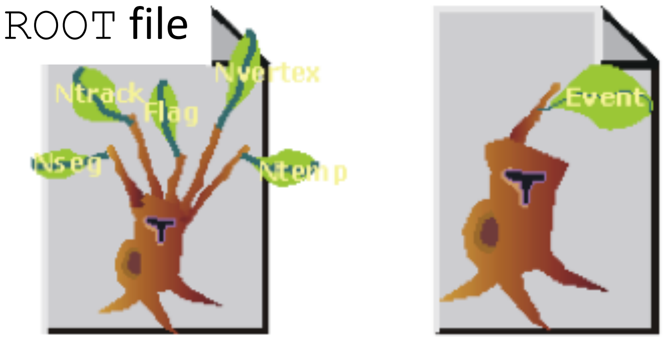
Some useful slides for the introduction to `Unix`

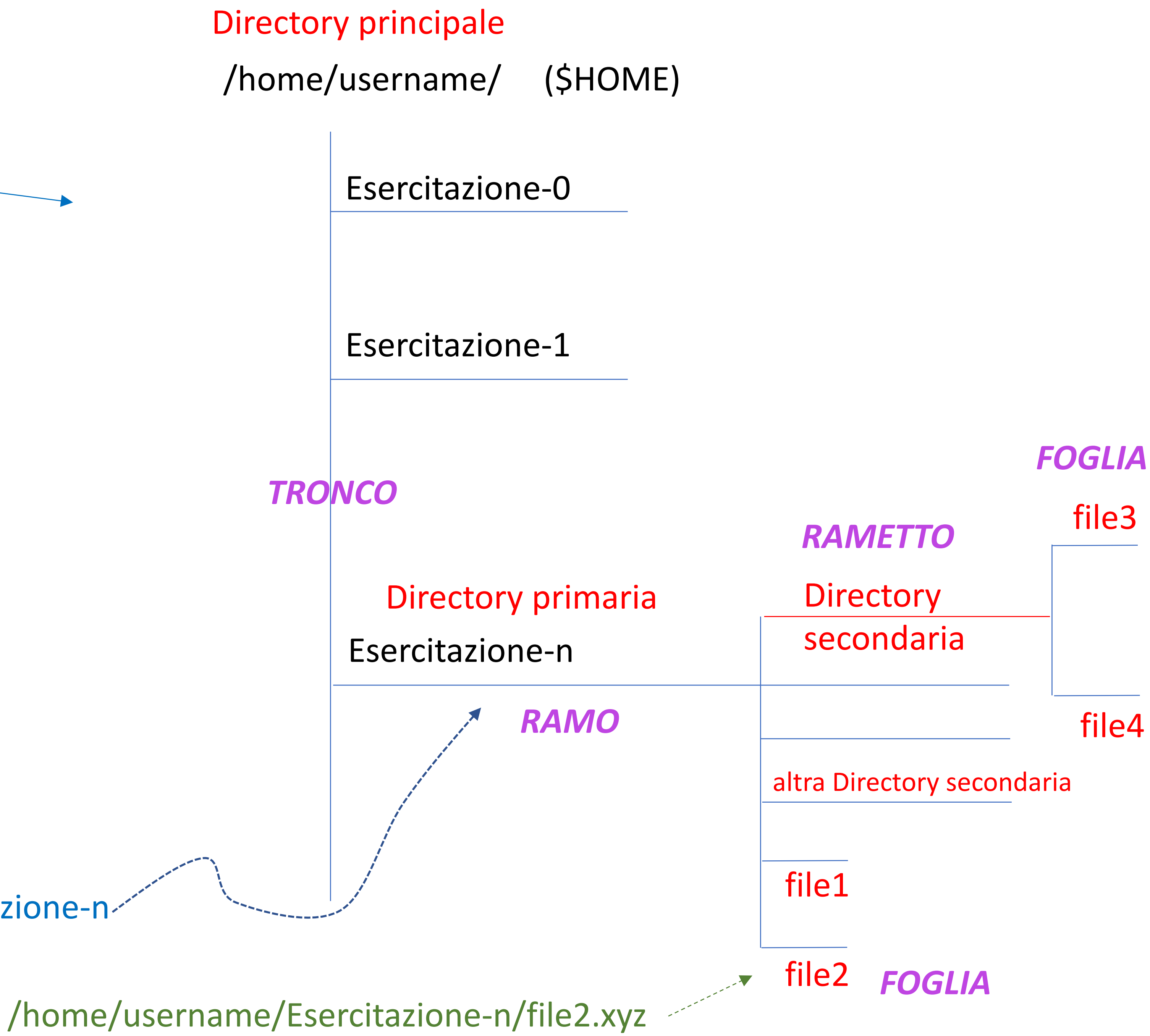
by **A.Pompili**

Tree structure - based:

- Unix filesystem
- **ROOT/TTree**

ROOT file





Directory principale

/home/username/ ($HOME)

Esercitazione-0

Esercitazione-1

TRONCO

Directory primaria

Esercitazione-n

RAMO

RAMETTO

Directory secondaria

altra Directory secondaria

file1

file2

FOGLIA

file3

file4

FOGLIA

/home/username/Esercitazione-n

/home/username/Esercitazione-n/file2.xyz

Some useful Unix commands

`find ./SDAL-2021 -name "*.root" -print`

`find ./SDAL-2021 -name "*.C" -print -exec grep -i 'yield' {} \;`

`df -h` (shows the structure of the file system included the muted devices)

`du -k --max-depth=2` (-k : kilobytes)

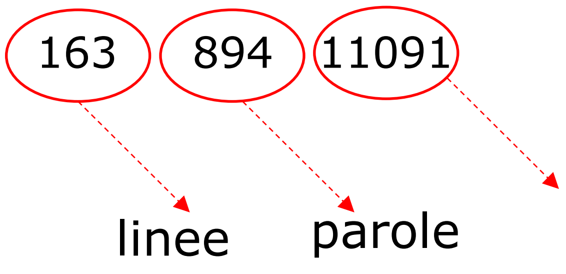
`du -hc --max-depth=2 /usr/` (-h : human readable, -c : display grand total; -hm in MegaBytes)

`tail -n20 history-5october2022.txt` (mostra le ultime 20 linee/righe di un file)

`sed "s/zanare/zanzare/g" file.txt > file-new.txt` (sostituisce e scrive in nuovo file)

Word count in un file (di testo):

`wc history-5october2022.txt` (opzioni: **`-l`** (solo le line), **`-c`** (solo i caratteri), **`-w`** (solo le parole))



```
163 894 11091
```

linee

parole

caratteri (contano gli spazi vuoti ed un carattere si aggiunge andando a capo)

(si puo anche fare: **`less file.txt | wc`**)

Some useful ways to use awk

```
$echo "Hello World" | awk '{$2 = "Universe"; print $0}'
```

Substitute 2nd word “World” with “Universe”

Pre- & post-processing of a text file (I use the commands BEGIN & END respectively):

```
$ less myfile.txt
Linux e' sistema operativo portabile di tipo proprietario per computer inizialmente
sviluppato da un gruppo di ricerca dei laboratori AT&T e Bell Laboratories
```

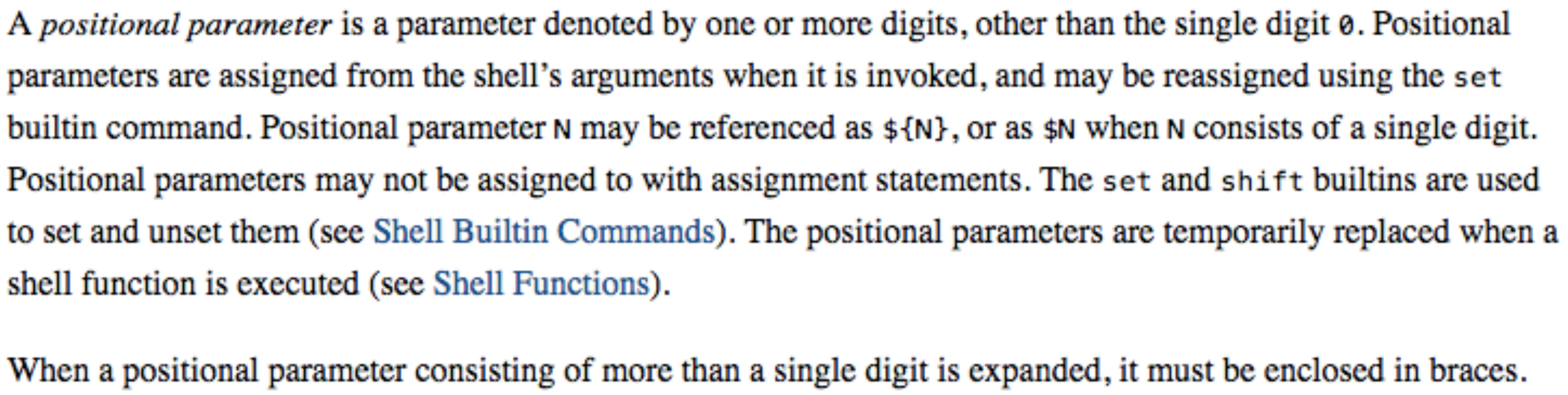
```
$ awk 'BEGIN {print “The file content starts now:”}; {print $0}; END {print “The file ends!"}' myfile.txt
```

```
The file content starts now:
Linux e' sistema operativo portabile di tipo proprietario per computer inizialmente
sviluppato da un gruppo di ricerca dei laboratori AT&T e Bell Laboratories
The file ends!
```

$0 is the executed script/file

```
$ awk '{if ($1 > 20) print $3}' myfile.txt
sistema
un
```

Nota: S1, $2, $3 are the so-called positional parameters (see next slide)

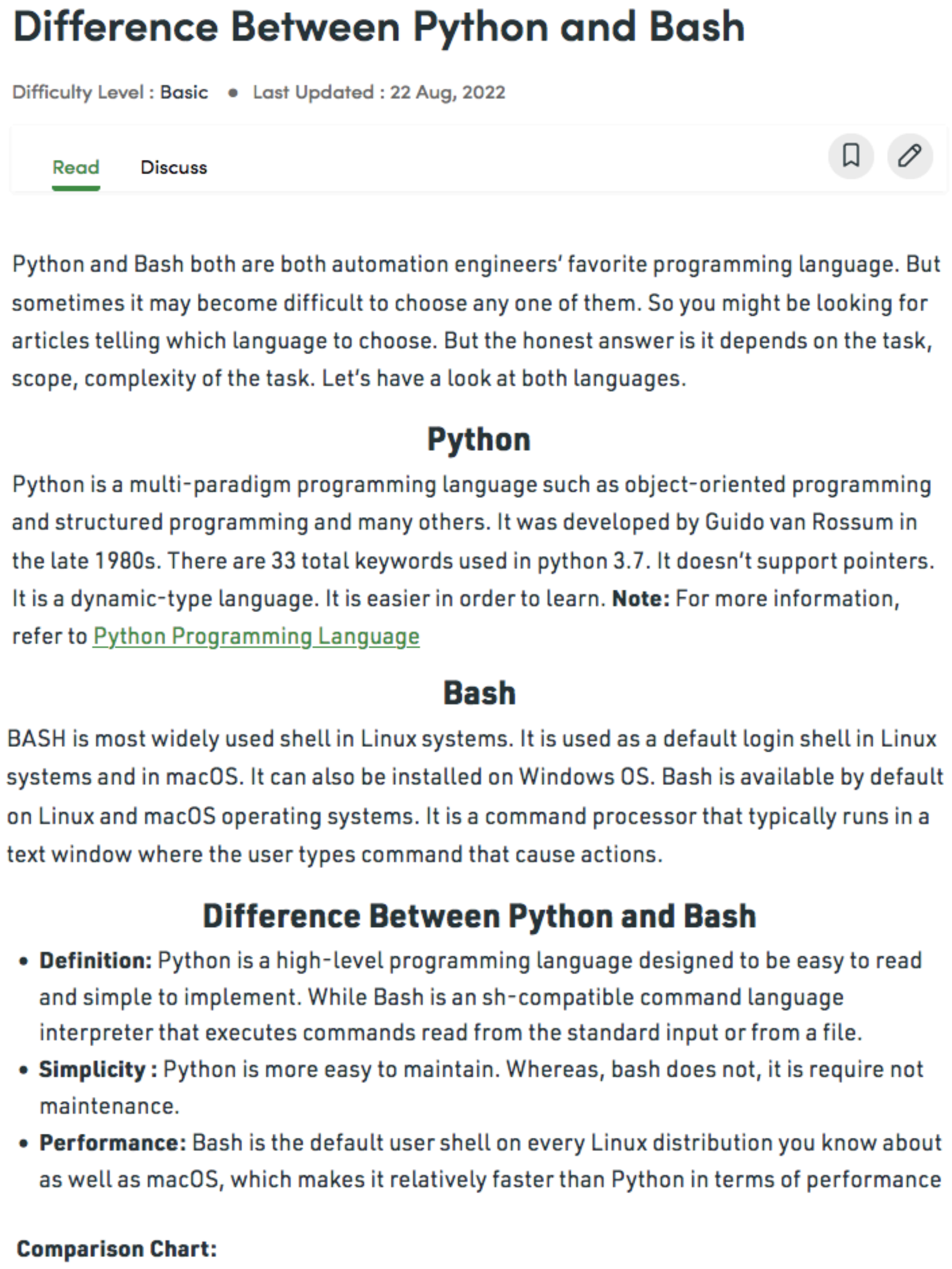


A *positional parameter* is a parameter denoted by one or more digits, other than the single digit `0`. Positional parameters are assigned from the shell's arguments when it is invoked, and may be reassigned using the `set` builtin command. Positional parameter `N` may be referenced as `${N}`, or as `$N` when `N` consists of a single digit. Positional parameters may not be assigned to with assignment statements. The `set` and `shift` builtins are used to set and unset them (see Shell Builtin Commands). The positional parameters are temporarily replaced when a shell function is executed (see Shell Functions).

When a positional parameter consisting of more than a single digit is expanded, it must be enclosed in braces.

In general the Unix shell scripting and python are "competitors"

(see next slide; from https://www.geeksforgeeks.org/difference-between-python-and-bash/)



Difference Between Python and Bash

Difficulty Level : Basic • Last Updated : 22 Aug, 2022

Python and Bash both are both automation engineers' favorite programming language. But sometimes it may become difficult to choose any one of them. So you might be looking for articles telling which language to choose. But the honest answer is it depends on the task, scope, complexity of the task. Let's have a look at both languages.

Python

Python is a multi-paradigm programming language such as object-oriented programming and structured programming and many others. It was developed by Guido van Rossum in the late 1980s. There are 33 total keywords used in python 3.7. It doesn't support pointers. It is a dynamic-type language. It is easier in order to learn. **Note:** For more information, refer to Python Programming Language

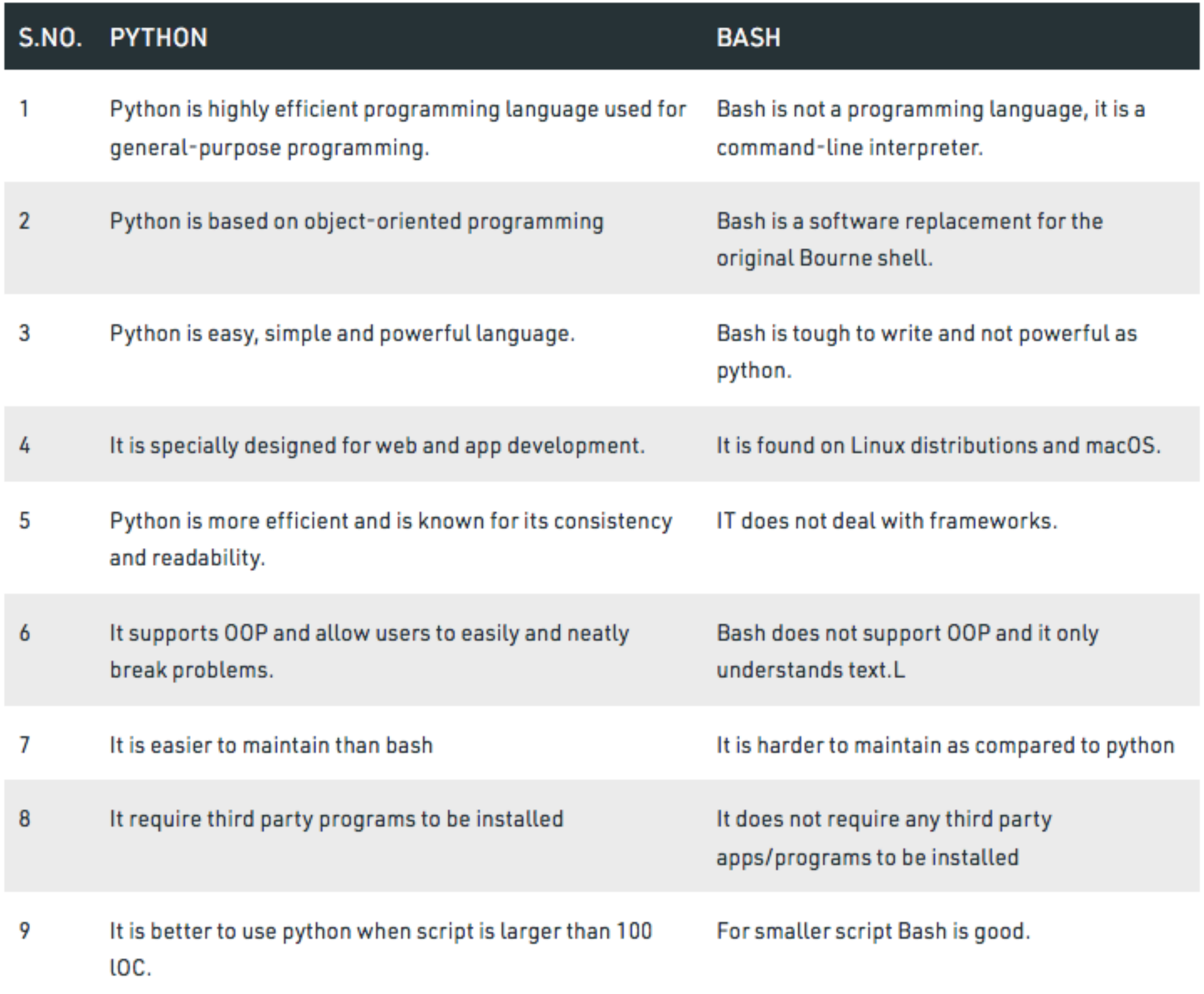
Bash

BASH is most widely used shell in Linux systems. It is used as a default login shell in Linux systems and in macOS. It can also be installed on Windows OS. Bash is available by default on Linux and macOS operating systems. It is a command processor that typically runs in a text window where the user types command that cause actions.

Difference Between Python and Bash

- **Definition:** Python is a high-level programming language designed to be easy to read and simple to implement. While Bash is an sh-compatible command language interpreter that executes commands read from the standard input or from a file.
- **Simplicity :** Python is more easy to maintain. Whereas, bash does not, it is require not maintenance.
- **Performance:** Bash is the default user shell on every Linux distribution you know about as well as macOS, which makes it relatively faster than Python in terms of performance

Comparison Chart:



S.NO.	PYTHON	BASH
1	Python is highly efficient programming language used for general-purpose programming.	Bash is not a programming language, it is a command-line interpreter.
2	Python is based on object-oriented programming	Bash is a software replacement for the original Bourne shell.
3	Python is easy, simple and powerful language.	Bash is tough to write and not powerful as python.
4	It is specially designed for web and app development.	It is found on Linux distributions and macOS.
5	Python is more efficient and is known for its consistency and readability.	IT does not deal with frameworks.
6	It supports OOP and allow users to easily and neatly break problems.	Bash does not support OOP and it only understands text.L
7	It is easier to maintain than bash	It is harder to maintain as compared to python
8	It require third party programs to be installed	It does not require any third party apps/programs to be installed
9	It is better to use python when script is larger than 100 lOC.	For smaller script Bash is good.

(*) per usare bash utilmente consultare: http://fmgroup.polito.it/quer/teaching/so/lucidi/u06-shell/u06s02-script.pdf