

User Input

Summary

 [Watch the video introduction before you begin.](#)

What am I learning about today?

- How do I receive, store, and print User Input?
- How do I use the Python input method?

When I complete this lesson:

- I can:
 - receive, store, and print User Input
 - use the Python input method

Materials You Will Need

- Your Laptop/Computer
- Chrome Browser
- Awesome Inc wifi - ask a coach if you need the password
- A curious mind and a little persistence!

Let's Get Started

Getting User Input

Our programs in Python have been pretty one-sided. We, the programmers, have coded programs that present information or data to our users, the people who run and view our programs. This is pretty limiting and a little boring. Our programs do not actually interact with our users. To make our programs more interesting, we need to add user input.

User input allows us to request and receive data from our users to use in our programs. To help us do this, Python has a built-in `input()` method.

The `input()` method prints a prompt and retrieves text from the user. When Python runs the `input()` method, it pauses our program until the user hits the return key. The `input()` method retrieves any text the user types.

To store the text, we need to create a variable for the `input()` method to pass the text into. The `input()` method prompts the user, receives the text when the user presses return, and passes it into the variable we create.

```
var_name = input("prompt")
```

You may type any text inside quotation marks to use for the prompt. Be sure to give your user clear directions about the data you would like them to enter.

```
user_name = input("What is your name? ")
```

Python will place a cursor on the screen at the end of your printed prompt. You may leave an extra space after your prompt inside quotation marks to improve readability.

When the user enters her text and hits return, the exact text she entered including any extra spaces, are passed into and stored in the variable that is on the left of the = assignment variable.

We can then use this newly created variable in our program.

```
print("Hi,", user_name, "It's nice to meet you.")
```

If I entered my name, Gina Beth, at the prompt, this would result in the following output to the screen:

```
Hi, Gina Beth, It's nice to meet you.
```

 [Let's review together before trying some practice exercises.](#)

Stop and Do:

Let's stop and practice. Complete the following tasks before moving on.

- ☐ Create a new program in repl.it
- ☐ Name the program user_input
- ☐ # Add a comment at the top of your program

```
# This program uses the Python input() method to  
get user input
```

- ☐ Use variables and the `input` method to ask the user for her name, her age, her favorite food, and her lucky number
- ☐ Use `print()` statements to print the user's data

User Input and Data Types

No matter what the user types at the prompt for the `input()` method, Python will ALWAYS pass in the data to our variable as a string.

```
age = input("How old are you? ")
```

For example, if I enter 21 at the prompt and hit enter after the statement above, Python will save the 21 value in the age variable as the string "21" not as an int 21 which I might expect. This is important to remember if we are planning to do any calculations with the values the user enters.

```
print(type(age))
```

Results in:

```
<class 'str'>
```

To get around this, we have two options, we can convert the value to a number using the Python methods `int()` or `float()` after we store it in our variable or before. Check out the examples of each below.

```
age = input("How old are you? ")
```

```
age = int(age)
```

Or

```
age = int(input("How old are you? "))
```

Now if we recheck the type of our age variable. It will be an int.

```
print(type(age))
```

Results in:

```
<class 'int'>
```

Python has a float method to convert user input to a float if you would expect the user to enter a value with a decimal point.

```
body_temp = float(input("What is your temperature? "))  
  
print(type(body_temp))
```

Results in:

```
<class 'float'>
```

Printing User Input

Printing user input is easy. Once the input is stored in a variable we can use the value in our `print` statements with other text and variables using concatenation with the `+` operator or with a comma.

Remember when you use the `+` operator between text and a variable, the variable must be a string. Python will give you an error if you try to use concatenation with a `string` and an `int` or a `string` and a `float`. If you need to include a variable that is a number in your `print` statement, wrap the number in the `str()` method inside your `print` statement.

```
print("I am " + str(age) + " years old.")
```

 Let's review these topics together.

[User Input is a String](#)

[Using `int\(\)` and `float\(\)` with User Input](#)

[Printing with User Input](#)

Stop and Do:

Let's stop and practice. Complete the following tasks before moving on.

- ☐ Open the starter repl.it program [chat_bot](#)
- ☐ Fork the program to make your own copy and rename the program `my_chat_bot`
- ☐ # Complete the comment at the top of your program explaining the Python `input` method

```
# I can use the Python input method to get user  
input by ... (finish this statement)
```

- ☐ Create a Chat Bot program that has a short conversation with your user.
- ☐ Follow the directions in the comments in the repl.it starter program

Are you stuck? No problem, ask your Coding Coach.

Check-In and Level Up

Check with a Coding Coach for the Challenges for this lesson.

