# The os module

Insert the correct functions into the gaps.

The Python module □ is very useful for interactions with the operating system. In fact, it is a combination of two modules: os and □. Before any of them can be used, the modules need to be activated by the □ statement.
Among the most frequently used operations is the □ function, that returns a list of all files in the given directory. If a program already has a filename, but it needs to be checked whether the file really exists, the function □ will return **True** or **False**. If a file needs to be deleted, this can be done using □ .
A very useful feature of the **os.path** module is that it helps operating with directory names. Probably the most frequently used function is □, that separates a file name from directory names.
But **os** can do even more: You can use any shell command from a Python program with □ - However, this method has disadvantages: it depends on the operating system, and is a potentially insecure.

1 `os.access(fn,os.F_OK)`

2 `os.remove(filename)`

3 `os.path`

4 `os.listdir()`

5 `os.system(command)`

6 `os.path.split(os.getcwd())`

7 `import os`

8 `os`

K. Rother, A. Philips
& K. Milanowska