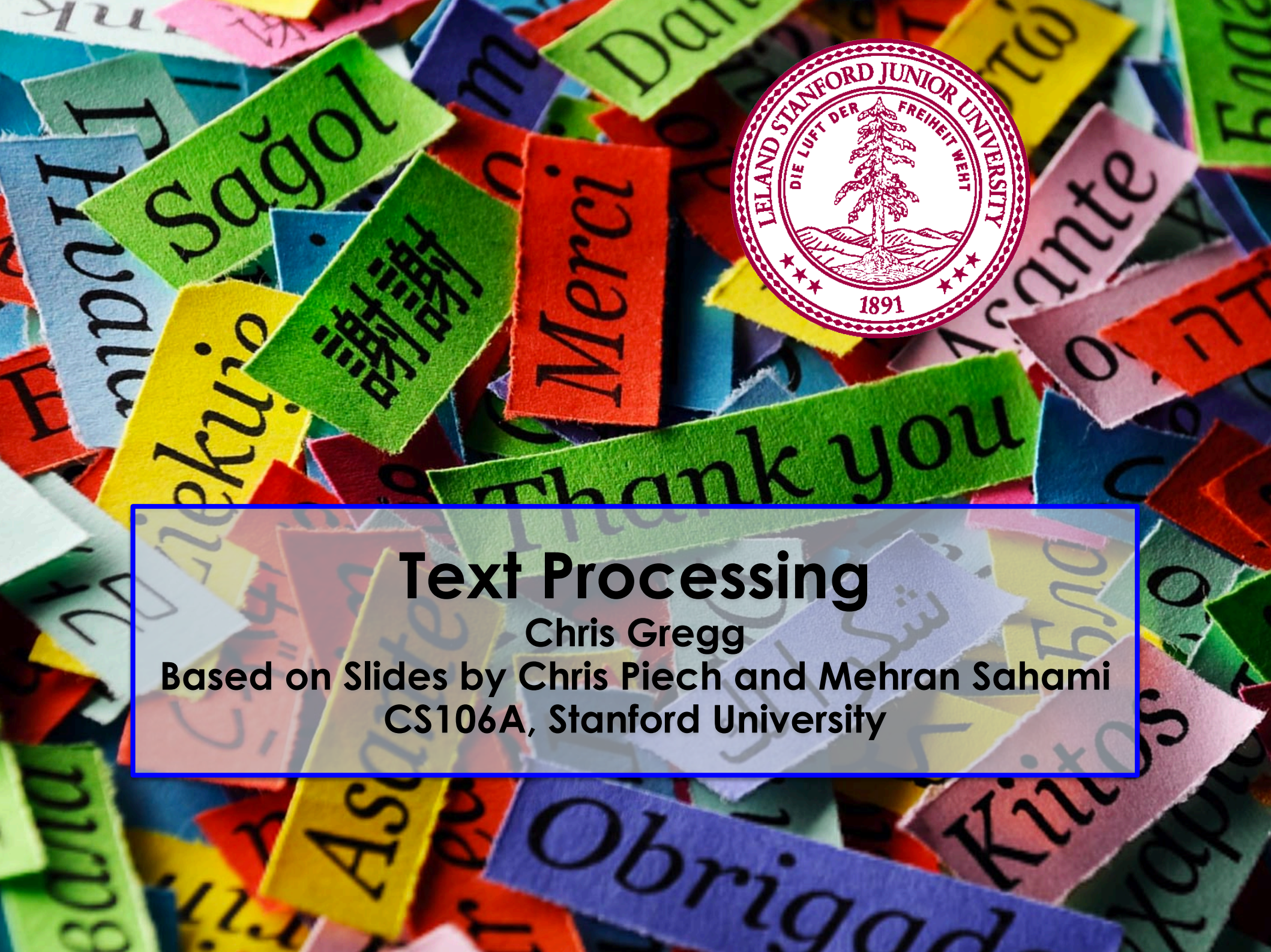
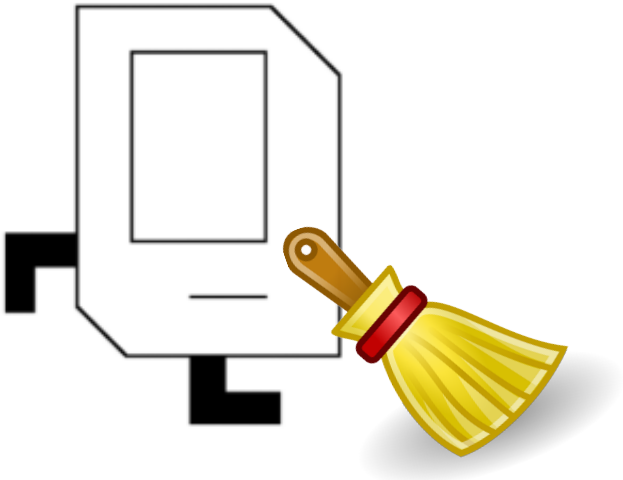


Text Processing

Chris Gregg
Based on Slides by Chris Piech and Mehran Sahami
CS106A, Stanford University

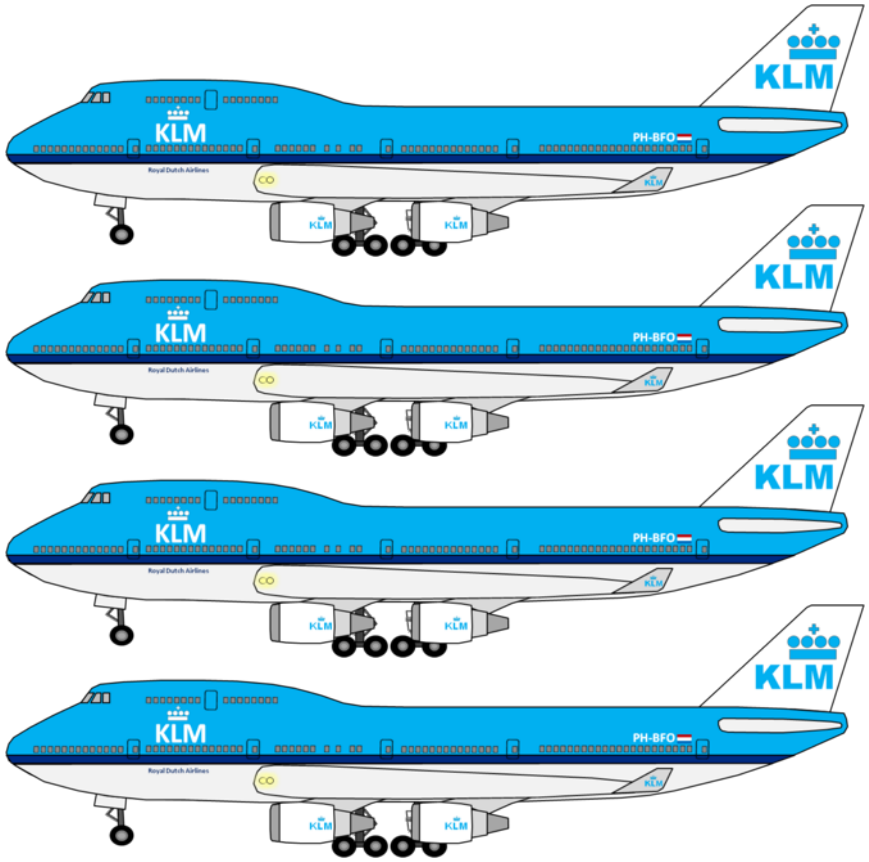
Housekeeping



- Chris AMA today at 11:30am. Come ask me things!
  - https://stanford.zoom.us/j/323735448?pwd=U2VMZlRXWlRxNnQ5MWZIeDRvNi8wdz09

Fake Medicine was a Problem

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



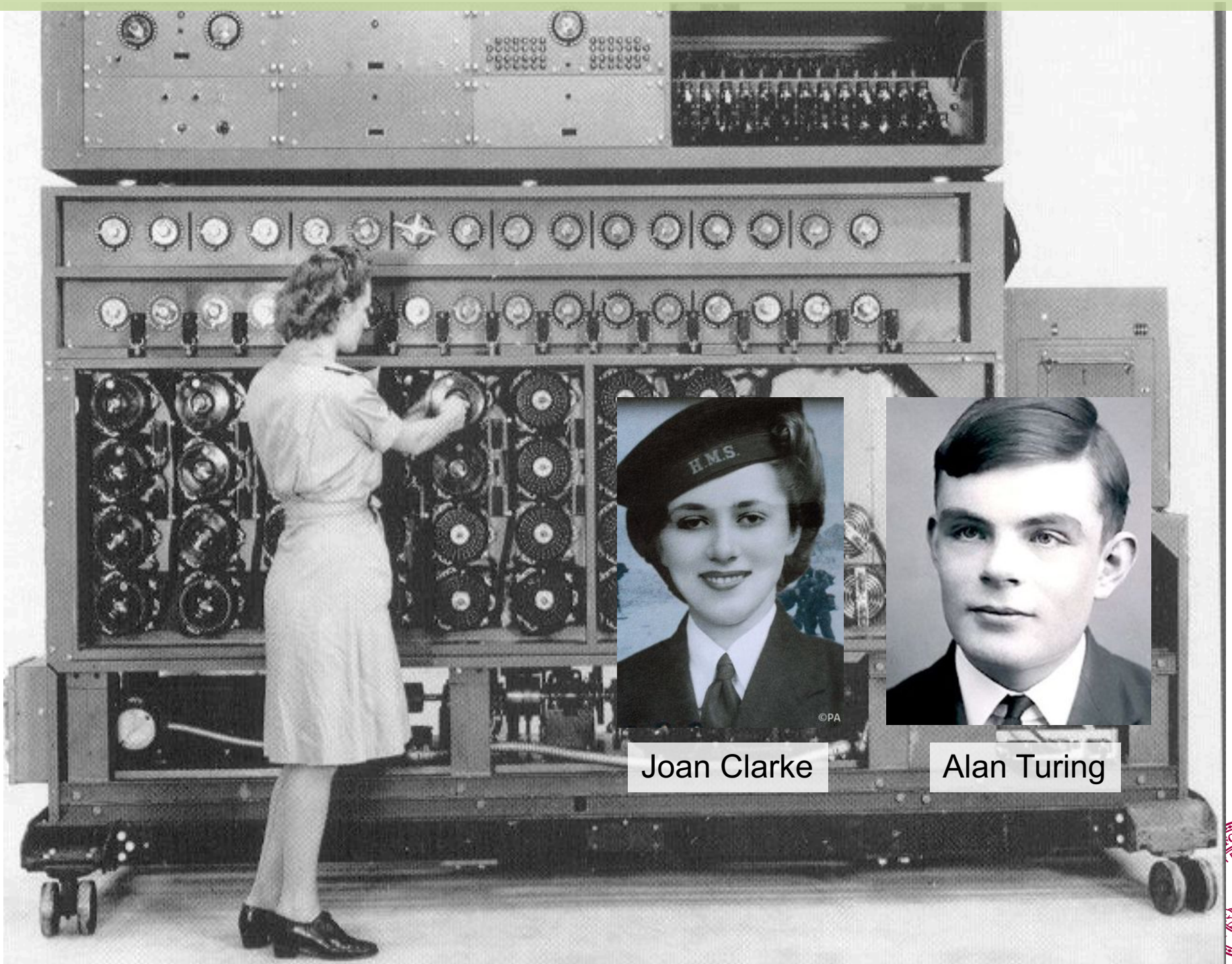
Equivalent of this many crashes per day

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

Learning Goals

1. Write string algorithms that loops over each character

Text Problem: Decryption



Text Problem: Translation

The spirit is willing but the flesh is weak.

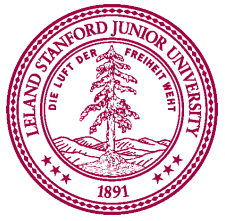
↓

(Russian)

↓

The vodka is good but the meat is rotten.

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



Text Problem: DNA Analysis

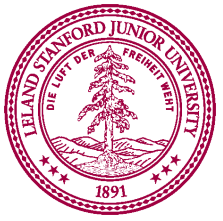
How is text
represented?

The variable type **`string`**

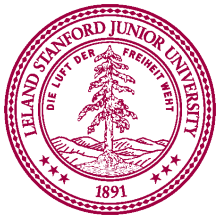
Text is stored using the variable type **`string`**.

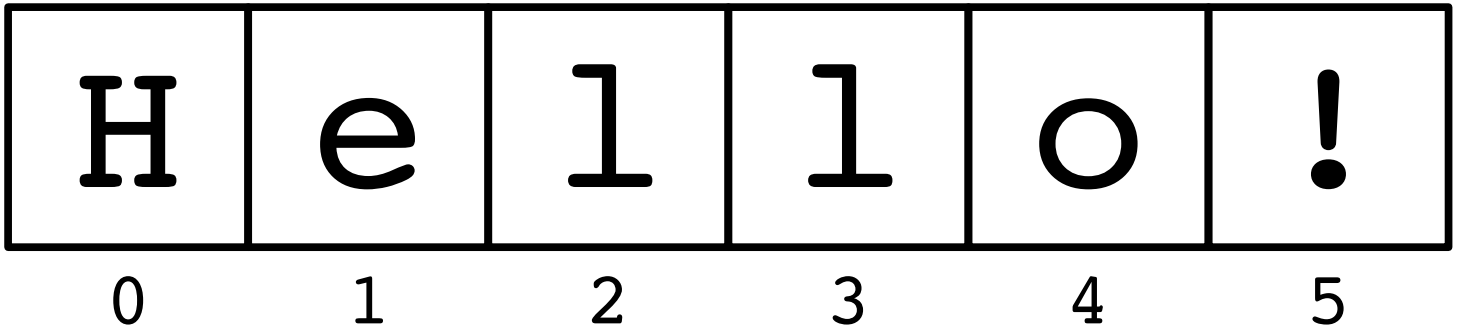
A **`string`** is a sequence of characters.

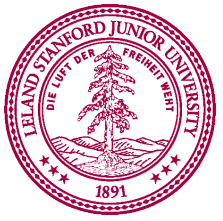
```
def main():
    text = "hello!"
    print(text)
```



H e l l o !

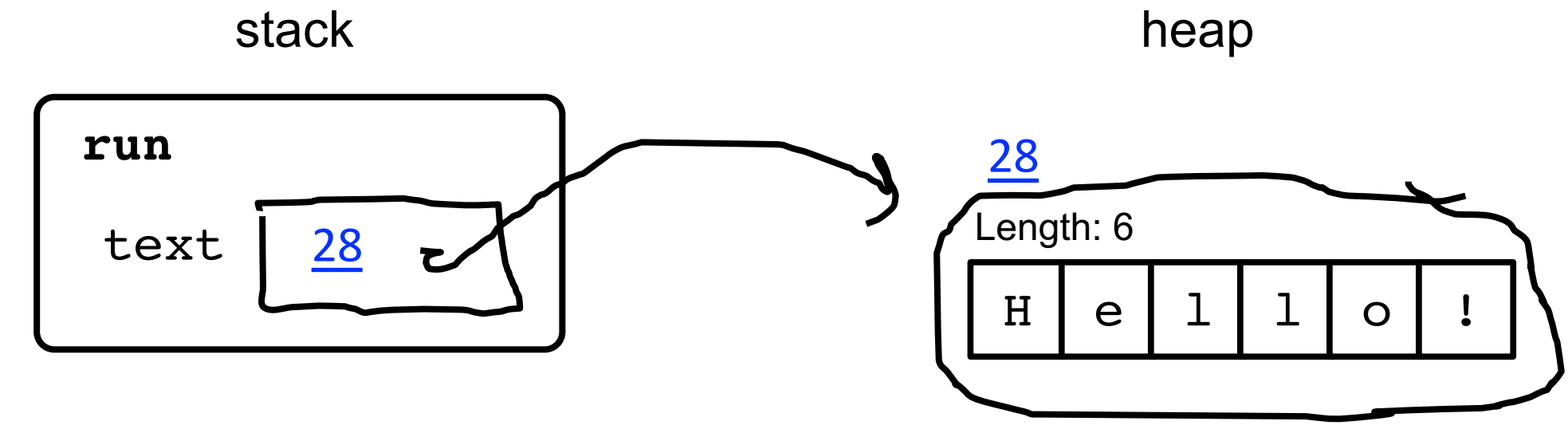

LELAND STANFORD JUNIOR UNIVERSITY
DIE LUFT DER FREIHEIT WEHT
1891


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

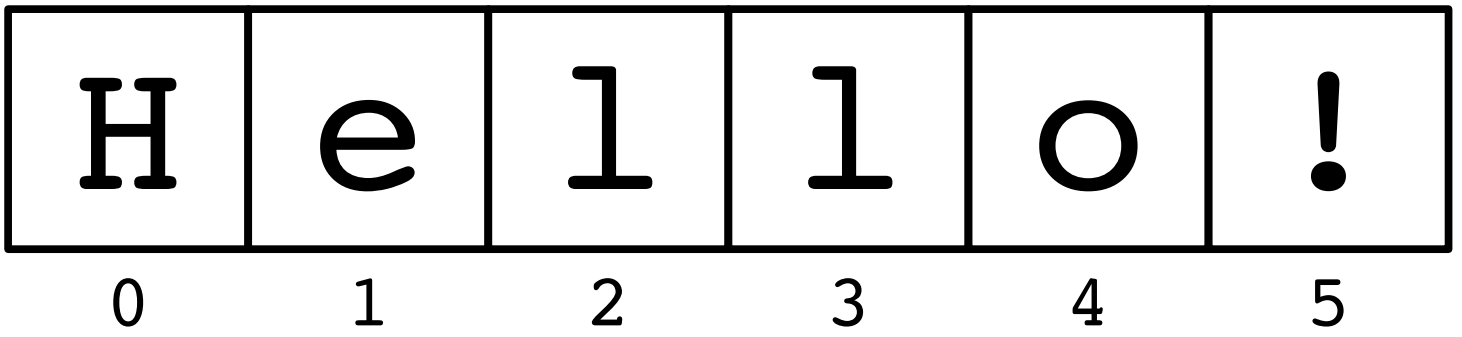

LELAND STANFORD JUNIOR UNIVERSITY
DIE LUFT DER FREIHEIT WEHT
1891

How it is actually stored

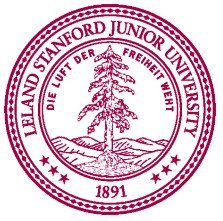
```
def main():
    text = "hello!"
```



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

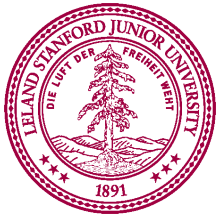


***text*[*index*]**



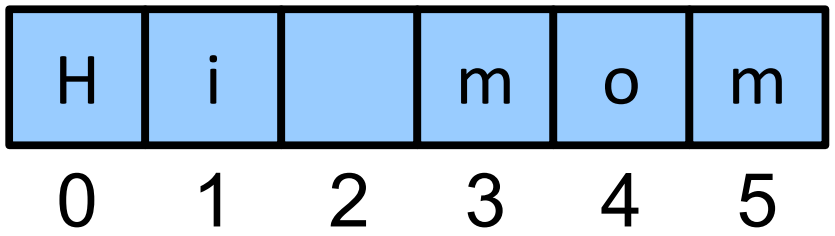
All characters in a string have an index.

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

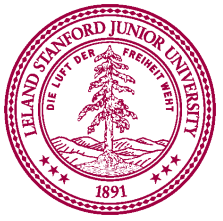


String Functions

- 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.

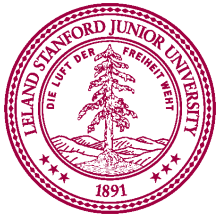


(length 6)

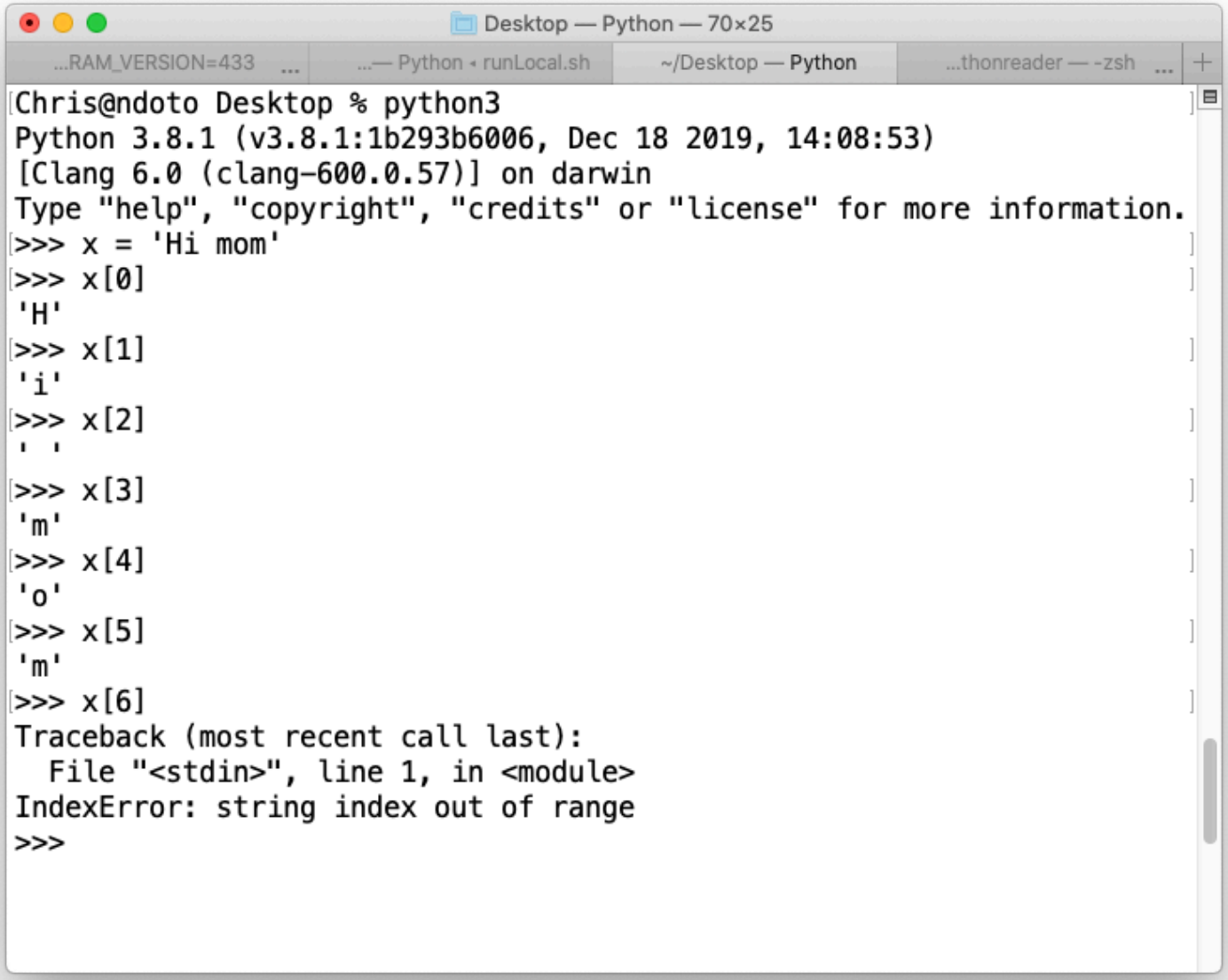


A string is indexed just like a list! Slices work too.

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

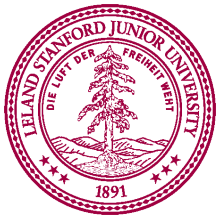


Lets play!


Desktop — Python — 70×25
...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

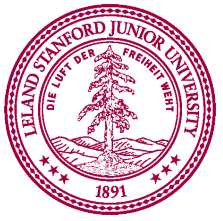
```
def main():
    example = "Hi mom"

    # 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)
```

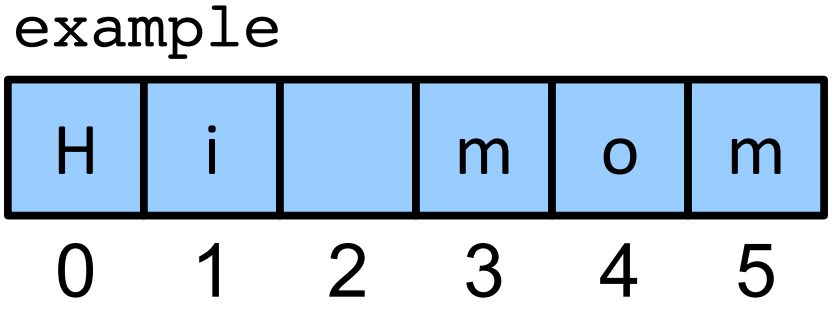
Console

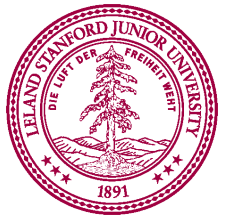


String Functions

```
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



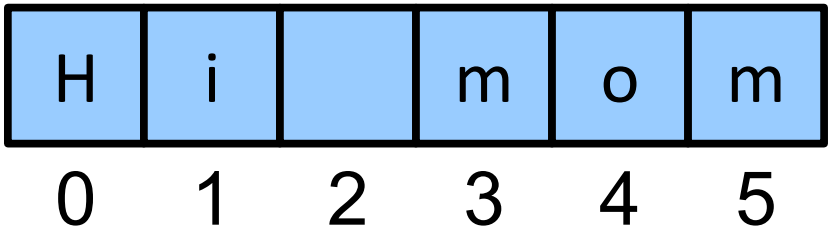


String Functions

```
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



length

6

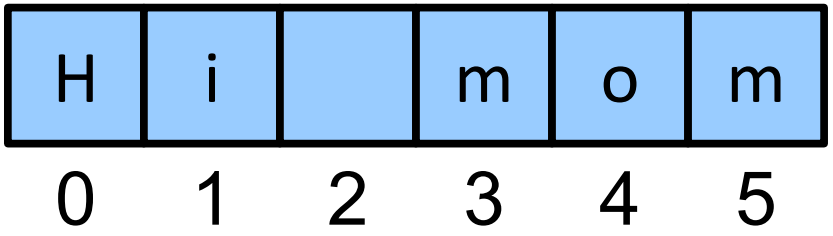
String Functions

```
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

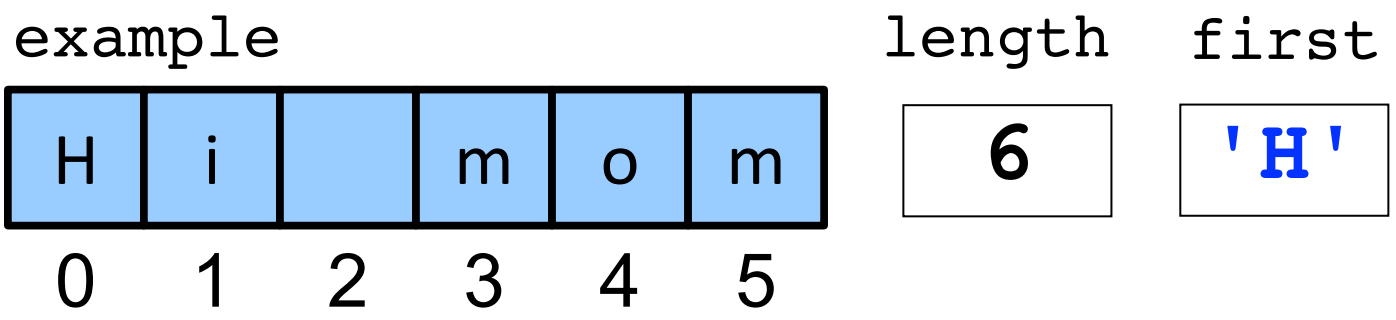
6

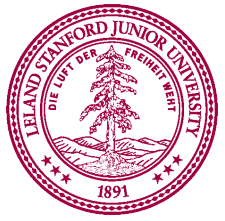
String Functions

```
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
```



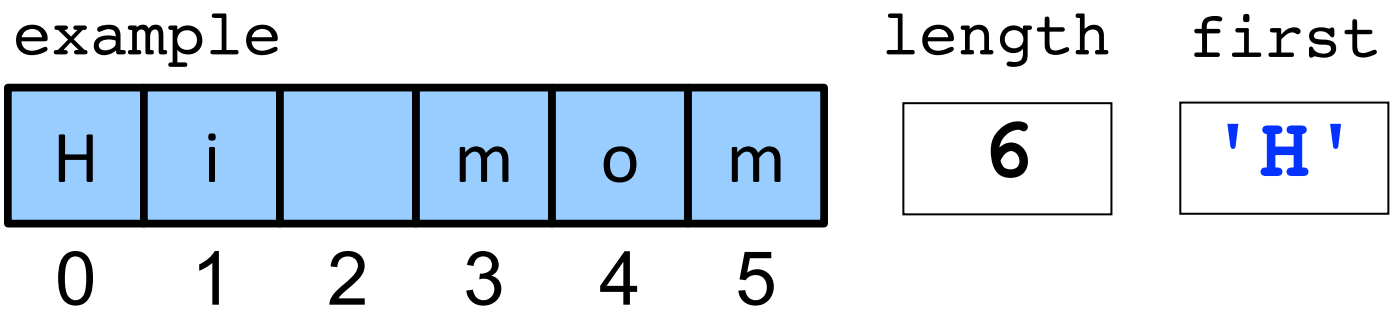


String Functions

```
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
```

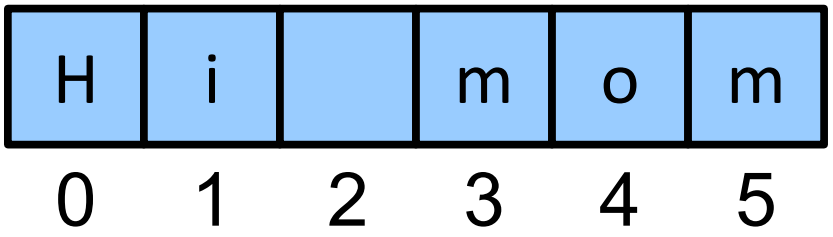


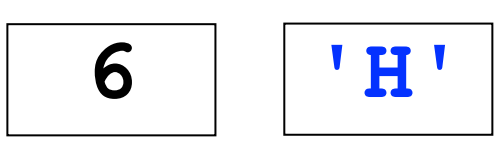
String Functions

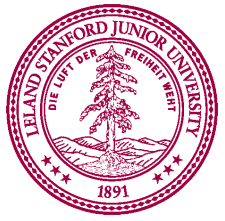
```
def main():
    example = "Hi mom"

    # 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
```





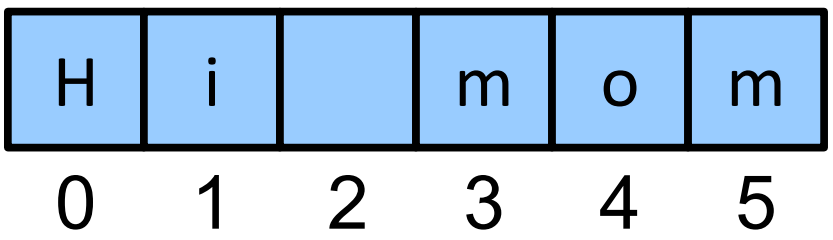


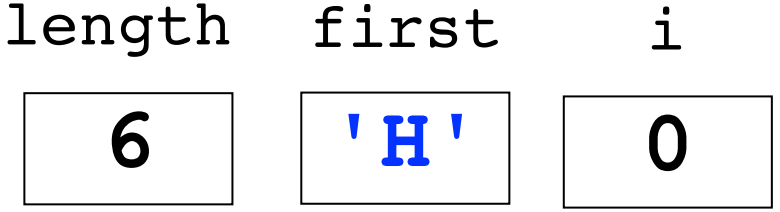
String Functions

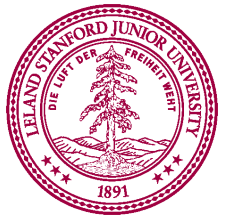
```
def main():
    example = "Hi mom"

    # 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
```





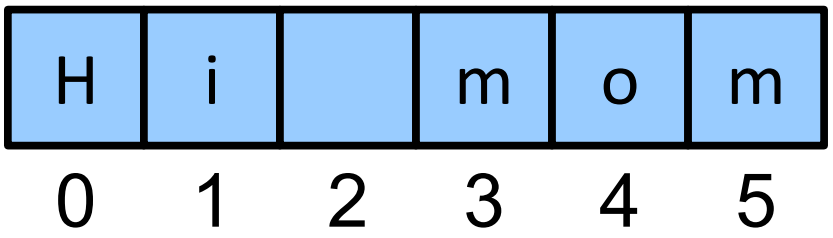


String Functions

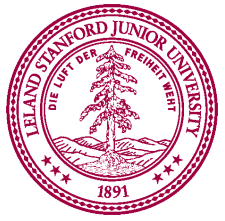
```
def main():
    example = "Hi mom"

    # 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
```



length	first	i
6	'H'	0



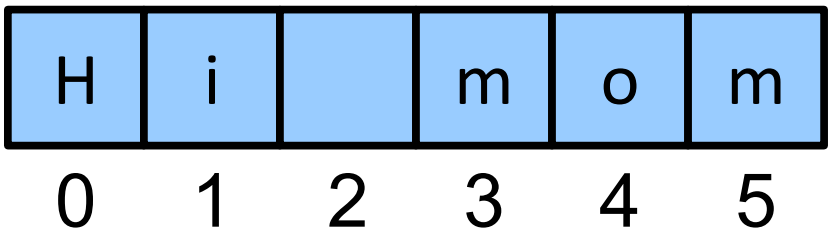
String Functions

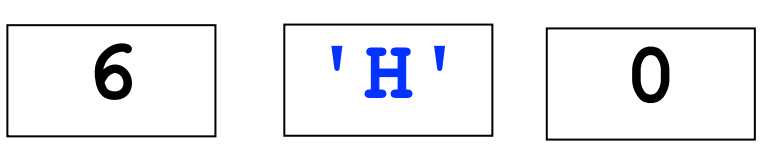
```
def main():
    example = "Hi mom"

    # 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)
```

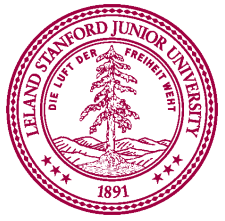
[0, 1, 2, 3, 4, 5]

Console

```
6
H
```





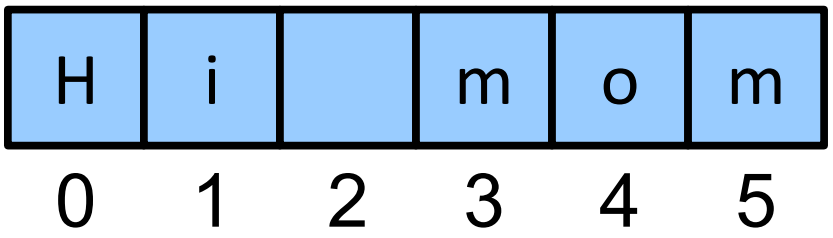


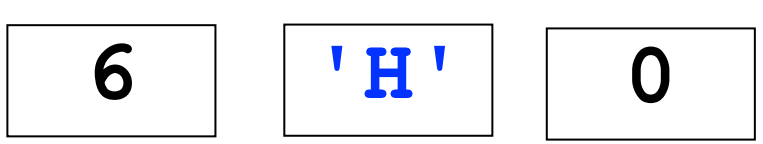
String Functions

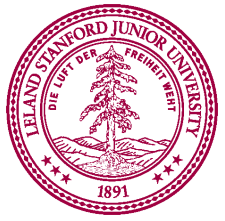
```
def main():
    example = "Hi mom"

    # 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
```





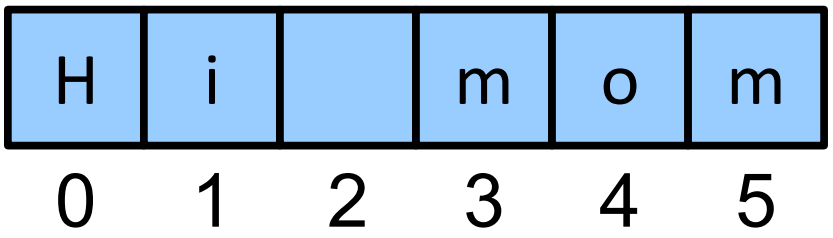


String Functions

```
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
```



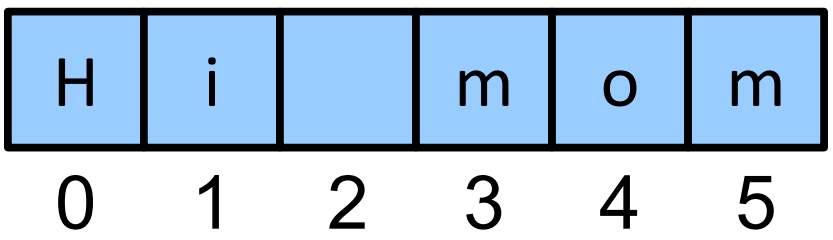
length	first	i	ch
6	'H'	0	'H'

String Functions

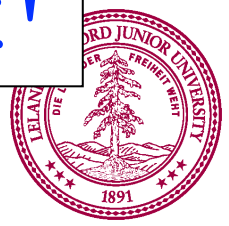
```
def main():
    example = "Hi mom"

    # 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
```



length	first	i	ch
6	'H'	0	'H'

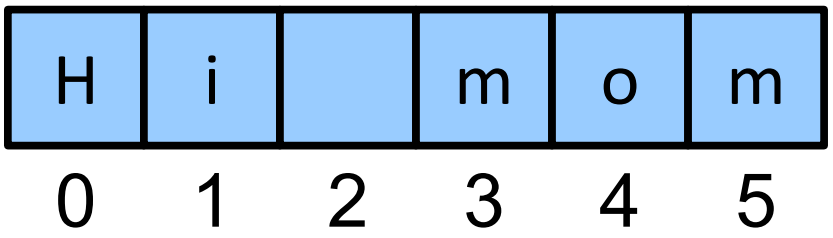


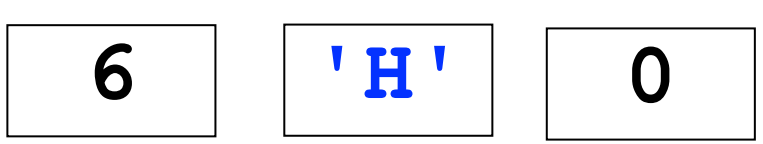
String Functions

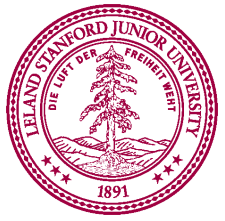
```
def main():
    example = "Hi mom"

    # 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
```





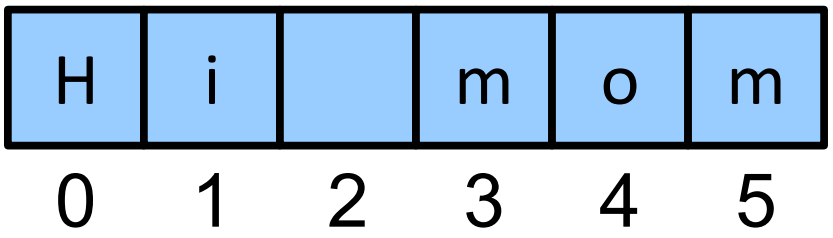


String Functions

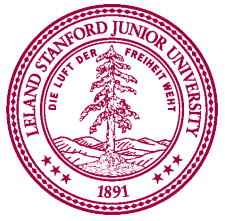
```
def main():
    example = "Hi mom"

    # 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
```



length	first	i
6	'H'	1

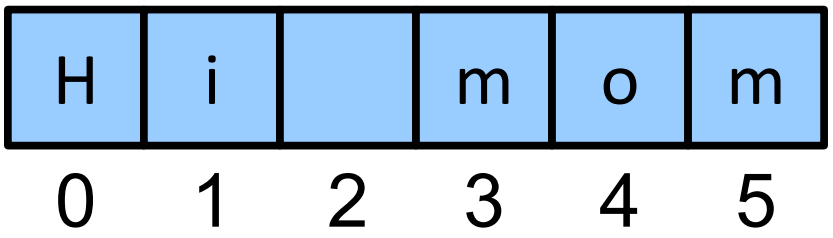


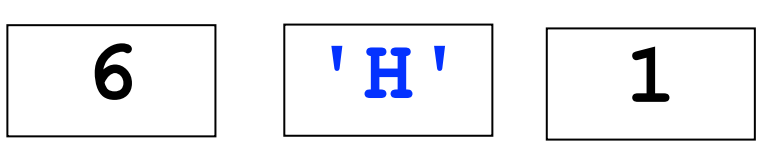
String Functions

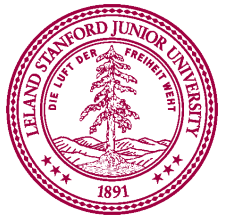
```
def main():
    example = "Hi mom"

    # 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
```





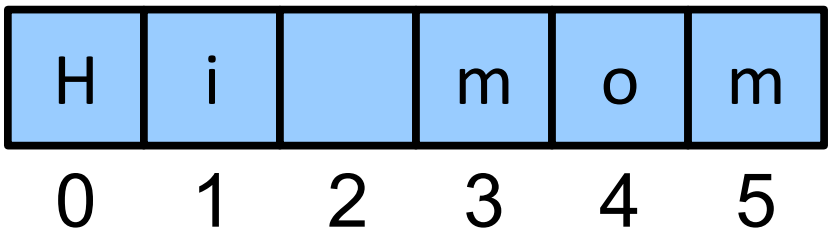


String Functions

```
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
```



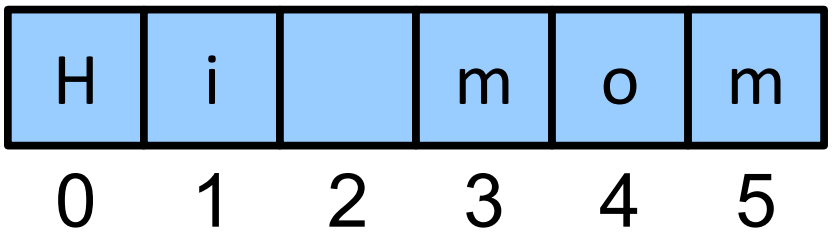
length	first	i	ch
6	'H'	1	'i'

String Functions

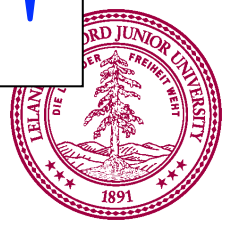
```
def main():
    example = "Hi mom"

    # 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
```



length	first	i	ch
6	'H'	1	'i'

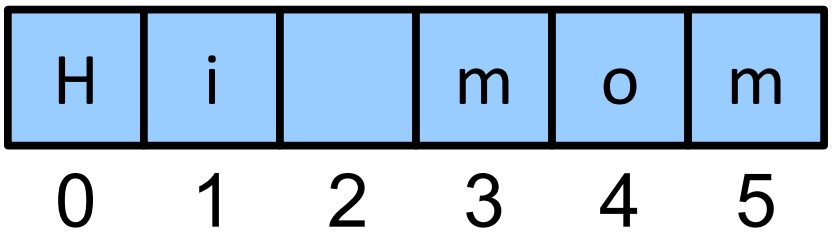


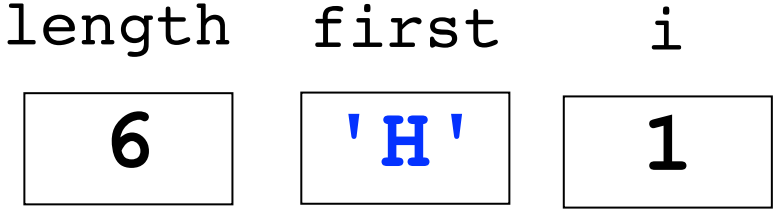
String Functions

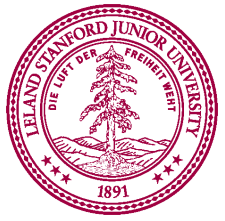
```
def main():
    example = "Hi mom"

    # 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
```





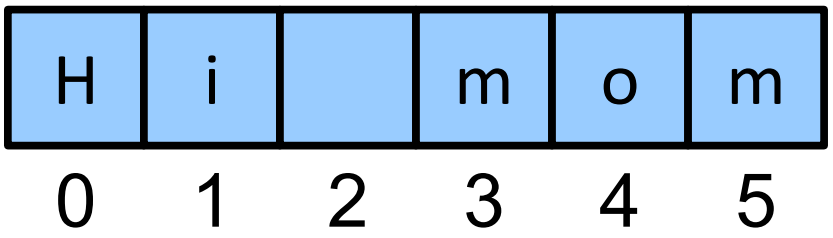


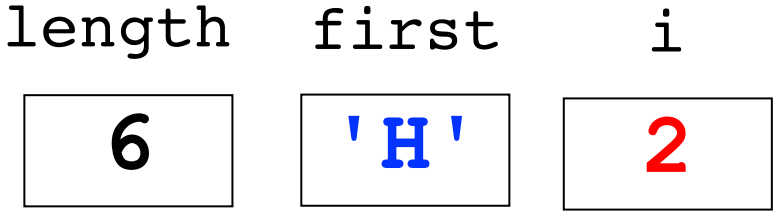
String Functions

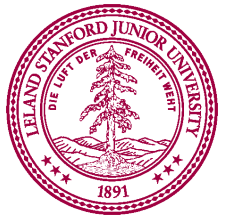
```
def main():
    example = "Hi mom"

    # 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
```





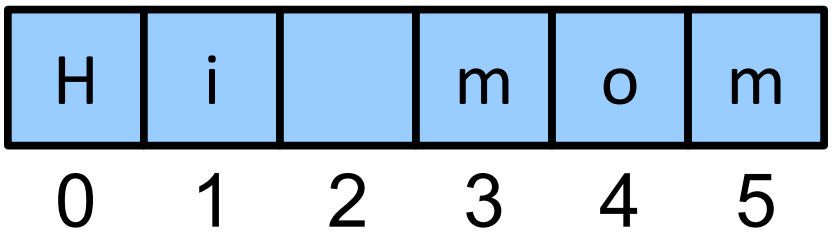


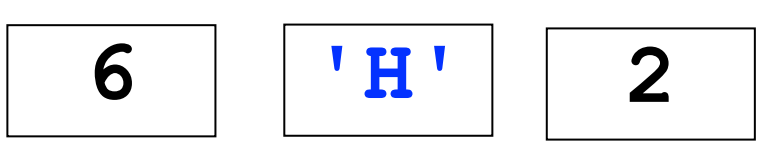
String Functions

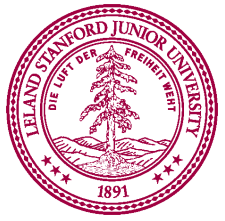
```
def main():
    example = "Hi mom"

    # 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
```





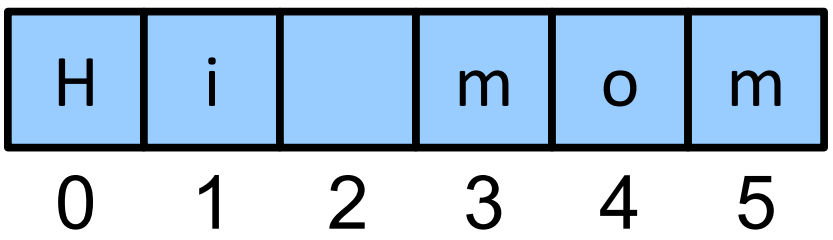


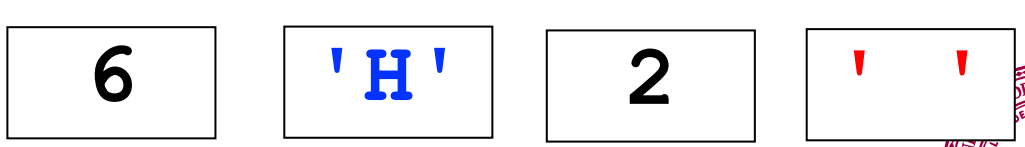
String Functions

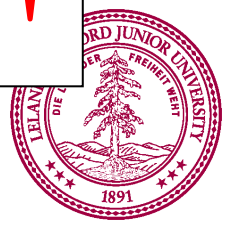
```
def main():
    example = "Hi mom"

    # 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
```





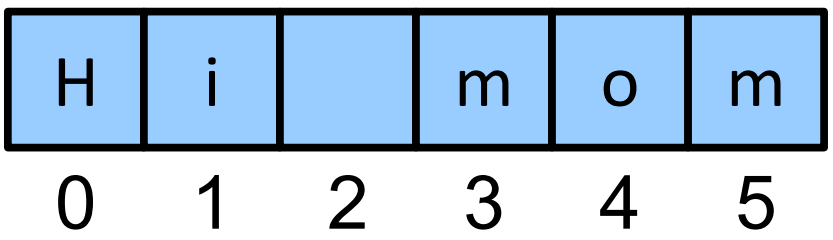


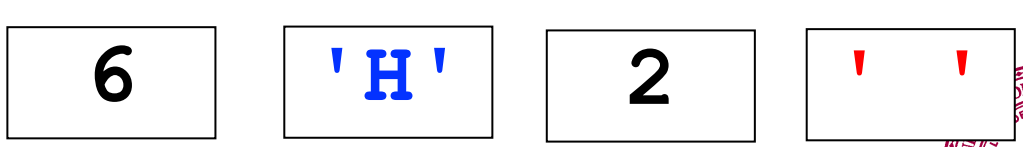
String Functions

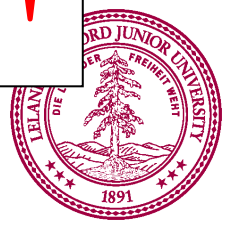
```
def main():
    example = "Hi mom"

    # 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
```





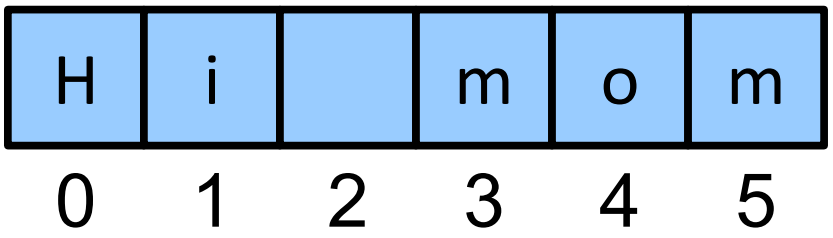


String Functions

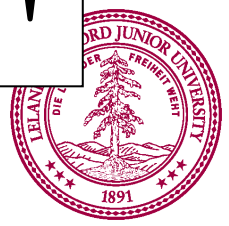
```
def main():
    example = "Hi mom"

    # 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
```



length	first	i	ch
6	'H'	2	' '

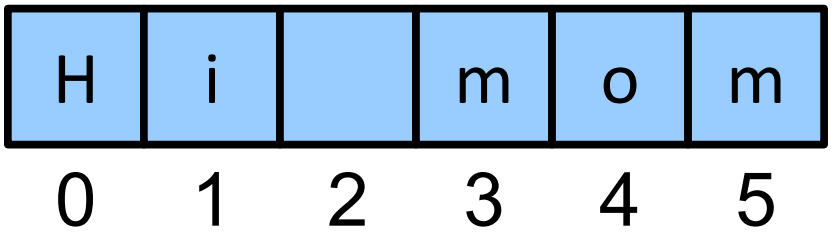


String Functions

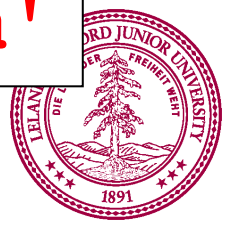
```
def main():
    example = "Hi mom"

    # 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

m
o
m
```



length	first	i	ch
6	'H'	5	'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

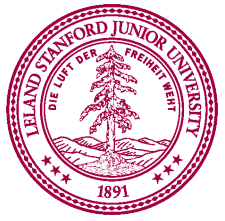
```
x = 'this is a Test '
```

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      x.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

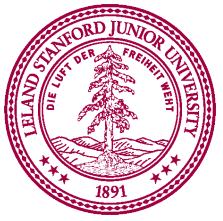
How are characters represented?

Single charaters

- Some examples:

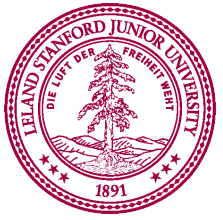
```
letter_A   = 'A'
plus       = '+'
zero       = '0'
space      = ' '
korean_ch  = '보'
new_line   = '\n'  # special
tab        = '\t'  # special
backslash  = '\\'  # special
backslash  = '\''  # special
emoji      = '😀'
first      = text[0]
```

\n newline char
\t tab char
\\ backlash char
\' single quote
\" double quote



Advanced course

Chars are just a giant enumeration!

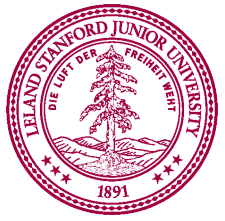


ASCII

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



Using portions of slides by Eric Roberts

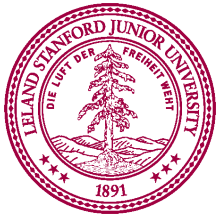
Unicode (bigger ASCII)

3200 **Enclosed CJK Letters and Months** **32FF**

	320	321	322	323	324	325	326	327	328	329	32A	32B	32C	32D	32E	32F
0	㈀ 3200	㈁ 3210	㈂ 3220	㈃ 3230	㈄ 3240	PTE 3250	㉠ 3260	㉡ 3270	㉢ 3280	㉣ 3290	㉤ 3A0	㉥ 32B0	1月 32C0	㋀ 32D0	㋐ 32E0	㋠ 32F0
1	㈁ 3201	㈂ 3211	㈃ 3221	㈄ 3231	㈅ 3241	㉑ 3251	㉠ 3261	㉡ 3271	㉢ 3281	㉣ 3291	㉤ 32A1	㉥ 32B1	2月 32C1	㋁ 32D1	㋑ 32E1	㋡ 32F1
2	㈂ 3202	㈃ 3212	㈄ 3222	㈅ 3232	㈆ 3242	㉒ 3252	㉡ 3262	㉢ 3272	㉣ 3282	㉤ 3292	㉥ 32A2	㉦ 32B2	3月 32C2	㋂ 32D2	㋒ 32E2	㋢ 32F2
3	㈃ 3203	㈄ 3213	㈅ 3223	㈆ 3233	㈇ 3243	㉓ 3253	㉢ 3263	㉣ 3273	㉤ 3283	㉥ 3293	㉦ 32A3	㉧ 32B3	4月 32C3	㋃ 32D3	㋓ 32E3	㋣ 32F3
4	㈄ 3204	㈅ 3214	㈆ 3224	㈇ 3234	㈈ 3244	㉔ 3254	㉣ 3264	㉤ 3274	㉥ 3284	㉦ 3294	㉧ 32A4	㉨ 32B4	5月 32C4	㋄ 32D4	㋔ 32E4	㋤ 32F4
5	㈅ 3205	㈆ 3215	㈇ 3225	㈈ 3235	㈉ 3245	㉕ 3255	㉤ 3265	㉥ 3275	㉦ 3285	㉧ 3295	㉨ 32A5	㉩ 32B5	6月 32C5	㋅ 32D5	㋕ 32E5	㋥ 32F5
6	㈆ 3206	㈇ 3216	㈈ 3226	㈉ 3236	㈊ 3246	㉖ 3256	㉥ 3266	㉦ 3276	㉧ 3286	㉨ 3296	㉩ 32A6	㉪ 32B6	7月 32C6	㋆ 32D6	㋖ 32E6	㋦ 32F6
7	㈇ 3207	㈈ 3217	㈉ 3227	㈊ 3237	㈋ 3247	㉗ 3257	㉦ 3267	㉧ 3277	㉨ 3287	㉩ 3297	㉪ 32A7	㉫ 32B7	8月 32C7	㋇ 32D7	㋗ 32E7	㋧ 32F7
8	㈈ 3208	㈉ 3218	㈊ 3228	㈋ 3238	㈌ 3248	㉘ 3258	㉧ 3268	㉨ 3278	㉩ 3288	㉪ 3298	㉫ 32A8	㉬ 32B8	9月 32C8	㋈ 32D8	㋘ 32E8	㋨ 32F8
9	㈉ 3209	㈊ 3219	㈋ 3229	㈌ 3239	㈍ 3249	㉙ 3259	㉨ 3269	㉩ 3279	㉪ 3289	㉫ 3299	㉬ 32A9	㉭ 32B9	10月 32C9	㋉ 32D9	㋙ 32E9	㋩ 32F9
A	㈊ 320A	㈋ 321A	㈌ 322A	㈍ 323A	㈎ 324A	㉚ 325A	㉩ 326A	㉪ 327A	㉫ 328A	㉬ 329A	㉭ 32AA	㉮ 32BA	11月 32CA	㋊ 32DA	㋚ 32EA	㋪ 32FA
B	㈋ 320B	㈌ 321B	㈍ 322B	㈎ 323B	㈏ 324B	㉛ 325B	㉪ 326B	㉫ 327B	㉬ 328B	㉭ 329B	㉮ 32AB	㉯ 32BB	12月 32CB	㋋ 32DB	㋛ 32EB	㋫ 32FB
C	㈌ 320C	㈍ 321C	㈎ 322C	㈏ 323C	㈐ 324C	㉜ 325C	㉫ 326C	㉬ 327C	㉭ 328C	㉮ 329C	㉯ 32AC	㉰ 32BC	Hg 32CC	㋌ 32DC	㋜ 32EC	㋬ 32FC
D	㈍ 320D	㈎ 321D	㈏ 322D	㈐ 323D	㈑ 324D	㉝ 325D	㉬ 326D	㉭ 327D	㉮ 328D	㉯ 329D	㉰ 32AD	㉱ 32BD	erg 32CD	㋍ 32DD	㋝ 32ED	㋭ 32FD
E	㈎ 320E	㈏ 321E	㈐ 322E	㈑ 323E	㈒ 324E	㉞ 325E	㉭ 326E	㉮ 327E	㉯ 328E	㉰ 329E	㉱ 32AE	㉲ 32BE	eV 32CE	㋎ 32DE	㋞ 32EE	㋮ 32FE
F	㈏ 320F		㈐ 322F	㈑ 323F	㈒ 324F	㉟ 325F	㉮ 326F	㉯ 327F	㉰ 328F	㉱ 329F	㉲ 32AF	㉳ 32BF	LTD 32CF	㋏ 32DF	㋟ 32EF	㋯ 32FF

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

```
ord(ch)
```



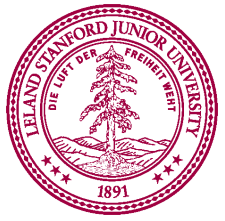
Functions which take strings

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

```
len
ord
hash
<
==
in
```

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

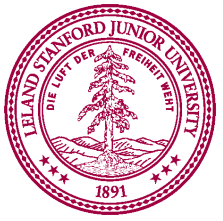
Won't need these too much in CS106A, but they are super clutch in cryptography and datastructures



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'
```

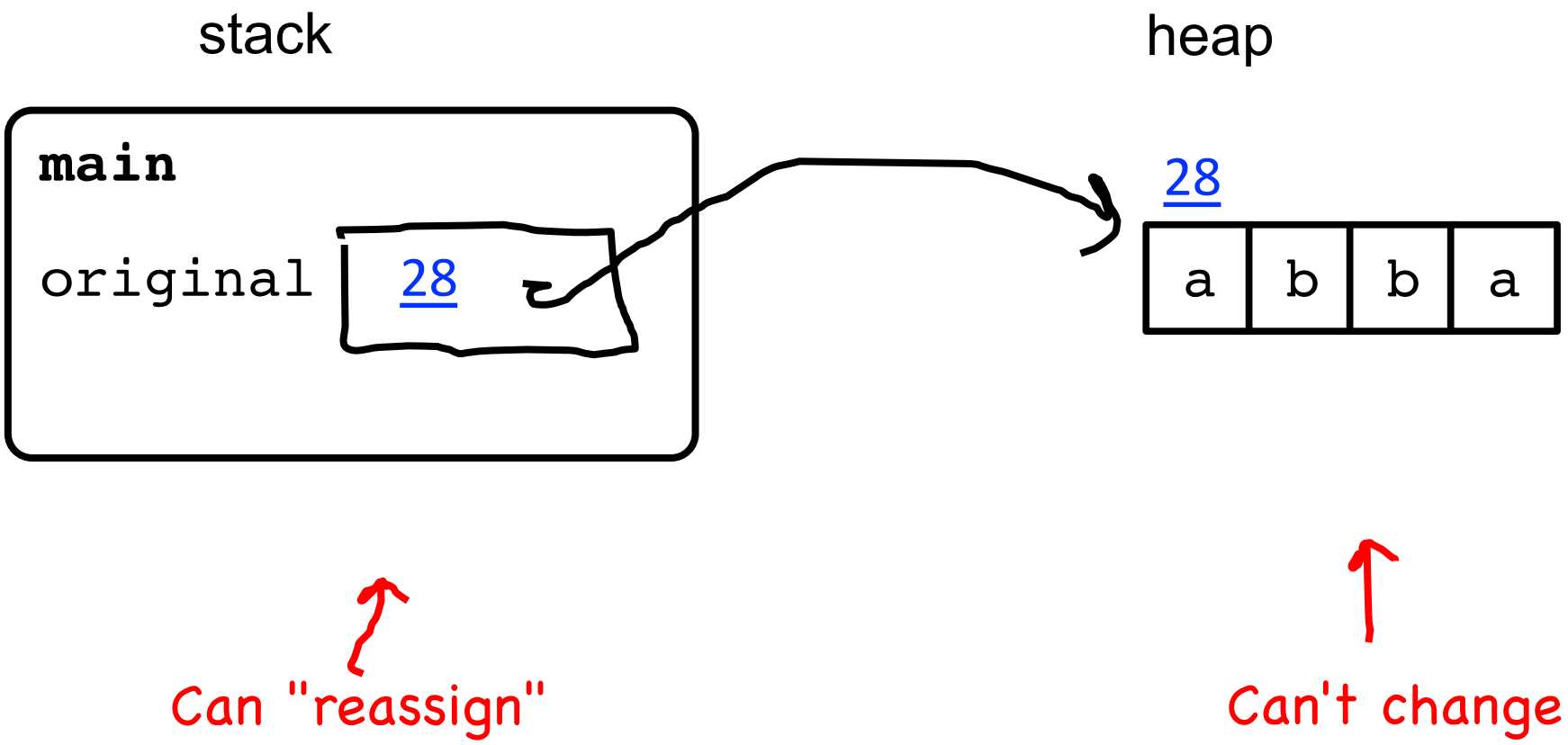
```
x[1] = 'z'
```

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

```
x = 'azc'
```

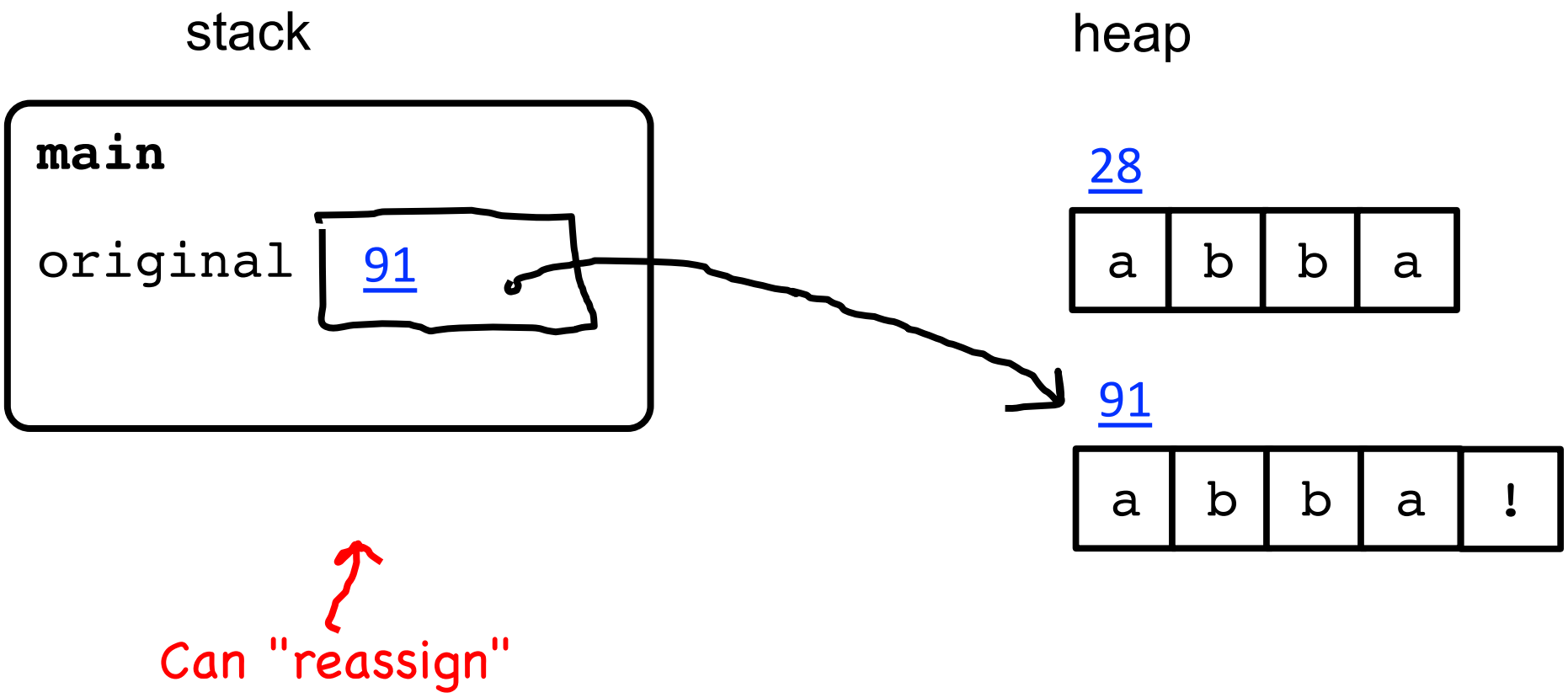
Strings are Immutable (Take 2)

```
original = 'abba'
```



Strings are Immutable (Take 1)

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

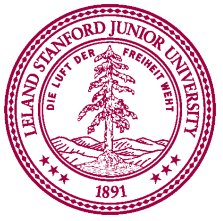


Pro tip: use id function to check a reference!

Strings are often made through concatenation

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

    print(s3)
```

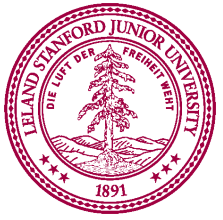
```
I got an A in CS106A
```



Lists are **mutable**

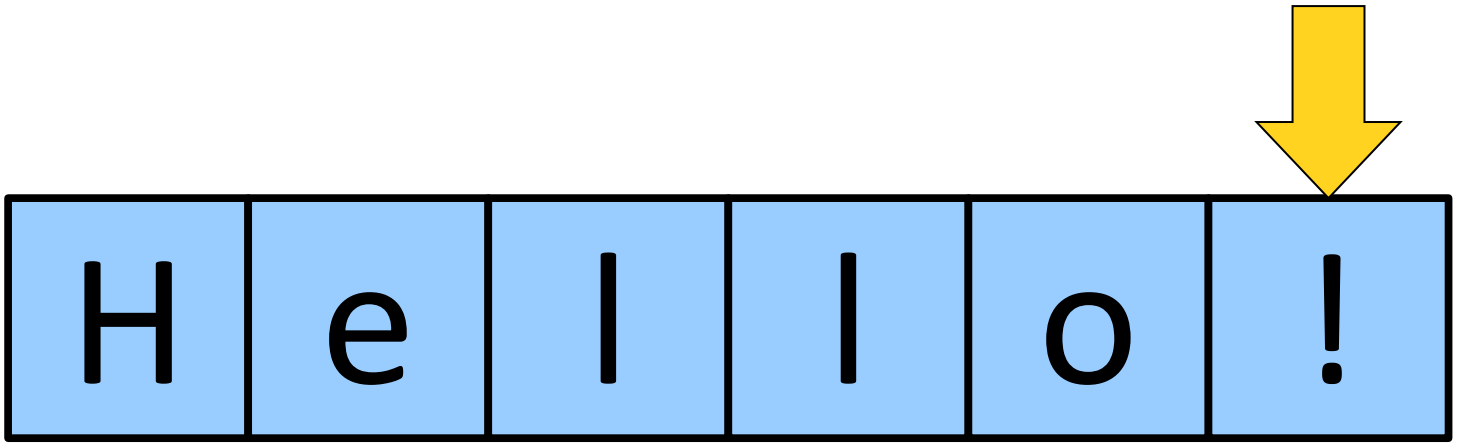
Strings are **immutable**

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

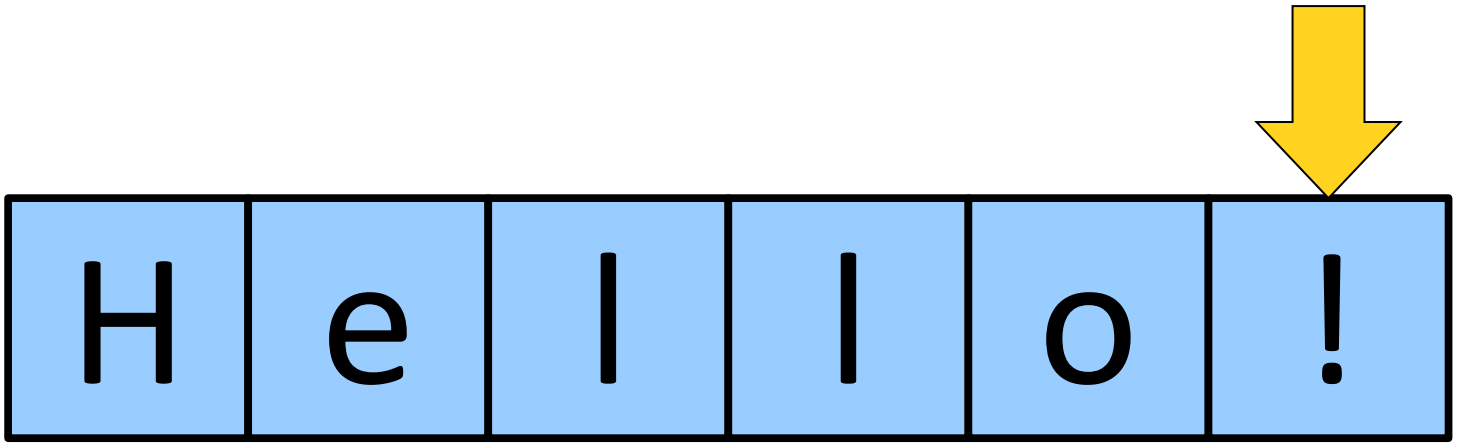


Many string algorithms use the "loop and construct" pattern.

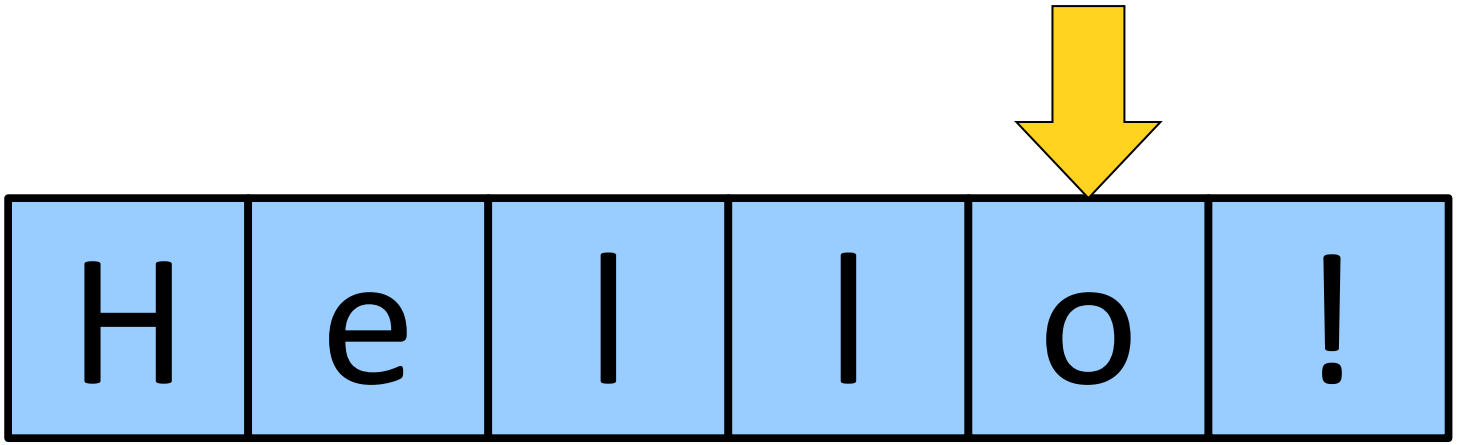
Reversing a String



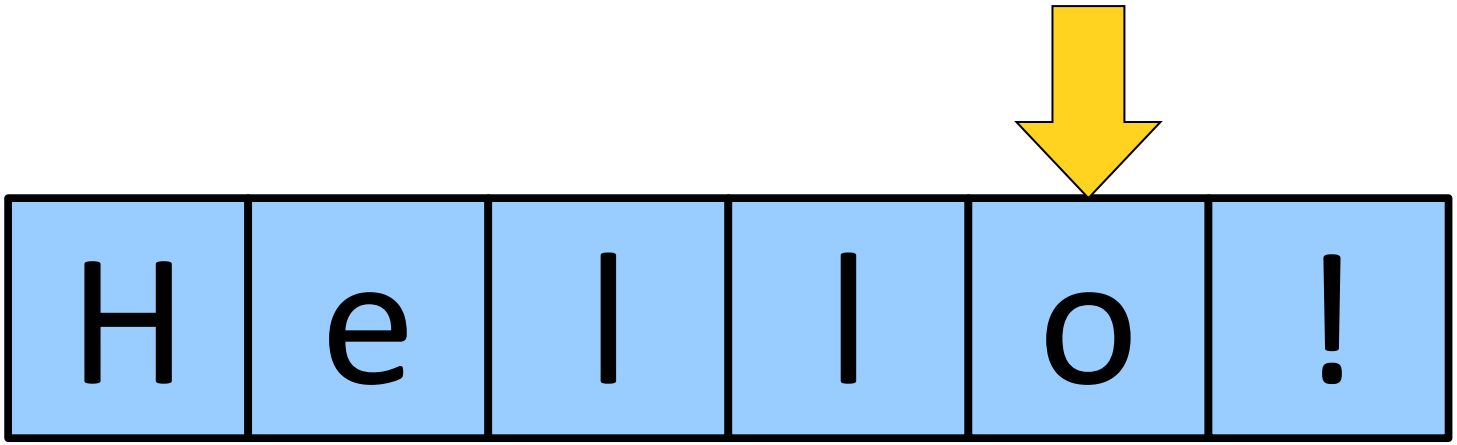
Reversing a String

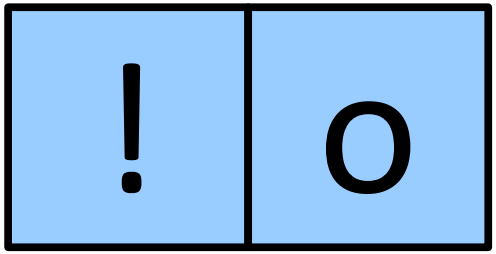


Reversing a String

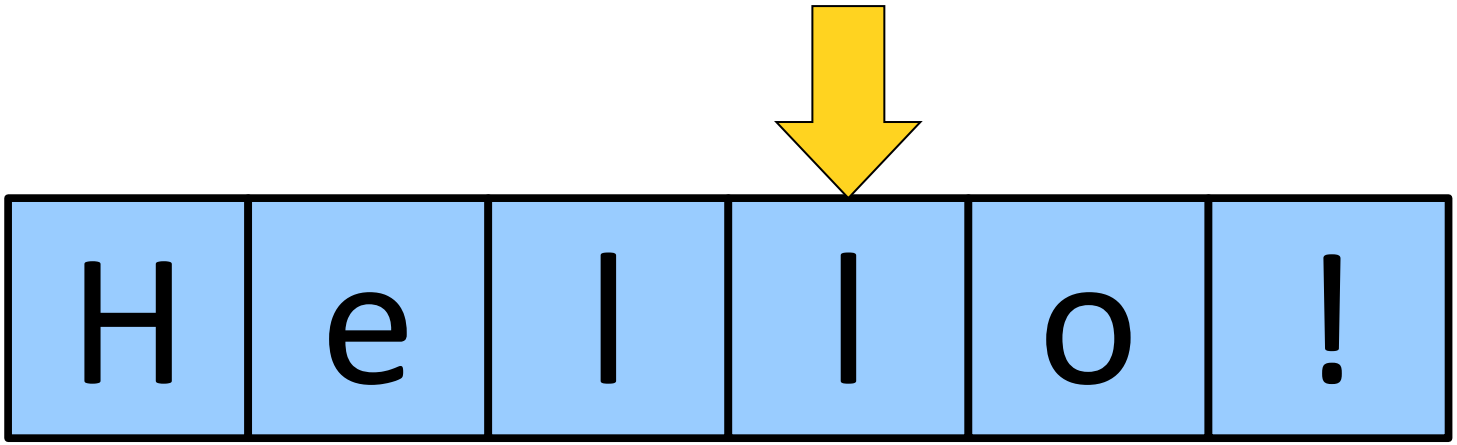


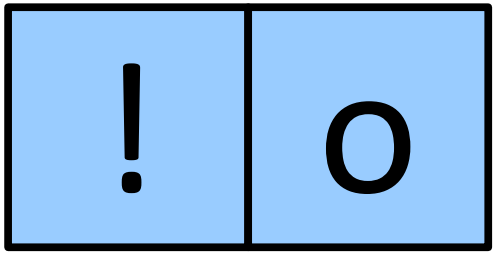
Reversing a String

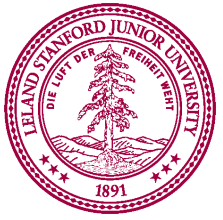




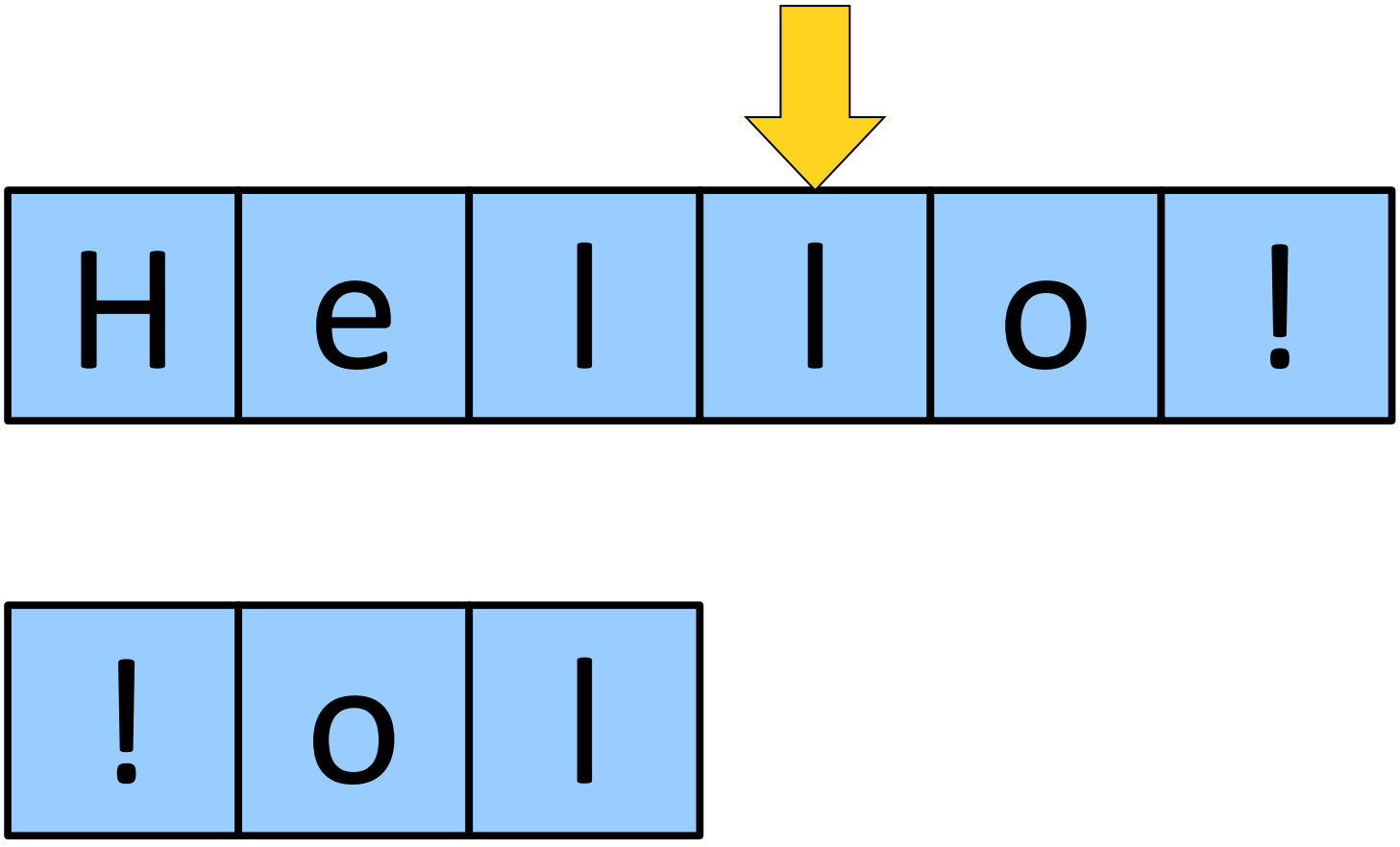
Reversing a String



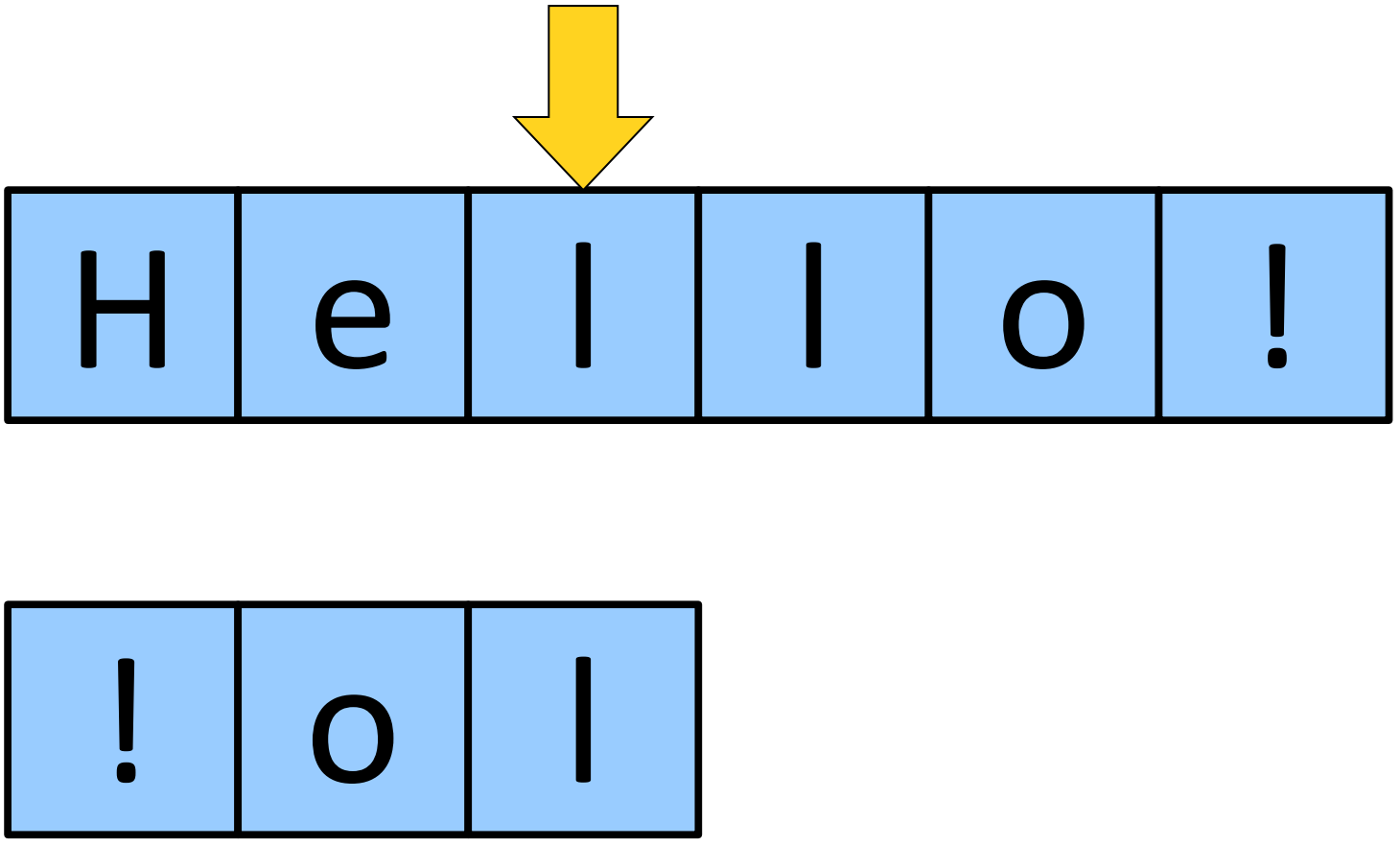




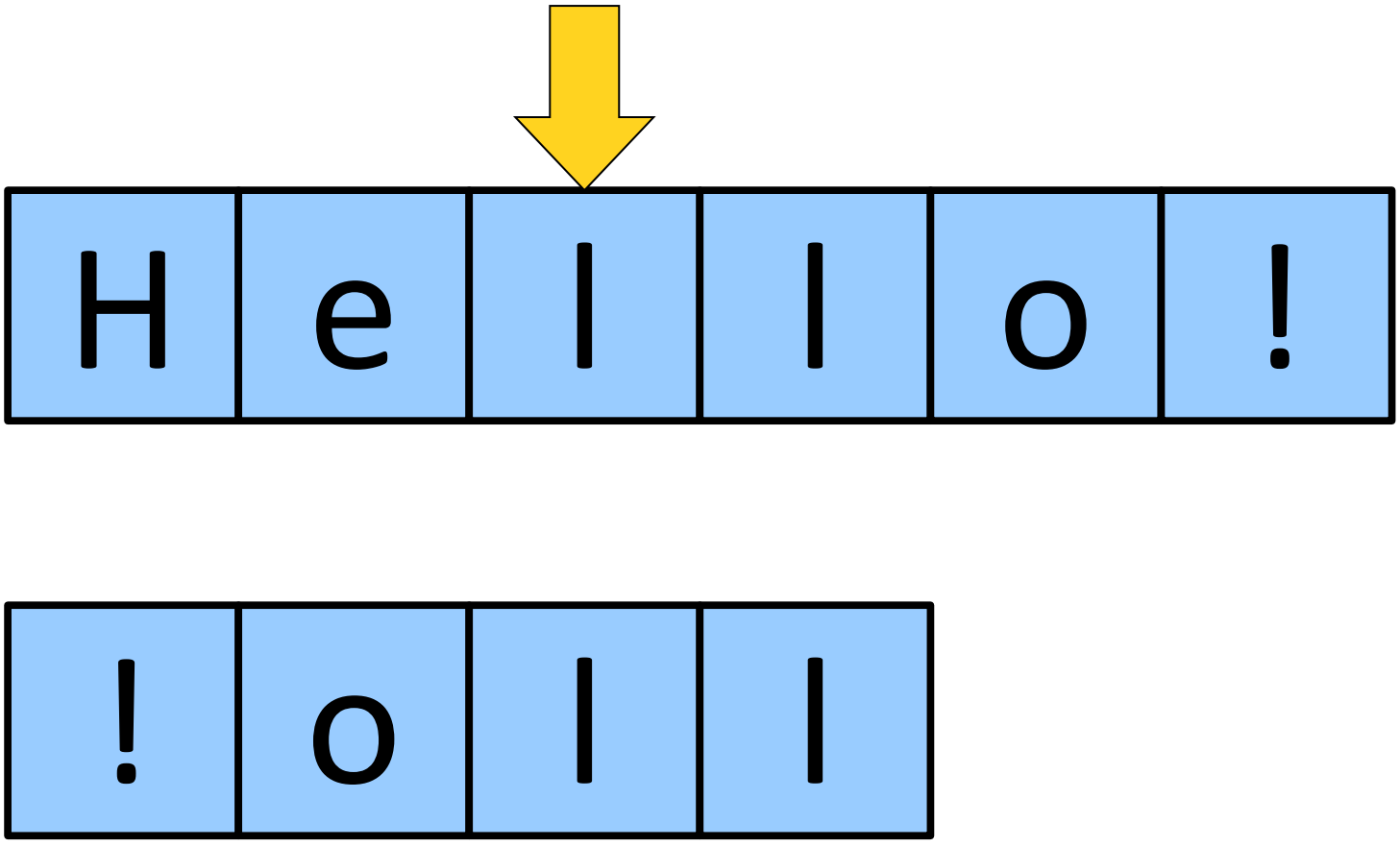
Reversing a String



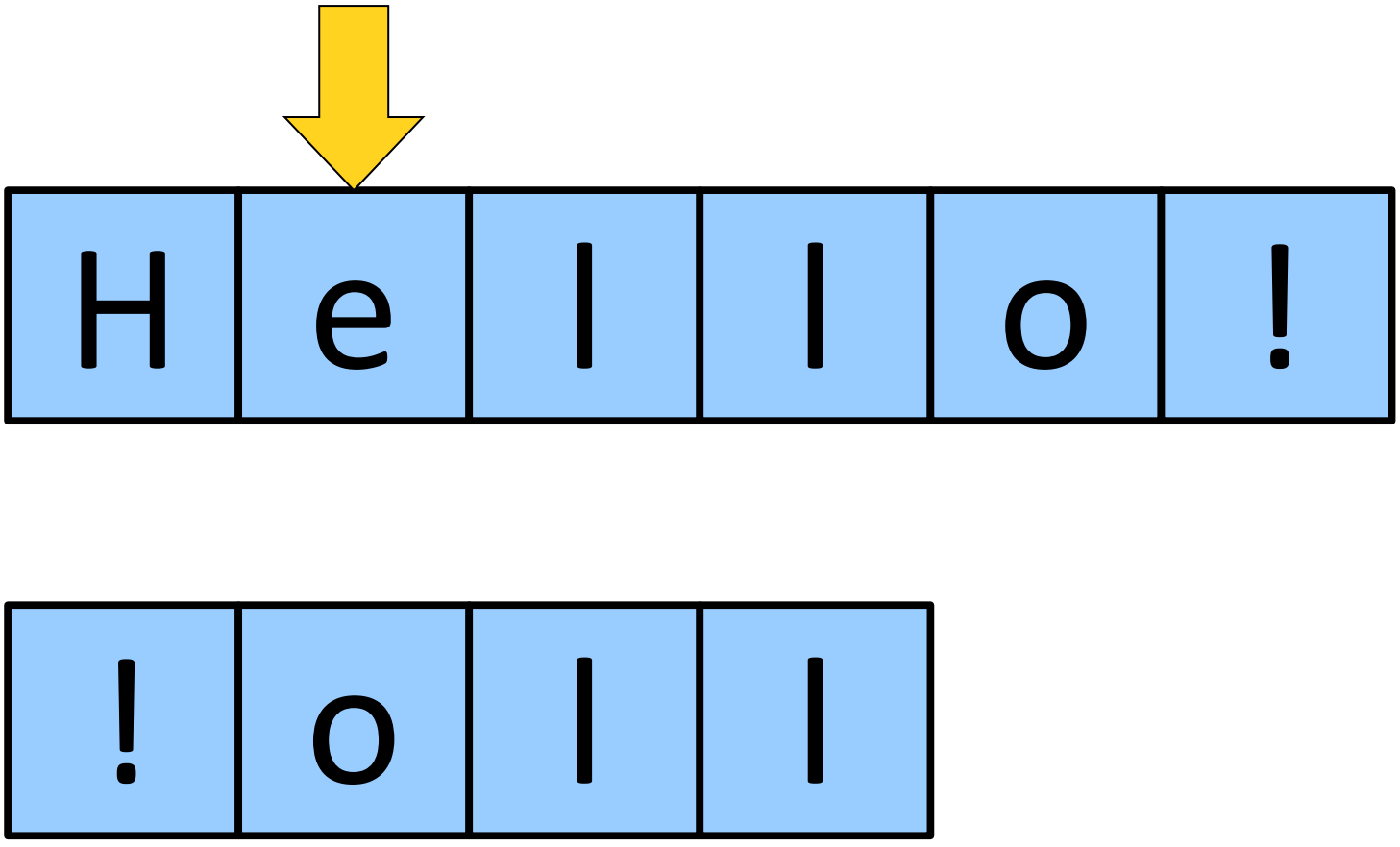
Reversing a String



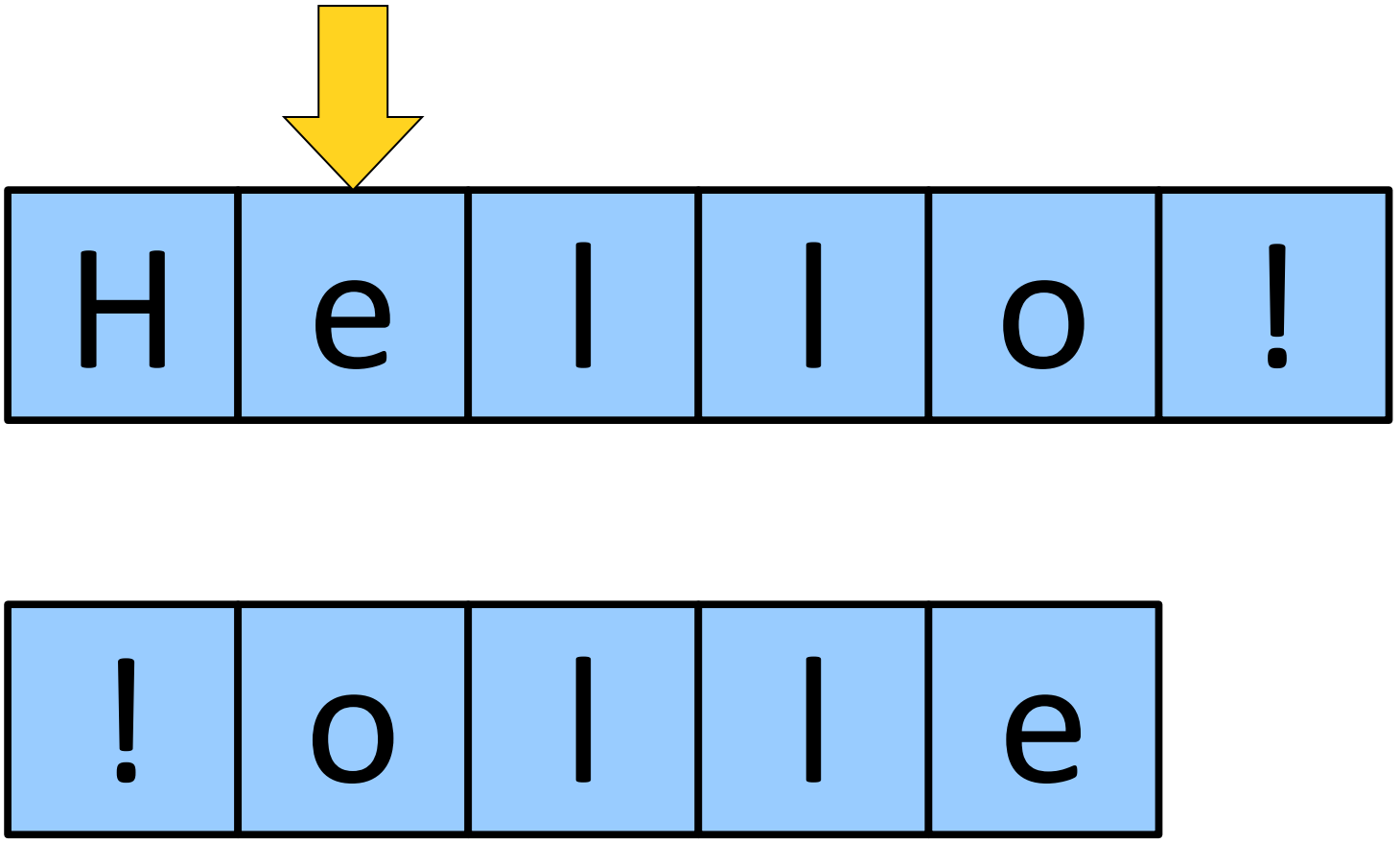
Reversing a String



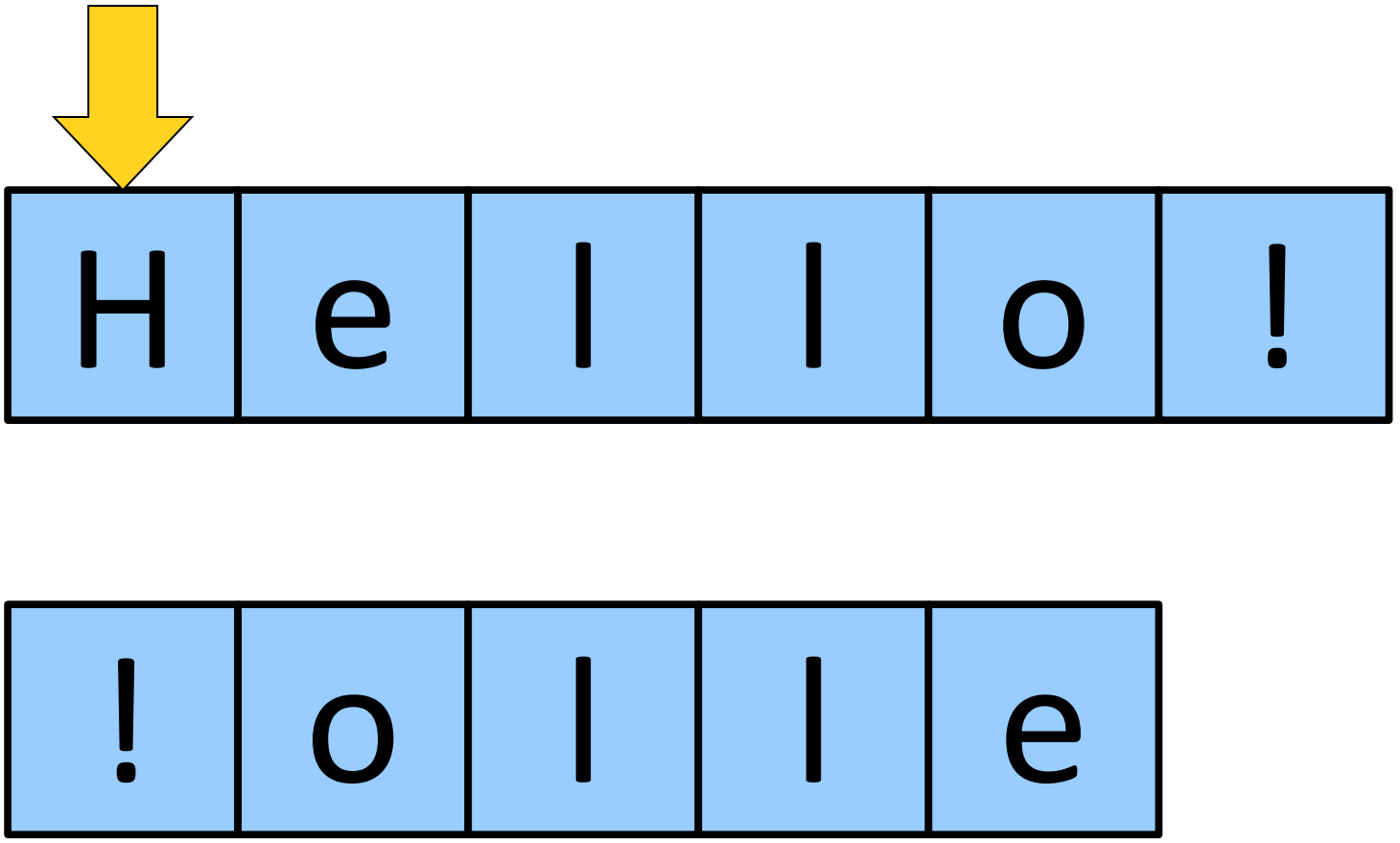
Reversing a String



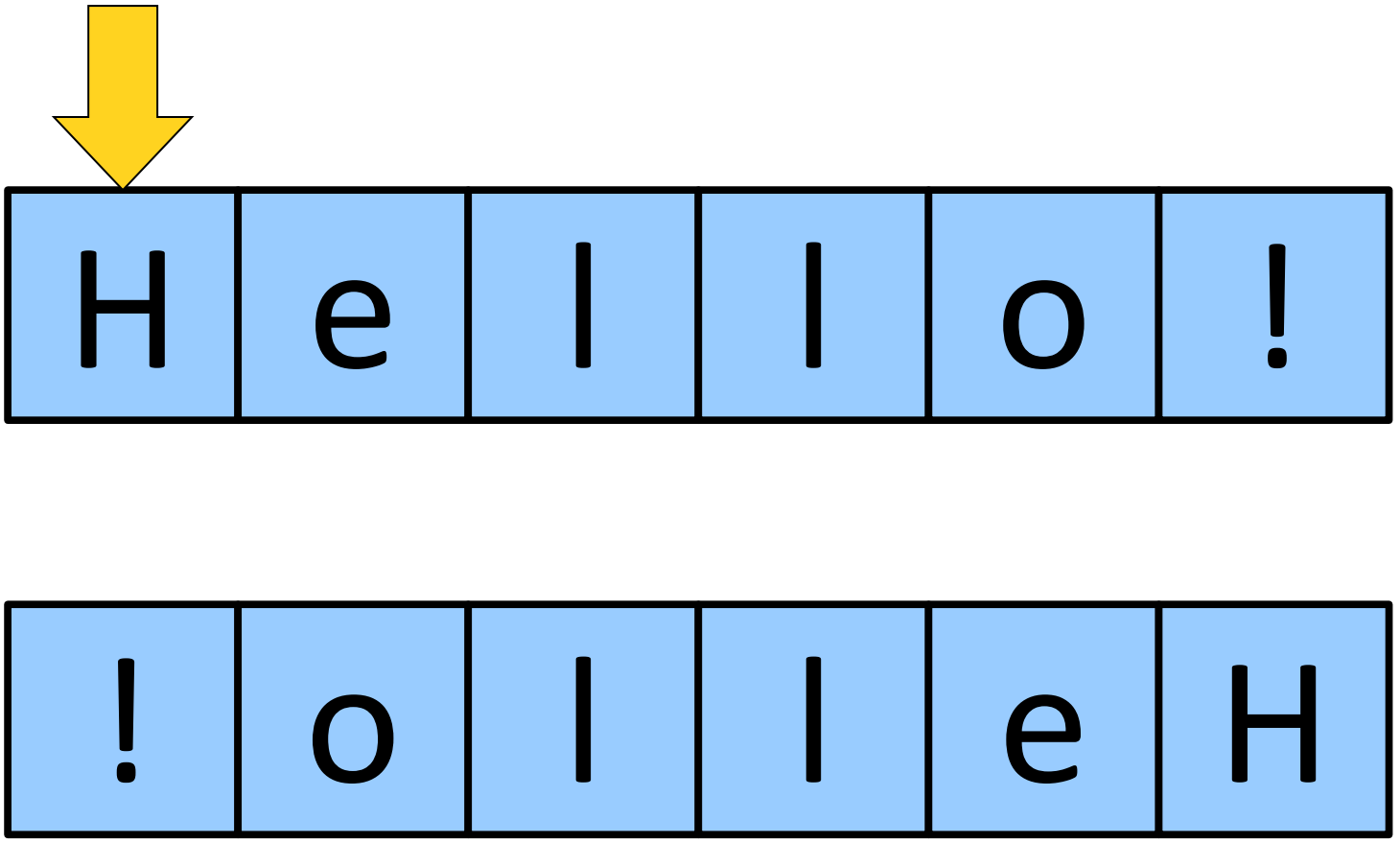
Reversing a String



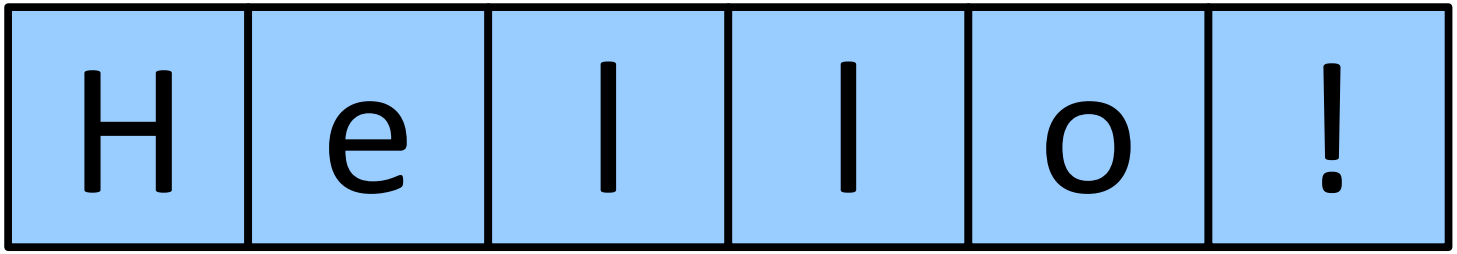
Reversing a String

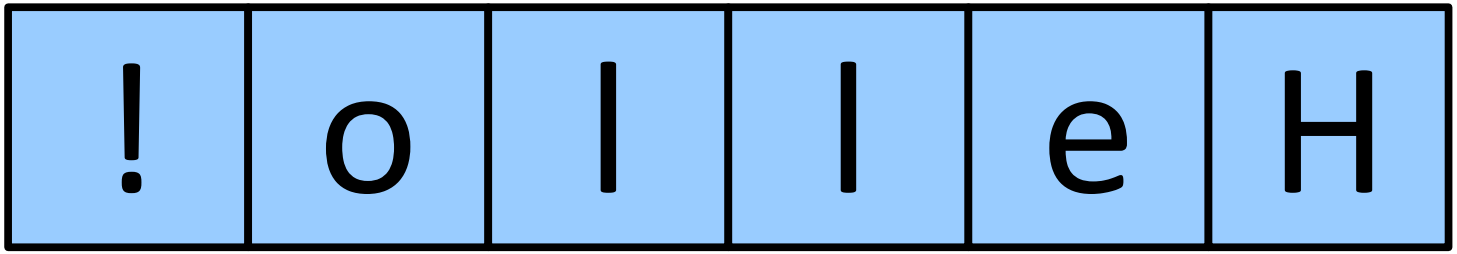


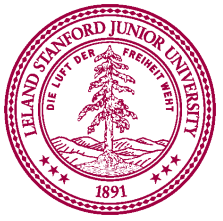
Reversing a String



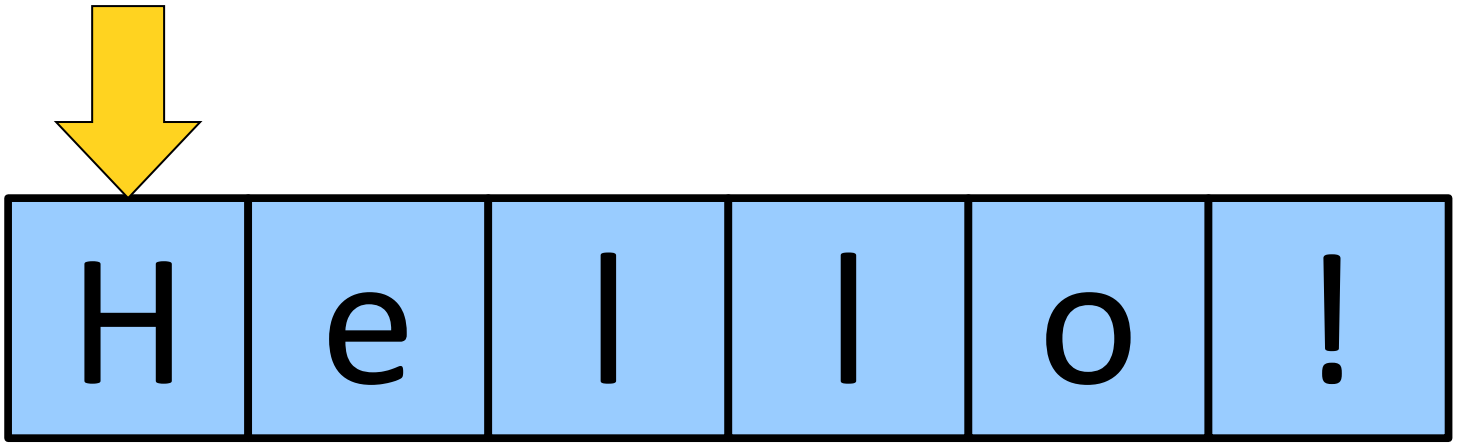
Reversing a String

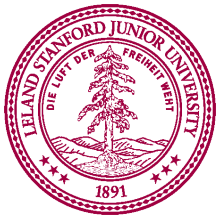




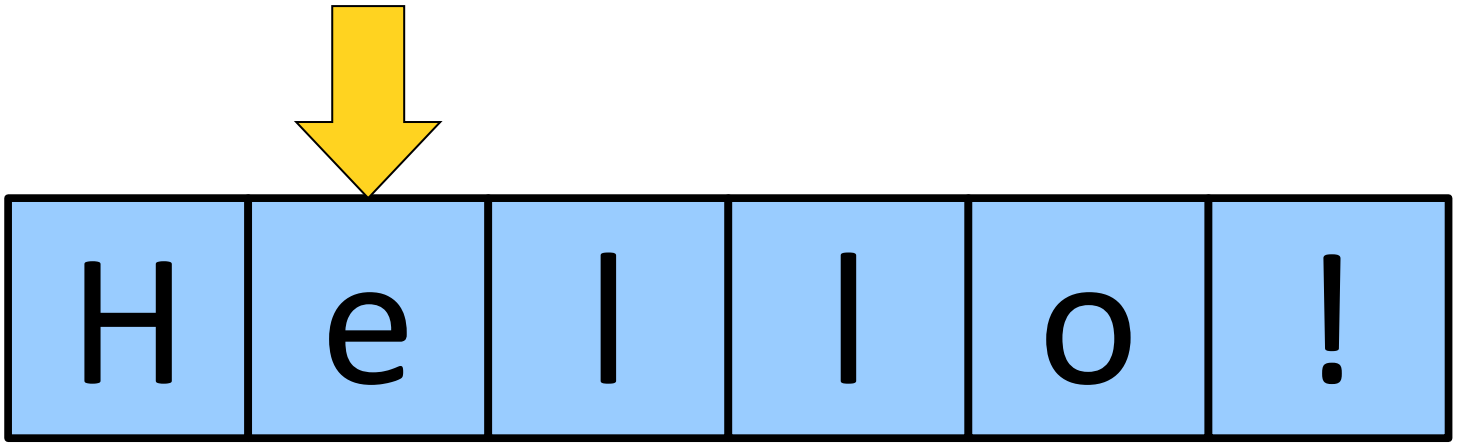


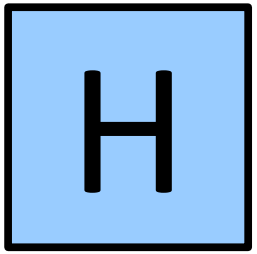
Reversing a String



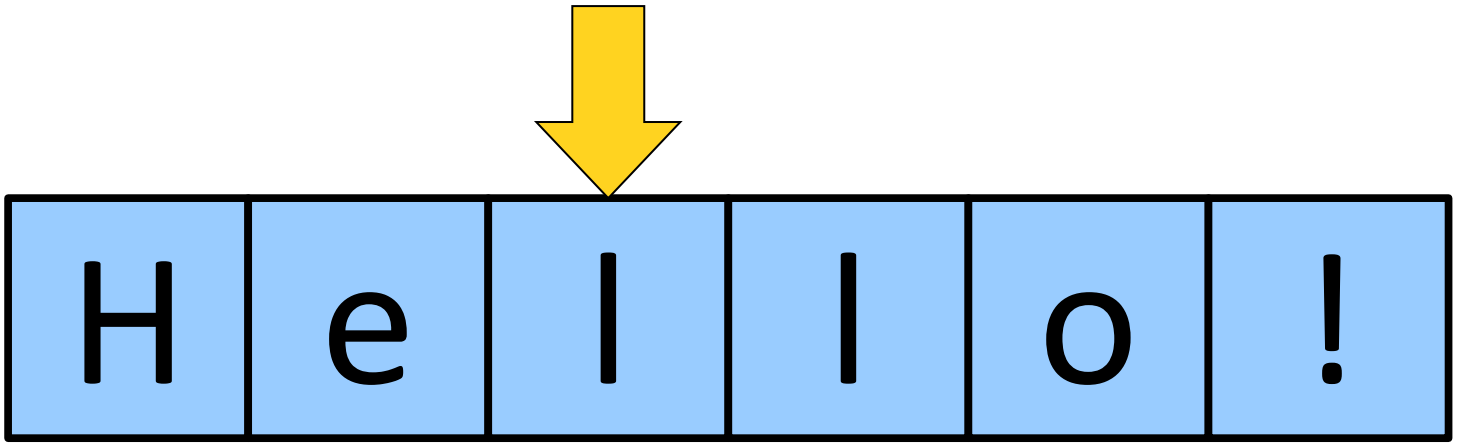


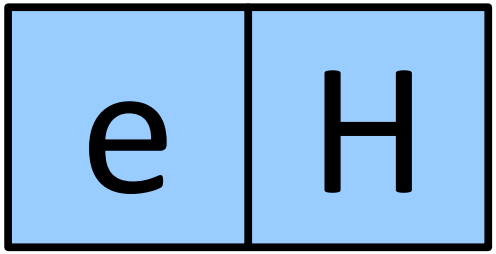
Reversing a String

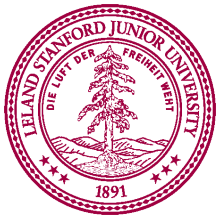




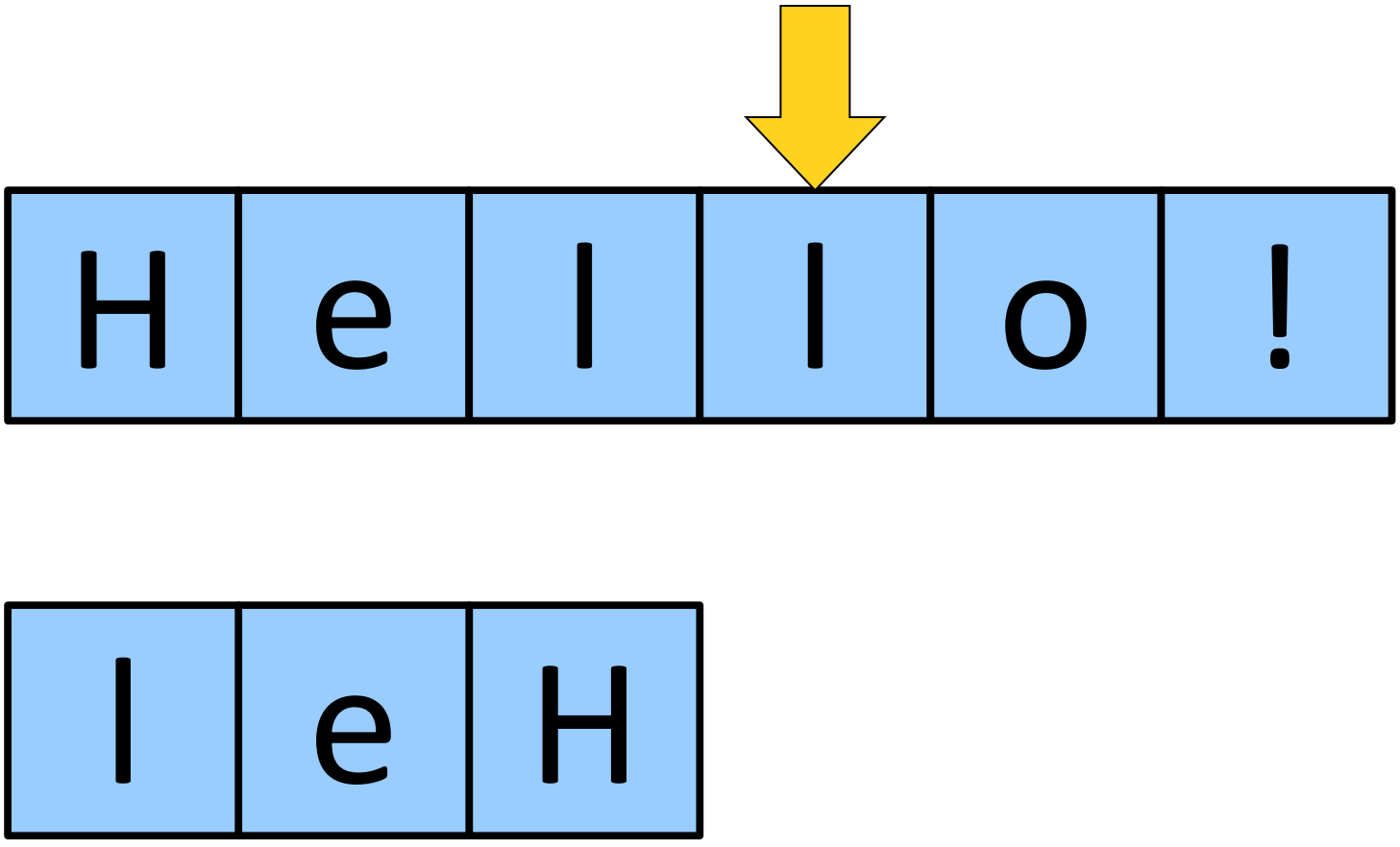
Reversing a String



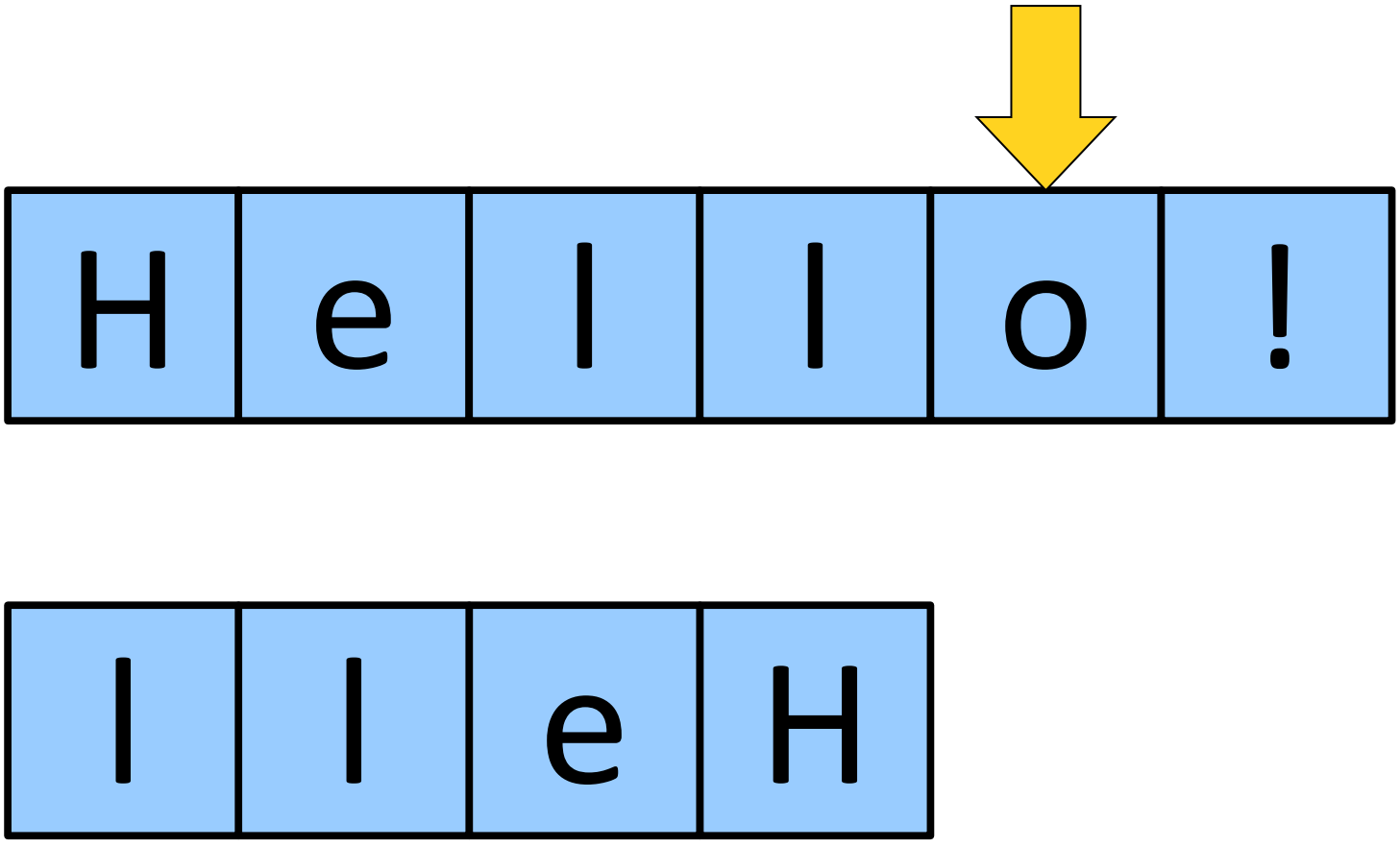




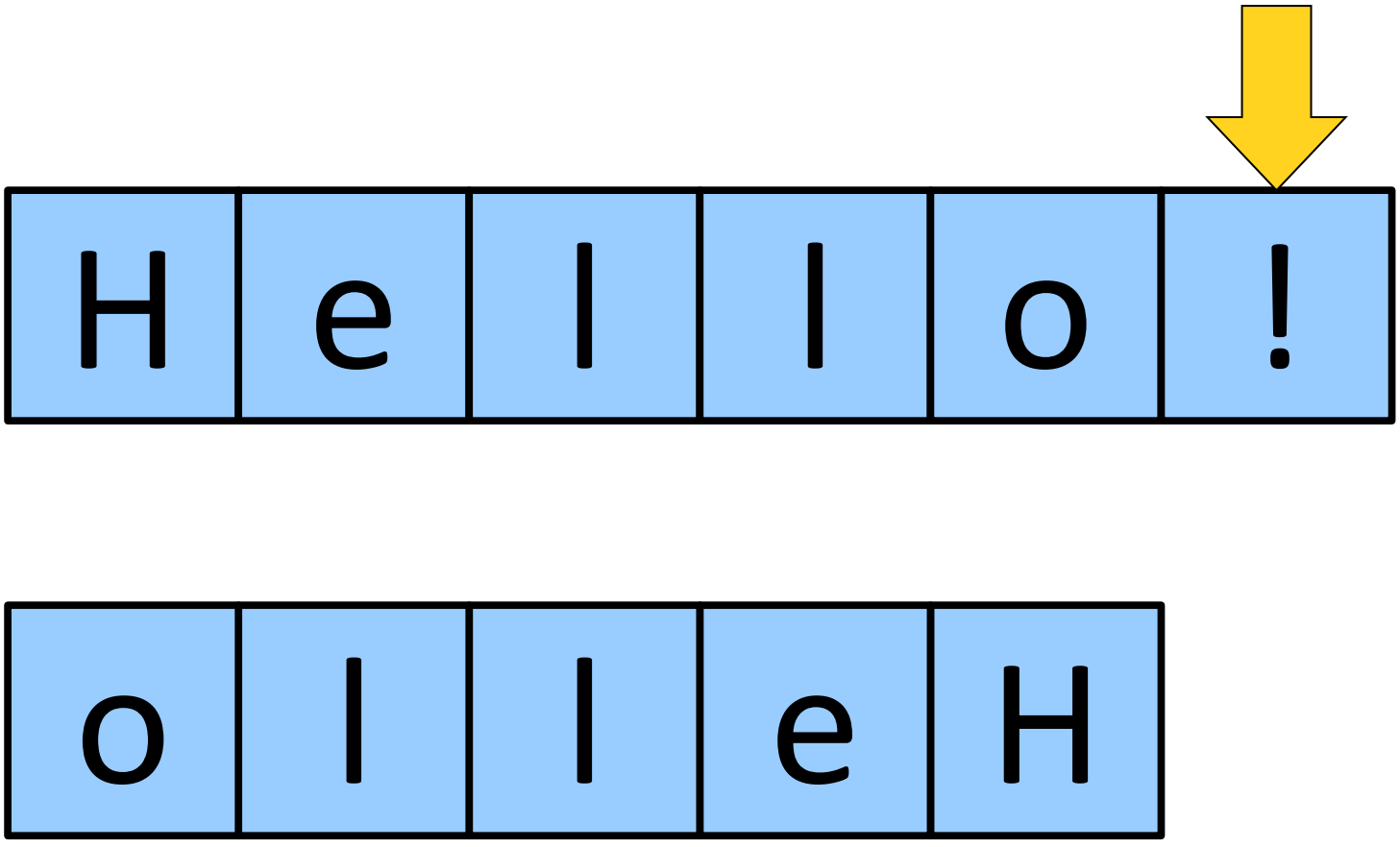
Reversing a String



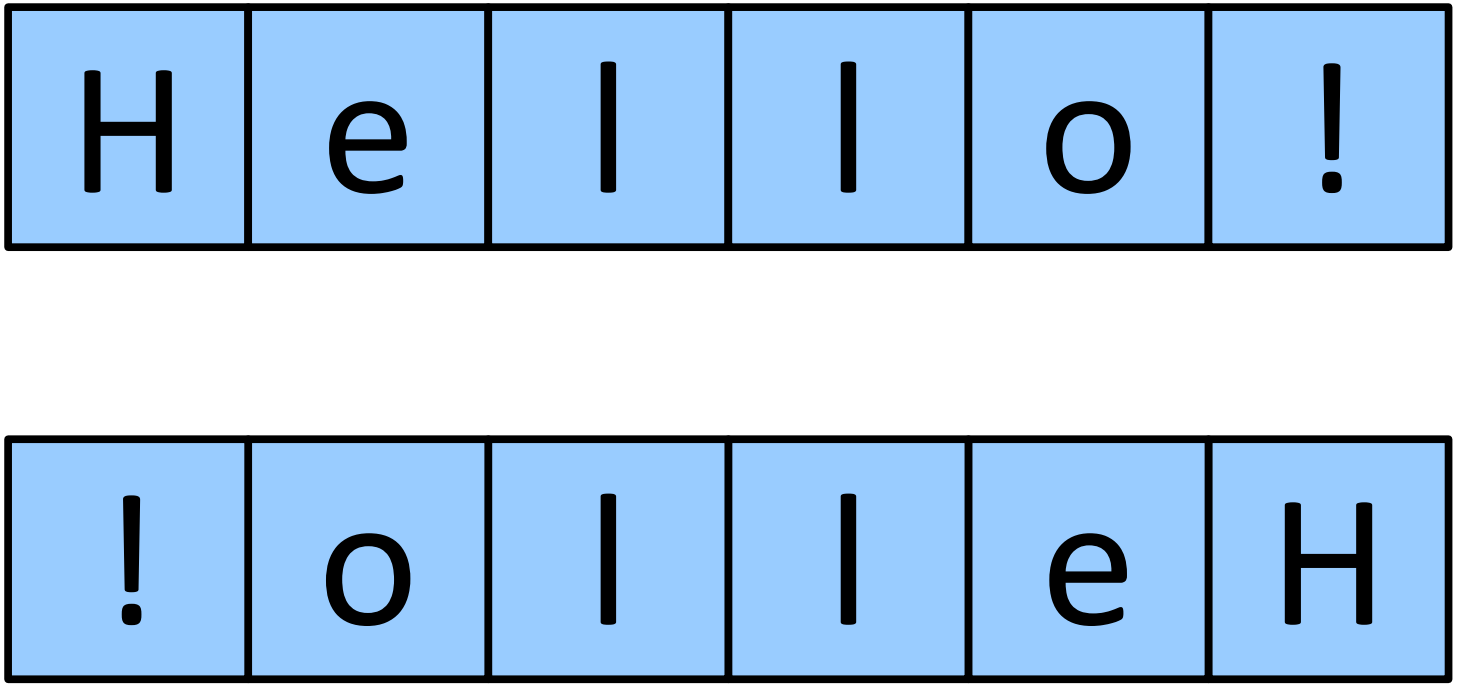
Reversing a String

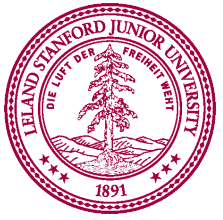


Reversing a String



Reversing a String



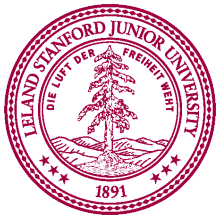


reverse_string

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

        return result
```

result	str	i
DESSERTS	STRESSED	8

```
ReverseString
This program reverses a string.
Enter a string: STRESSED
STRESSED spelled backwards is DESSERTS
```



Using portions of slides by Eric Roberts

reverse_string redux

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

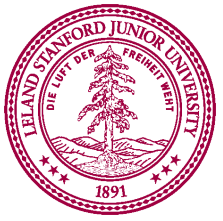
Swiss army knife pattern

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

Medium bear.

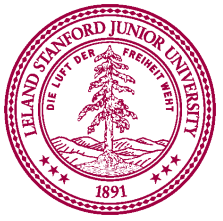
```
def reverse_string_v3(str):
    '''
    This uses the slice operator in a
    special way. With no start, no
    end and a delta of -1, slice reverses.
    '''
    return str[::-1]
```

One cool trick that will blow your mind



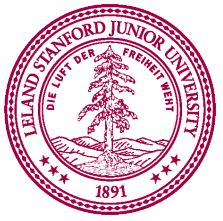
Palindrome

- 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)
- 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!