Ejercicios básicos de scripting en Bash

---

0) Preparación

Crea un directorio de trabajo (por ejemplo `bash-practica`) y sitúate dentro. Salvo que se indique lo contrario, usa un *shebang* portátil en los scripts:

```
#!/usr/bin/env bash
```

---

1) “Hola Mundo” y permisos

1. Crea `mi_script.sh` con el *shebang* y una línea que imprima `¡Hola Mundo!`.
2. Dale permisos de ejecución y ejecútalo de **dos formas** distintas: con `./mi_script.sh` y con `bash mi_script.sh`.
3. Añade un comentario al principio explicando qué hace el script.

Entrega: el contenido del archivo y la salida en pantalla.

---

2) Variables sencillas

En `mi_script.sh`, añade:

- una variable `nombre="MiApp"` y otra `saludo="Hola"`
- un `echo` que muestre: `Hola MiApp` (usando expansión de variables).

Entrega: la salida exacta del script.

---

3) Argumentos por línea de comandos

Crea `argumentos.sh` que:

- Muestre el primer y segundo argumento (`$1` y `$2`).
- Muestre **todos** los argumentos con `$@`.
- Muestre el **nombre del script** con `$0`.

Ejecútalo pasando tres argumentos cualesquiera.

Entrega: el contenido de `argumentos.sh` y la salida al ejecutarlo con tres argumentos.

---

4) Entrada del usuario con `read`

Modifica `mi_script.sh` para que:

- Si **no** se pasa el primer argumento, pregunte al usuario su nombre con `read -p` y luego salude.
- Si **sí** hay primer argumento, lo use como nombre y salude sin pedir entrada.

Entrega: dos capturas de salida (sin y con argumento).

---

5) Arrays básicos

Crea `arrays.sh` que:

- Defina `mi_array=("valor 1" "valor 2" "valor 3" "valor 4")`.
- Imprima: el segundo elemento, el **último** elemento, **todos** los elementos y el **número total** de elementos.

Entrega: el contenido y la salida del script.

---

6) Slicing de arrays (Bash ≥ 4)

En `arrays.sh`, añade líneas que impriman:

- Los elementos desde el índice `1` **tres** posiciones.
- Los elementos desde el índice `2` hasta el final.

Entrega: las nuevas líneas y su salida.

---

7) Slicing de cadenas

Crea `cadenas.sh` con una variable `texto="¡Hola, Mundo!"` y muestra, usando slicing:

- `Mundo` (5 caracteres a partir del índice correcto).
- Los dos primeros caracteres del string.
- Desde el índice 7 hasta el final.

Entrega: el contenido y la salida del script.

---

8) (NUEVO) Clasificador de edad con `if/elif/else`

Crea `edad.sh` que pida una edad (entero) y muestre:

- Si es `< 0`: “Edad no válida”.
- Si es `< 18`: “Menor de edad”.
- Si es `< 65`: “Adulto”.
- En otro caso: “Mayor (65+)”.

Entrega: cuatro ejecuciones cubriendo cada rama.

---

9) (NUEVO) Comparador de dos números

Crea `comparar.sh` que lea **dos** números y muestre:

- "A es mayor que B", "A es menor que B" o "A es igual a B".
  Usa `if/elif/else` con `-gt`, `-lt` y `-eq`.

Entrega: tres ejecuciones (cada caso).

---

10) Pruebas de archivos

Crea `archivos.sh` que reciba una **ruta** como `$1` y:

- Indique si **existe** (`-e`).
- Si es **directorio** (`-d`), **archivo regular** (`-f`) o **enlace simbólico** (`-L` / `-h`).
- Si es **legible** (`-r`), **escribible** (`-w`) y **ejecutable** (`-x`).

Entrega: varias ejecuciones con rutas distintas para cubrir varios casos.

---

11) Comparaciones numéricas y `if/elif/else`

Crea `numeros.sh` que pida un número:

- Si es `> 0` imprima "positivo".
- Si es `< 0` imprima "negativo".
- Si es `0` imprima "cero".

Entrega: tres ejecuciones mostrando cada caso.

---

12) Operadores lógicos `&&` y `||`

Crea `usuarios.sh` que:

- Pida un nombre de usuario y lo guarde en `usuario`.
- Si `usuario` **es** `usuario_admin`, muestre "¡Eres el usuario administrador!".
- Si `usuario` **no** es `usuario_admin` **y** el `EUID` **no** es 0, muestre "No eres admin ni root, ¡ten cuidado!".
- En otro caso, muestre que es admin o root.

Entrega: tres ejecuciones cubriendo cada rama.

---

13) No ejecutar como root

Crea `no_root.sh` que **salga** si `EUID == 0` mostrando: "Por favor no ejecutes como root".
(Colócalo al inicio del script).

Entrega: una ejecución como usuario normal y una demostración (texto) de qué ocurriría si fuese root.

---

14) Estado de salida (`$?`)

Crea `estado.sh` que ejecute un comando que falle (por ejemplo `ls /noexiste`) y, si el estado de salida es **mayor que 0**, muestre “Hubo errores”. En caso contrario, “Comando OK”.

Entrega: salida del script.

---

15) Sentencia `case`

Crea `frutas.sh` que pida una fruta y, usando `case`, muestre un mensaje distinto para:

- `Manzana`
- `Banana` o `Limon` (mismo caso)
- `Naranja`
- Cualquier otra (caso por defecto)

Entrega: al menos cuatro ejecuciones (una por caso).

---

16) Comentarios útiles

Añade comentarios descriptivos a **todos** los scripts creados, explicando entradas, salidas y casos contemplados.

Entrega: ejemplo de comentarios añadidos en dos scripts.

---

17) Extra (opcional): nombre del propio script

Crea `self.sh` que imprima “El nombre del archivo es: ”. **(No borres el propio script).**

Entrega: salida del script.