Strings

- String basics
- String formatting

String basics

A *string* is a sequence of characters
Example: abrakadabra,
We can store it and associate a name/variable with it.

To strings like "hi" or 'broom' we refer to as a *string literal* in a Python program.

String basics

A *string* is a sequence of characters
Example: abrakadabra,
We can store it and associate a name/variable with it.

To strings like "hi" or 'broom' we refer to as a *string literal* in a Python program.

```
>>> firstPart = “abra”
>>> secondPart = “kadabra”
```

string variables

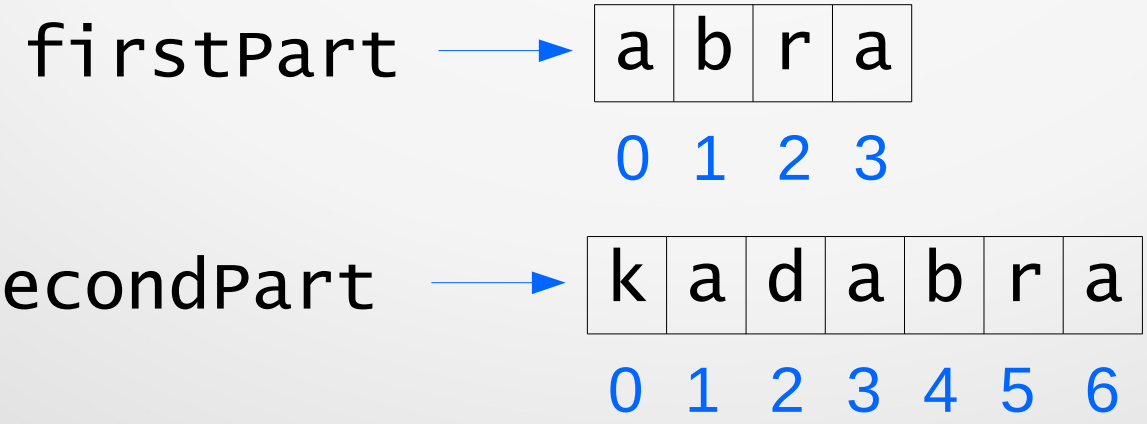
string literals

String basics

A *string* is a sequence of characters
Example: abrakadabra,
We can store it and associate a name/variable with it.

To strings like "hi" or 'broom' we refer to as a *string literal* in a Python program.

```
>>> firstPart = “abra”
>>> secondPart = “kadabra”
```



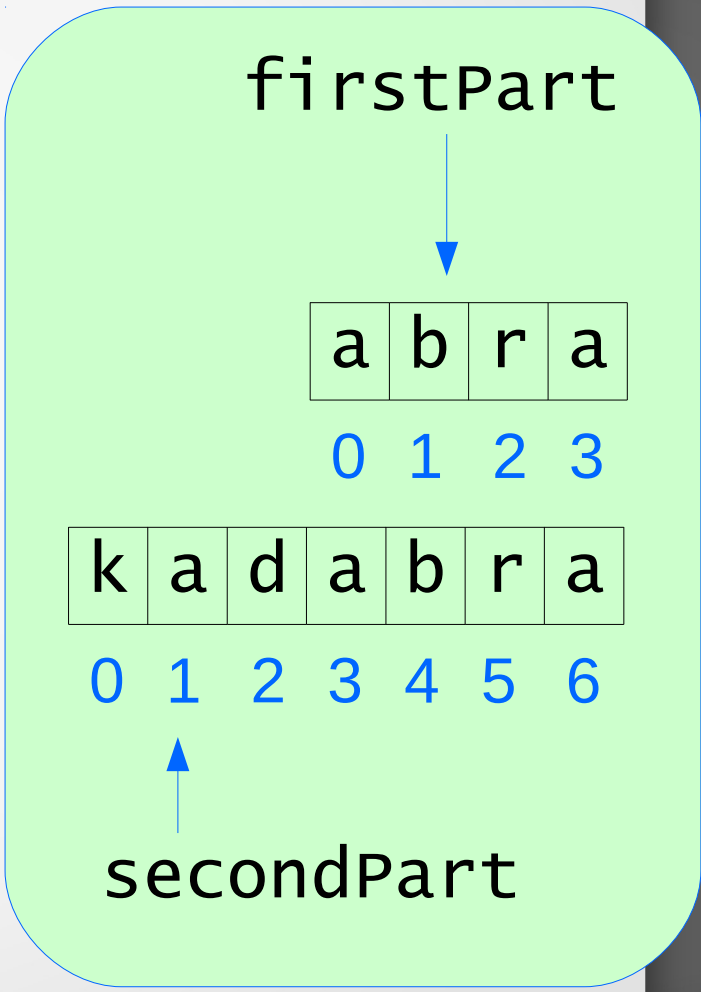
String basics

What can I do with strings in Python?

String basics

What can I do with strings in Python?

```
>>> firstPart = “abra”
>>> secondPart = “kadabra”
```

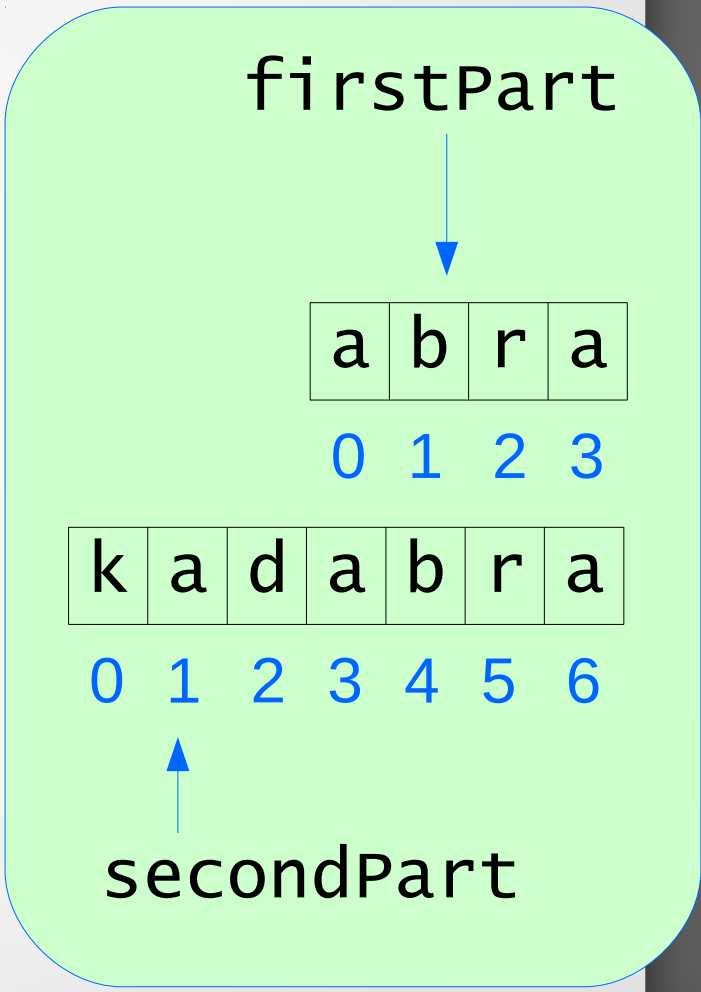


String basics

What can I do with strings in Python?

```
>>> firstPart = “abra”
>>> secondPart = “kadabra”

>>> firstPart[2]
‘r’
>>> secondPart[0]
‘k’
>>> secondPart[2:]
‘dabra’
>>> secondPart[1:4]
‘ada’
>>> firstPart + secondPart
‘abrakadabra’
>>> len(firstPart)
4
```



String basics

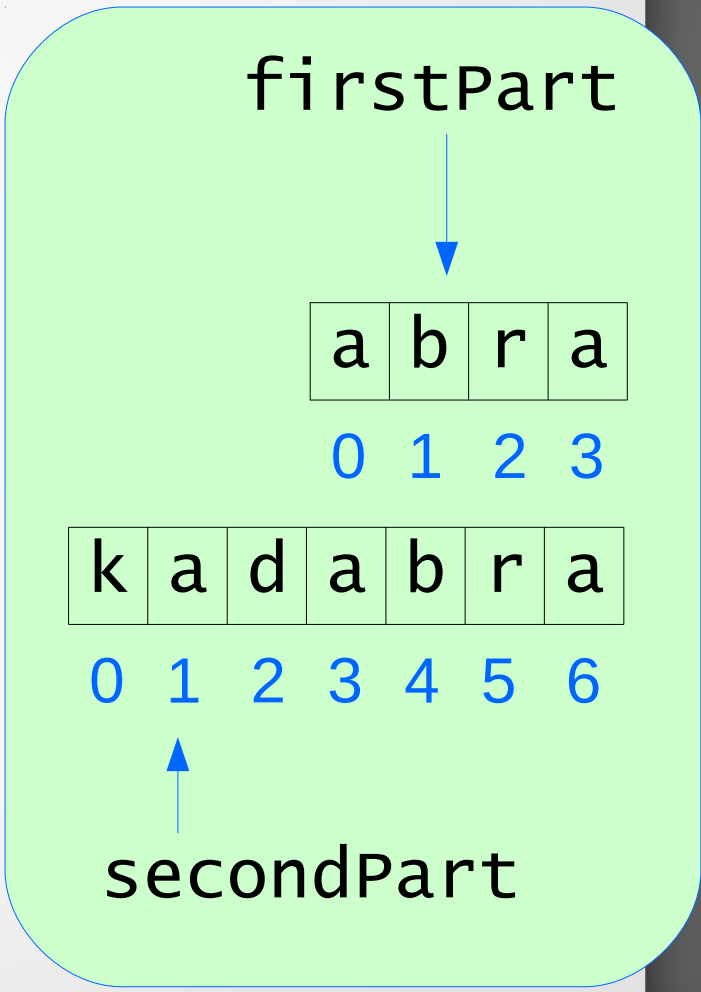
What can I do with strings in Python?

```
>>> firstPart+”cloud”
‘abracloud’

>>> firstPart+’/’+secondPart
‘abra/kadabra’

>>> firstPart*3
‘abraabraabra’

>>> firstPart
‘abra’
```



String basics: in-class work

Do the activities 1-3 in the in-class activity handout

String basics

Download the following three programs from our web-page and run them one by one. See what happens!

stringsProg1.py stringsProg2.py stringsProg3.py

String basics: in-class work

Write a program that gets three words from the user, then displays the length of each word and then displays all the words merged together separated by -(dash).

For example, assume that the user entered: “hat”,“my”,”track”
Program’s output:
3
2
5
Hat-my-track

- this is in-class activity 4) in the handout

String formatting

Program output commonly includes the value of variables as a part of the text.

Example: for the following code

```
num = 18
tum = 9.8
print("I have a number",num,"and a number",tum)
```

String formatting

Program output commonly includes the value of variables as a part of the text.

Example: for the following code

```
num = 18
tum = 9.8
print("I have a number",num,"and a number",tum)
```

will produce:

```
I have a number 18 and a number 9.8
```

Note that we have to keep track of those double quotes (") and commas in the print statement.

String formatting

Compare the following two `print` statements:

```
num = 18,tum = 9.8
```

```
print("I have a number",num,"and a number",tum)
```

```
print("I have a number %d and a number %f" %
(num,tum))
```

The first one produced:

```
I have a number 18 and a number 9.8
```

The second one produced:

```
I have a number 18 and a number 9.800000
```

String formatting

A *string formatting expression* allows a programmer to create a string with *placeholders* that are replaced by the *value of variables*.
Such a placeholder is called a *conversion specifier*.
Different conversion specifiers are used to perform a conversion of the given variable value to a different type when creating the string.

String formatting

A *string formatting expression* allows a programmer to create a string with *placeholders* that are replaced by the *value of variables*.
Such a placeholder is called a *conversion specifier*.
Different conversion specifiers are used to perform a conversion of the given variable value to a different type when creating the string.

Example:

```
num = 5.5
print("The integer part is %d" % num)
```

will yield:

```
The integer part is 5
```

String formatting

Example: Consider the following code fragment

```
price = 119 # in dollars
discount = 30 # in percent %
print("A $%d jacket at %d%% discount is now
priced at %f" % (price,discount,price*0.7))
```

Produces the output:

```
A $119 jacket at 30% discount is now priced at
83.300000
```

String formatting

Example: Consider the following code fragment

```
price = 119 # in dollars
discount = 30 # in percent %
print("A $%d jacket at %d%% discount is now
priced at %f" % (price,discount,price*0.7))
```

Produces the output:

```
A $119 jacket at 30% discount is now priced at
83.300000
```

String formatting

Example: Consider the following code fragment

```
name1 = "Chris"
name2 = "John"
print("%s and %s are heading to the movies
tonight" % (name1,name2)
```

Produces the output:

```
Chris and John are heading to the movies
tonight
```

This OER material was produced as a result of the CS04ALL CUNY OER project.