Name: _____ Date: _____

Python Functions: The Building Blocks of Code

Goals & Success Criteria

By the end of this worksheet, you will be able to:

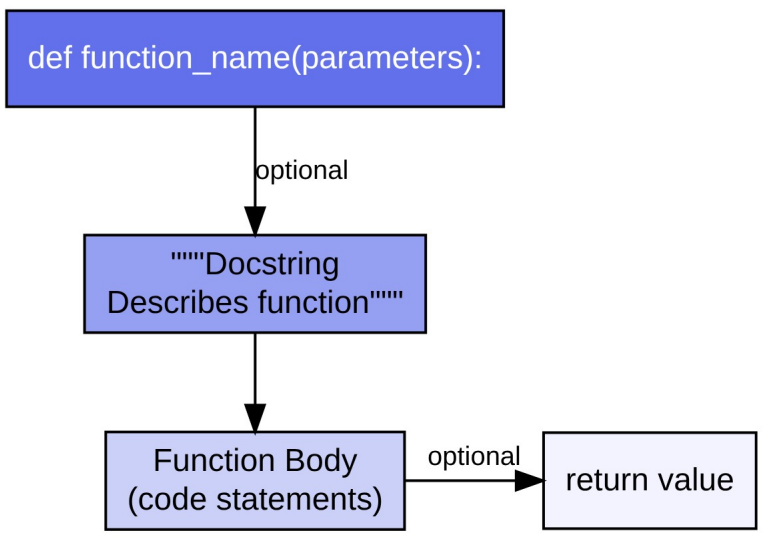
- Define and implement Python functions.
- Use parameters to pass data into functions.
- Use return values to get data out of functions.

Key Concepts

What is a function?
A function is a reusable block of code that performs a specific task. Functions help organize code, making it more readable, reusable, and easier to debug. You define a function using the def keyword.

Parameters vs. Arguments
Parameters are the variables listed inside the parentheses in the function definition. They are placeholders.
Arguments are the actual values you pass to the function when you call it.

Return Values
A function can send a value back to the code that called it using the return statement. This is useful for getting the result of a calculation or process.



Anatomy of a Python Function

Check Your Understanding

1. In your own words, what is a Python function and why is it useful?

2. Explain the difference between a parameter and an argument.

3. What is the purpose of the return statement in a function?

Guided Practice: Your First Function

Your goal is to create a function called **greet_user** that prints a welcome message. Follow the steps below to build and test your code.

Step 1: Worked Example

Here is the complete function. Notice the **def** keyword, the function name, parentheses **()**, a colon **:**, and the indented print statement.

```
def greet_user():
    print("Welcome to the world of Python functions!")
```

Step 2: Faded Practice

The function definition is started for you. Add the correct indented line of code inside the function to print the welcome message.

```
def greet_user():

```

Step 3: Independent Practice

Now, write the entire greet_user function from scratch in the box below.

Step 4: Test Your Function

After defining your function, you need to 'call' it to make it run. Add the line below to your code (after the function definition) to test it.

```
greet_user()
```

What should the output be? Write the exact text that will be printed in the box below.

Advanced Practice: Parametric Functions

1. Code Your Function

Implement a function called **greet_by_name** that accepts one input parameter (*name*) and **returns** a personalized greeting string.

2. Metacognitive Calibration

Predict the output for each function call, then run your code to verify. This helps calibrate your understanding of how parameters and return values work.

Function Call	Your Prediction	Actual Output
greet_by_name("Alice")		
greet_by_name("Bob")		

3. Spaced & Cumulative Review

Combine your new function with previous concepts. Write one line of code that calls **greet_by_name** with your own name and prints the result.

Function Challenges: Transfer & Debugging

Apply Your Skills: Far Transfer

Let's apply what you've learned about functions to solve two new problems. This is called 'far transfer'—using your skills in a completely new context.

1. Calculate Area

Write a function `calculate_area` that takes `length` and `width` as arguments and returns the area of a rectangle.

2. Check if Even

Write a function `is_even` that takes a `number` and returns `True` if it's even and `False` otherwise. (Hint: Use the modulo operator `%`)

Find the Bug! Error Analysis

Finding and fixing mistakes is a critical skill called debugging. Analyze the code below to find the error and suggest a fix.

```
def multiply_by_two(num):
    result = num * 2

print(multiply_by_two(5))
```

1. What does the code above print to the console, and why?

2. How would you fix the code? Rewrite the function correctly below.