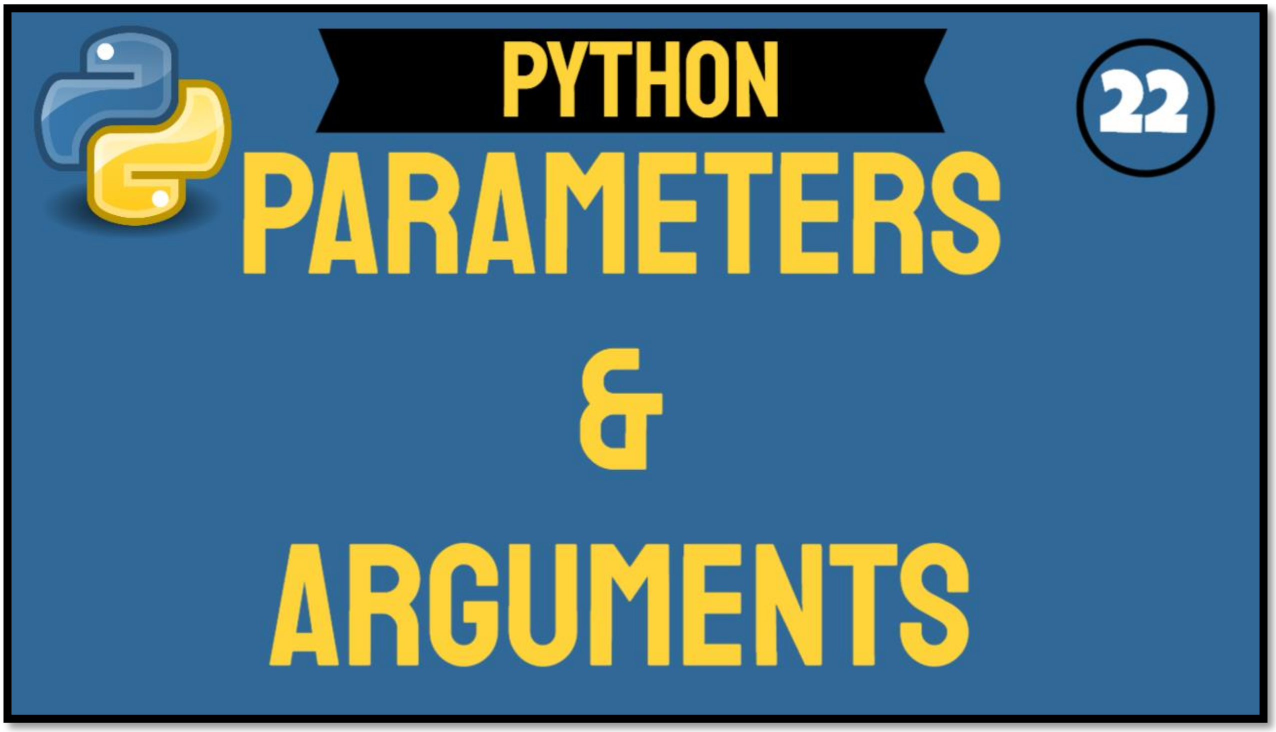
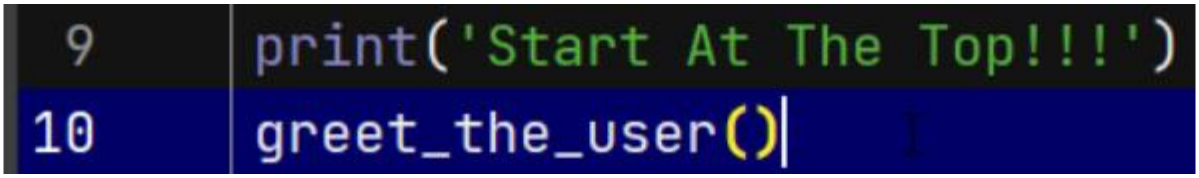
Python Parameters & Arguments



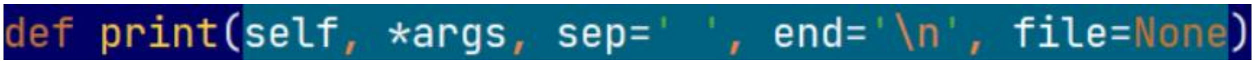
Python Video = https://youtu.be/3k1jmrVYyVk

In this session, we are going to talk about Python parameters and arguments. Sometimes an engineer may call a parameter an argument and call an argument a parameter. However, there's a difference between parameters and arguments. A parameter is a variable that stores information so a function can perform its job. An argument is also information but its passed to the parameter when calling the function. Let's use the same example from our last session. What is the difference between Python's built-in print() function our defined greet_the_user function?



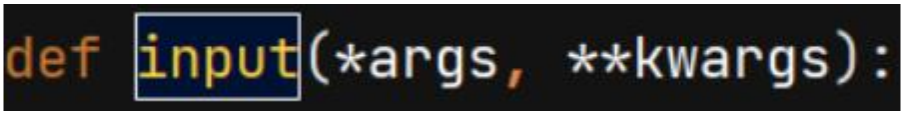
```
print('Start At The Top!!!')
greet_the_user()
```

The built-in print() function contains information and greet_the_user() does not contain information. Let's investigate the print() function. Do you see the values within the parenthesis?



```
def print(self, *args, sep=' ', end='\n', file=None)
```

Those are called parameters. We are going to create our own parameters. The information in green is the docstring comment. If you look at the filename, we see builtins.py because this complete file contains a list of built in Python functions. If I search for input(), we see it also has parameters.

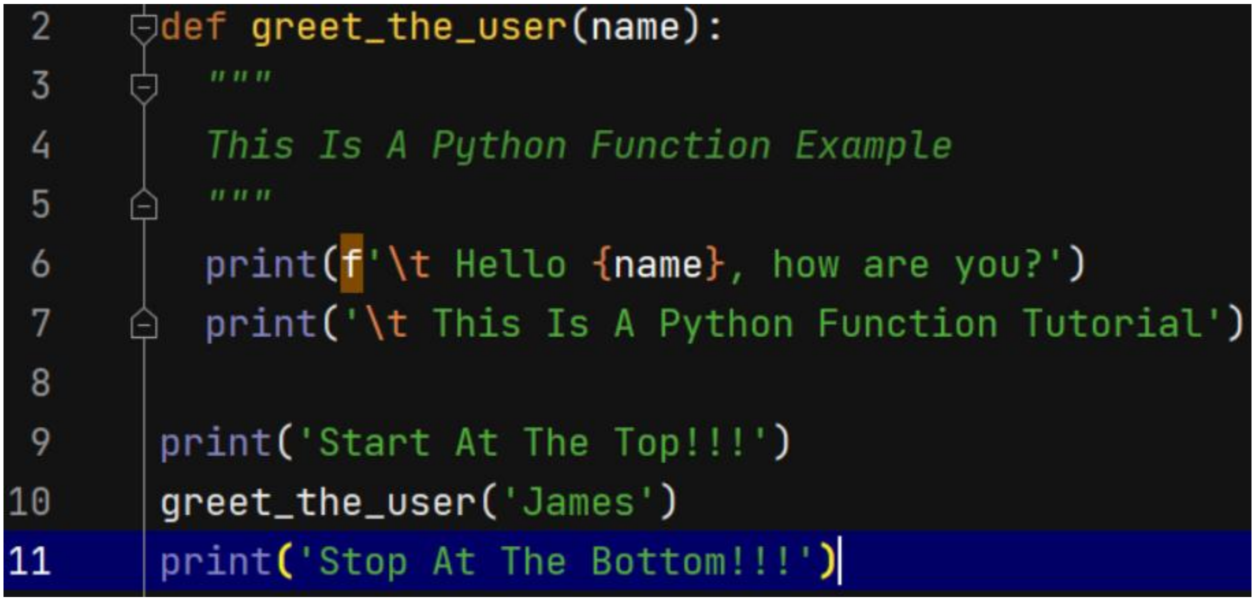


```
def input(*args, **kwargs):
```

Now, let go back to our file. We are going to create our own parameters and arguments. Notice, all of the print statements have data within the quotes. Hello, how are you, This Is A Python Function Tutorial, Start At The Top, Stop At The Bottom. These are the arguments that get passed to the print() function.

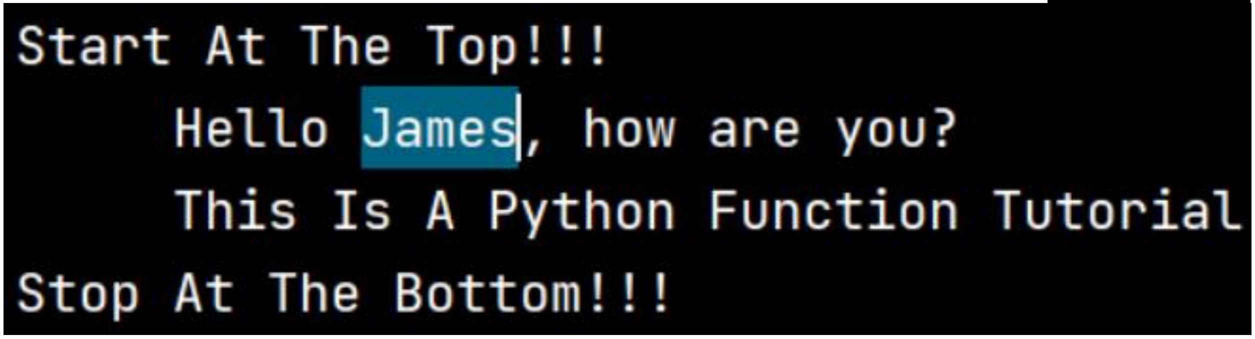
So, with our parameters. We are going to write name in between the parenthesis. The purpose of this name parameter is to operate as a placeholder so it can receive information. Recall an argument is passed to the parameter. We pass information to the parameter from our function call greet_the_user. 'James' will be the name we pass. After our function receives the name value, we can print James by formatting the name.

Place an 'f' before the quotes then the curly brackets {} before the comma. This how we format the name, we place name between each curly bracket. Did you notice that 'f' has a brown background. That's because this Python version does not support the prefix 'f'. It's okay, it's won't cause an error. Our print statement says Hello {name}, how are you?



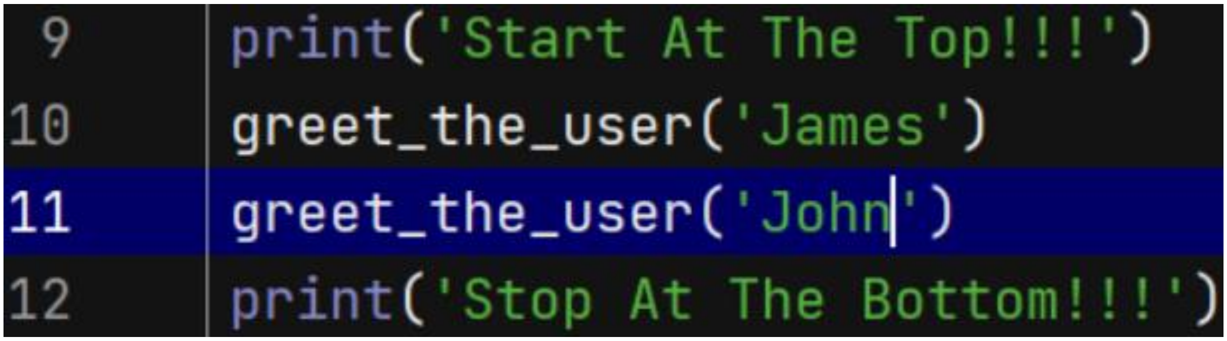
```
def greet_the_user(name):
    """
    This Is A Python Function Example
    """
    print(f'\t Hello {name}, how are you?')
    print('\t This Is A Python Function Tutorial')

print('Start At The Top!!!')
greet_the_user('James')
print('Stop At The Bottom!!!')
```

Let's Run and we see the name James was printed to the console.

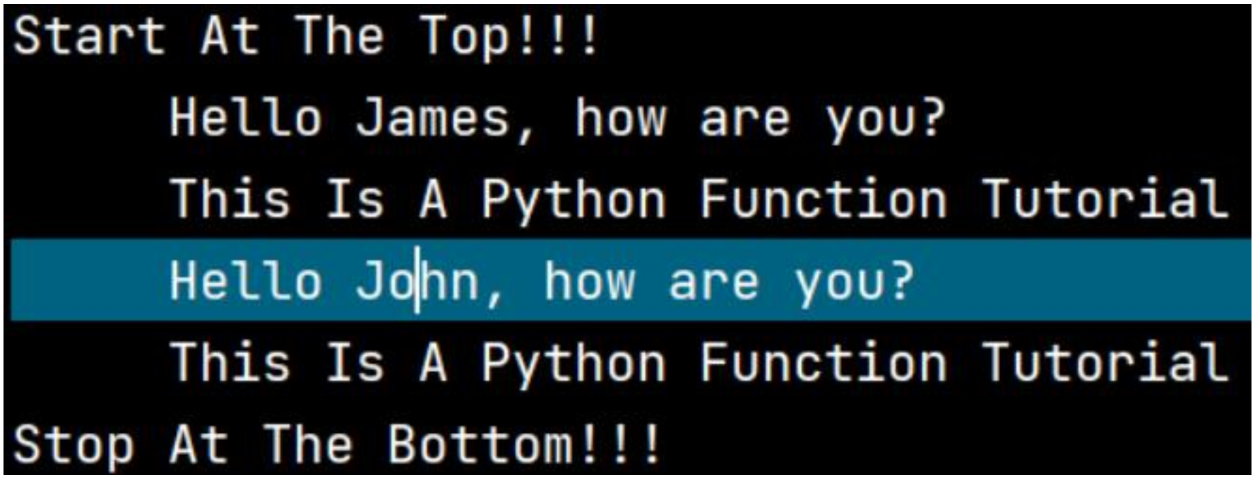


```
Start At The Top!!!
	Hello James, how are you?
	This Is A Python Function Tutorial
Stop At The Bottom!!!
```

You see we used the print statement 4 times. If we wanted to, we could also print greet_the_user more than 1 time and we see James and John.

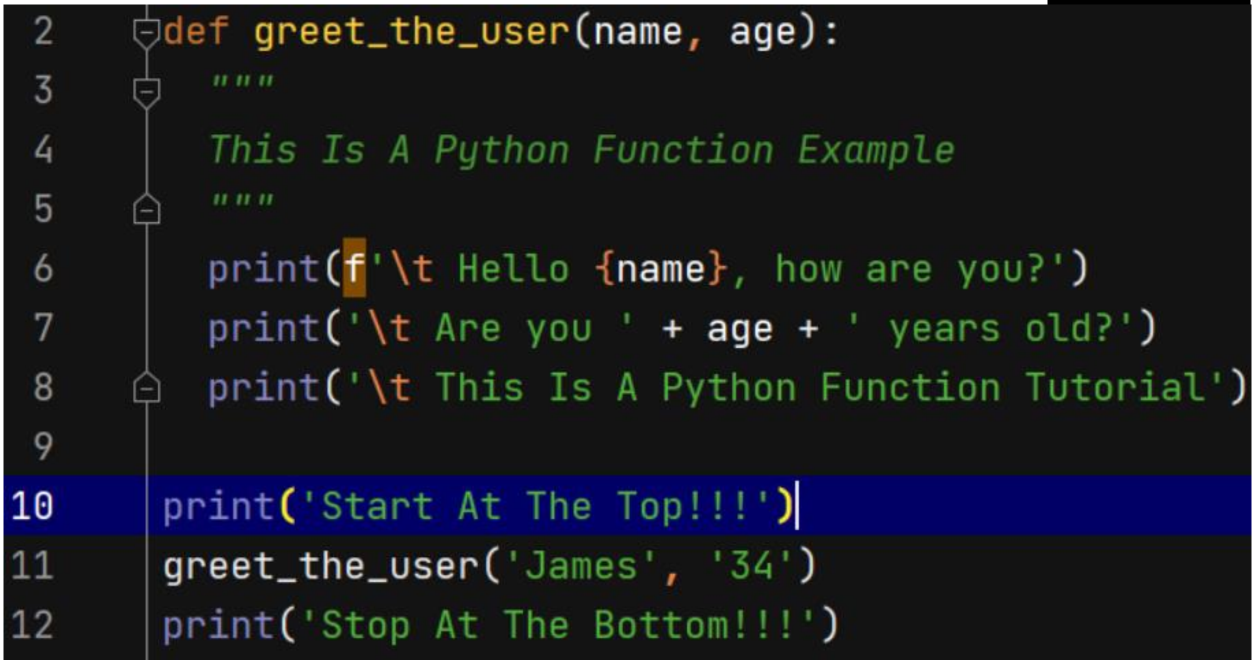


```python
print('Start At The Top!!!')
greet_the_user('James')
greet_the_user('John')
print('Stop At The Bottom!!!')
```



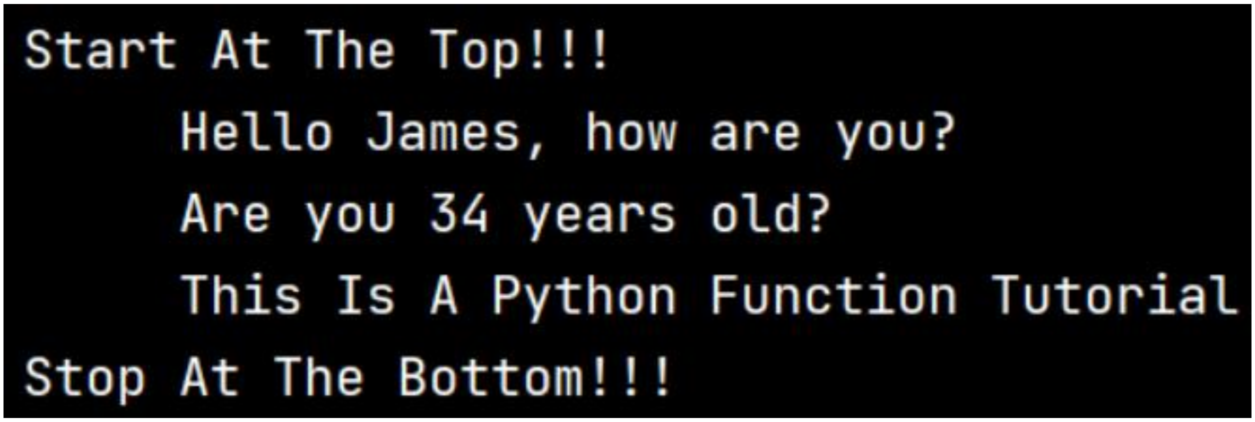
```
Start At The Top!!!
	Hello James, how are you?
	This Is A Python Function Tutorial
	Hello John, how are you?
	This Is A Python Function Tutorial
Stop At The Bottom!!!
```

I’m going to remove John. In the same way, how the print() function had more than 1 parameter. We can also add more than 1 parameter to the greet_the_user() function, by separating name with a comma and writing age as the other parameter. Let’s add one more print statement for age print() statement will “\t Are you “ + age + “ years old?” Let’s pass ‘34’ as the value.



```python
def greet_the_user(name, age):
    """
    This Is A Python Function Example
    """
    print(f'\t Hello {name}, how are you?')
    print('\t Are you ' + age + ' years old?')
    print('\t This Is A Python Function Tutorial')

print('Start At The Top!!!')
greet_the_user('James', '34')
print('Stop At The Bottom!!!')
```

Now, let’s run and we see both values in the console. James and 34.



```
Start At The Top!!!
	 Hello James, how are you?
	 Are you 34 years old?
	 This Is A Python Function Tutorial
Stop At The Bottom!!!
```

At this point, we can reuse this function over and over.

Contact Info

- ✔ Email Rex.Jones@Test4Success.org
- ✔ YouTube https://www.youtube.com/c/RexJonesII/videos
- ✔ Facebook https://facebook.com/JonesRexII
- ✔ Twitter https://twitter.com/RexJonesII
- ✔ GitHub https://github.com/RexJonesII/Free-Videos

- LinkedIn https://www.linkedin.com/in/rexjones34/