Python Lambda/Anonymous Function

In Python, a lambda function is a special type of function without the function name. For example,

```
lambda : print('Hello World')
```

Python lambda Function Declaration

We use the `lambda` keyword instead of `def` to create a lambda function. Here's the syntax to declare the lambda function:

```
lambda argument(s) : expression
```

Here,

- `argument(s)` - any value passed to the lambda function
- `expression` - expression is executed and returned

Let's see an example,

```
greet = lambda : print('Hello World')
```

```
# call the lambda
greet()
```

Example: Python lambda Function

```
# declare a lambda function
greet = lambda : print('Hello World')

# call lambda function
greet()

# Output: Hello World
```

Python lambda Function with an Argument

Similar to normal functions, the `lambda` function can also accept arguments.

```
# lambda that accepts one argument
greet_user = lambda name : print('Hey there,', name)

# lambda call
greet_user('Delilah')

# Output: Hey there, Delilah
```

In the above example, we have assigned a lambda function to the `greet_user` variable.

Here, `name` after the `lambda` keyword specifies that the lambda function accepts the argument named `name`.
Notice the call of lambda function,

```
greet_user('Delilah')
```

Here, we have passed a string value `'Delilah'` to our lambda function.

And finally, the statement inside the lambda function is executed.