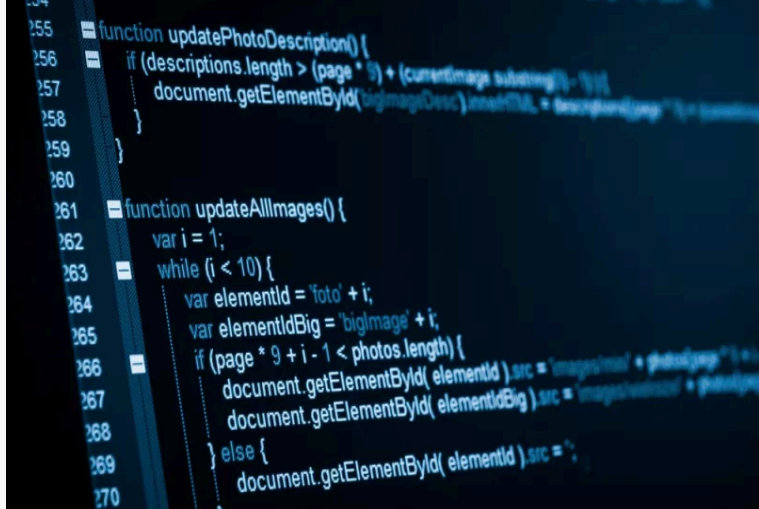




Introducción a la línea de comandos y a la programación para análisis bioinformáticos.



Teórico 7

Programación en el Shell I:
Manejo de variables,
salidas y entradas.
Integración con otros
lenguajes de programación

Qué es un script en bash?

- Cuando escribimos un *script* en lenguaje Bash le decimos al programa interpretador **bash** que hacer y como se debe hacer una o varias acciones.
- Todo lo que se hace en la línea de comando puede hacerse en un script en **bash** y viceversa.
- Si sabemos manejar la línea de comando podremos programar en **bash**.

Qué es un script en bash?

- Usualmente se le asigna la extensión “.sh” a los *scripts* en **bash**. El equivalente a “.pl” y “.py” para **perl** y **python**, respectivamente.
 - Yo utilizo “.scp”
- El mismo *script* puede correr en distintas terminales (o en diferentes instancias) en paralelo.
 - Hay que controlar la incompatibilidad de las corridas en paralelo.

Permiso de ejecución.

1. Permiso de ejecución a un archivo de texto con el código.
2. **chmod** es el comando que cambia los permisos de un archivo.
 - o Ej: **chmod** 755 myscript.sh

3. **4 códigos definen los permisos:**

- o "1": Puede ejecutar
- o "2": Puede escribir
- o "4": Puede leer

```
ls -l script.sh
```

4. **El número octal (la suma total) define el permiso:**

- o 3 (1+2): puede ejecutar y escribir
- o 6 (2+4): puede escribir y leer
- o 7 (1+2+4): puede ejecutar, leer y escribir.

5. **La posición del dígito en el valor total definen el o los usuarios:**

- o 1ra - Permisos para el dueño
- o 2da - Permisos para los usuarios del grupo del archivo
- o 3ra - Permisos para todos los usuarios del

script1.scp

Estructura general

1. `#!/bin/bash`
2. `# Un ejemplo muy muy sencillo.`
3. `echo "Hola mundo"`
4. `exit`

script3.scp

- 
1. *Shebang*
 2. Comentario
 3. El comando echo ejecuta la acción.
 4. Sale del script

“\$Variables”

1. Cuando nos referimos o leemos una variable ponemos un signo \$ antes del nombre de la variable.
2. Al establecer una variable no ponemos el signo \$.
3. Es costumbre escribir siempre nombres de variables en mayúsculas para que se destaquen. Aunque pueden ser mayúsculas, minúsculas o una mezcla.
4. Una variable puede colocarse en cualquier lugar de un script (o en la línea de comandos) y, cuando se ejecuta, Bash lo reemplazará por el valor de la variable.

Argumentos desde la línea de comando.

1. Se usan las variables \$1, \$2, \$3, para utilizar (pasar) los valores de los argumentos dentro del scripts.

script4.scp

Definir valores para una variable.

1. Variable_x=14
2. Variable_y=value
3. Variable_r="valor 54"
 - Tomar el resultado de un pipeline y ponerlo dentro de un resultado:
4. Variable_z=\$(ls *.fas | sed 's/.fas//g')

Entradas:

1. Modo interactivo, pedir al usuario por un valor de entrada

- **read variable_x**
- **echo \$variable_x**

script9.scp

2. Entrada estándar, incluir el script en un pipeline

- **STDIN** - /dev/stdin or /proc/self/fd/0
- **STDOUT** - /dev/stdout or /proc/self/fd/1
- **STDERR** - /dev/stderr or /proc/self/fd/2

script5.scp

Los scripts en otros lenguajes:

1. Los scripts en otros lenguajes se “llaman” o utilizan como se utilizarían en la línea de comando”
2. `./script.pl` **script8.scp**