

## 2\_TRUCOS\_EDITOR\_COLAB\_START\_sebassilvap

January 31, 2024

### TRUCOS DEL EDITOR JUPYTER / COLAB

1) Ejecutar una celda de Código:

**0.0.1 CTRL + ENTER : Ejecutar y quedarse en la celda**

**0.0.2 SHIFT + ENTER : Ejecutar y avanzar a la siguiente celda, y si no hay crear una**

**0.0.3 ALT + ENTER : Ejecutar y CREAR nueva celda, exista o no una después**

[ ]:

[ ]:

[ ]:

```
[46]: # =====  
      # Output sin necesidad de print()  
  
      # - colab / jupyter notebook  
      # - presenta el output sin necesidad de un print  
      # =====
```

2) Guardar Cuaderno:

**0.0.4 CTRL + S**

```
[47]: # Al pulsar CTRL + S => el cuaderno queda guardado hasta ese punto
```

3) Comandos Clásicos de Edición de Texto: Copiar / Pegar / Cortar

0.0.5 CTRL + C : Copiar Texto / Código

0.0.6 CTRL + V : Pegar Texto / Código

0.0.7 CTRL + X : Cortar Texto / Código

0.0.8 WINDOWS + V : Abrir Portapapeles de Windows

```
[48]: # =====  
# Ejercicio  
# =====  
  
print('pizza')  
print('cola')  
print('banana')  
print('hamburguesa')  
print('helado')
```

```
pizza  
cola  
banana  
hamburguesa  
helado
```

```
[49]: # =====  
# Práctica y Solución  
# =====
```

4) Transformar / Quitar Comentarios

- **OJO !!!** Esto sirve tanto para celda de código como de markdown

0.0.9 CTRL + / : Transformar o Quitar Comentario

```
[50]: # =====  
# Ejercicio  
# =====  
  
print('Buenos días')  
print('Espero que se encuentre bien')  
print('Esto no va en consola')  
print('Que tenga un buen día. Chao!!')
```

```
Buenos días  
Espero que se encuentre bien  
Esto no va en consola  
Que tenga un buen día. Chao!!
```

```
[6]: # =====  
# Práctica y Solución  
# =====
```

## 5) Comandos Clásicos de Deshacer & Rehacer

### 0.0.10 CTRL + Z : Deshacer Cambios

### 0.0.11 CTRL + Y : Rehacer lo modificado con CTRL + Z

```
[7]: # =====  
# Ejercicio  
# =====  
  
print('Hola Python')  
print('Hola Java')  
print('Hola C++')  
print('Hola Javascript')
```

Hola Python  
Hola Java  
Hola C++  
Hola Javascript

```
[8]: # =====  
# Práctica y Solución  
# =====
```

## 6) Desplazamiento y Selección Horizontal

### 0.0.12 HOME : Inicio de línea

### 0.0.13 END : Final de línea

### 0.0.14 CTRL + HOME : Selección hasta inicio de línea

### 0.0.15 CTRL + END : Selección hasta final de línea

```
[9]: print('aquí viene un texto largo')  
print('aquí viene un texto largo')  
print('aquí viene un texto largo')  
print('aquí viene un texto largo')
```

aquí viene un texto largo  
aquí viene un texto largo  
aquí viene un texto largo  
aquí viene un texto largo

## 7) Borrar + Copiar Línea:

### 0.0.16 SHIFT + DELETE

```
[10]: # =====  
# Ejercicio  
# =====
```

```
print('Pablo')
print('Sebas')
print('Andrea')
print('Carla')
print('Marcelo')
```

Pablo  
Sebas  
Andrea  
Carla  
Marcelo

```
[11]: # =====
      # Práctica y Solución
      # =====
```

8) Mover línea de código dentro de la celda

**0.0.17** SHIFT + : Mover línea hacia arriba

**0.0.18** SHIFT + : Mover línea hacia abajo

```
[12]: # =====
      # Ejercicio
      # =====

print( 'Esta es una línea que se imprime en la consola' )
print( 'Esta es una línea que se imprime en la consola' )
print( 'Esta es una línea que se imprime en la consola' )
print( 'Esta es una línea que se imprime en la consola' )
print( 'Esta es una línea que se imprime en la consola' )
print( 'Esta es una línea que se imprime en la consola' )
print( '*** LÍNEA A MOVER ***' )
print( 'Esta es una línea que se imprime en la consola' )
print( 'Esta es una línea que se imprime en la consola' )
```

Esta es una línea que se imprime en la consola  
Esta es una línea que se imprime en la consola  
Esta es una línea que se imprime en la consola  
Esta es una línea que se imprime en la consola  
Esta es una línea que se imprime en la consola  
Esta es una línea que se imprime en la consola  
\*\*\* LÍNEA A MOVER \*\*\*  
Esta es una línea que se imprime en la consola  
Esta es una línea que se imprime en la consola

9) Duplicar línea de código:

**0.0.19** SHIFT + ALT + : Duplicar línea hacia arriba

**0.0.20** SHIFT + ALT + : Duplicar línea hacia abajo

```
[13]: # =====  
# Ejercicio  
# =====  
  
print('hola estoy en consola')  
  
print('hola sebas')
```

```
hola estoy en consola  
hola sebas
```

```
[14]: # =====  
# Práctica y Solución  
# =====
```

10) Seleccionar TODAS las coincidencias:

**0.0.21** CTRL + SHIFT + L

```
[15]: # =====  
# Ejercicio  
# =====  
  
comida_favorita = 'pizza'  
  
print( 'Mi comida favorita es la ' + comida_favorita )  
print( comida_favorita.upper() )  
print( comida_favorita.title() )
```

```
Mi comida favorita es la pizza  
PIZZA  
Pizza
```

```
[16]: # =====  
# Práctica y Solución  
# =====
```

11) Seleccionar coincidencia de un punto en adelante:

**0.0.22** CTRL + D

```
[17]: # =====  
# Ejercicio  
# =====  
  
# - Cambiar la segunda parte a "bebida_favorita"
```

```
comida_favorita = 'pizza'

print( 'Mi comida favorita es la ' + comida_favorita )
print( comida_favorita.upper() )
print( comida_favorita.title() )

print('-----')

comida_favorita = 'pizza'

print( 'Mi comida favorita es la ' + comida_favorita )
print( comida_favorita.upper() )
print( comida_favorita.title() )
```

```
Mi comida favorita es la pizza
PIZZA
Pizza
-----
Mi comida favorita es la pizza
PIZZA
Pizza
```

```
[18]: # =====
# Práctica y Solución
# =====
```

## 12) Selección Avanzada de Elementos

**0.0.23** SHIFT + CTRL + / : Selección por nombres de variables / Se corta en espacios (MUY MUY USADO !!!)

**0.0.24** SHIFT + ALT + / : Selección por palabras estándar (No muy usado)

```
[19]: # =====
# Ejercicio
# =====

# shift + ctrl => selecciona: nombre_completo (OK)
# shift + alt => selecciona: nombre (X)

nombre_completo_de_persona = 'Sebastián Silva'

print( nombre_completo_de_persona.upper() )
print( nombre_completo_de_persona + '!' )
print( nombre_completo_de_persona + ' ' + nombre_completo_de_persona )
```

```
SEBASTIÁN SILVA
Sebastián Silva!
```

Sebastián Silva Sebastián Silva

```
[20]: # =====  
# Práctica y Solución  
# =====  
  
# seleccionar rápidamente una variable y cambiarla con CTRL + D
```

13) Abrir pantalla de Shortcuts

**0.0.25 CTRL + M + H : Con el teclado**

**0.0.26 Menú + Herramientas + Combinaciones de Teclas : Por medio del menú de edición de Colab**

```
[21]: # Abrir la PANTALLA DE SHORTCUTS de Colab  
# Usar cualquiera de las 2 maneras explicadas arriba
```

14) Abrir la Paleta de Comandos

**0.0.27 CTRL + SHIFT + P : Con el teclado**

**0.0.28 Menú + Herramientas + Paleta de Comandos : Por medio del menú de edición de Colab**

```
[22]: # Abrir la PALETA DE COMANDOS de Colab  
# Usar cualquiera de las 2 maneras explicadas arriba  
  
# ==> La paleta de comandos nos muestra todos los shortcuts a manera de lista  
# ==> Y como una búsqueda rápida de algo que queramos hacer
```

15) Crear Multicursores con el Teclado

- Una de las funcionalidades MÁS ÚTILES que se tiene al momento de escribir código !!!

**0.0.29 CTRL + ALT + /**

```
[23]: # =====  
# EJERCICIO  
  
# - Cambiar los valores de las variables  
# - Anteponer "numero_"  
# - Imprimir luego los valores en consola  
# - Utilizar multicursores con el teclado  
# =====  
  
a = 10  
b = 15  
c = 5.5  
d = 1.66  
e = 0.5
```

```
[24]: # =====  
# SOLUCIÓN  
# =====
```

16) Crear Multicursores con el Mouse

- También útil !! Sobre todo cuando se dificulta la navegación con el teclado

#### 0.0.30 ALT + CLICK IZQUIERDO

```
[25]: # =====  
# EJERCICIO  
  
# - A los números a, c y e  
# - Aumentar " * 2 "  
# - Para multiplicar por 2 estos valores  
# - Y luego imprimir en consola  
# - Y transformar todo a float  
# =====  
  
numero_a = 10  
numero_b = 15  
numero_c = 5.5  
numero_d = 1.66  
numero_e = 0.5
```

```
[26]: # =====  
# SOLUCIÓN  
# =====
```

17) Ejecutar todas las celdas ANTERIORES a la actual

- **OJO !!!** Tener cuidado con este comando.
- Podemos tener varias celdas antes que involcuren inputs, bucles, etc...
- En nuestro libro de ejercicios es preferible no usar esto.

#### 0.0.31 CTRL + F8

```
[27]: print('Ejecutar esta celda con CTRL + F8')
```

Ejecutar esta celda con CTRL + F8

18) Ejecutar todas la celda ACTUAL y las siguientes

- **OJO !!!** Tener cuidado con este comando.
- Podemos tener varias celdas antes que involcuren inputs, bucles, etc...
- Puede ser útil en nuestro Notebook de Ejercicios, pero en nuestro caso ejecutaremos celda por celda para hacerlo seguro



### 0.0.32 CTRL + F10

```
[28]: # =====  
# Primera Celda  
# =====  
  
nombre = 'Sebastián'  
edad = 36
```

```
[29]: # =====  
# Segunda Celda  
# =====  
  
edad += 1
```

```
[30]: # =====  
# Tercera Celda  
# =====  
  
print( '{} tendrá {} el siguiente año !!'.format(nombre, edad) )
```

Sebastián tendrá 37 el siguiente año !!

19) Mover una CELDA Arriba o Abajo

- **OJO !!!** Tener cuidado con este comando.
- Mover una celda puede afectar el orden de ejecución del código.
- Y causar problemas en nuestro programa
- También se puede usar la barra derecha que aparece en la celda

### 0.0.33 CTRL + M + K : Mover Celda hacia Arriba

### 0.0.34 CTRL + M + J : Mover Celda hacia Abajo

```
[31]: # Esta es una celda de código
```

```
[32]: # Esta es una celda de código
```

```
[33]: # Esta es una celda de código
```

```
[34]: # ***** MOVER CELDA DE CÓDIGO *****
```

```
[35]: # Esta es una celda de código
```

```
[36]: # Esta es una celda de código
```

```
[37]: # Esta es una celda de código
```

20) Eliminar una CELDA de código & Deshacer Cambios en CELDA

- **OJO !!!** Tener cuidado con este comando.

- Mover / Eliminar una celda puede afectar el orden de ejecución del código.
- Y causar problemas en nuestro programa
- También se puede usar la barra derecha que aparece en la celda
- Si algún cambio no deseado se ha hecho en una celda (ej: borrarla por error) se puede deshacer estos cambios también.

#### 0.0.35 CTRL + M + D : Mover Celda hacia Arriba

#### 0.0.36 CTRL + M + Z : Deshacer cambios realizados a nivel de CELDA

[38]: # Celda de Código # 1

[39]: # Celda de Código # 2

[40]: # Celda de Código # 3

[41]: # Celda de Código # 4

[42]: # Celda de Código # 5

[43]: # Celda de Código # 6

[44]: # Celda de Código # 7

[ ]:

[45]: # \*\*\*\*\* THE END \*\*\*\*\*