



PYTHON IS FOR EVERYONE

Tutorial 6:

PYTHON PROGRAMMING - FUNCTIONS IN GOOGLE COLAB



Jeff Gentry

[.at www.linkedin.com/in/jefferycharlesgentry.](https://www.linkedin.com/in/jefferycharlesgentry)



Objectives

- Understand what functions are and why they are useful.
- Learn how to define and call functions in Python.
- Explore function parameters and return values.
- Practice writing and using functions.



What are Functions?

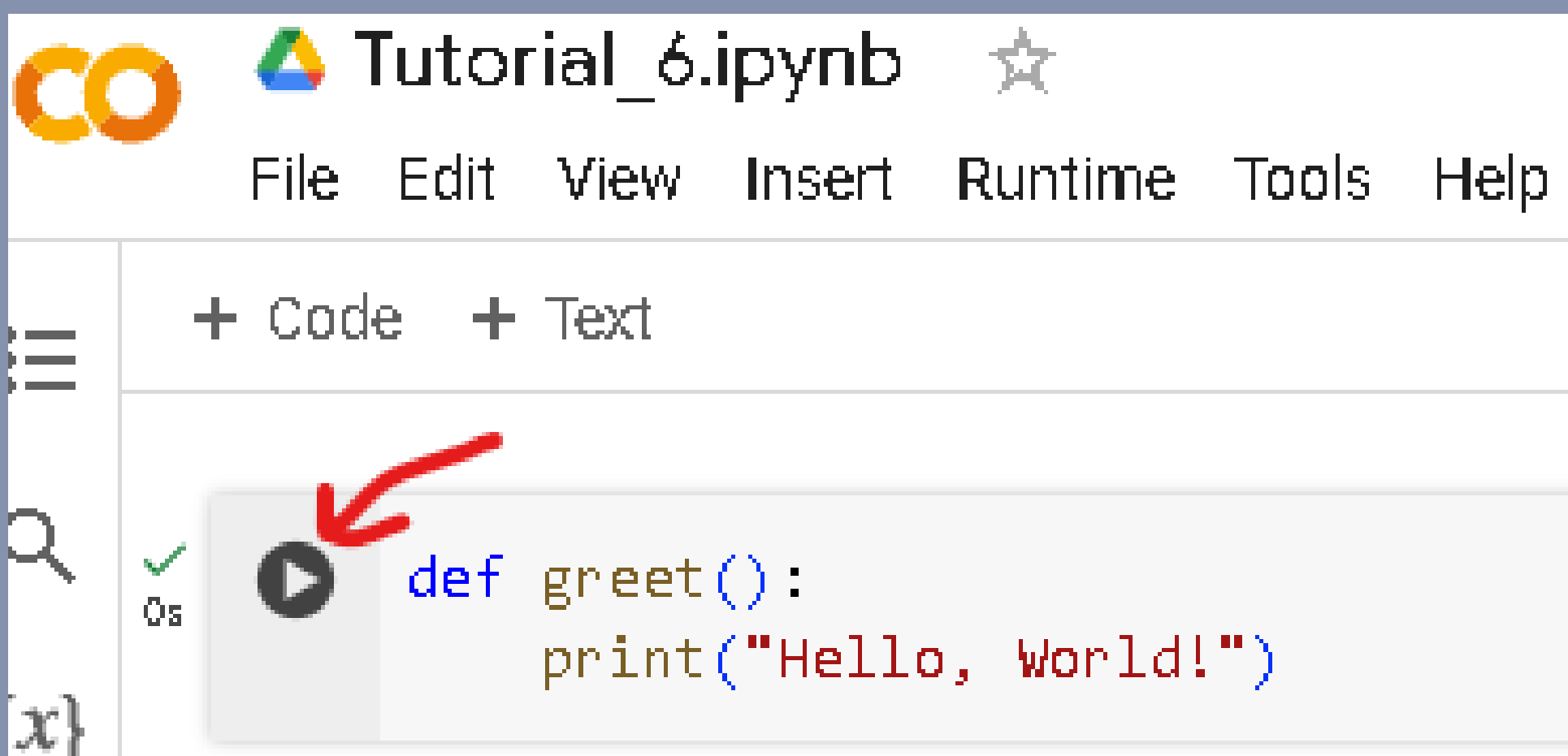


A function is a block of code that performs a specific task. Functions help you organize your code, make it more readable, and allow you to reuse code without rewriting it.



Defining a Function

You can define a function using the “def” keyword, followed by the function name and parentheses. The code block within the function is indented.

A screenshot of a Jupyter Notebook interface. The top bar shows the Colab logo, the file name 'Tutorial_6.ipynb', and a star icon. Below this is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. The main area has a toolbar with '+ Code' and '+ Text' buttons. Below the toolbar, there is a code cell. On the left of the code cell is a play button icon with a red arrow pointing to it, and a green checkmark with '0s' below it. The code cell contains the following Python code:

```
def greet():  
    print("Hello, World!")
```

It is important that you “run” this code. You will see no output, but it will be used shortly.

Calling a Function

To execute the code inside a function, you need to call it by its name followed by parentheses.

```
[2] def greet():  
    print("Hello, World!")
```

 `greet()` # Output: Hello, World!

 Hello, World!

The second code cell can execute because you have loaded what's in the first cell by running it previously.

Function Parameters

Functions can take parameters (also known as arguments) that allow you to pass data into the function.



```
def greet(name):  
    print("Hello, " + name + "!")  
  
greet("Alice")    # Output: Hello, Alice!  
greet("Bob")      # Output: Hello, Bob!
```



```
Hello, Alice!  
Hello, Bob!
```

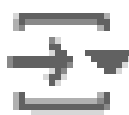
Return Values



Functions can return values using the return statement. This allows you to capture the output of a function.



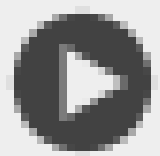
```
def add(a, b):  
    return a + b  
  
result = add(5, 3)  
print("Sum:", result)    # Output: Sum: 8
```



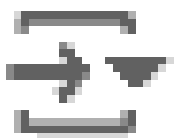
```
Sum: 8
```

Practice Exercises

Write a function called “square” that takes a number as a parameter and returns its square.



```
def square(num):  
    return num ** 2  
  
print(square(4))    # Output: 16
```



16

Practice Exercises

Write a function called “area_of_rectangle” that takes the length and width as parameters and returns the area.



```
def area_of_rectangle(length, width):  
    return length * width  
  
print(area_of_rectangle(5, 3)) # Output: 15
```



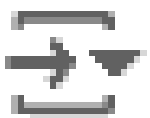
15

Practice Exercises

Write a function called `is_even` that takes a number as a parameter and returns “True” if the number is even, and “False” if it is odd.



```
def is_even(num):  
    return num % 2 == 0  
  
print(is_even(10))    # Output: True  
print(is_even(7))    # Output: False
```



```
True  
False
```

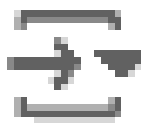
Practice Exercises



Write a function called `factorial` that takes a non-negative integer as a parameter and returns its factorial.



```
def factorial(n):  
    if n == 0:  
        return 1  
    else:  
        return n * factorial(n - 1)  
  
print(factorial(5))    # Output: 120
```



120



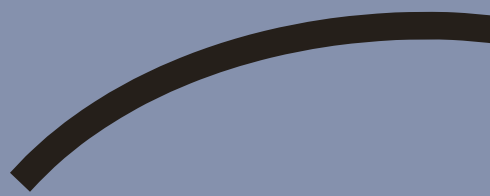
Conclusion

In this tutorial, you learned about functions in Python, including how to define and call them, use parameters, and return values. Functions are a powerful way to organize your code and make it more modular and reusable.



Next Steps

In tutorial 7, we will cover how to create and manipulate lists, which are one of the most versatile data structures in Python.



FOLLOW ME

for more tips you
didn't know you
needed



Jeff Gentry

[.@www.linkedin.com/in/jefferycharlesgentry](https://www.linkedin.com/in/jefferycharlesgentry)