

Procedures and functions

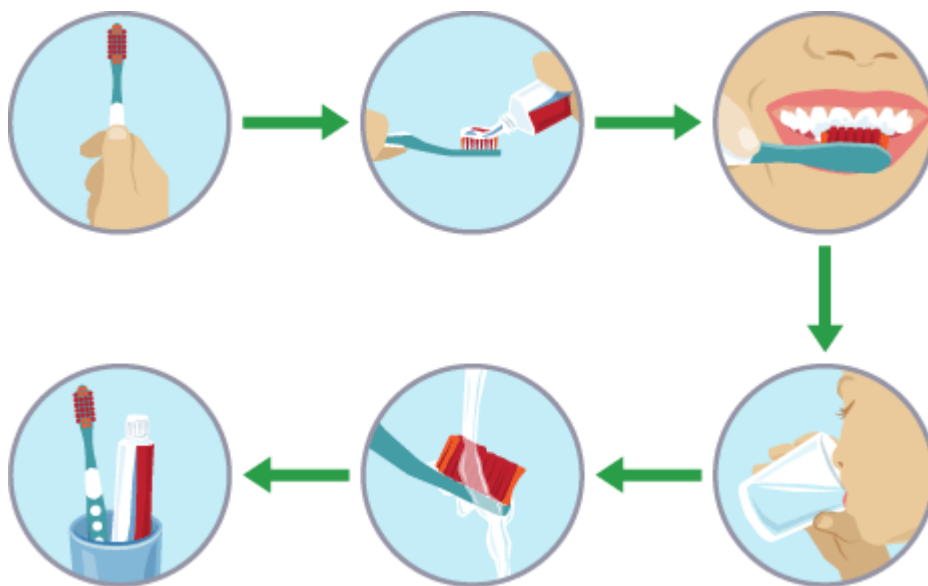
When writing programs, we should avoid long, repetitive code. Procedures and functions help to keep our programs simple and short.

What is a procedure?

Just like algorithms, programs are designed to solve a problem. In order to solve the problem, the program needs to perform several tasks. Code is written to perform each of these tasks.

A procedure is a small section of a program that performs a specific task. Procedures can be used repeatedly throughout a program.

A real-life example of a procedure is brushing your teeth. There are several steps involved in this process – getting your toothbrush, squeezing the toothpaste onto the brush, brushing all your teeth, rinsing your mouth out and putting everything back. Each of these steps isn't usually described, they just form part of the whole 'brushing teeth' procedure. The next time you need to brush your teeth, you remember the 'brushing teeth' procedure and know exactly what to do.



Why use procedures?

Program code is easier to read and understand when it is broken up into smaller sections. By breaking a program up into these sections, or procedures, code can be made shorter and simpler.

When writing programs it is very easy to end up with long-winded, repetitive code which is hard to understand and debug. This is because the same task often needs to be carried out in different places in the program. Procedures can be used throughout a program, making them simpler and quicker to code.

Using procedures has an added benefit. If something needs to be changed in a procedure, it only needs to be changed once, within the procedure code. This change

will then appear wherever the procedure is used in the program. If procedures weren't used, the code would have to be amended at every point that it appears in the program.

Procedures in Python

Procedures can make code shorter, simpler and easier to write.

Consider this excerpt from a Python game program which prints player information on the screen:

```
print("Your score: " + str(score))
time.sleep(1)
print("High score: " + str(high_score))
time.sleep(1)
print("Lives remaining: " + str(lives))
time.sleep(1)
```

The program uses six lines of code to print out the player information.

Suppose you wanted to print out the player information at these different points in the game:

- at the end of a level
- when the player loses a life
- when the player beats the high score
- when the game is over

You would need to repeat those six lines of code on each occasion, giving a total of 24 lines of code simply to display the player information:

```
# End of level
print("Your score: " + str(score))
time.sleep(1)
print("High score: " + str(high_score))
time.sleep(1)
print("Lives remaining: " + str(lives))
time.sleep(1)
```

```
# Lose a life
print("Your score: " + str(score))
```

```
time.sleep(1)
print("High score: " + str(high_score))
time.sleep(1)
print("Lives remaining: " + str(lives))
time.sleep(1)
```

```
# New high score
print("Your score: " + str(score))
time.sleep(1)
print("High score: " + str(high_score))
time.sleep(1)
print("Lives remaining: " + str(lives))
time.sleep(1)
```

```
# Game over
print("Your score: " + str(score))
time.sleep(1)
print("High score: " + str(high_score))
time.sleep(1)
print("Lives remaining: " + str(lives))
time.sleep(1)
```

This is repetitive and a waste of time. It would be better to write a procedure and simply run that procedure whenever it is needed.

Writing a procedure

Writing a procedure is extremely simple. Every procedure needs:

- a name
- the program code to perform the task

Writing a procedure in Python

Consider this excerpt from a Python game program which prints player information on the screen:

```
print("Your score: " + str(score))
time.sleep(1)
print("High score: " + str(high_score))
time.sleep(1)
print("Lives remaining: " + str(lives))
time.sleep(1)
```

To create a procedure, first give the procedure a name. A good name for the player information procedure could be **'update_display'**. Python uses the statement `def` to name a procedure. Note the brackets at the end of the procedure's name:

```
def update_display():
```

The procedure code is then written indented beneath the `def` statement:

```
def update_display():
    print("Your score: " + str(score))
    time.sleep(1)
    print("High score: " + str(high_score))
    time.sleep(1)
    print("Lives remaining: " + str(lives))
    time.sleep(1)
```

Indented code tells Python that the code belongs to the procedure.