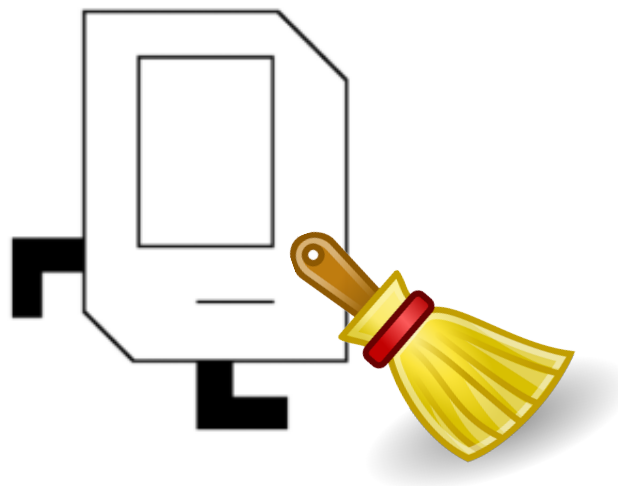


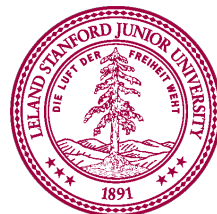


Chris Piech and Mehran Sahami
CS106A, Stanford University

Housekeeping



- Happy Friday
- Mehran made an extra example video!
- Chris AMA tomorrow at 9am. Come ask me things!
 - <https://us.edstem.org/courses/325/discussion/59604>
- Diagnostic review sessions
- Mid quarter evaluations



Fake Medicine was a Problem

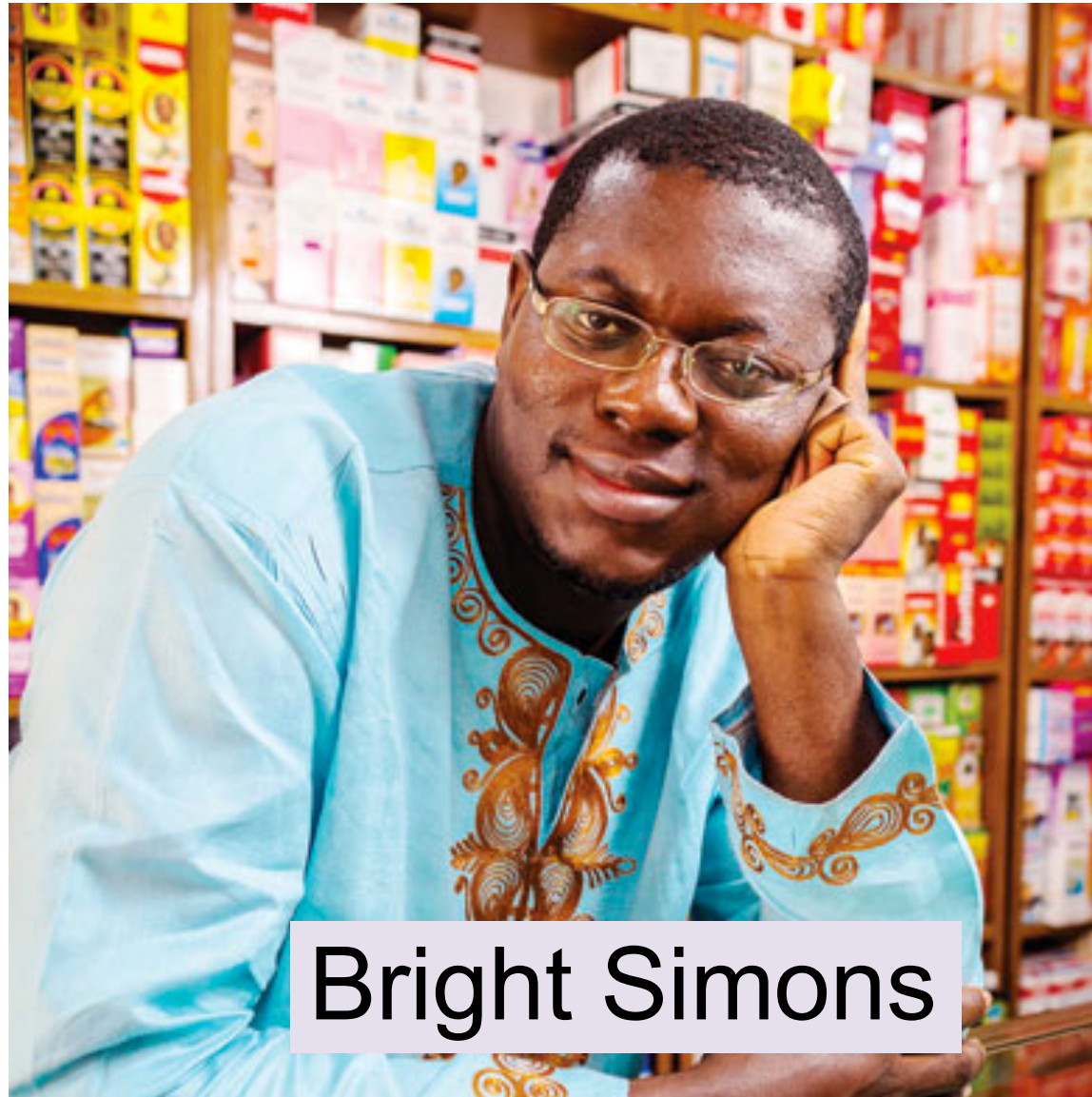
700,000 deaths a year from **fake** malaria and tuberculosis drugs [1]



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

Underlying Puzzle

Counterfeiter



User



You (Distributor)



Underlying Puzzle

Counterfeiter



You (Distributor)



User



Revisit this problem..

Learning Goals

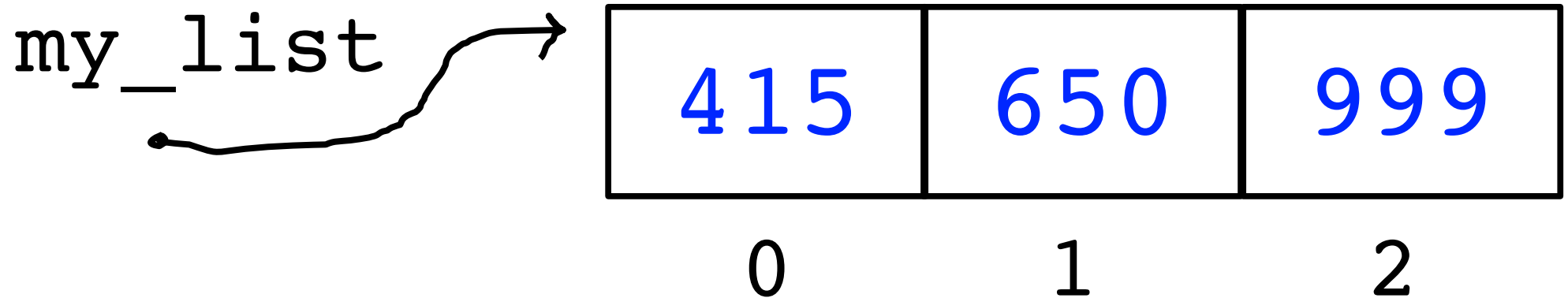
1. Write string algorithms that loops over each character



Review

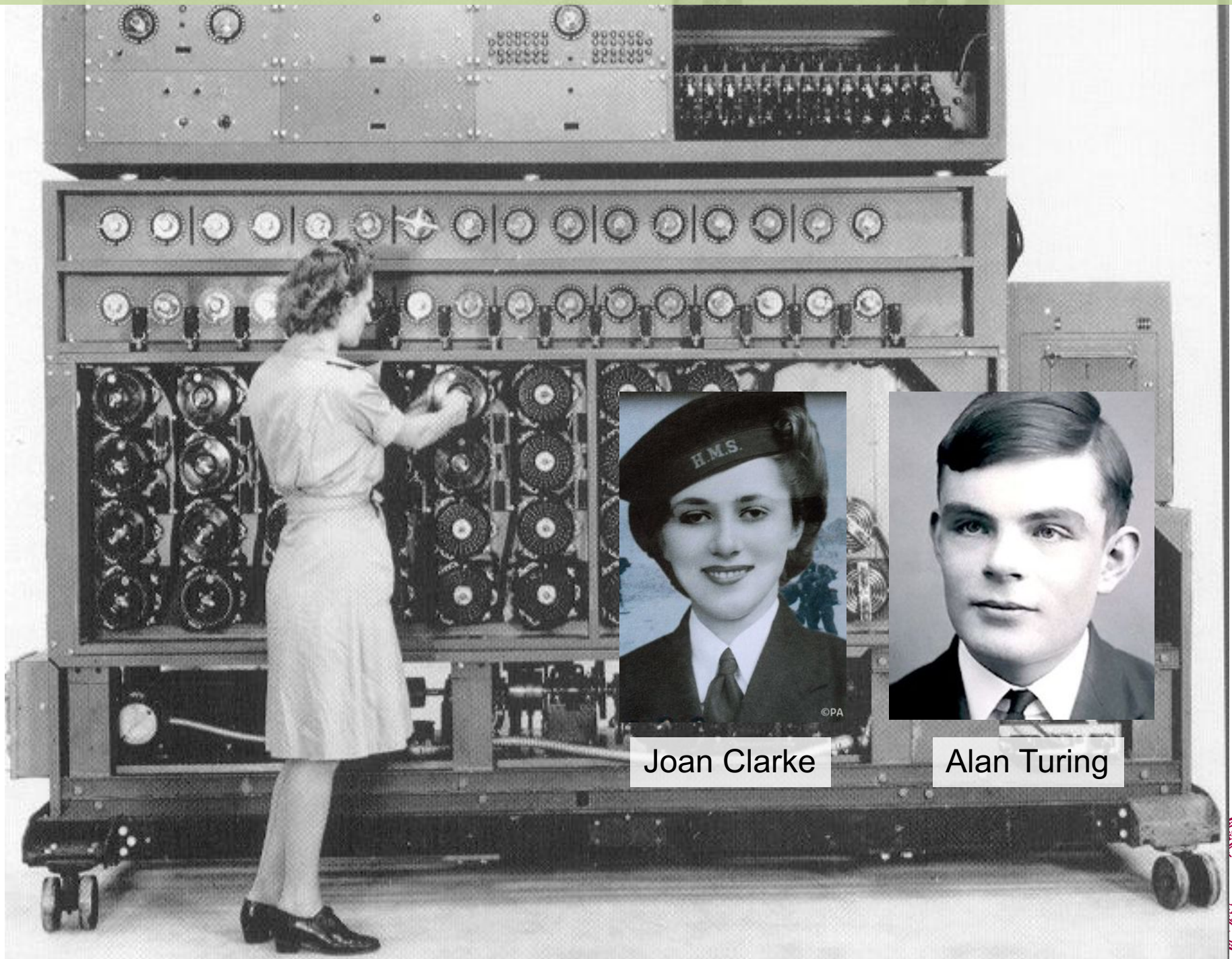
Review of lists

```
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



```
AGGTCAGTCAGATTTACCCTGGCTCACC  
TGTTTCGTACAACCAATTTAGGTGAGTTC  
TTCGGAAAGACTCCCTGGTACCATCCCCG  
CCGGGGGGTTGGAATTTACGGGTCAGAAG  
ACCAATCGTAACATATGAGAGCCACTGC  
ATAATAGGGGAGGGTTCATTTTCGTCTGT  
CTAACTTTGCTTAATACCCGACCACCAC  
CCACCCTGGCATTATAGTACCCCGAATC  
CGTAGAGCCAGATGTATGCAATGCCCCG  
GTAAGATCTCCAAAAAGGTCGACGATGA  
AGTGGGTACTTTGGATACCATCATTGGT  
ATCCGCTGATTGCTGGTTAATTCGTATG  
TCCCGGTTTCAAGTTTCAGACACTAGTT  
CCTAGGGGCGTCGACTGCGCACCATACT  
TCAATAGGGTATCGGGAGGTTTCATTAC  
TGGCACCCGTCCGTCAACGGTGTGCGCA  
GCCGCAGCTACCTCGAAAGTCATAGCGA  
CCTGATCGTCCATTACCGGGATGTGTGC  
GCGGGAGGGTCAACACAGGTAACCTCTC  
CTAACGCGTCCCCATCCCATCCGAGATT  
TTTTTTAGAAATGTTTTGTAGAATGGGAT  
TCGAGGGGTCTTCGTTACCCATGGCGA  
AGTTCGCAGTATTGACAAACGAGCATGG  
CCAGGAATTCCGATGCCGATCGTCTGAC  
ACACCTTTGTCCAATAACAAAGTAACG  
TGTGAAAGTTTTACCTAGATGGTCTAG
```

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 !

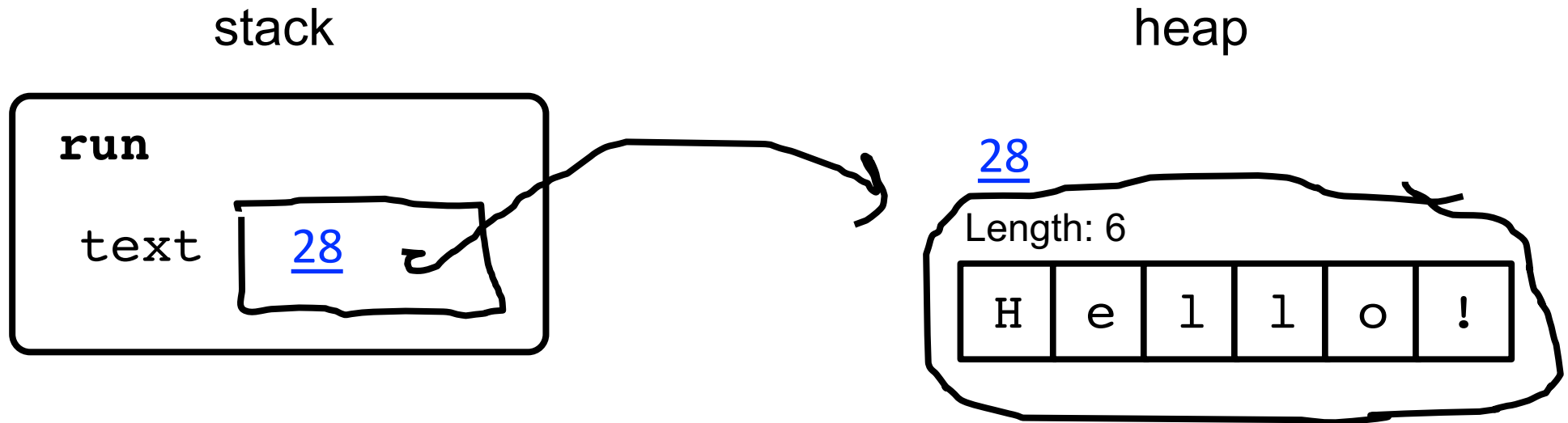


H	e	l	l	o	!
0	1	2	3	4	5



How it is actually stored

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



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

H	e	l	l	o	!
0	1	2	3	4	5

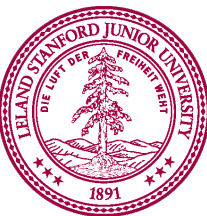
text [*index*]





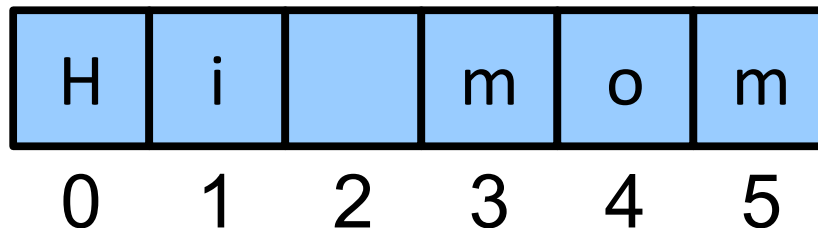
All characters in a string have
an index.

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.



(length 6)





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

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



Lets play!

```
Desktop — Python — 70x25
...RAM_VERSION=433 ... — Python • runLocal.sh ~/Desktop — Python ...thonreader — -zsh ... +
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]
' '
>>> x[3]
'm'
>>> 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

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

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

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

H	i		m	o	m
---	---	--	---	---	---

0 1 2 3 4 5

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



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

H	i		m	o	m
---	---	--	---	---	---

0 1 2 3 4 5

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



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

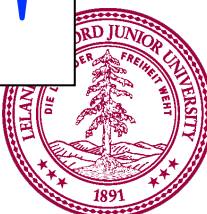
'H'

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

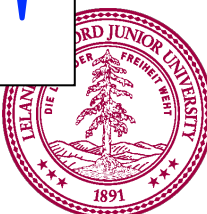
'H'

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

H	i		m	o	m
---	---	--	---	---	---

0

1

2

3

4

5

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

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



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

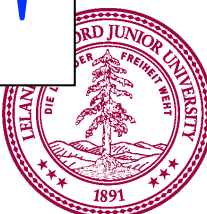
'H'

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

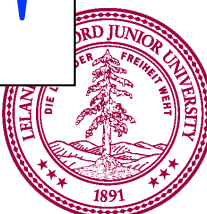
'H'

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

H	i		m	o	m
---	---	--	---	---	---

0

1

2

3

4

5

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



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

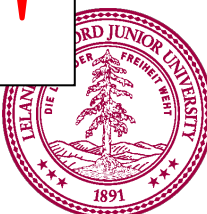
'H'

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

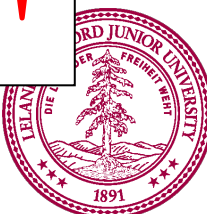
'H'

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

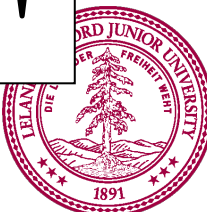
'H'

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

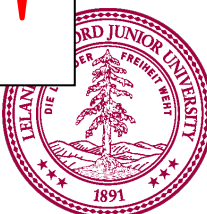
'H'

i

5

ch

'm'

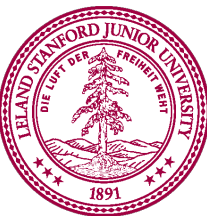


Again there are two for loops



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

good to know:

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



Just the number please

DNA $\rightarrow$ mRNA

Advanced version

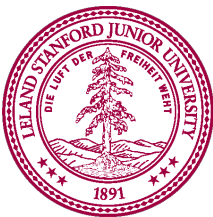
How are characters
represented?

Single characters

- Some examples:

```
letter_A    = 'A'
plus        = '+'
zero        = '0'
space       = ' '
korean_ch   = '보'
new_line    = '\n' # special
tab         = '\t' # special
backslash   = '\\ ' # special
backslash   = '\\'  # special
emoji       = '😊'
first       = text[0]
```

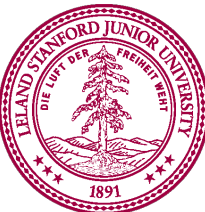
\n newline char
\t tab char
\\ backslash char
' single quote
' double quote





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



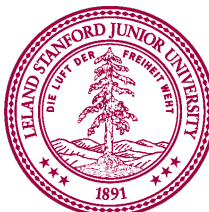
Unicode (bigger ASCII)

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)	(一)	(日)	祭	PTE	ㄱ	ㄷ	一	日	項	夜	1月	ア	チ	ム
1	(ㄴ)	(ㄹ)	(二)	株	休	㉑	ㄴ	ㄹ	二	株	休	36	2月	イ	ツ	メ
2	(ㄷ)	(ㅁ)	(三)	有	自	㉒	ㄷ	(ㅁ)	三	有	写	37	3月	ウ	テ	モ
3	(ㄷ)	(ㅁ)	(四)	社	至	㉓	ㄷ	(ㅁ)	四	社	正	38	4月	エ	ト	ヤ
4	(ㄷ)	(ㅁ)	(五)	名	問	㉔	ㄷ	(ㅁ)	五	名	上	39	5月	オ	ナ	ユ
5	(ㄷ)	(ㅁ)	(六)	特	幼	㉕	ㄷ	(ㅁ)	六	特	中	40	6月	カ	ニ	ヨ
6	(入)	(자)	(七)	財	文	㉖	(入)	(자)	七	財	下	41	7月	キ	ヌ	ラ
7	(ㅇ)	(차)	(八)	祝	筆	㉗	(ㅇ)	(차)	八	祝	左	42	8月	ク	ネ	リ
8	(스)	(카)	(九)	勞	10	㉘	(스)	(카)	九	勞	右	43	9月	ケ	ノ	ル
9	(え)	(타)	(十)	代	20	㉙	(え)	(타)	十	秘	医	44	10月	コ	ハ	レ
A	(フ)	(파)	(月)	呼	30	㉚	(フ)	(파)	月	男	宗	45	11月	サ	ヒ	ロ
B	(ㅍ)	(하)	(火)	学	40	㉛	(ㅍ)	(하)	火	女	学	46	12月	シ	フ	ワ
C	(ㄷ)	(주)	(水)	監	50	㉜	(ㄷ)	(주)	(水)	適	監	47	Hg	ス	ヘ	中
D	(ㅎ)	(오전)	(木)	企	60	㉝	(ㅎ)	(오전)	(木)	優	企	48	erg	セ	ホ	エ
E	(가)	(오후)	(金)	資	70	㉞	(가)	(오후)	(金)	印	資	49	eV	ソ	マ	ヲ
F	(나)		(土)	協	80	㉟	(나)		(土)	注	協	50	LTD	タ	ミ	誦





'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'
```

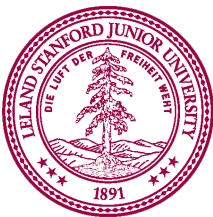
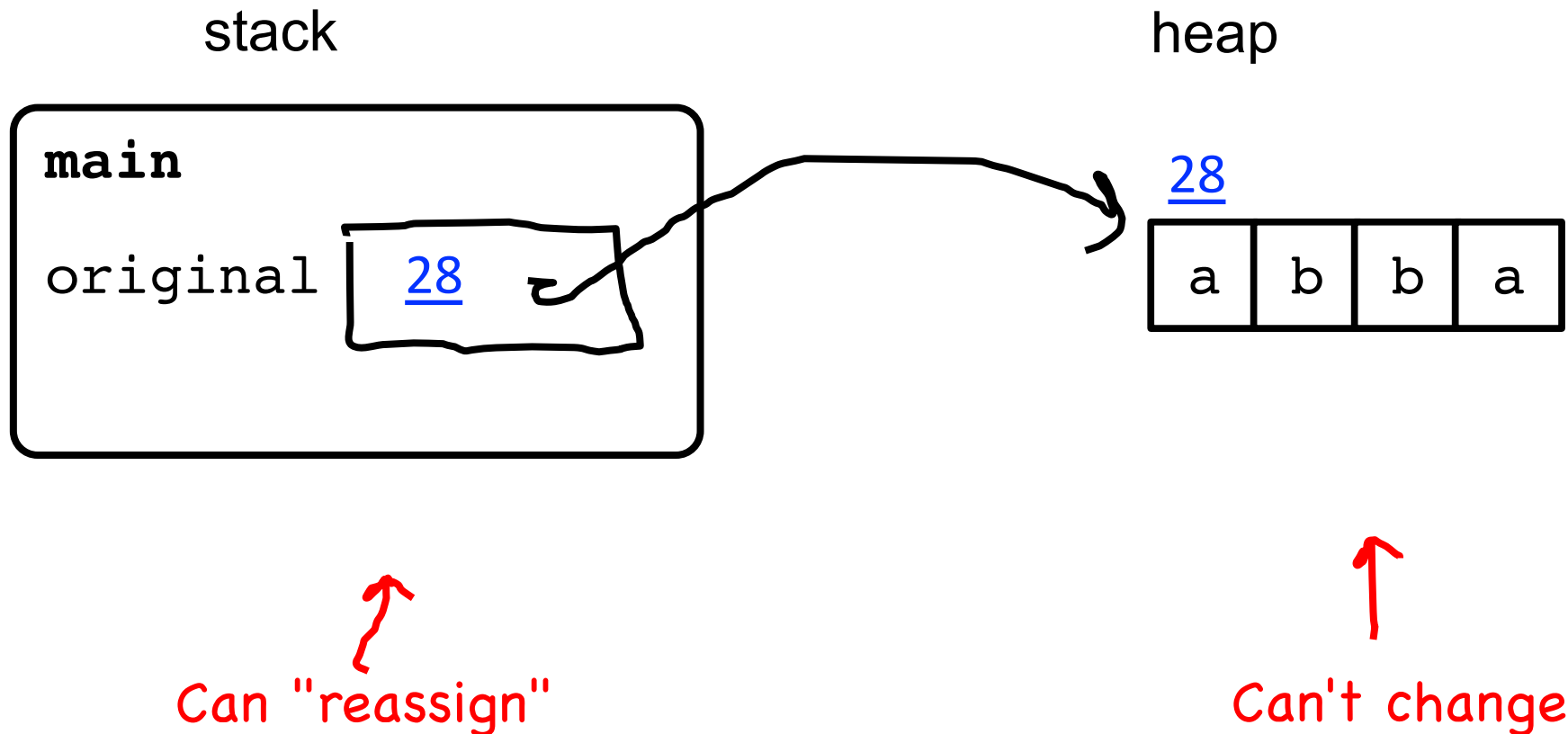
```
Traceback (most recent call last):  
...  
TypeError: 'str' object does not  
support item assignment
```



```
x = 'azc'
```

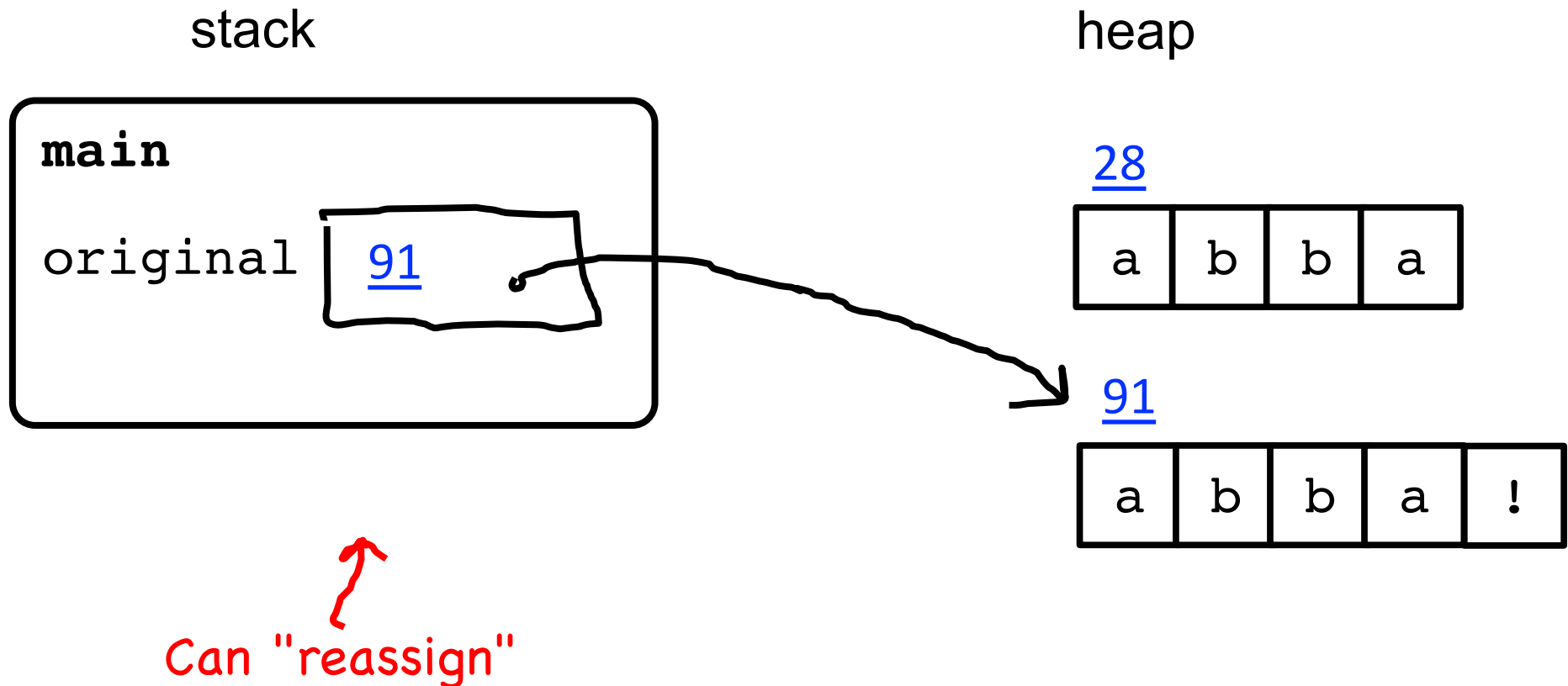
Strings are Immutable (Take 2)

```
original = 'abba'
```



Strings are Immutable (Take 1)

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



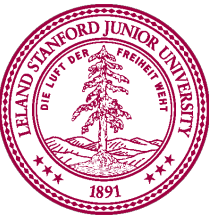
Pro tip: use id function to check a reference!



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*

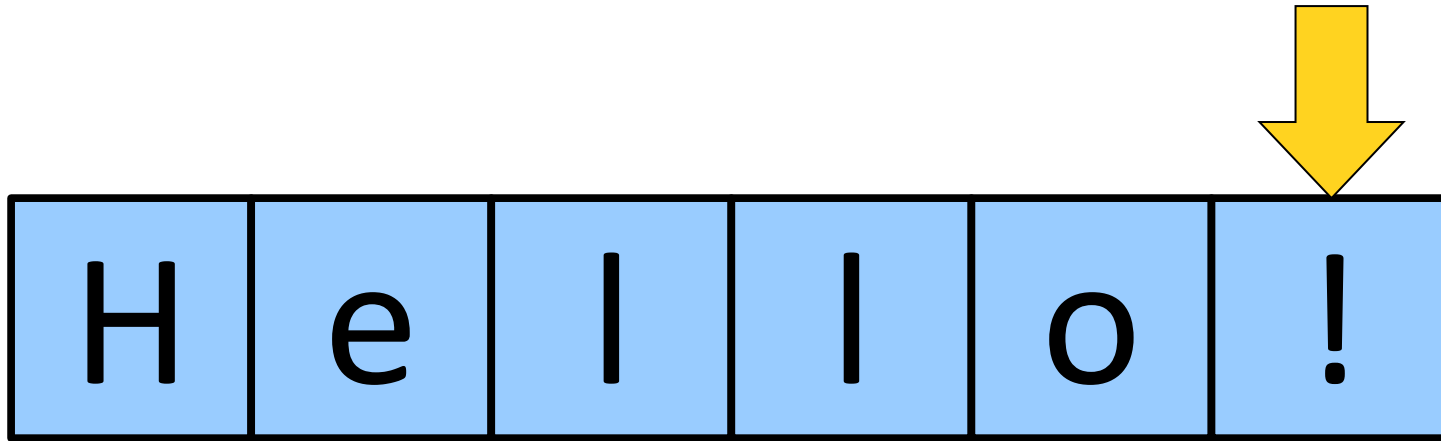




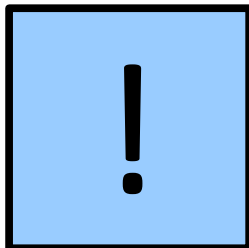
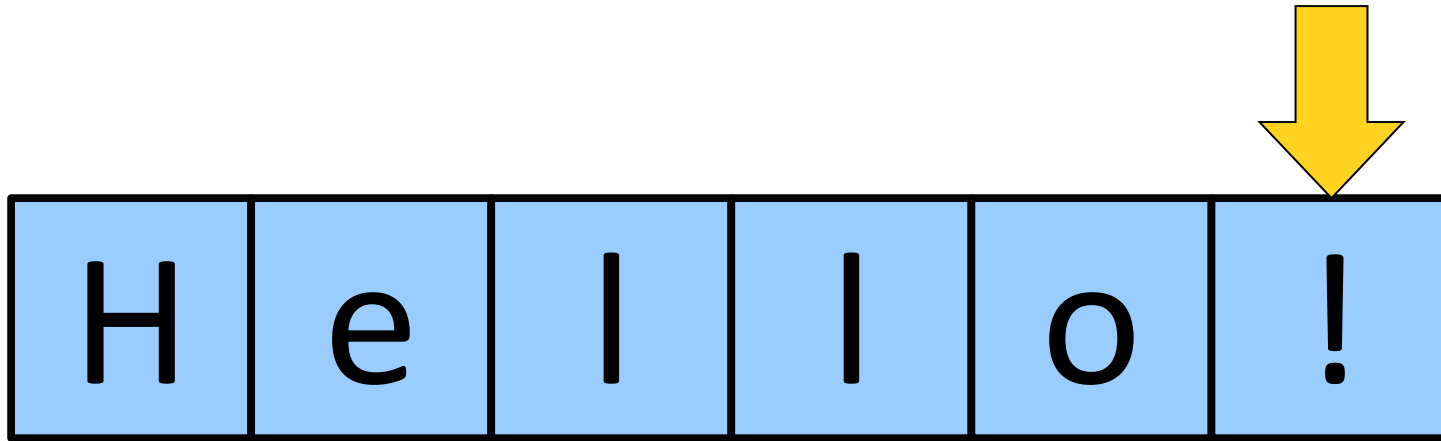
Many string algorithms use the "loop and construct" pattern.



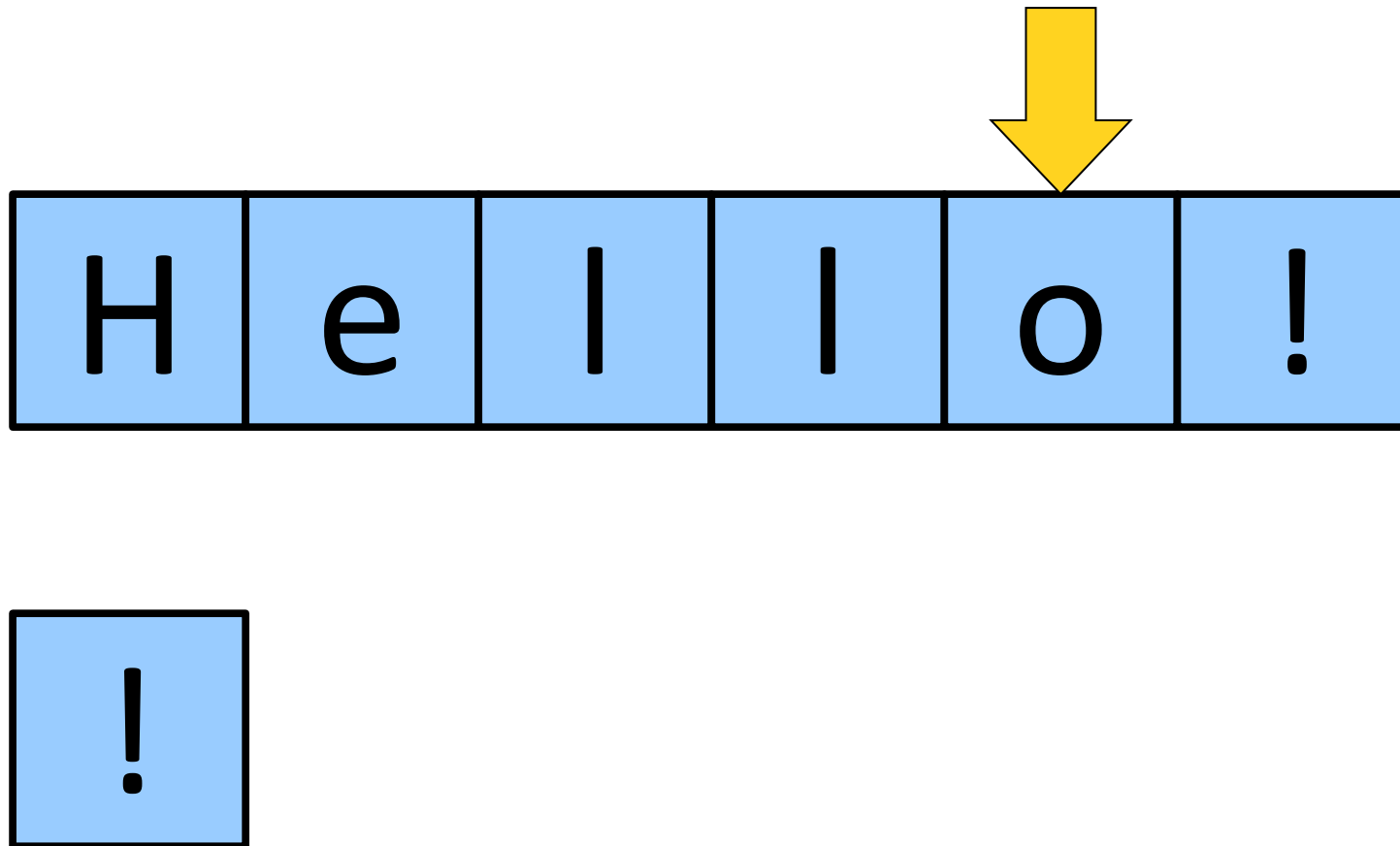
Reversing a String



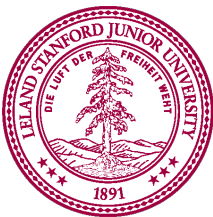
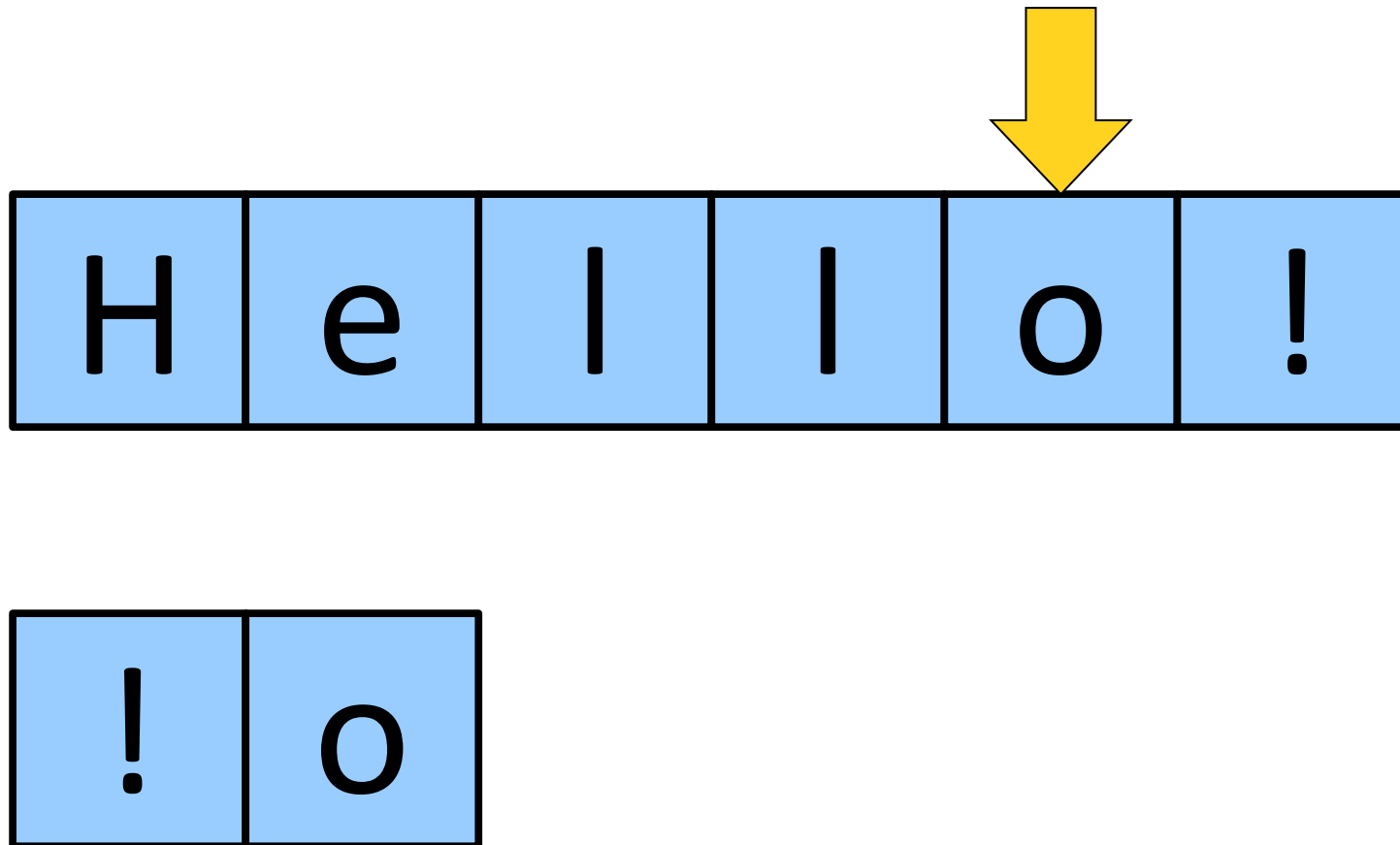
Reversing a String



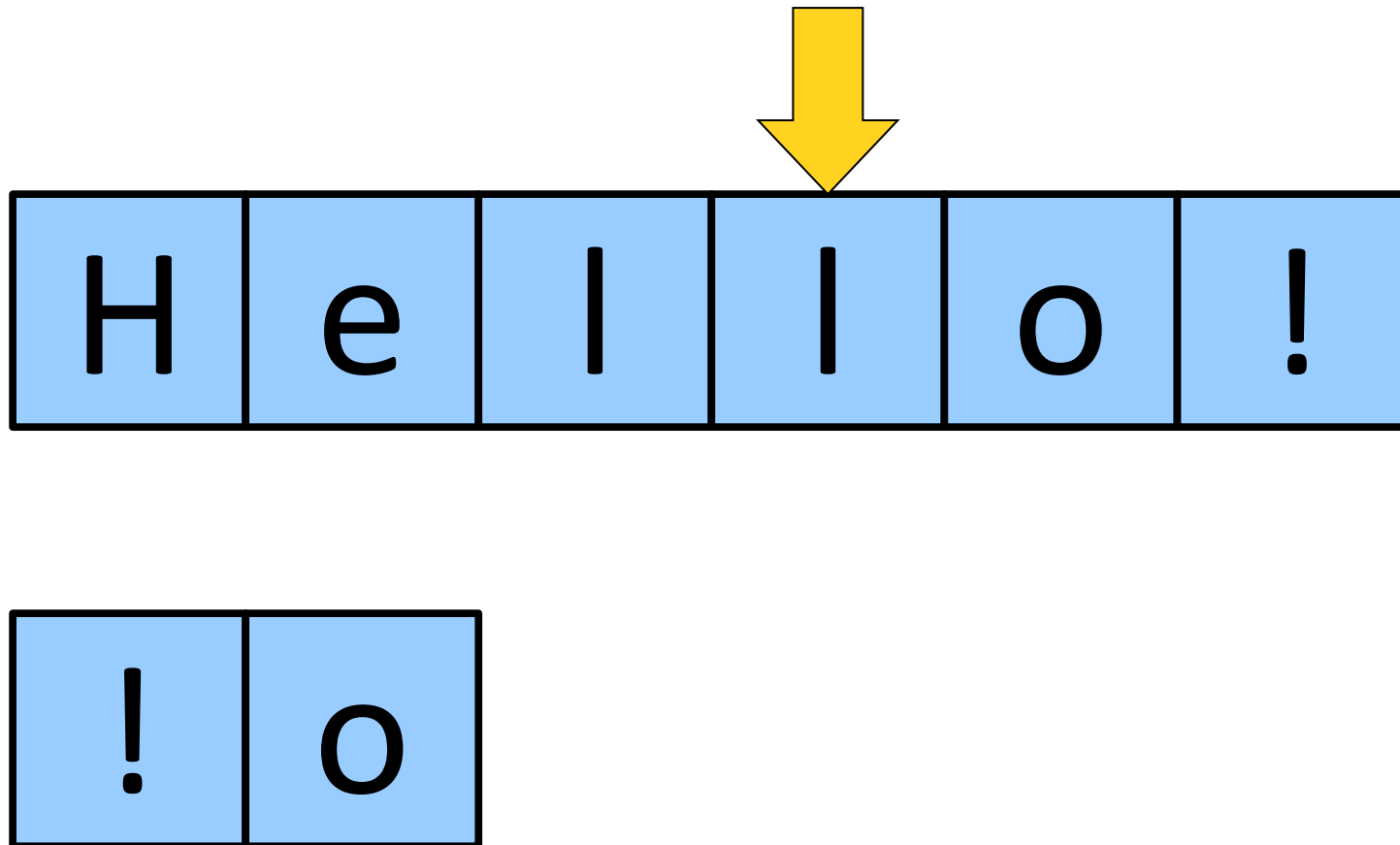
Reversing a String



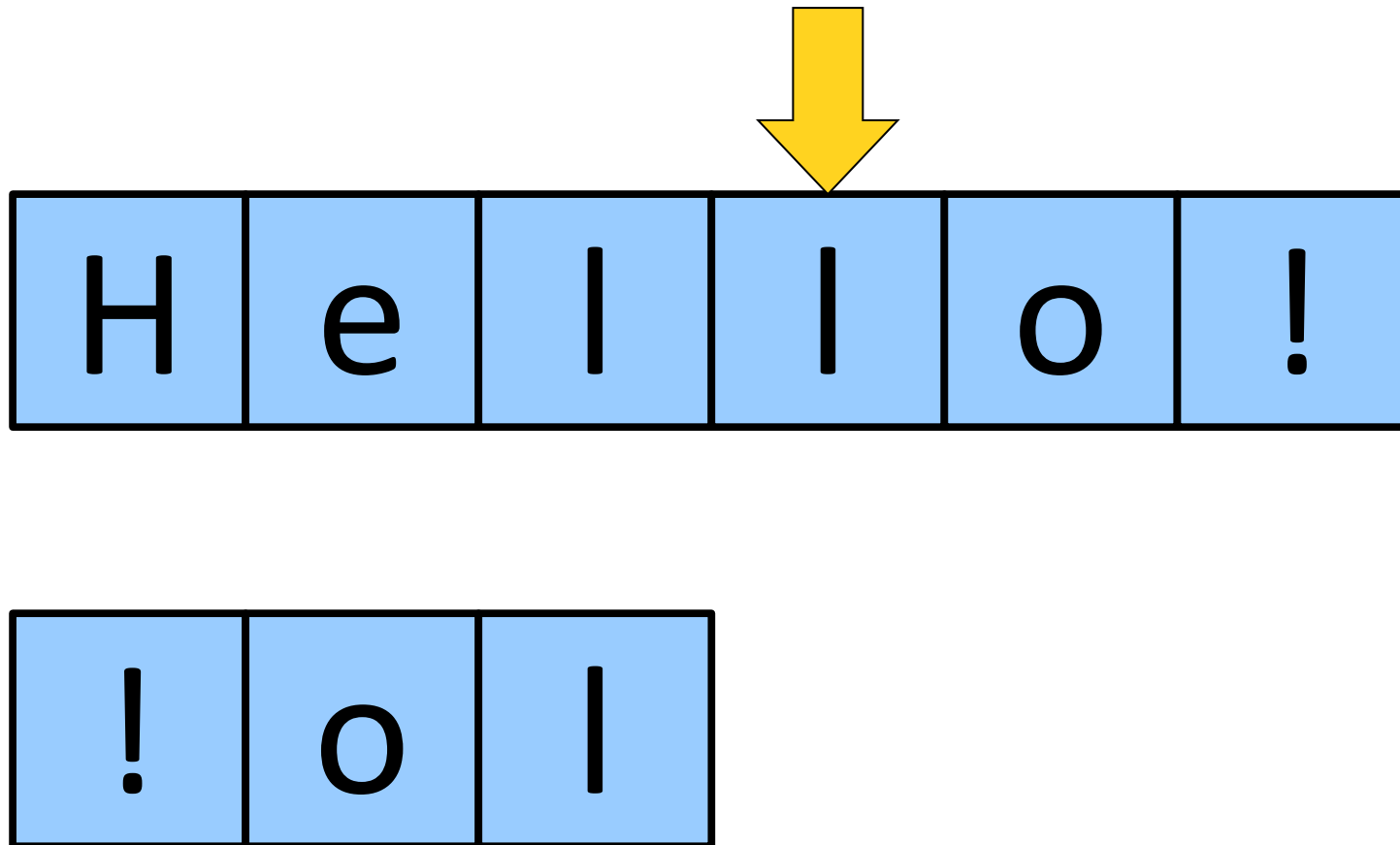
Reversing a String



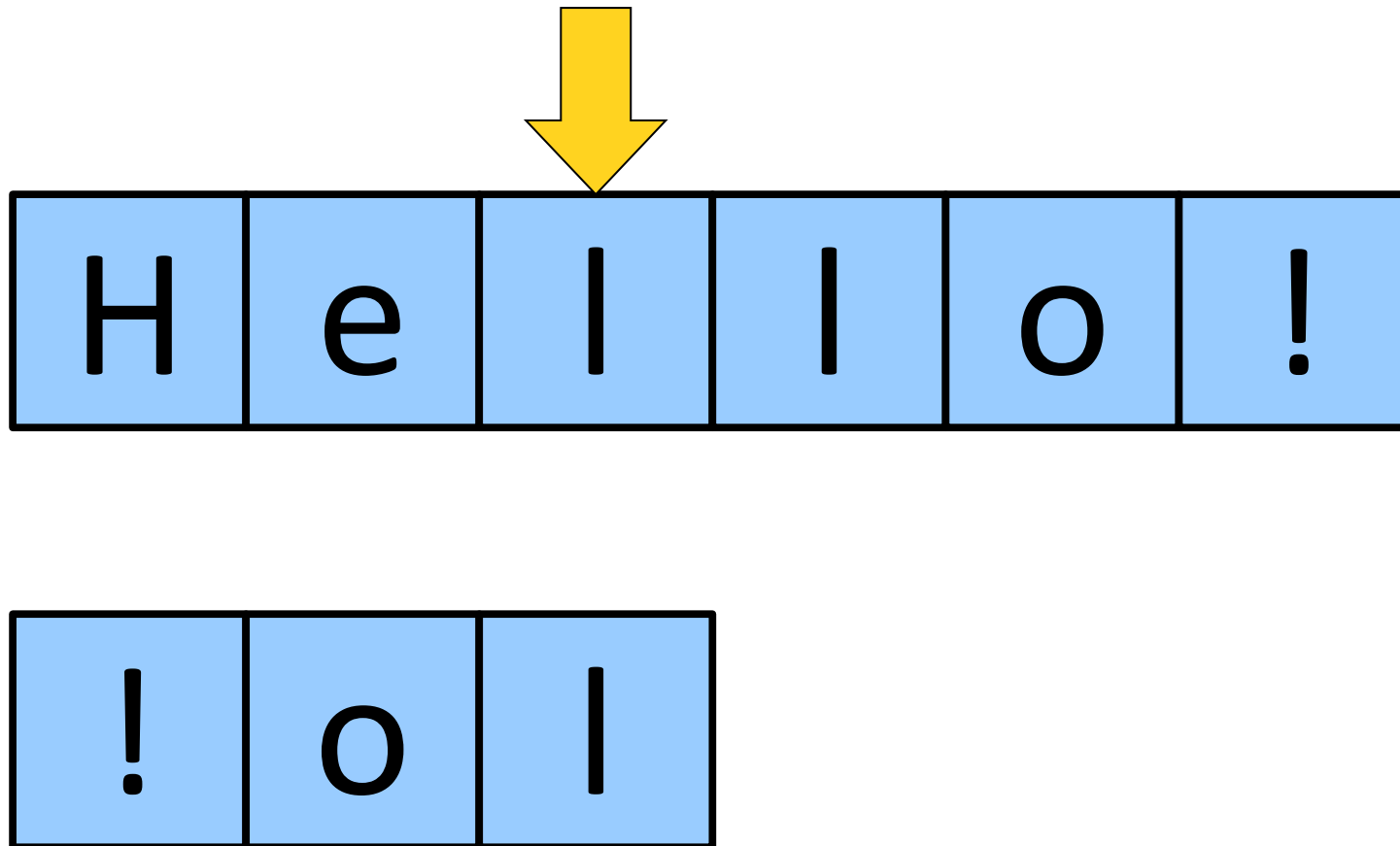
Reversing a String



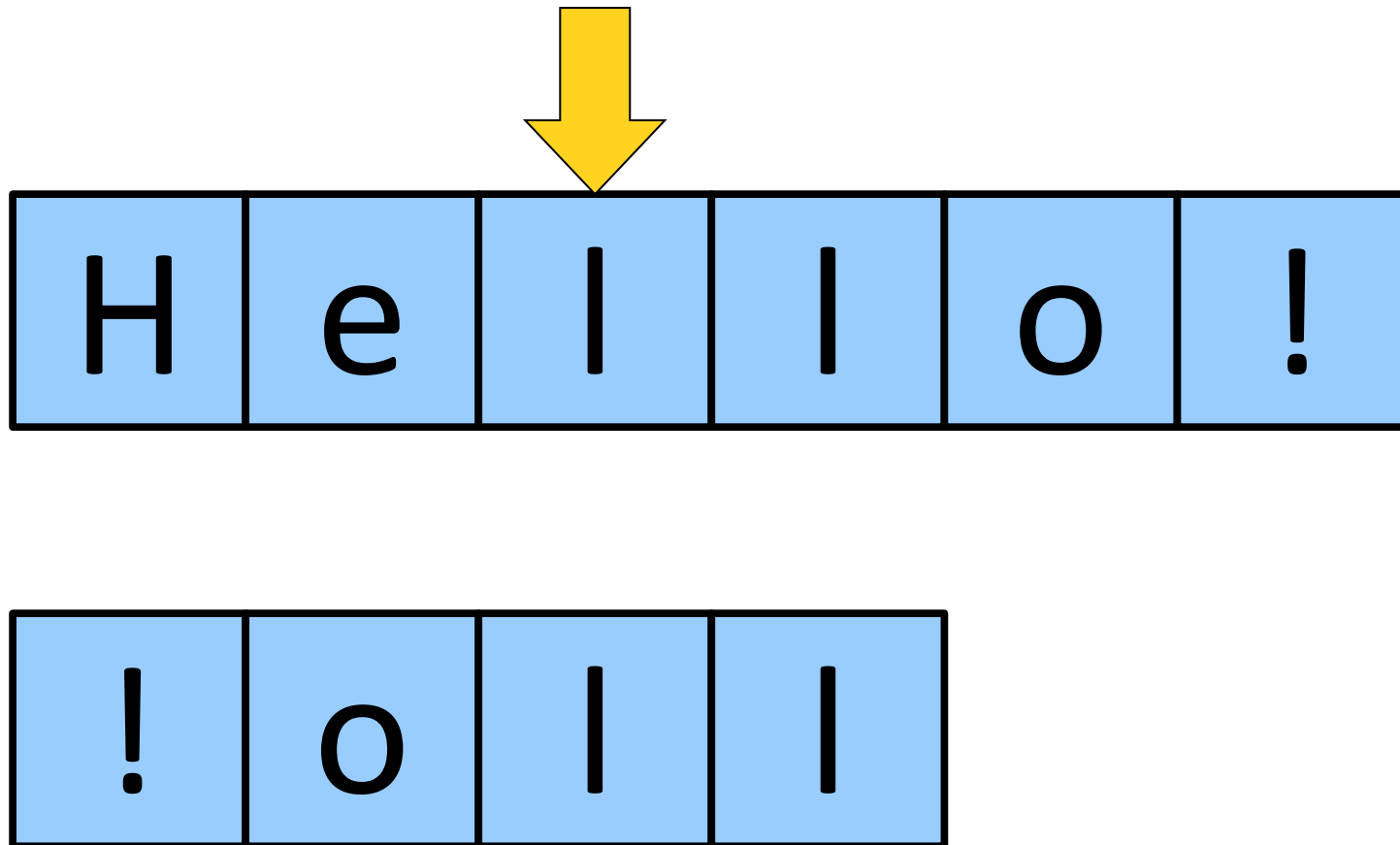
Reversing a String



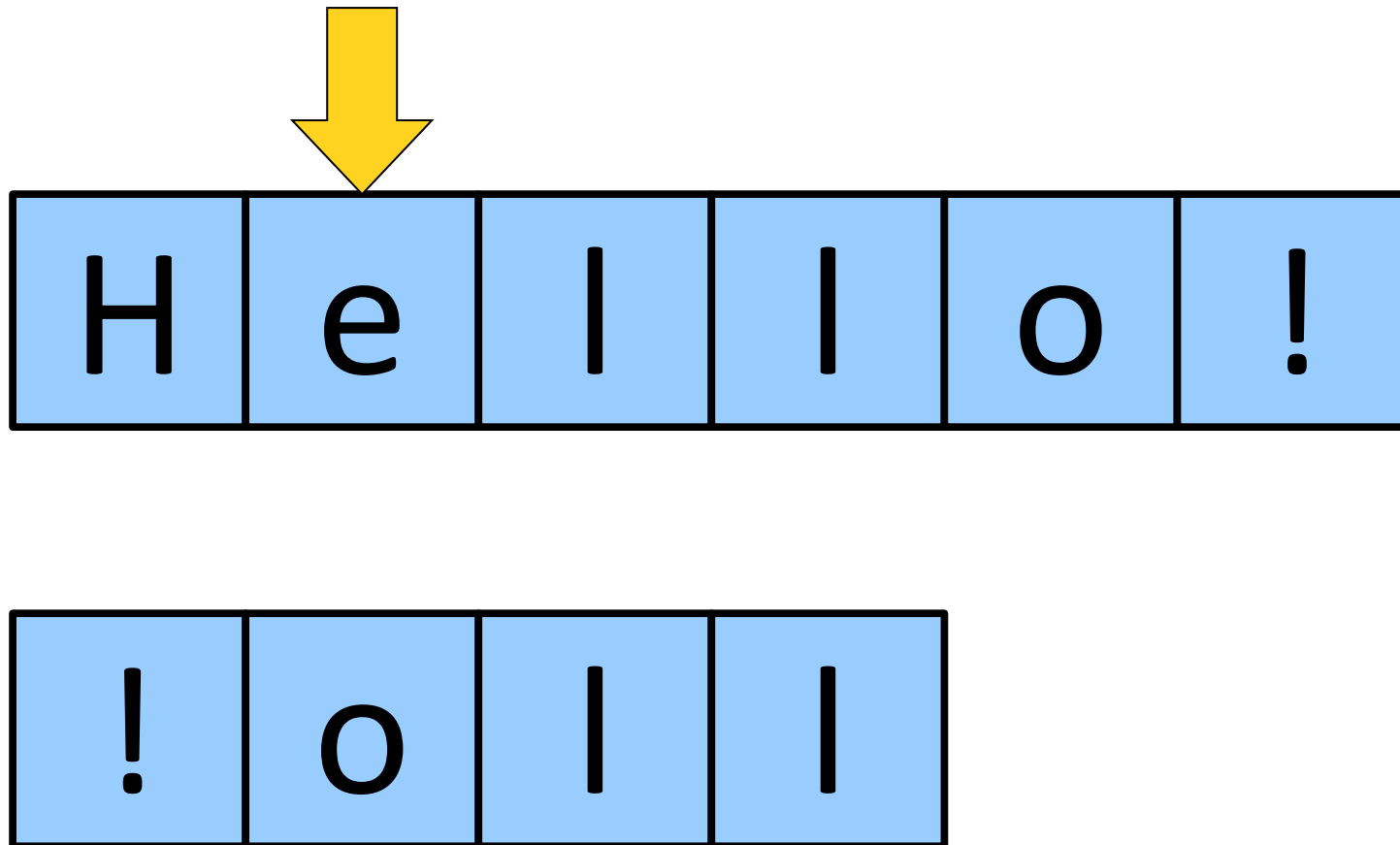
Reversing a String



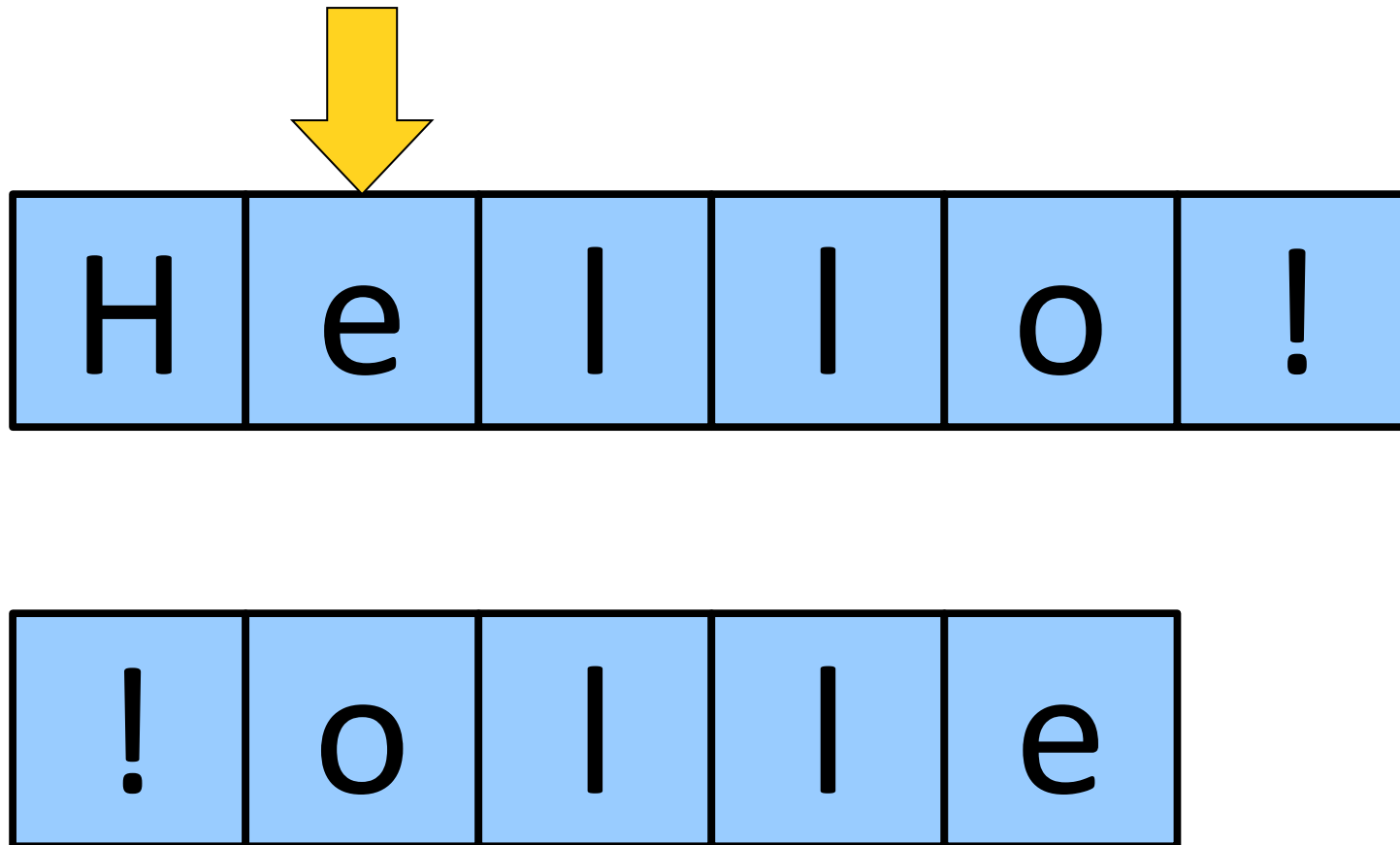
Reversing a String



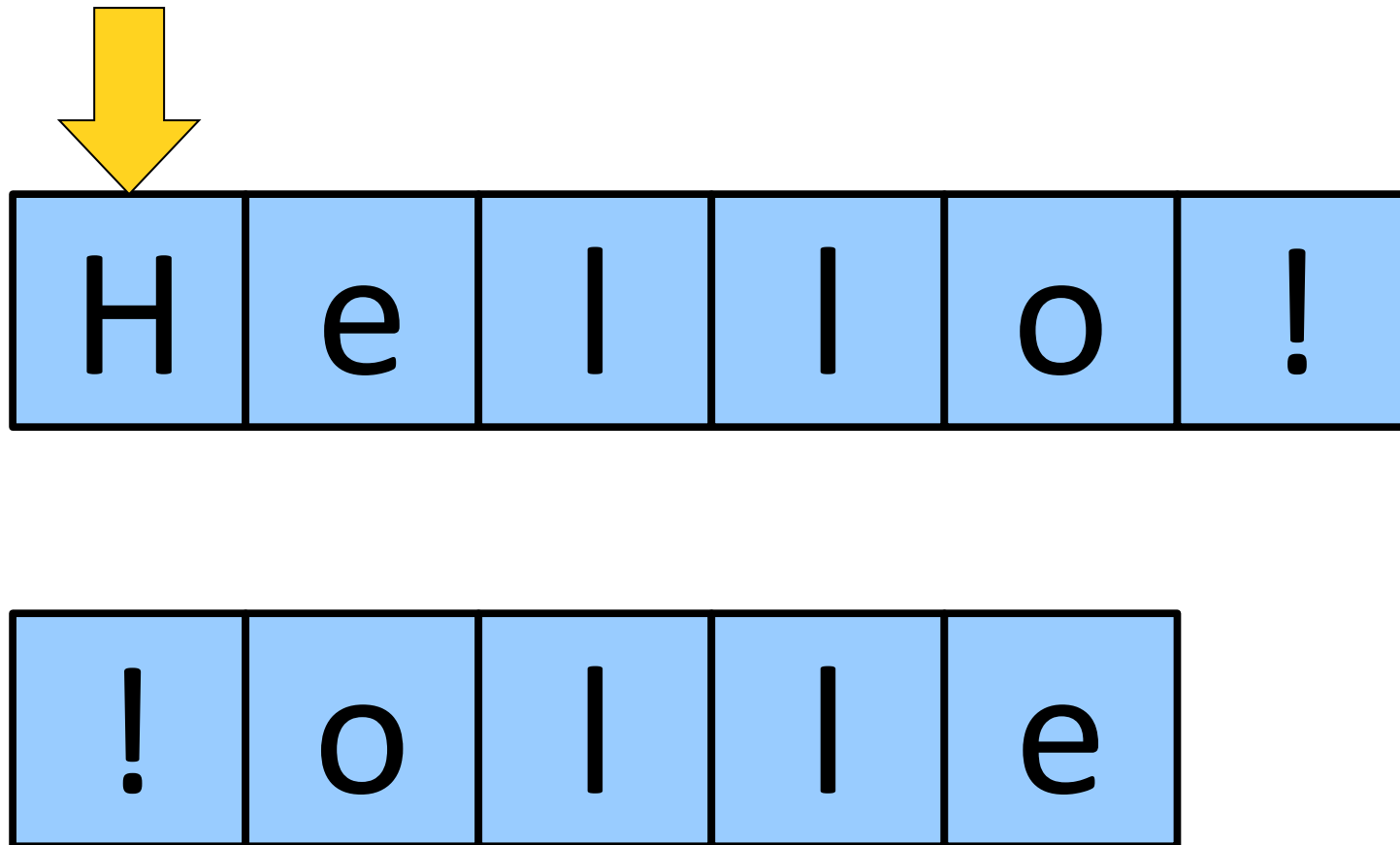
Reversing a String



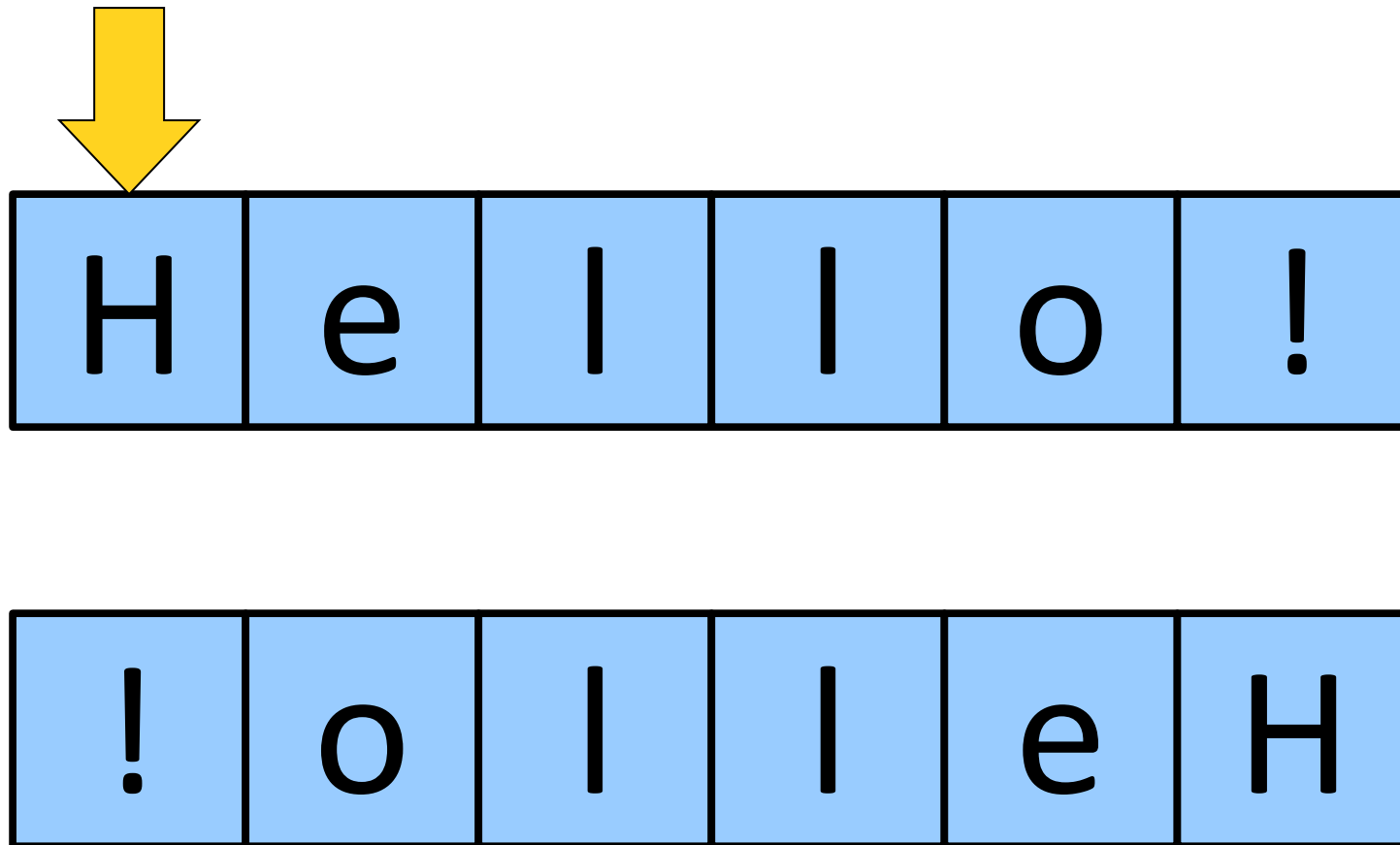
Reversing a String



Reversing a String



Reversing a String



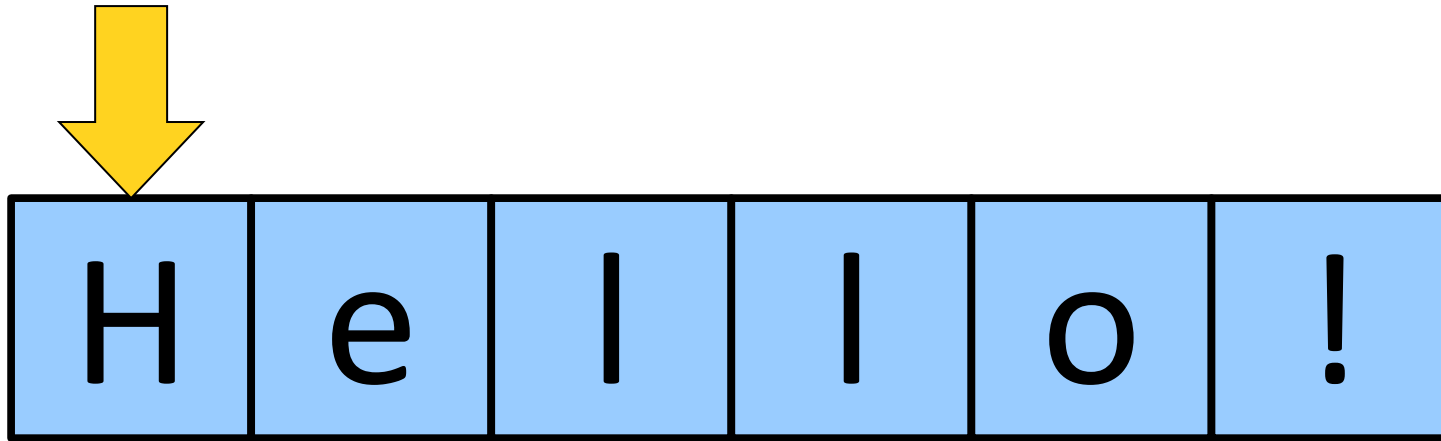
Reversing a String

H	e	l	l	o	!
---	---	---	---	---	---

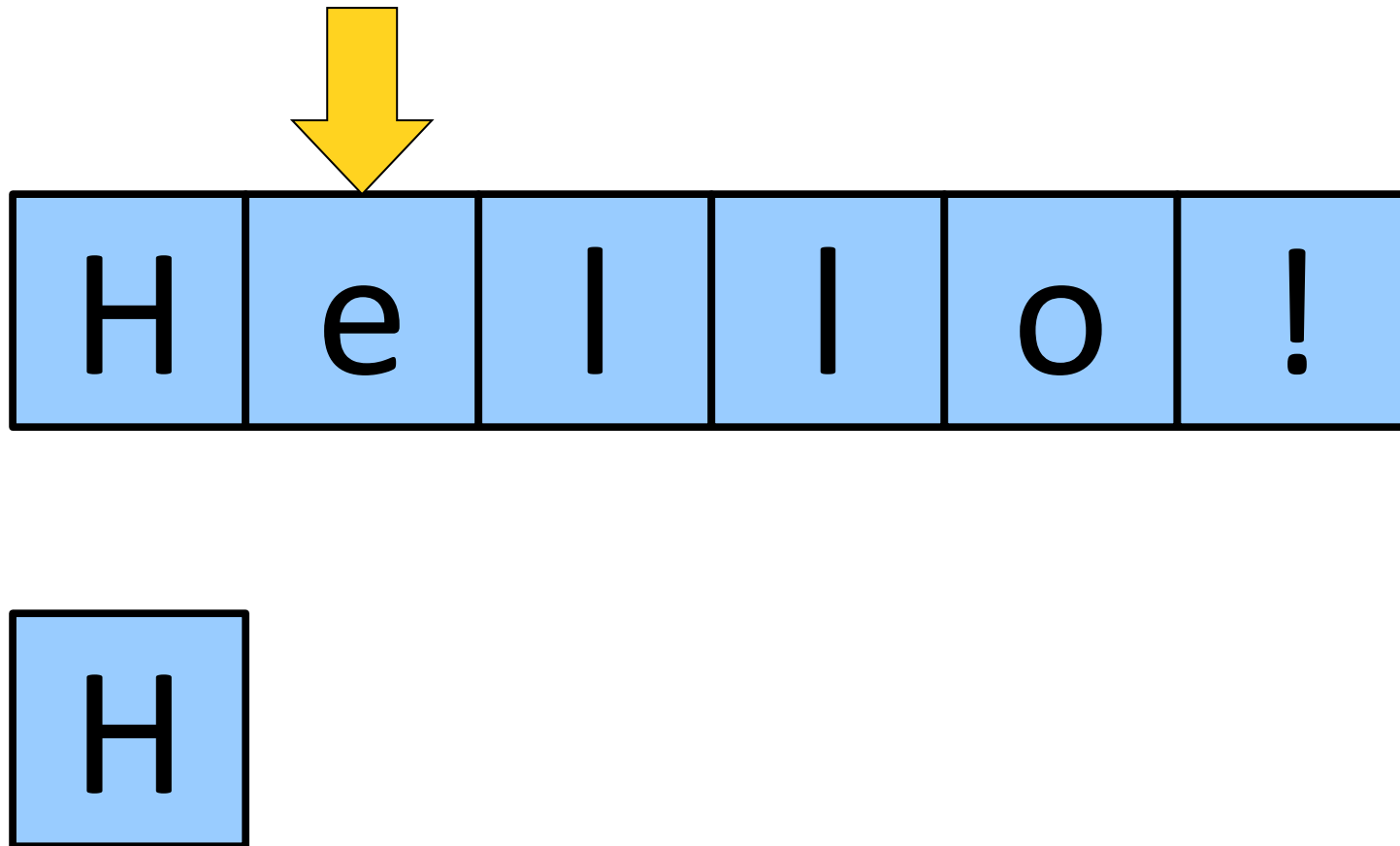
!	o	l	l	e	H
---	---	---	---	---	---



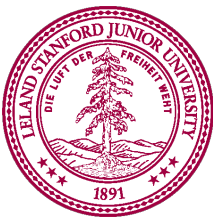
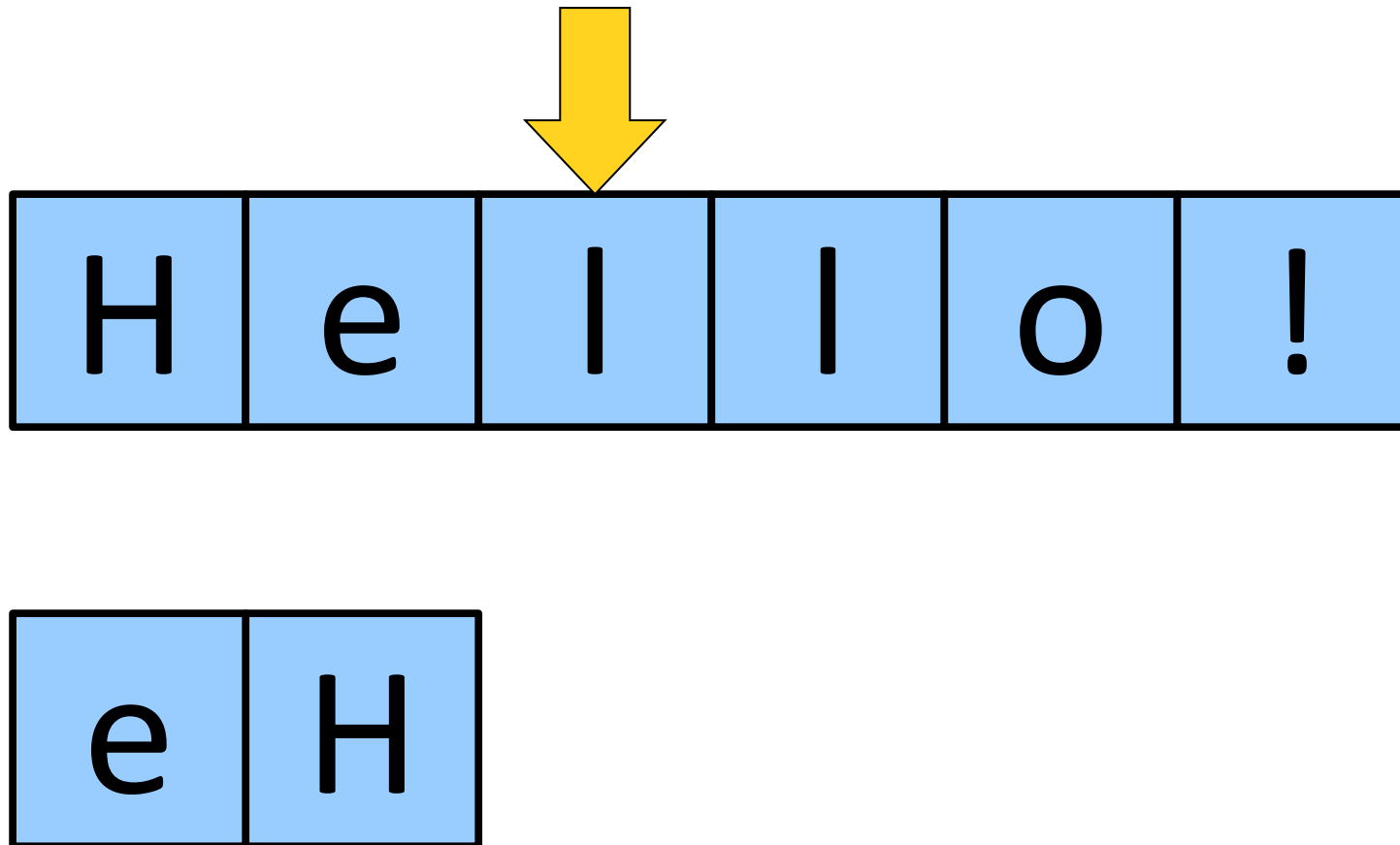
Reversing a String



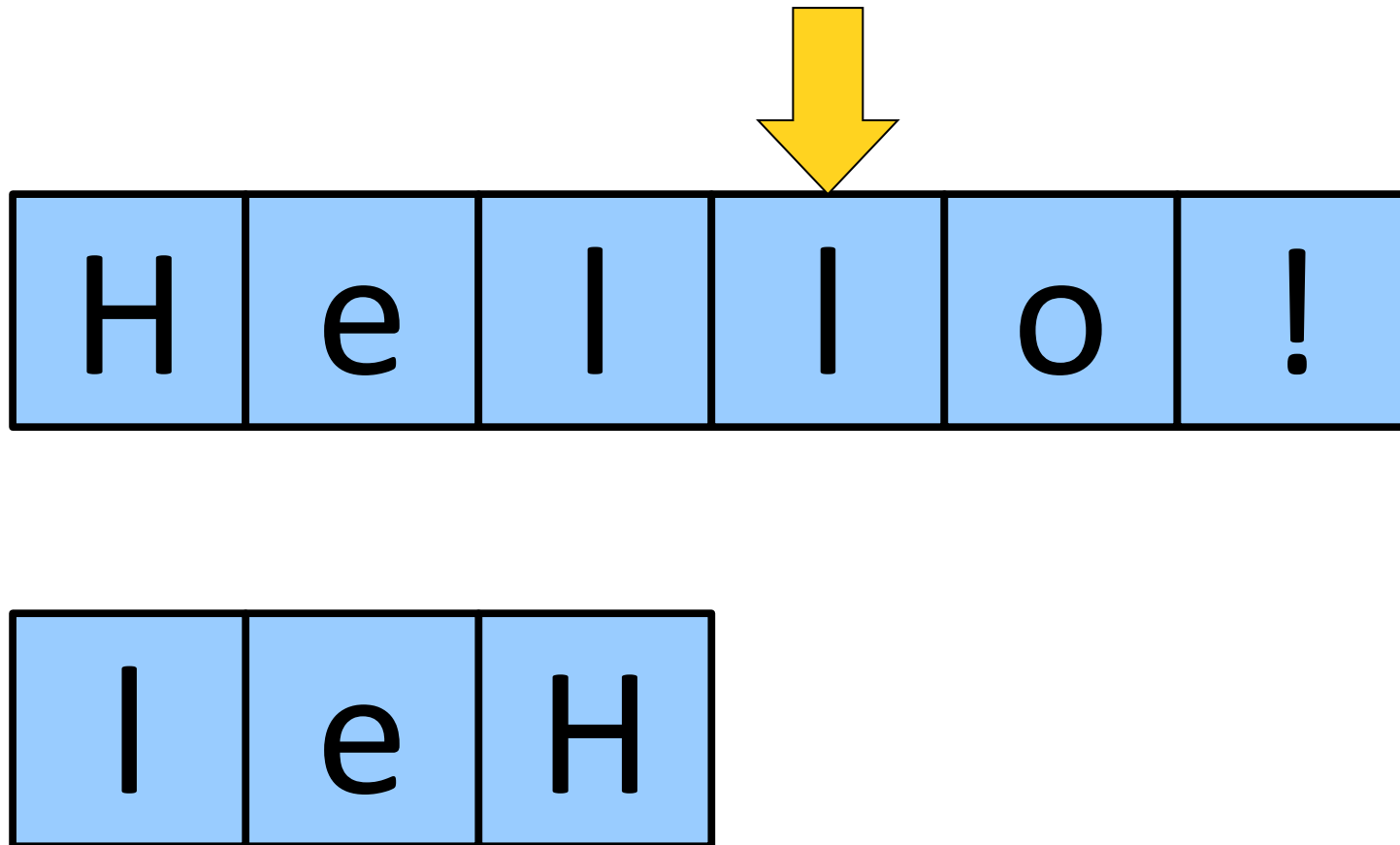
Reversing a String



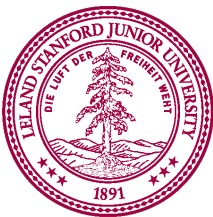
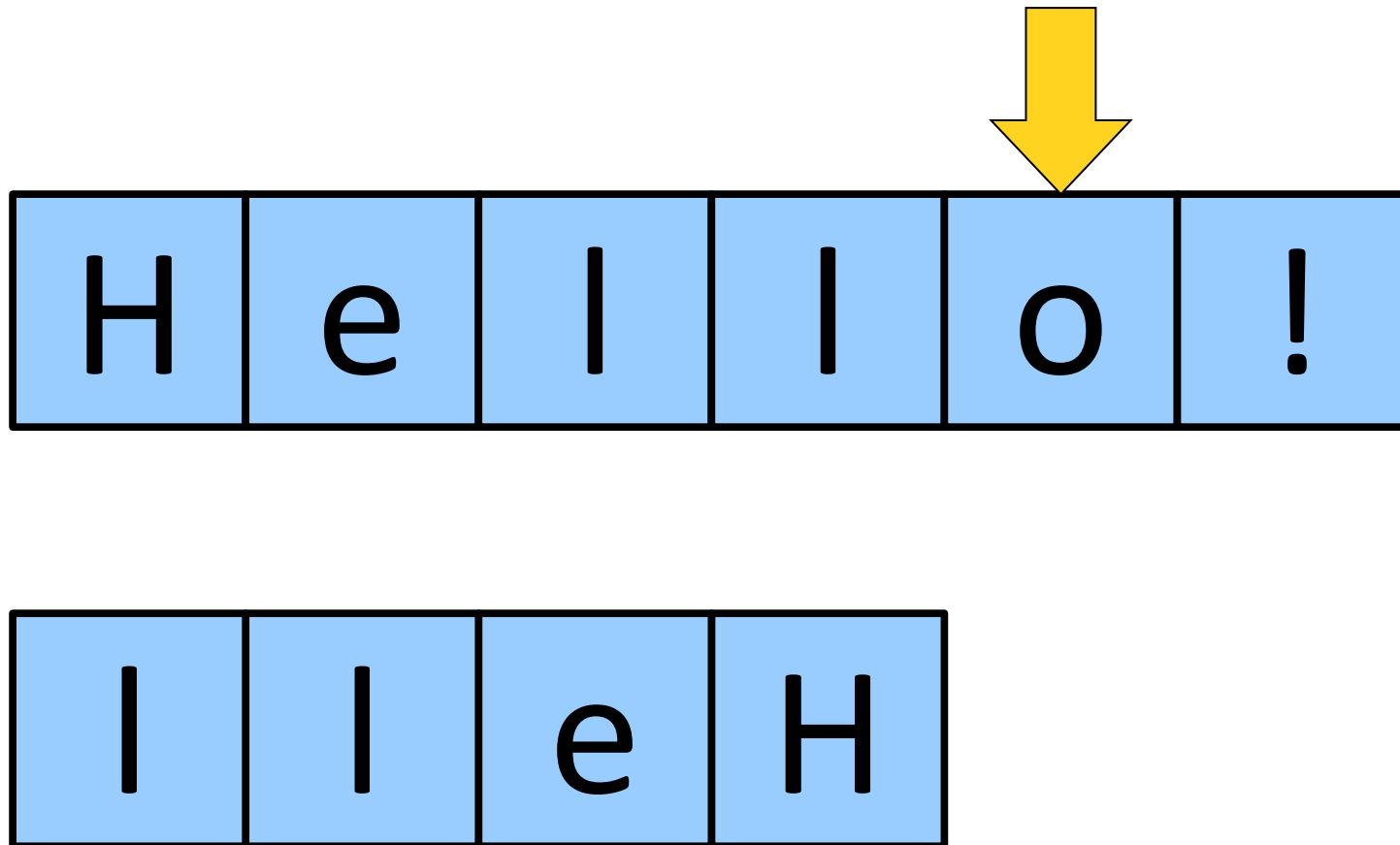
Reversing a String



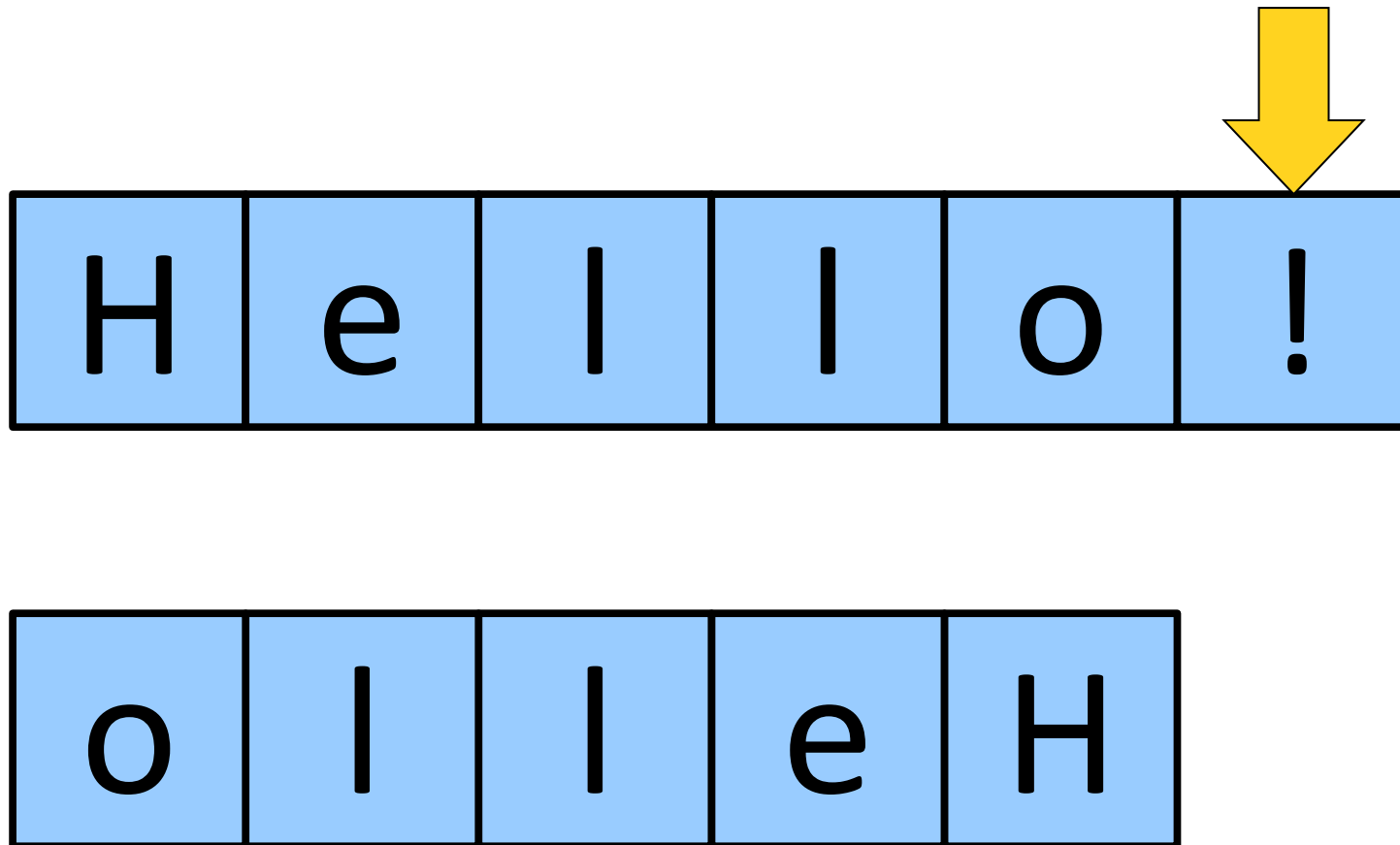
Reversing a String



Reversing a String



Reversing a String



Reversing a String

H	e	l	l	o	!
---	---	---	---	---	---

!	o	l	l	e	H
---	---	---	---	---	---



reverse_string

```
def main():
```

```
    def reverse_string(str):
```

```
        result = ""
```

```
        for i in range(len(str)):
```

```
            result = str[i] + result
```

```
        return result
```

result

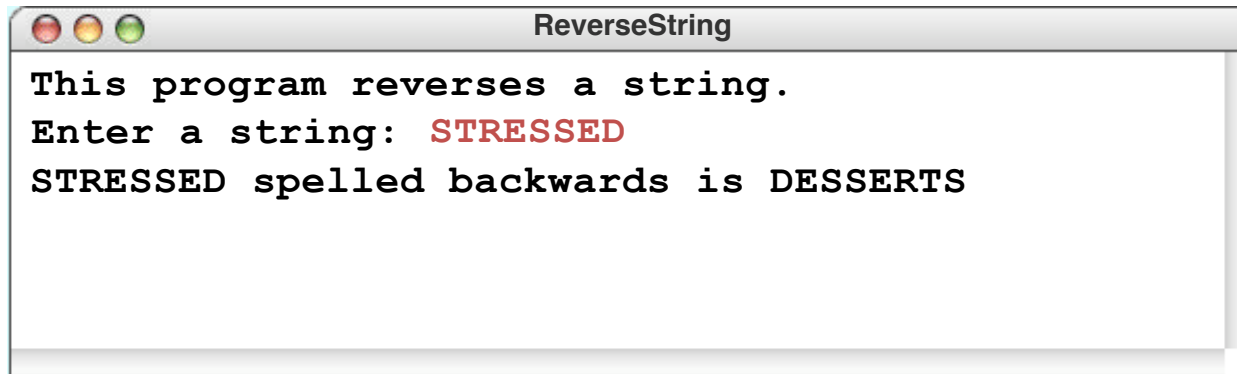
DESSERTS

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.

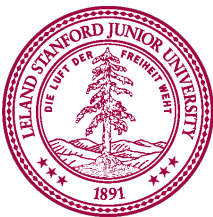
```
def reverse_string_v3(str):  
    '''
```

One cool trick that
will blow your mind

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

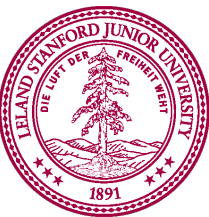
```
'''
```

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



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:

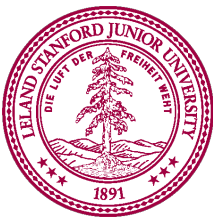
- بلح تعلق تحت قلعة حلب (Dates hang underneath a castle in Halab)
- 여보, 안경 안보여 (Honey, I can't see my glasses)
- कड़क (a loud thunderous sound)
- 上海自來水來自海上 (Shanghai tap water originates from "above" the ocean)

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



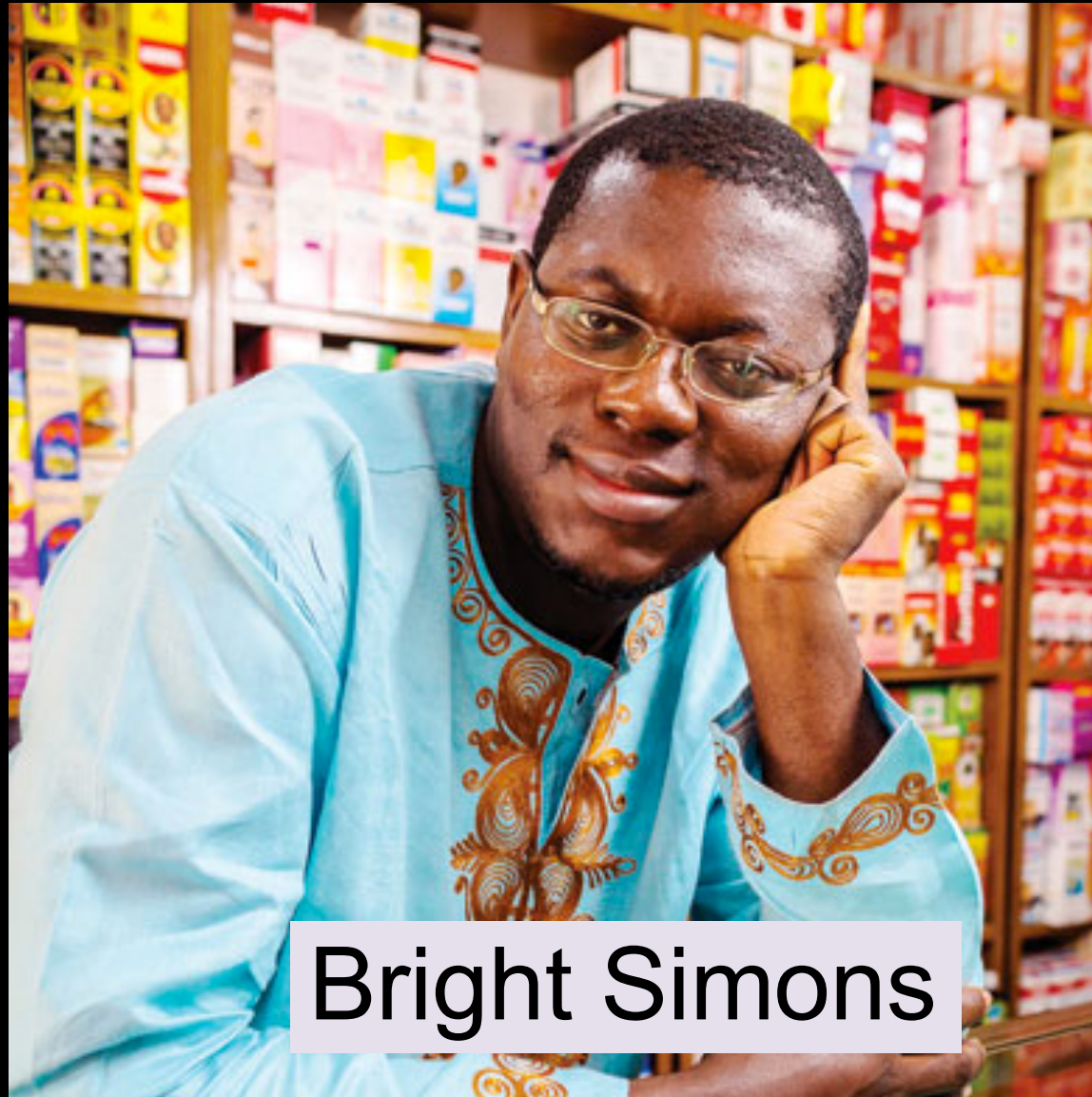
User



You (Distributor)



Can you solve it?



Bright Simons