CSCI 141

Scott Wehrwein

Calling Functions
`print` and `input`

Goals

- Know the syntax used to to call a function with or without arguments.

- Understand the behavior of the `print` function with multiple arguments.

- Know how to use the `input` function with no arguments to pause a program.

- Know how to use the `input` function with an argument.

Function Calls

```python
print("Hello, world!")

print(3 * 4 + 2)

print("Hello", 3 * 4 + 2)
```

Each of these lines makes a call to the `print` function.

What exactly is a function? More on this later.

For now: it's a thing that does stuff for us.

Function Calls

```python
print("Hello, world!")

print(3 * 4 + 2)

print("Hello", 3 * 4 + 2)
```

Each of these lines makes a call to the `print` function.

What exactly is a function? More on this later.

For now: it's a thing that does stuff for us.
(We make it do stuff for us by calling it.)
Here's another example of a function call:

```python
input()
```

Let's see if we can figure out what stuff it does...

Demo: print and input

- the Shell pane in Thonny

- `print(3 * 4 + 2)`

- Print with multiple arguments

  - A space is printed between each

- input() to pause the program

- Input with a prompt

Function Calls

- We've seen two functions so far:

  - `print` and `input`

- Functions can take inputs, called arguments

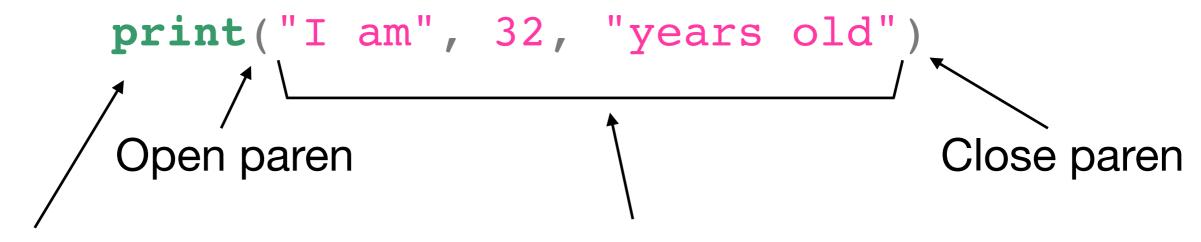 **`print`**`(``"some text"``)`

  `"some text"` is an argument to the `print` function call

- or not:

  `input``()`

  `input` is called with no arguments here

Function Calls

- Syntax for a function call:

```
print("I am", 32, "years old")
```

Function name

Open paren

Comma-separated list of arguments

Close paren

Function Calls

- Functions have rules about how many arguments they take, what they mean, and how the function behaves.

  - Example: print takes any number of arguments. It prints each of its arguments separated by spaces, followed by a line break.

  - Example: `input` takes zero or one argument.

Function Calls

- Functions have rules about how many arguments they take, what they mean, and how the function behaves.

- Need to know anything about a function? You can look it up in the documentation!

- My approach: google search for
  `python 3 <yourfunction> function`
  and pick the first result from python.org

- It may look intimidating now - don't panic!