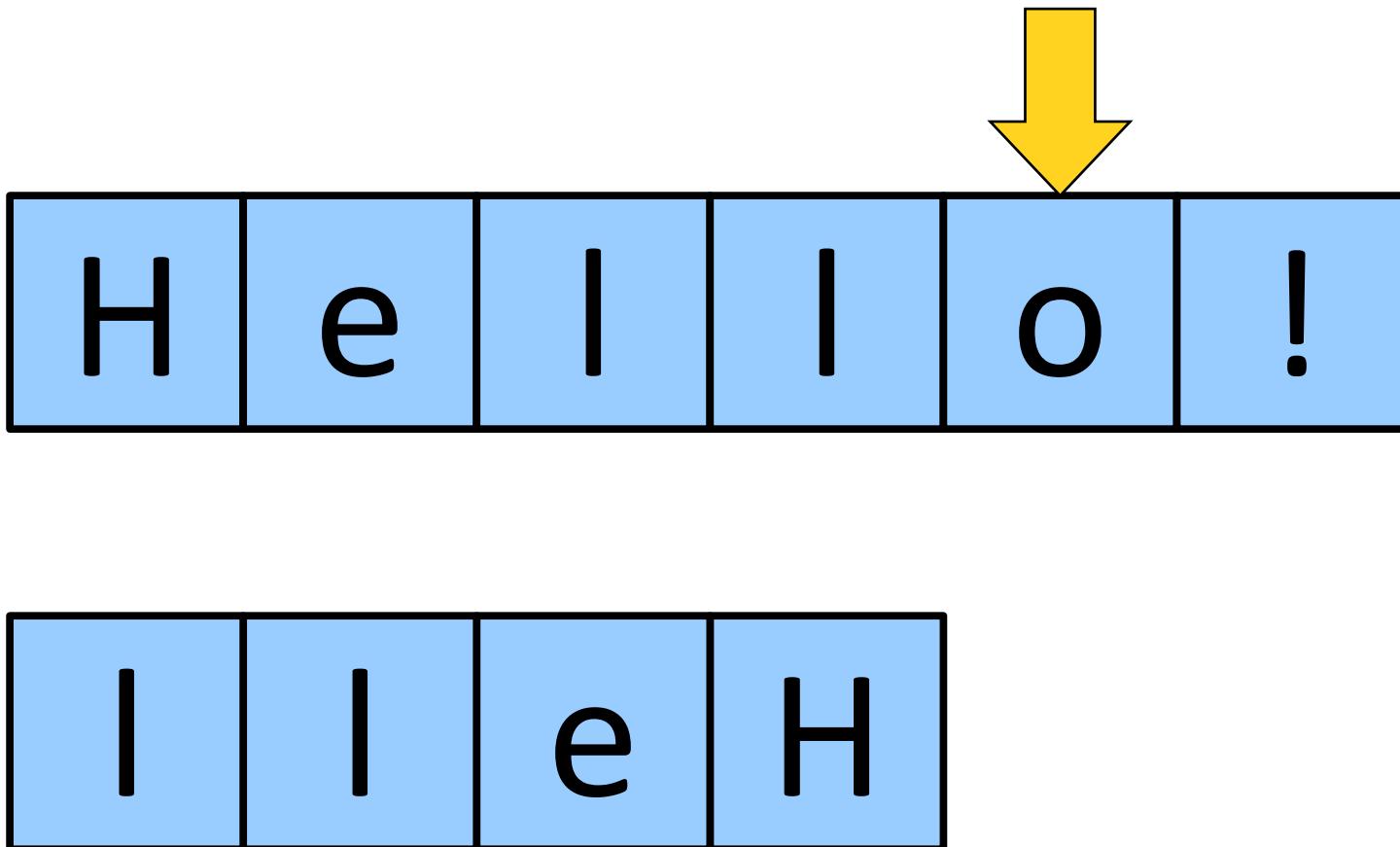
Reversing a String



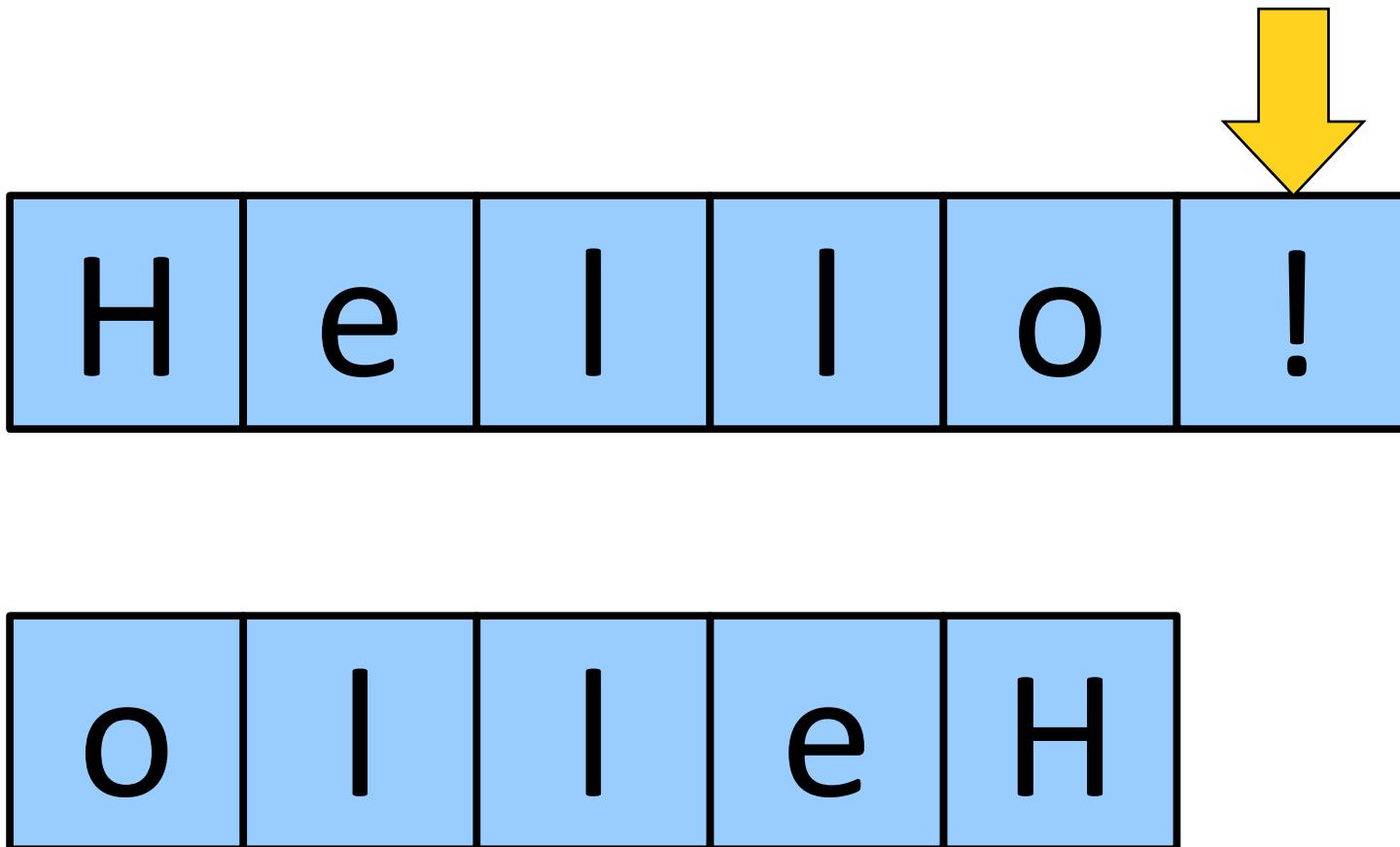
Reversing a String



Reversing a String



Reversing a String



Reversing a String

H	e	I	I	o	!
---	---	---	---	---	---

!	o	I	I	e	H
---	---	---	---	---	---



reverse_string

```
def main():
    def reverse_string(str):
        result = ""
        for i in range(len(str)):
            result = str[i] + result
    return result
```

result

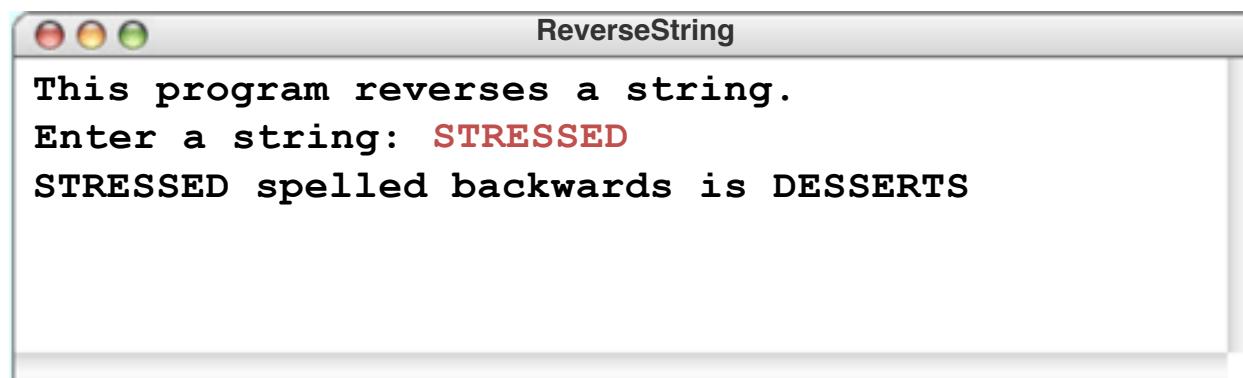
DESSERTS

str

STRESSED

i

8



reverse_string redux

```
def reverse_string(str):  
    result = ""  
    for i in range(len(str)):  
        result = str[i] + result  
    return result
```

Swiss army knife pattern

```
def reverse_string_v2(str):  
    result = ""  
    for ch in str:  
        result = ch + result  
    return result
```

Medium bear.

```
def reverse_string_v3(str):  
    """
```

This uses the slice operator in a special way. With no start, no end and a delta of -1, slice reverses.

```
    """
```

```
    return str[::-1]
```

One cool trick that will blow your mind



Palindrome

- A **palindrome** is a string that reads the same forwards and backwards.
- For example:
 - Abba
 - Racecar
 - Kayak
 - Mr. Owl ate my metal worm.
 - Go hang a salami! I'm a lasagna hog.
 - Elu par cette crapule



Some test cases

- Let's test our program on some examples:
 - Racecar
 - Kayak
 - Mr. Owl ate my metal worm.
 - Go hang a salami! I'm a lasagna hog.
- Will it work?



More Palindromes

Here are some palindromes in other languages:

- بلح تعلق تحت قلعة حلب (Dates hang underneath a castle in Halab)
- 여보, 안경 안보여 (Honey, I can't see my glasses)
- কড়ক (a loud thunderous sound)
- 上海自來水來自海上 (Shanghai tap water originates from "above" the ocean)

The comedian Dmitri Martin also has a routine about palindromes check it out at
<https://www.youtube.com/watch?v=0hUHDIOazIU>



Stress Test

A man, a plan, a caret, a ban, a myriad, a sum, a lac, a liar, a hoop, a pint, a catalpa, a gas, an oil, a bird, a yell, a vat, a caw, a pax, a wag, a tax, a nay, a ram, a cap, a yam, a gay, a tsar, a wall, a car, a luger, a ward, a bin, a woman, a vassal, a wolf, a tuna, a nit, a pall, a fret, a watt, a bay, a daub, a tan, a cab, a datum, a gall, a hat, a tag, a zap, a say, a jaw, a lay, a wet, a gallop, a tug, a trot, a trap, a tram, a torr, a caper, a top, a tonk, a toll, a ball, a fair, a sax, a minim, a tenor, a bass, a passer, a capital, a rut, an amen, a ted, a cabal, a tang, a sun, an ass, a maw, a sag, a jam, a dam, a sub, a salt, an axon, a sail, an ad, a wadi, a radian, a room, a rood, a rip, a tad, a pariah, a revel, a reel, a reed, a pool, a plug, a pin, a peek, a parabola, a dog, a pat, a cud, a nu, a fan, a pal, a rum, a nod, an eta, a lag, an eel, a batik, a mug, a mot, a nap, a maxim, a mood, a leek, a grub, a gob, a gel, a drab, a citadel, a total, a cedar, a tap, a gag, a rat, a manor, a bar, a gal, a cola, a pap, a yaw, a tab, a raj, a gab, a nag, a pagan, a bag, a jar, a bat, a way, a papa, a local, a gar, a baron, a mat, a rag, a gap, a tar, a decal, a tot, a led, a tic, a bard, a leg, a bog, a burg, a keel, a doom, a mix, a map, an atom, a gum, a kit, a baleen, a gala, a ten, a don, a mural, a pan, a faun, a ducat, a pagoda, a lob, a rap, a keep, a nip, a gulp, a loop, a deer, a leer, a lever, a hair, a pad, a tapir, a door, a moor, an aid, a raid, a wad, an alias, an ox, an atlas, a bus, a madam, a jag, a saw, a mass, an anus, a gnat, a lab, a cadet, an em, a natural, a tip, a caress, a pass, a baronet, a minimax, a sari, a fall, a ballot, a knot, a pot, a rep, a carrot, a mart, a part, a tort, a gut, a poll, a gateway, a law, a jay, a sap, a zag, a tat, a hall, a gamut, a dab, a can, a tabu, a day, a batt, a waterfall, a patina, a nut, a flow, a lass, a van, a mow, a nib, a draw, a regular, a call, a war, a stay, a gam, a yap, a cam, a ray, an ax, a tag, a wax, a paw, a cat, a valley, a drib, a lion, a saga, a plat, a catnip, a pooh, a rail, a calamus, a dairyman, a bater, a canal – Panama!

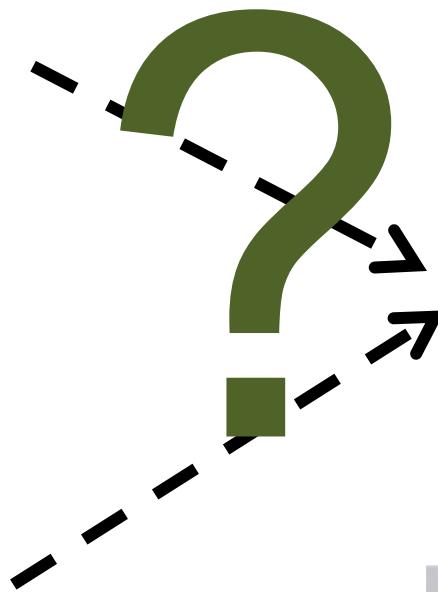


Remember!

Counterfeiter



You (Distributor)



User



Piech and Sahami, CS106A, Stanford University



Can you solve it?



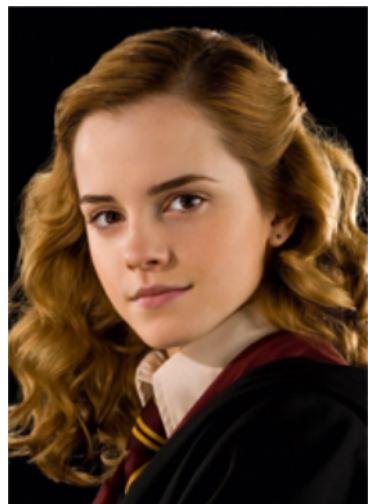
Bright Simons

Piech and Sahami, CS106A, Stanford University



Underlying Puzzle

Counterfeiter



You (Distributor)

User



Underlying Puzzle

Counterfeiter



You (Distributor)



User



Make a code to
put on every box



1. Unique
2. Impossible to guess

Insight

Code = RandomNum + UniqueNum

So that it is
impossible to guess

Concatenation

UniqueNum

So that no two
codes are the same



M-Pedigree

Every string
should be the
same length

MPedigree
4843220000
9861230001
2330240002
8047970003
1543690004
2787880005
9838840006
5224750007
2661390008
3482180009
4249170010
4133400011
1984670012
8917780013
6907970014
9829370015
3775510016
9956230017
0649500018
4208970019
1740950020
7023530021
0679450022

```
def main():
    # prints a set of unique labels
    for i in range(N_LABELS):
        rand_value = random.randint(0, MAX)
        rand_part = pad(rand_value, 6)
        unique_part = pad(i, 4)
        print(rand_part + unique_part)

# prepends 0s to a str version of a num
# until the str has goal_length len
def pad(number, goal_length):
    number_string = str(number)
    while len(number_string) < goal_length:
        number_string = '0' + number_string
    return number_string
```

