



CSCI 141

Scott Wehrwein

Functions with inputs

Goals

- Know how to use parameters to refer to the input arguments in a function definition

Writing Functions: Syntax

```
def name(parameters):  
    statements
```

Two important questions:

- 1. How does the function use the arguments (inputs) passed to it?**
2. How does the function return a value?

Demo: Function to sum 2 numbers

Input(s):

- Two numbers, a and b



Return value:

- none

Effects: prints the sum of a and b to the screen

- define the function in a program, then test it on the Thonny shell
 - need to re-run the program to update the function definition if you've changed the code
- passing arguments:
 - pass raw values as inputs (e.g., `add2(4, 6)`)
 - pass variables as inputs (`add2(x, y)`)
 - pass expressions as inputs (`add2(x+y, 10)`)

Writing Functions: Syntax

1. How does the function use the arguments (inputs) passed to it?

def keyword

function name

def *name*(*parameters*):
statements

inputs

comma-separated
list of **parameters**:
variable names that
will refer to the input
arguments

Demo: Function to print a rectangle of a symbol passed in as an argument.

Input(s):

- width
- character

Return value:

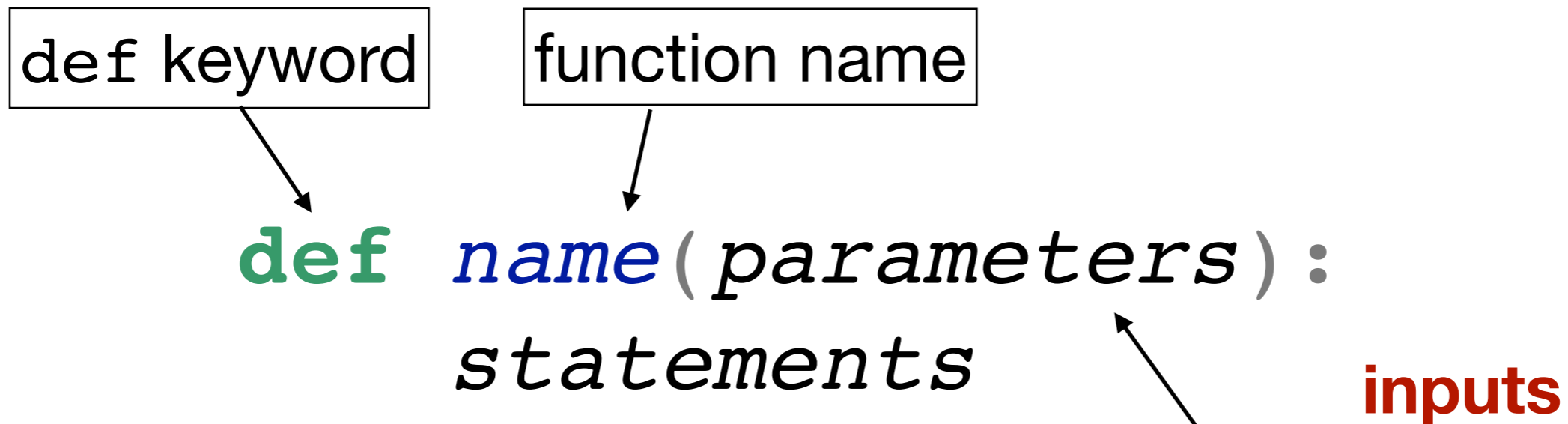
- none

→ `print_rectangle` →

Effects: prints a 2-x-width rectangle of characters to the screen

Writing Functions: Syntax

1. How does the function use the arguments (inputs) passed to it?



Inside the function, the parameters act as **local variables** that refer to the arguments passed into the function.

comma-separated list of **parameters**: variable names that will refer to the input arguments

Demo: Function to draw a square using a turtle

- `turtle_square_fn.py`
 - refactor `turtle_square` to use a function to draw the square
 - modify the function to take a side length