¿Qué es un atajo de teclado?

Normalmente, para realizar tareas en una computadora hacemos uso de dos recursos principales: el **mouse** y la **interfaz gráfica** (lo que se muestra en nuestra pantalla). Y cuando requerimos de algún texto o palabra específica, involucramos el teclado de manera secundaria.

Pero hay tareas para las cuales podríamos utilizar el **teclado únicamente**, en vez de utilizar el mouse y la interfaz. Realizando tareas con más rapidez y de una manera más sencilla.

Pero.. ¿Cómo podríamos utilizar el teclado para realizar las mismas instrucciones que con el mouse? Utilizando los llamados "**atajos de teclado**" o "**shortcuts**" en inglés.

Como su nombre indica, son un atajo y nos ahorran una parte del camino para llegar a un lugar o, en este caso, a una determinada instrucción. Pero **¿qué son en realidad?**

Un atajo de teclado implica una **combinación de teclas específicas** para realizar una determinada tarea. En cada entorno de trabajo tenemos determinados y específicos shortcut que pueden ser utilizados. Por ejemplo, en Zoom tenemos algunos atajos de teclado, para silenciarnos, prender la cámara, entre otros, y en PseInt, también tenemos otros que nos ayudan a realizar determinadas tareas.

Su importancia

Los shortcuts son importantes a la hora de hablar sobre el performance de un programador. Ya que, combinando determinadas teclas, se ahorra un esfuerzo y un tiempo determinado que conlleva realizarlo de otra manera. Esto impacta en la productividad de un desarrollador, debido a que, realizaría las tareas en menos tiempo.

¿Qué es identar?

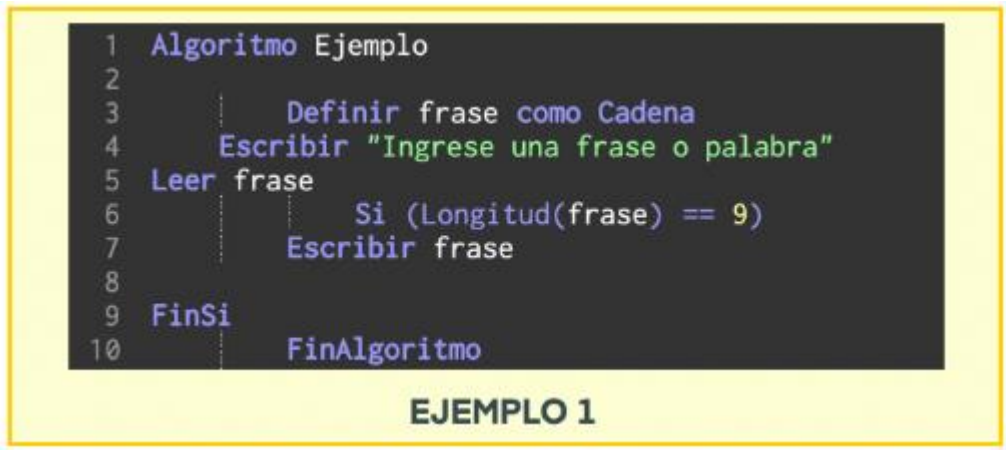
Básicamente, este término significa **mover un bloque de texto hacia la derecha insertando espacios o tabulaciones**, para así separarlo del margen izquierdo y distinguirlo mejor.

En los lenguajes de programación, se lo considera un tipo de notación secundaria utilizada para mejorar la **legibilidad del código**, teniendo en cuenta que los compiladores o intérpretes raramente consideran los espacios en blanco entre las sentencias de un programa.

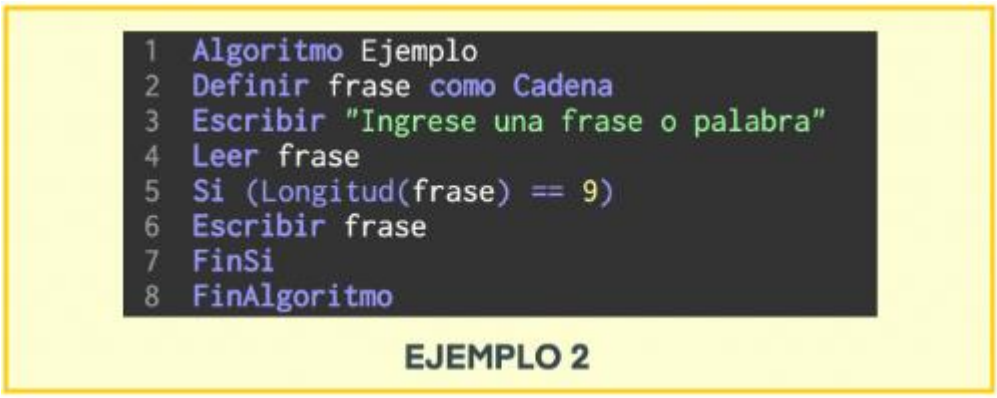
El indentado comprende una parte importante de la presentación de nuestro código y es un punto muy relevante en los lineamientos del **Código Limpio***(Clean Code)*. Un código bien indentado es **legible** y ayuda a que otras personas puedan **comprender** mejor nuestro código. ¿No nos crees? Hagamos una prueba.

Te dejamos un pequeño desafío: Intenta descifrar cuál es la funcionalidad del siguiente código. Al final reflexionaremos un poco.

Ejemplo 1:



Ejemplo 2:



Ejemplo 3:

Reflexión

Reflexionamos: ¿Cuál de los tres ejemplos fue más sencillo leer e interpretar?

¿Es fácil y práctico analizar dónde se encuentran las palabras claves “FinSi” y “FinAlgoritmo”?

¿Cuándo el indentado no es adecuado?

¿Es perceptible las etapas que tiene el código o cuántos procesos ocurren?

¿Cuál de los tres ejemplos tiene mejor presentación?

El indentado es una herramienta para que nuestro código se vea prolijo y agradable. Resultando en un código legible y facilitando un poco la interpretación del mismo.

Comando corregir identado

A la hora de trabajar el código de manera legible PseInt nos da un comando que cuando seleccionamos una porción de codigo nos corrige el identado y nos deja el codigo de la mejor manera posible.

Como podemos ver nuestro código estaba todo amontonado y al usar el comando identar, el codigo se separó y quedó mucho más legible.

El comando lo podemos usar con el **click derecho** o en el teclado tocar **Ctrl + L (Windows) o Cmd + L(Mac)**

Un proyecto no involucra a UN solo programador

Este es un punto muy importante a comprender: los programadores **trabajan en equipo**. En casi la totalidad de los trabajos actuales, un desarrollador comparte su código con otras personas. Por este motivo, es importante que aprendamos a desarrollar códigos legibles y claros, para que, **al compartirlos**, otras personas logren entender nuestro código con claridad. Además un código limpio incrementa la eficiencia y productividad del proyecto.

Comentar nuestro código

Agregar comentarios a nuestro código es una buena práctica de programación. A través de comentarios podemos especificar el propósito y finalidad de nuestro código, para que otras personas puedan interpretar y comprender mejor nuestro código.

A continuación vamos a ver una **serie de consejos** a la hora de **utilizar comentarios**:

1. Ser claro y preciso

Nuestros comentarios deben ser herramientas para que comprendamos mejor nuestro código, para que otras personas logren comprender rápidamente de qué se trata. Para ello, lo mejor es que nuestros comentarios sean precisos y claros. Pero atención, que corto no es lo mismo que **claro y preciso**: se debe entender el mensaje que se quiere transmitir sin confusiones. Recordemos que los comentarios son herramientas para esclarecer y ayudar a que los desarrolladores tengan un mejor rendimiento.

2. Comenta como si fuese para ti (lo es)

No tengas miedo a comentar demasiado: ¡comentá lo que sea necesario para que comprendas mejor la finalidad de tu código! Puedes comentar un párrafo completo, puedes agregar anotaciones teóricas, ¡lo que quieras!

3. Identar el párrafo comentado

Nuestro código debe ser claro y legible, eso ya lo sabemos, pero debemos tener en cuenta que los comentarios también son parte de nuestro código y por lo tanto, también deben ser claros y legibles, aguardando una buena presentación general. Por ello recuerda **alinearlo acorde al código**: es preferible tener un párrafo de un par de líneas en vez de una única línea y que sea muy extensa.

4. Comenta el propósito de una función o estructura

Cuando creamos una función o una estructura determinada que cumplirán UN propósito, es acorde que comentemos al inicio de la estructura cuál es esa finalidad. De esta manera, quien lea nuestro código asimilará mejor cuál es la finalidad de ese bloque de código, en lugar de adivinar cuál podría ser.

5. Actualiza comentarios junto con el código

Si realizamos cambios en nuestro código, debemos analizar si los comentarios también requieren de cambios o actualizaciones. Debemos estar atentos a esto, ya que podría confundir al lector.

Shortcuts en PseInt

Te dejamos algunos atajos de teclado para que empieces a implementar en Pseint:

FINALIDAD	ATAJO DEL TECLADO (Windows)	ATAJO DEL TECLADO (Mac)
Ejecutar el código	F9	F9
Seleccionar Todo	Ctrl + A	Cmd + A
Copiar	Ctrl + C	Cmd + C
Comentar Líneas	Ctrl + D	Cmd + D
Descomentar Líneas	Ctrl + Shift + D	Cmd + Shift + D
Buscar	Ctrl + F	Cmd + F
Duplicar Líneas	Ctrl + L	Cmd + L
Eliminar Líneas	Ctrl + Shift + L	Cmd + Shift + L
Rehacer	Ctrl + Shift + Z	Cmd + Shift + Z
Pegar	Ctrl + V	Cmd + V
Cortar	Ctrl + X	Cmd + X
Deshacer	Ctrl + Z	Cmd + Z
Corregir Indentado	Ctrl + L	Cmd + L
Buscar siguiente	F3	F3
Ejecutar Paso a Paso	F5	F5