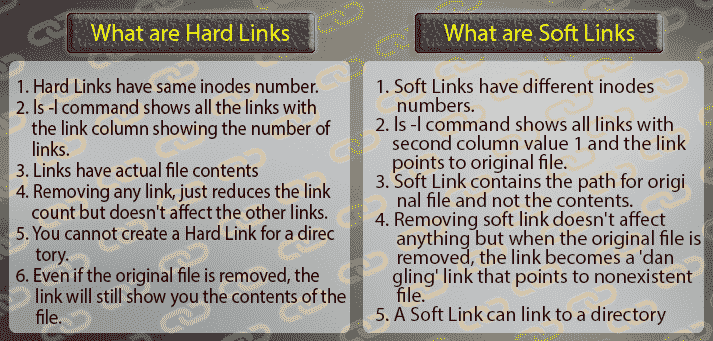
**Hard links** cannot link directories however **soft links** can



**Hard links**

$ mkdir Test

$ cd Test

$ touch sample1

create a hard link to sample1. Name the hard link sample2.

**$ ln sample1 sample2**

Display inodes for both files using i argument of the ls command.

**$ ls -il sample1 sample2**

output

**1482256** -rw-r--r-- 2 bruno bruno 21 May 5 15:55 sample1

**1482256** -rw-r--r-- 2 bruno bruno 21 May 5 15:55 sample2

Remove the original sample1

**$ rm sample1**

After removing Hard Link just have a look at the content of the "link" sample2.

**$ cat sample2**

You will still be able to see the contents of the file.

**Symbolic links**

create a soft link for the file sample2 using below command.

**$ ln -s sample2 sample3**

Display inodes for both using i argument of ls command.

**$ ls -il sample2 sample3**

output

**1482256** -rw-r--r-- 1 bruno bruno 21 May 5 15:55 FileB

**1482226** lrwxrwxrwx 1 bruno bruno 5 May 5 16:22 FileC -> FileB

From the output, you can notice that the **inodes are different and the symbolic link has an "l"** before the rwxrwxrwx.

The permissions are different for the link and the original file because it is just a symbolic link.

Now list the contents:

$ cat sample2

$ cat sample3

Now remove the original file

**$ rm sample2**

And then check the Test directory:

$ ls

It will still display symbolic link sample3 but if you try to list the contents,

it will tell you that there is no such file or directory.

**$ cat sample3**

Now you know about some of the key differences between Hard Links and Soft Links

that will make it easier for you to access files and run programs.