



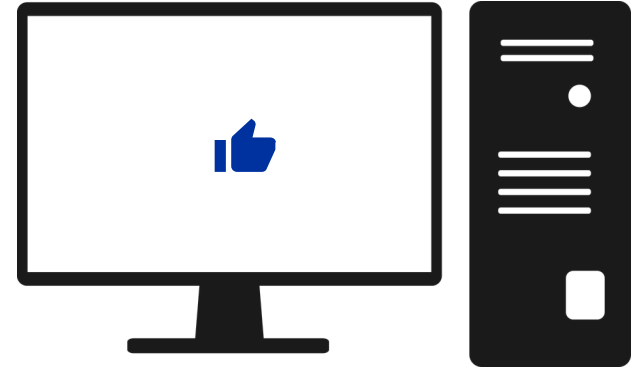
1. Hello World
2. Variables
3. Constantes

Introducción a Bash

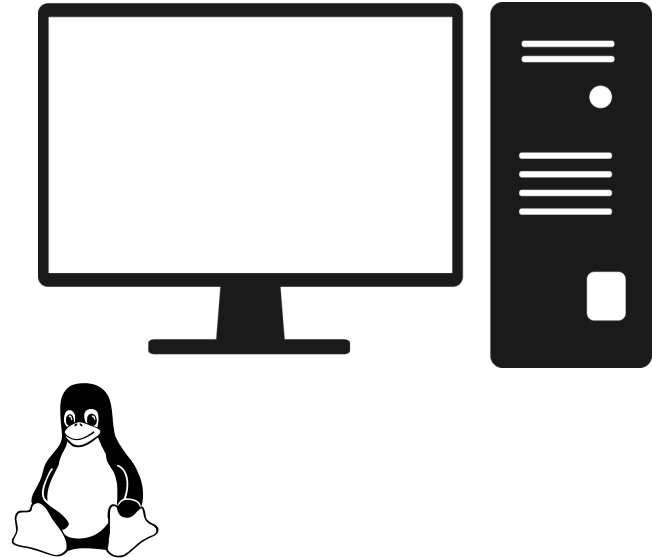
Hello World

1. Hello World

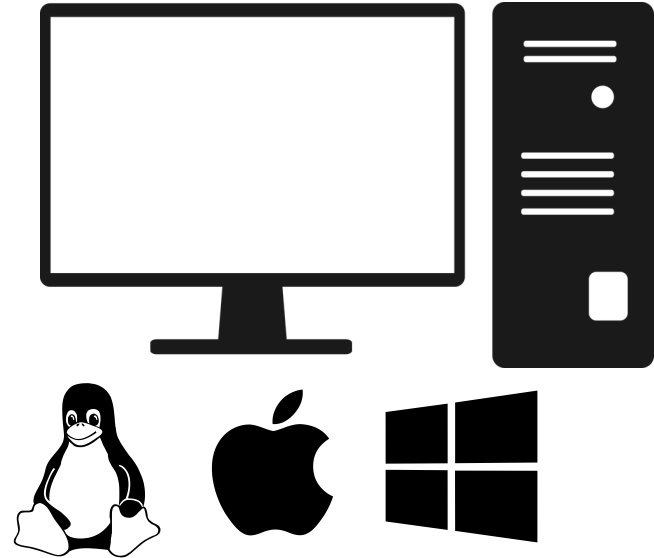
1. Abre esta carpeta
2. Copia los ficheros
3. Pégalos en una nueva ubicación



```
#!/bin/bash
```



```
#!/bin/bash
```



HELLO WORLD

```
#!/bin/bash  
echo -e "Hello World"
```

-e del echo

echo -e "Hello World"

Se utiliza para
activar caracteres
especiales

Salto de línea


```
echo -e "Hello World \n"
```



Para hacer un salto de línea

Tabulado

```
echo -e "Hello World \t"
```

A yellow callout box with a pointer directed at the backslash and 't' in the command above.

Para hacer un
tabulado

Bash es case sensitive



Echo -e "Hello World"

A terminal window with a black background. A horizontal line represents the command prompt. The text "Echo -e \"Hello World\"" is written on the line. A yellow cursor, shaped like an upward-pointing arrow, is positioned below the line, pointing to the first character of the word "Echo".

Bash no utiliza final de línea

```
echo -e "Hello World"; echo -e "Hello World"
```



Los espacios son muy importantes en Bash

```
echo - e "Hello World"
```



Comentarios en bash

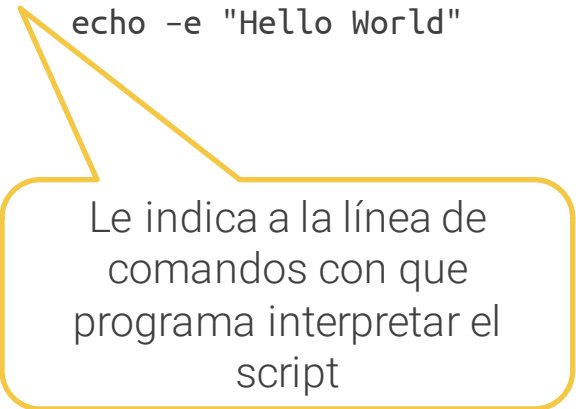
```
#!/bin/bash  
  echo -e "Hello World"  
#esto es un comentario
```

Bash no utiliza main

```
#!/bin/bash
  echo -e "Hello World"
  echo -e "Hello World"
  echo -e "Hello World"
  ls
  cat file
```

Shebang

```
#!/bin/bash  
echo -e "Hello World"
```



Le indica a la línea de comandos con que programa interpretar el script

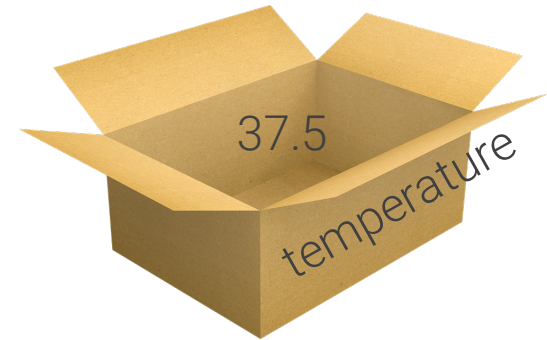
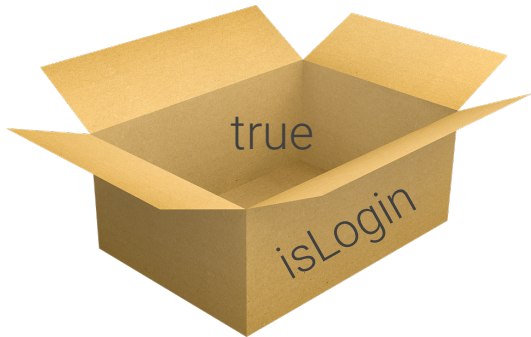
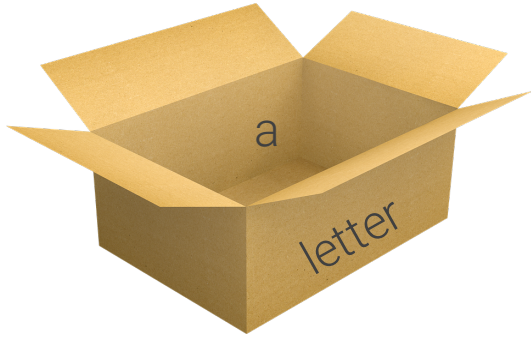
Shebang	Description
<code>#!/bin/sh</code>	Execute the file using sh, the Bourne shell, or a compatible shell
<code>#!/bin/csh</code>	Execute the file using csh, the C shell, or a compatible shell
<code>#!/usr/bin/perl</code>	Execute using Perl with the option for taint checks
<code>#!/usr/bin/php</code>	Execute the file using the PHP command line interpreter
<code>#!/usr/bin/python</code>	Execute using Python

Variables

2. Variables



2. Variables



2. Variables



Declarar una variable

`passengers=0`

Sin tipo de variable



Declarar una variable

```
passengers=0
```

Se declara y se inicializa en la misma línea



2. Variables

```
#!/bin/bash  
echo -e "Mi primera variable definida"  
passengers=0
```

Printar una variable

```
passengers = 0  
echo -e "$passengers"
```


2. Variables

```
#!/bin/bash
echo -e "Mi primera variable definida\n"
passengers=0
echo -e "$passengers"
```

Print output

Mi primera variable definida
0

2. Variables

```
#!/bin/bash
echo -e "Mi primera variable definida"
passengers=0
echo -e "El número de pasajeros es: $passengers"
```

Print output

El número de pasajeros es: 0

Nombre de las variables

```
passengers=10
```

```
witdh=25.5
```

```
i=0
```

```
1passengers = 10
```

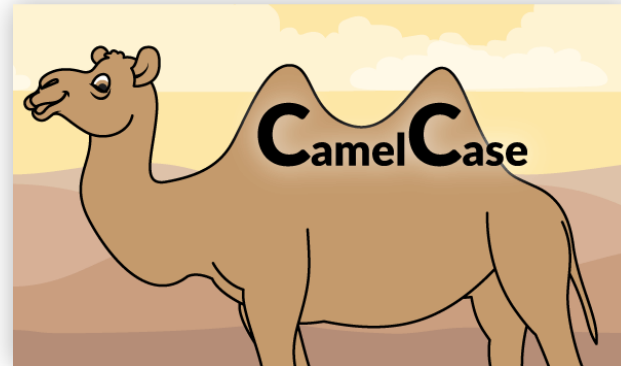
Los nombres de las variables deben comenzar con una letra minúscula o _

Nombre de las variables

isLogin=1

stopIdentification=A

lower Camel Case



Nombre de las variables

isLogin=1

~~stop Identification=A~~

Sin espacios en
blanco

Los nombres de variables son case sensitive

```
productPrice=20
```

```
echo -e "$productprice"
```

productprice como tal
está vacía

Constantes

`readonly varCar=1`

readonly indica que el valor no
puede cambiar

readonly TAX=21

~~TAX=2656~~

No se le puede asignar un
nuevo valor

Nombre de las constantes

readonly TOTAL_STOPS = 10

Upper Case

“Y te juro que vale la pena caer y romperse la cara,
que regalo tan grande aprender de lo que nos
dolió.”

Pablo Lopez

