

Intro to Coding with Python—User Input

Dr. Ab Mosca (they/them)

Slides based off slides courtesy of Jordan Crouser (<https://jcrouser.github.io/>)

Reminder!

- First quiz is out today!
- Quizzes (and homeworks) are *week long* assignments; expect to spend 5-7 hours on them (this is standard for a college class)

Plan for Today

- Learn how to get user input
- Write a few programs

Recall: the
`print()`
function

- `print()` outputs information to the console ("the shell")
- Works on lots of different **data types** (strings, integers, floats, and many more!)
- When `print()` is called on ("passed") a **variable**, it outputs the **contents**

```
Python 3.8.2 (default, Feb 26 2020, 02:56:10)
```

```
> x = 3
```

```
> print(x)
```

Recall: the `print()` function

- `print()` outputs information to the console ("the shell")
- Works on lots of different **data types** (strings, integers, floats, and many more!)
- When `print()` is called on ("passed") a **variable**, it outputs the **contents**

```
Python 3.8.2 (default, Feb 26 2020, 02:56:10)
```

```
> x = 3
```

```
> print(x)
```

```
3
```

Practice

- Write a short program that prints the following:

```
#####
```

```
# CAIS117 #
```

```
#####
```

Discussion

What if we wanted to be able to
print a banner around ANY word?
What would we need?

Discussion

What if we wanted to be able to
print a banner around ANY word?
What would we need?

- The word
- To be able to count how many characters are in the word

Another
function:
`len()`

- `len()` takes in a string and gives back the string's length (number of characters, including spaces)
- Can be called on string **literals** (`"stuff in quotes"`) or on **variables** whose contents are strings
- Unlike `print()`, `len()` returns a value

```
Python 3.8.2 (default, Feb 26 2020, 02:56:10)  
> len("Jordan")
```

Another
function:
`len()`

- `len()` takes in a string and gives back the string's length (number of characters, including spaces)
- Can be called on string **literals** (`"stuff in quotes"`) or on **variables** whose contents are strings
- Unlike `print()`, `len()` returns a value

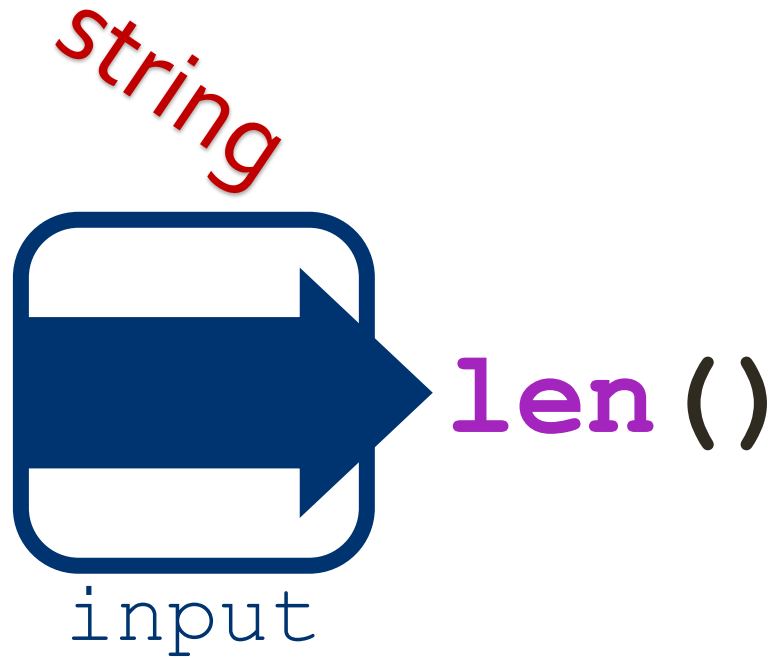
```
Python 3.8.2 (default, Feb 26 2020, 02:56:10)  
> len("Jordan")  
6
```

Another
function:
`len()`

- `len()` takes in a string and gives back the string's length (number of characters, including spaces)
- Can be called on string **literals** (`"stuff in quotes"`) or on **variables** whose contents are strings
- Unlike `print()`, `len()` returns a value

```
Python 3.8.2 (default, Feb 26 2020, 02:56:10)
> len("Jordan")
6
```

Back to the 4
basic tasks



Back to the 4
basic tasks



`len("hello")`
`> 5`



Back to the 4
basic tasks



```
len("hello")  
> 5
```



A red arrow points from the "output" diagram to the number 5 in the code snippet.

Recall: variables

- In CS, a **variable** is a place to store a piece of data
- In Python, variables are:
 - **declared** by giving them a name
 - **assigned** using the equals sign
- Example:

declaring a variable x

$x = 3$

assigning the value 3 to x

Recall: variables

- In CS, a **variable** is a place to store a piece of data
- In Python, variables are:
 - **declared** by giving them a name
 - **assigned** using the equals sign
- Example:

The diagram illustrates the components of the Python code `x = len("hello")`. The variable `x` is identified as the declaration of a variable. The expression `len("hello")` is identified as the assignment of the return value of the `len` function to the variable `x`.

```
x = len("hello")
```


declaring a variable `x`


assigning the return value to `x`

Recall: variables

- In CS, a **variable** is a place to store a piece of data
- In Python, variables are:
 - **declared** by giving them a name
 - **assigned** using the equals sign
- Example:

`myStrLen = len("hello")`

declaring
a variable 

 assigning
the return value

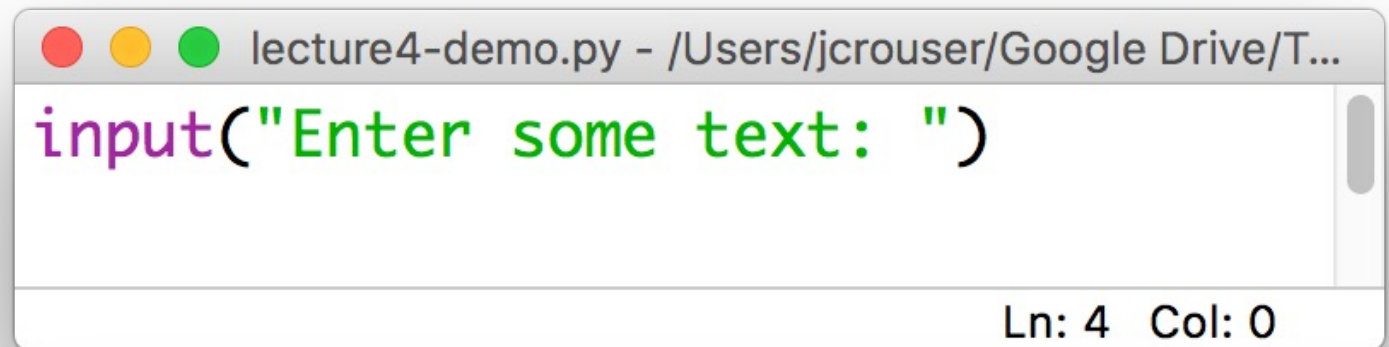
Discussion

What if we wanted to be able to
print a banner around ANY word?
What would we need?

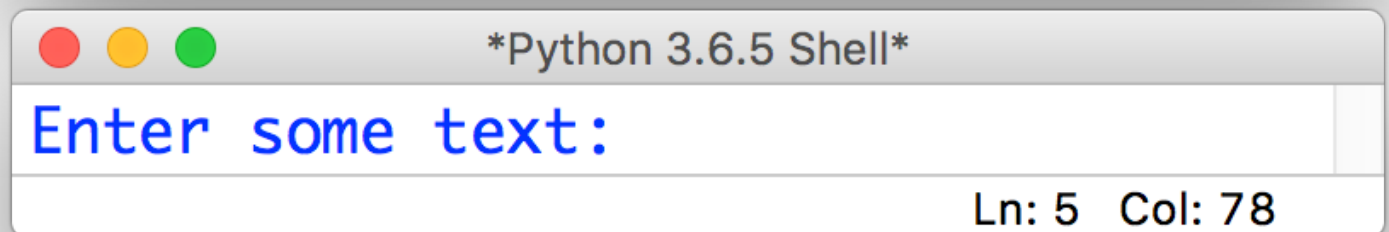
- The word
- ☒ • To be able to count how many characters are in the word (**and store answer**)

The `input()` function

- Python has a built-in `input()` function that allows us to ask the user to type in information
- The `input()` function takes in a value, which will be printed to the console as a prompt:



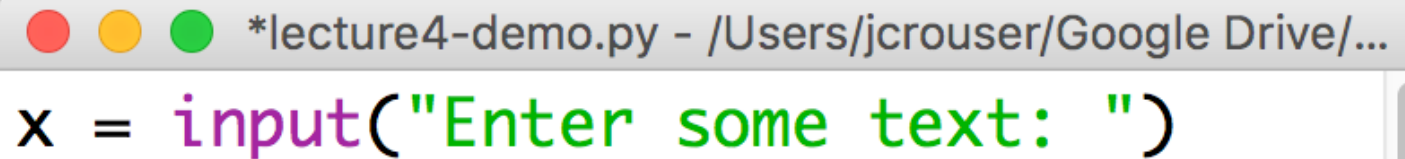
A screenshot of a code editor window titled "lecture4-demo.py - /Users/jcrouser/Google Drive/T...". The editor shows a single line of Python code: `input("Enter some text: ")`. The text "Enter some text: " is highlighted in green. The status bar at the bottom right indicates "Ln: 4 Col: 0".



A screenshot of a Python 3.6.5 Shell window titled "*Python 3.6.5 Shell*". The shell displays the prompt "Enter some text:" in blue text. The status bar at the bottom right indicates "Ln: 5 Col: 78".

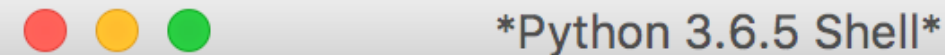
The `input()` function

- In general, we will want to **save** what the user enters so we can do something with it
- This means we need to **assign** the value **returned** by the `input()` function to some variable, e.g.



A screenshot of a code editor window titled '*lecture4-demo.py - /Users/jcrouser/Google Drive/...'. The code shown is `x = input("Enter some text: ")`. The `input` function is highlighted in purple, and the string argument is in green. A scrollbar is visible on the right side of the code area.

Ln: 4 Col: 0



A screenshot of a terminal window titled '*Python 3.6.5 Shell*'. It shows the prompt 'Enter some text:' in blue text, with the cursor positioned at the end of the line.

Enter some text:

Ln: 5 Col: 78

Discussion

What if we wanted to be able to
print a banner around ANY word?
What would we need?

- ✓ • The word
- ✓ • To be able to count how many characters are in the word (**and store answer**)
- Modify your program from earlier to work for any string

Note: The
`eval()`
function

- The user's input is always returned as a **string**, even if they enter only numeric characters
- If we want Python to interpret it as a number, we can use the `eval()` function



A screenshot of a code editor window titled '*lecture4-demo.py - /Users/jcrouser/Google Drive/...'. The editor contains the following Python code: `x = eval(input("Enter some text"))`. The code is color-coded: `x` is black, `=` is black, `eval` is purple, `(` is black, `input` is purple, `(` is black, `"Enter some text"` is green, `)` is black, and `)` is black. The status bar at the bottom right of the editor shows 'Ln: 3 Col: 0'.

- Then we can manipulate `x` using mathematical operations

Putting it all together

- In small groups, write a program that asks the user to **input ()** two strings:
 - a **word**
 - a **number**
- Store the user input in appropriate **variables** (remember: **eval ()** will return the *numeric value* of a *string*)
- **print ()** the **word** the user-specified **number** of times
 - Note: try multiplying a string by an int and see what happens
- Want a **challenge**? Also ask the user to input a **character** (a single letter or symbol), and use that to print a banner around the repeated word

Discussion

What did you come up with?