

Strings are Indexed (Question 1)

- `s = 'abc d'`

```
0 1 2 3 4
a b c d
```
- Access characters with `[]`
 - `s[0]` is 'a'
 - `s[4]` is 'd'
 - `s[5]` **causes an error**
 - `s[0:2]` is 'ab' (excludes `c`)
 - `s[2:]` is 'c d'
- Called "string slicing"

- `t = 'Hello all'`

```
0 1 2 3 4 5 6 7 8
H e l l o a l l
```
- What is `t[3:6]`?

```
A: 'lo a'
B: 'lo'
C: 'lo '
D: 'o'
E: I do not know
```

3

Other Things We Can Do With Strings

Operator in: `s1 in s2`

- Tests if `s1` "a part of" (or a *substring* of) `s2`
- Evaluates to a bool

Examples:

```
>>> s = 'abracadabra'
>>> 'a' in s
True
>>> 'cad' in s
True
>>> 'foo' in s
False
```

Built-in Function len: `len(s)`

- Value is # of chars in `s`
- Evaluates to an int

Examples:

```
>>> s = 'abracadabra'
>>> len(s)
11
>>> len(s[1:5])
4
>>> s[1:len(s)-1]
'bracadabr'
>>>
```

7

Defining a String Function

Want to write function middle which returns the middle 3rd of a string (length divisible by 3).

How we want it to behave:

```
>>> middle('abc')
'b'
>>> middle('aabbcc')
'bb'
>>> middle('aaabbccc')
'bbb'
```

Important Questions:

1. What are the parameters?
2. What is the return value?
3. What goes in the body?

```
def middle(text):
    ???
    return middle_third
```

8

Advanced String Features: Method Calls

- Strings have some useful *methods*
 - Like functions, but "with a string in front"
- **Format:** `<string name>.<method name>(x,y,...)`
- **Example:** `upper()` returns an upper case version

```
>>> s = 'Hello World'
>>> s.upper()
'HELLO WORLD'
>>> s
'Hello World'
>>> s[1:5].upper()
'ELLO'
>>> 'scream'.upper()
'SCREAM'
>>> 'cs1110'.upper()
'CS1110'
```

11

Examples of String Methods

- `s1.index(s2)`
 - Returns position of the first instance of `s2` in `s1`
- `s1.count(s2)`
 - Returns number of times `s2` appears inside of `s1`
- `s.strip()`
 - Returns a copy of `s` with white-space removed at ends

```
• s = 'abracadabra'
0 1 2 3 4 5 6 7 8 9 10
a b r a c a d a b r a

• s.index('a')           0
• s.index('rac')        2
• s.count('a')          5
• s.count('b')          2
• s.count('x')          0
• 'a b'.strip()         'a b'
```

See Python Docs for more

12

String Extraction Example

```
def firstparens(text):
    """Returns: substring in ()
    Uses the first set of parens
    Param text: a string with ()"""
    # Find the open parenthesis
    start = text.index('(')
    # Store part AFTER paren
    substr = text[start+1:]
    # Find the close parenthesis
    end = substr.index(')')
    return substr[:end]

>>> s = 'One (Two) Three'
>>> firstparens(s)
'Two'
>>> t = '(A) B (C) D'
>>> firstparens(t)
'A'
```

13

String Extraction Puzzle

```
def second(thelist):
    """Returns: second word in a list
    of words separated by commas, with
    any leading or trailing spaces from the
    second word removed
    Ex: second('A, B, C') => 'B'
    Param thelist: a list of words with
    at least two commas """
    1 start = thelist.index(',')
    2 tail = thelist[start+1:]
    3 end = tail.index(',')
    4 result = tail[:end]
    5 return result
```

```
>>> second('cat, dog, mouse, lion')
expecting: 'dog'      get: 'dog'

>>> second('apple, pear, banana')
expecting: 'pear'    get: 'pear'
```

Where is the error?

- A: Line 1
- B: Line 2
- C: Line 3
- D: Line 4
- E: There is no error

14

Not All Functions Need a Return

```
def greet(n):
    """Prints a greeting to the name n

    Parameter n: name to greet
    Precondition: n is a string"""
    print('Hello '+n+'!')
    print('How are you?')
```

Displays these strings on the screen

No assignments or return (returns None)

17

Exercise 1

Module Text	Python Interactive Mode
# module.py	>>> import module
	>>> module.x
def foo(x):	...
x = 1+2	What does Python give me?
x = 3*x	
	A: 9
	B: 10
	C: 1
	D: Nothing
	E: Error

25

Exercise 2

Module Text	Python Interactive Mode
# module.py	>>> import module
	>>> module.x
def foo(x):	...
x = 1+2	What does Python give me?
x = 3*x	
	A: 9
	B: 10
	C: 1
	D: Nothing
	E: Error

27

Exercise 3

Module Text	Python Interactive Mode
# module.py	>>> import module
	>>> module.x
def foo(x):	...
x = 1+2	What does Python give me?
x = 3*x	
return x+1	
	A: 9
	B: 10
	C: 1
	D: Nothing
	E: Error

29

Exercise 4

Function Definition	Function Call
def foo(a,b):	>>> x = 2
1 x = a	>>> foo(3,4)
2 y = b	>>> x
3 return x*y+y	What does Python give me?
	A: 2
	B: 3
	C: 16
	D: Nothing
	E: I do not know

31