

In python functions are 1st class citizens, meaning they can be passed as an argument to another function, they can be received as return value from a function

Now let say before printing nam we want to do something, and after printing name also we want to do something

So, for that we pass printName to a temp function

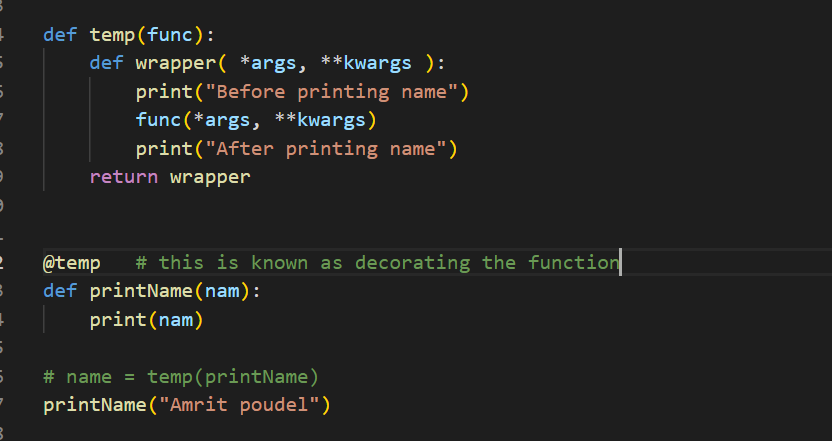
We printed “Before printing nam”

We then printed nam

We printed “After printing nam”

**Also here we used a wrapper function because we may need to pass argument like “Amrit poudel” in this case**

Now instead of doing all this we can simple **decorate our printName function** as:



We will get the same output

**Q) What if we need to do something before, after or both before and after the execution of certain function**

We can do that by decorating the function, without needing to modify the codes inside the function with the help of decorators

That was the point of creating decorators