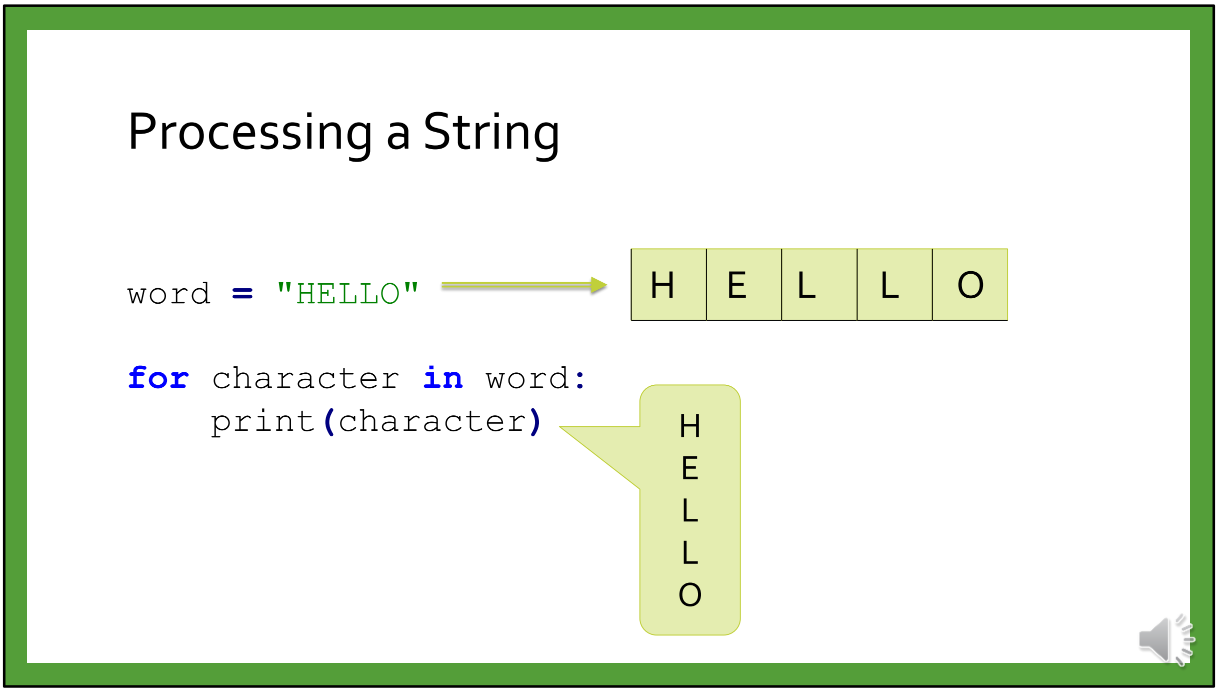
LISTS AND STRINGS

An Introduction to Computer Science

Let's learn more about Lists and Strings



Processing a String

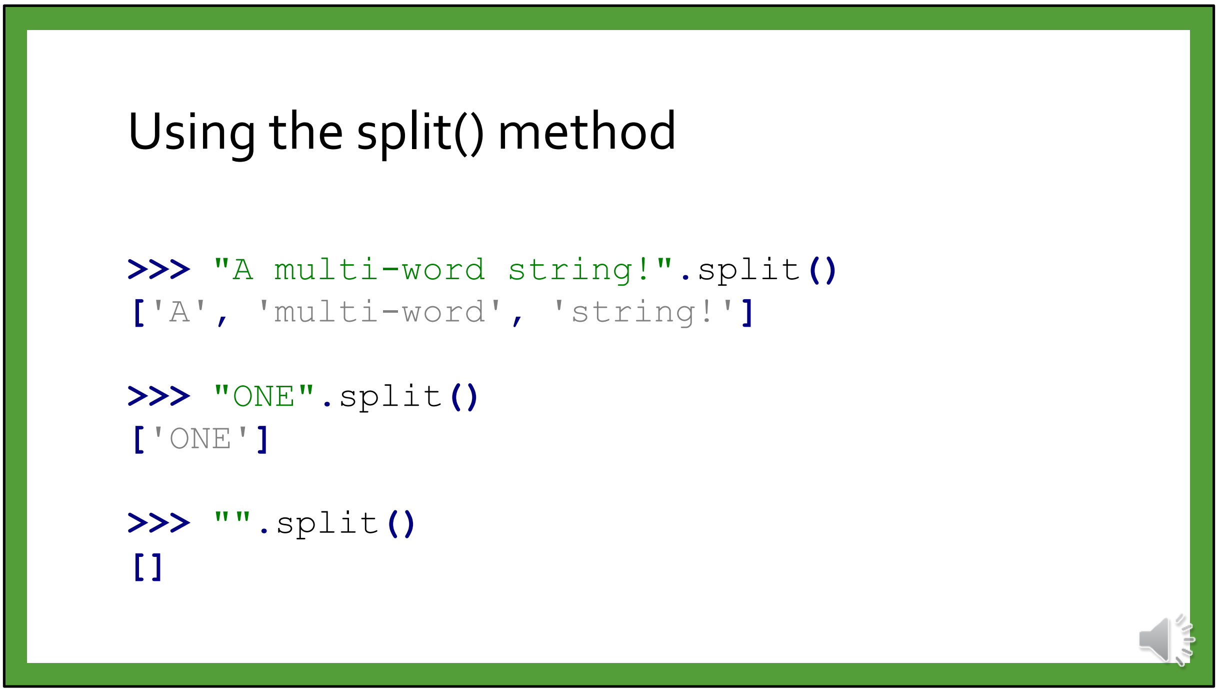
```
word = "HELLO"

for character in word:
    print(character)
```

H	E	L	L	O

H
E
L
L
O

Lists and Strings are somewhat similar, since they are both a sequence of things.
Strings are sequences of characters, but Lists can be a sequence of anything.
The key idea is that both Strings and Lists are sequences, which means that you can iterate over them with a FOR loop.
When you iterate over a list, you get each element.
When you iterate over a string, you get each character.



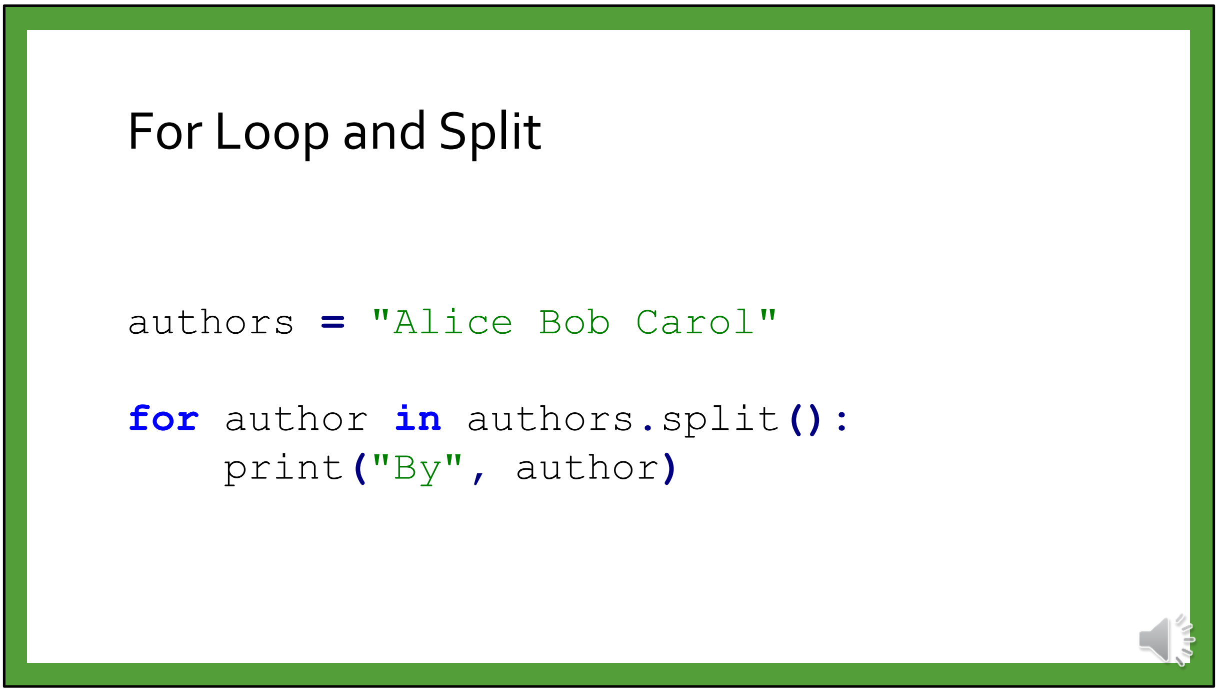
Using the split() method

```
>>> "A multi-word string!".split()
['A', 'multi-word', 'string!']

>>> "ONE".split()
['ONE']

>>> "".split()
[]
```

Often, instead of processing a list character-by-character, we want to process it word-by-word, or by some other chunking of characters.
To make this easy, strings have a method named split.
Split is an awesome method because it is easy to use.

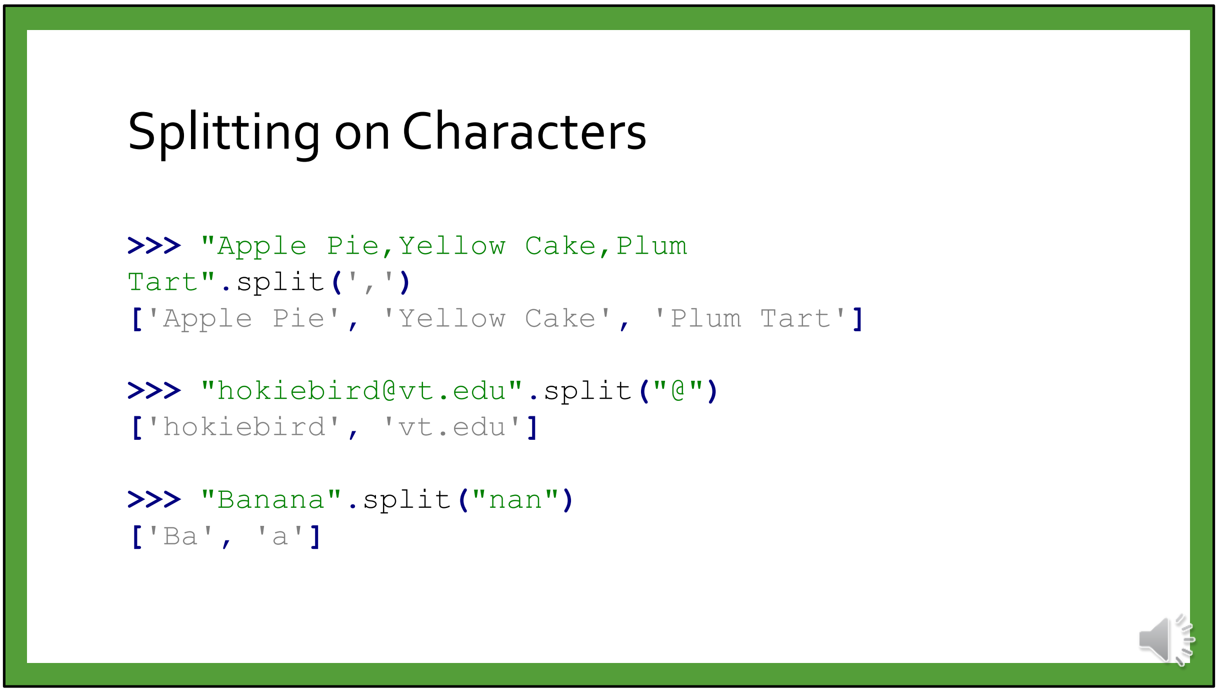


For Loop and Split

```
authors = "Alice Bob Carol"

for author in authors.split():
    print("By", author)
```

As you can see below, after we have split a string, it is easy to loop over each word.
In this example, we separate each author to print them individually.



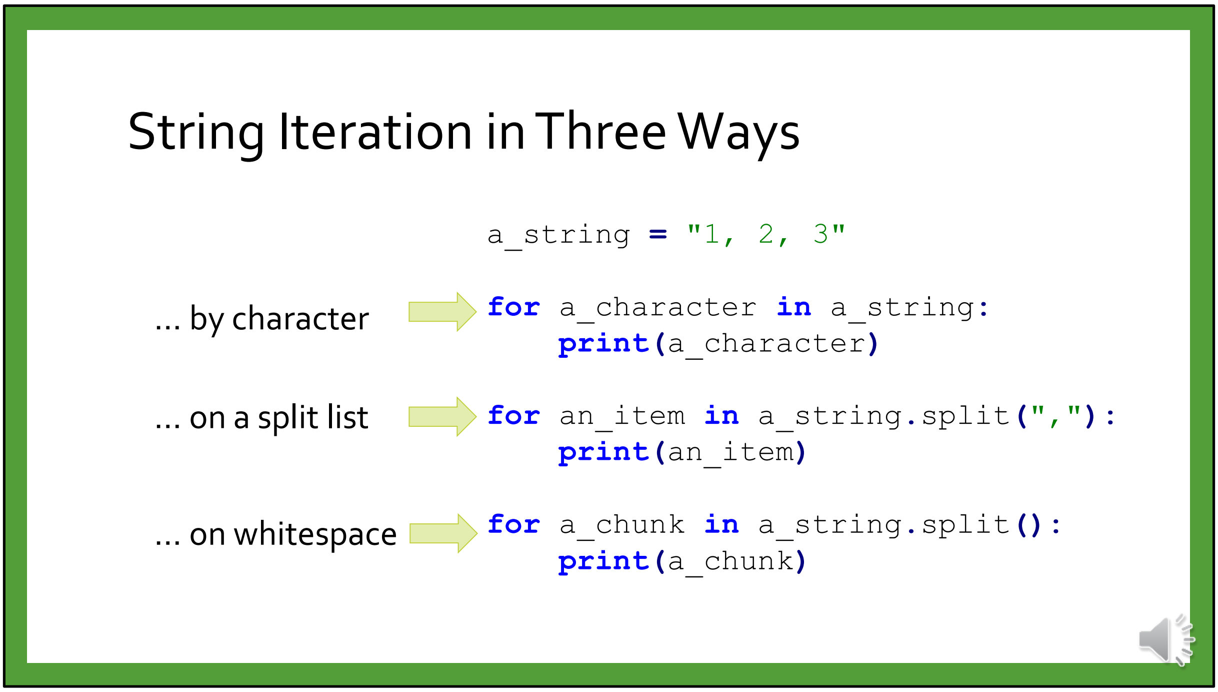
Splitting on Characters

```
>>> "Apple Pie,Yellow Cake,Plum
Tart".split(',')
['Apple Pie', 'Yellow Cake', 'Plum Tart']

>>> "hokiebird@vt.edu".split("@")
['hokiebird', 'vt.edu']

>>> "Banana".split("nan")
['Ba', 'a']
```

If you do not pass anything to split, then it splits on any kind of whitespace - spaces, tabs, new lines.
Sometimes, we want to split on other characters.
You can pass a string as an argument to split on a different character.



String Iteration in Three Ways

```
a_string = "1, 2, 3"
```

... by character

```
for a_character in a_string:
    print(a_character)
```

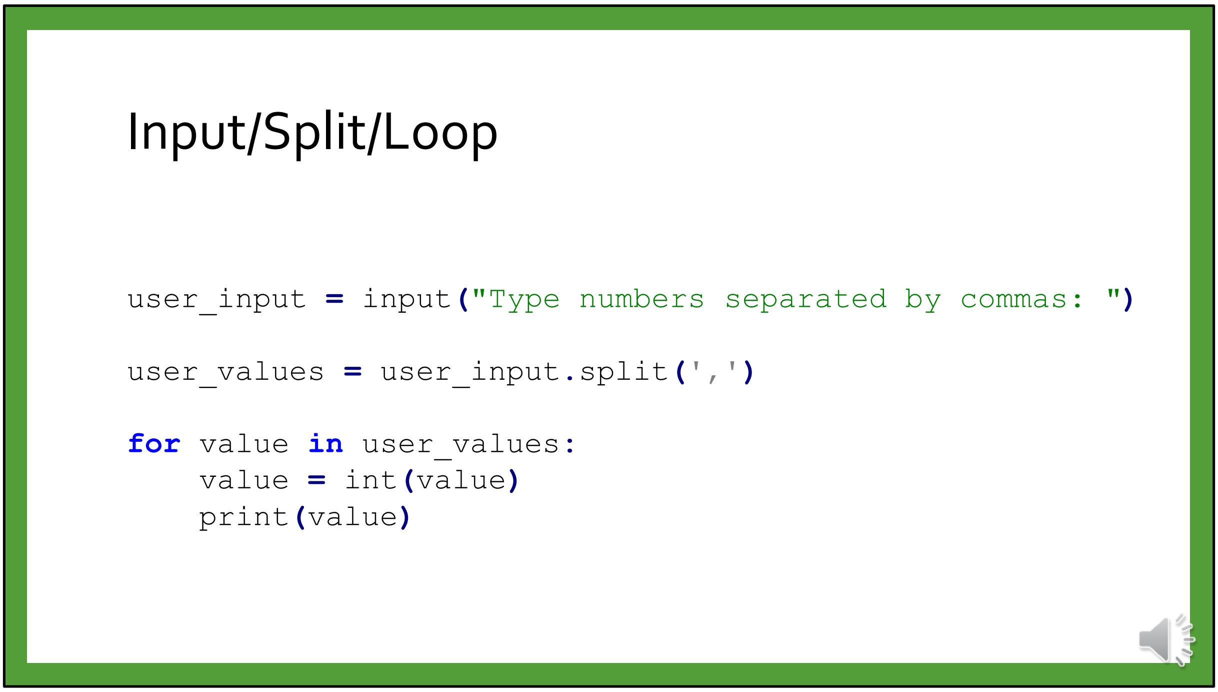
... on a split list

```
for an_item in a_string.split(","):
    print(an_item)
```

... on whitespace

```
for a_chunk in a_string.split():
    print(a_chunk)
```

Just to summarize briefly, there are three major ways to iterate over a string.
Without the split method, you iterate by character.
With a parameter in the split method, you iterate by splitting the string on the parameter.
And without a parameter in the split method, you iterate by splitting the string on whitespace.



Input/Split/Loop

```
user_input = input("Type numbers separated by commas: ")

user_values = user_input.split(',')

for value in user_values:
    value = int(value)
    print(value)
```

Here is a useful pattern.
You take a string separated by a specific character from the user, split the elements on that character, and process each component in turn.
Notice how we can use this to process a string of numbers separated with commas by converting them using the "int" function.
Study each statement of this pattern carefully, and make sure you understand it.