

## What are Functions and Methods?

### Functions and Methods

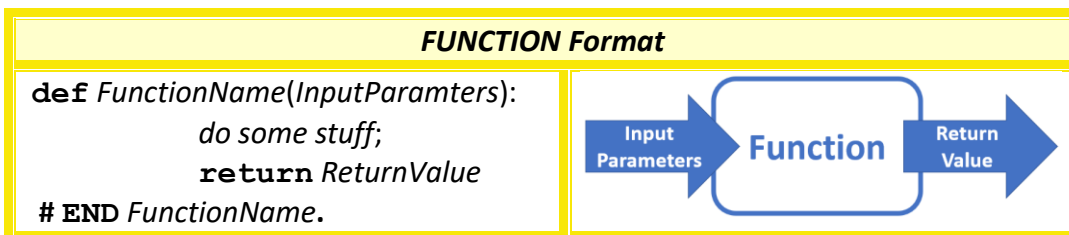
Imagine if we had a large Python program that has several sections of the code repeated throughout the program. It would be good if there was some way we could wrap up the frequently used commands into a single package, and instead of having to rewrite the same code over and over again, we could just call the package name instead. In Python we usually call these packages *functions*.

Functions are normally designed to accomplish a single, specific task, for example, check if a number is odd or even. In this case the function would be called with its package name and the number to be checked, and the function would return back to the main program whether the number is odd or even (as a Boolean).

Functions are sometimes called *Methods*, but only if they are defined as part of a larger structure called a *Class*. We won't be learning about classes in detail for the moment, but it is worth mentioning that classes are part of an approach to programming called *Object-Oriented design*, which focuses on defining the main functions of a program as general structures called *objects*.

```
print((range(1, 6)) + 30)
for a in range(1,6):
    # DO
    print(a)
    # ENDFOR;
a = int(input("Please input value:"))
b = a - 1
while b != 1:
    # DO
    if a % b == 0:
        # THEN
        IsPrime = False
        # ENDF;
        b = b - 1
    # ENDM;
    if IsPrime == True:
        # THEN
        print(a, "is a prime number")
    else:
        print(a, "is not a prime number")
    # ENDM;
for a in range(1,6):
    # DO
    print(a)
    # ENDFOR;
a = int(input("Please input value:"))
b = a - 1
IsPrime = True
a = int(input("Please input value:"))
Firstnum = 1
Secondnum = 1
while a != 1:
    # DO
    total = Secondnum + Firstnum
    Firstnum = Secondnum
    Secondnum = total
    a = a - 1
    # ENDM;
print(total)
# END.
```

The first thing we need to do is decide on a name for the function, use the **def** command with that name. Following the name are a pair of parenthesis (which may or may not include the names of values (parameters) that need to be included in the function). Next the full set of commands that are part of the function are included (and they are indented), followed by an optional **return** command, which lets the function pass a value back to the main program. We normally finish the function with a commented **END** command.



Functions are great because they allow us to reduce the amount of code in a program, and also allow us to focus on *what* the overall program is doing instead of concentrating on *how* the overall program is doing it (this is called *abstraction*).