

JAVA.

Java es el lenguaje con el que vamos a realizar nuestras primeras aplicaciones de escritorio, lo que quiere decir que solo se podrán ejecutar a través de un ordenador y un ejecutable.

Las variables.

Las variables son pequeños contenedores de datos en los que guardamos la información o datos que reciba y use el programa. Hay varios tipos de variables, pero los primeros con los que vamos a trabajar, son estos:

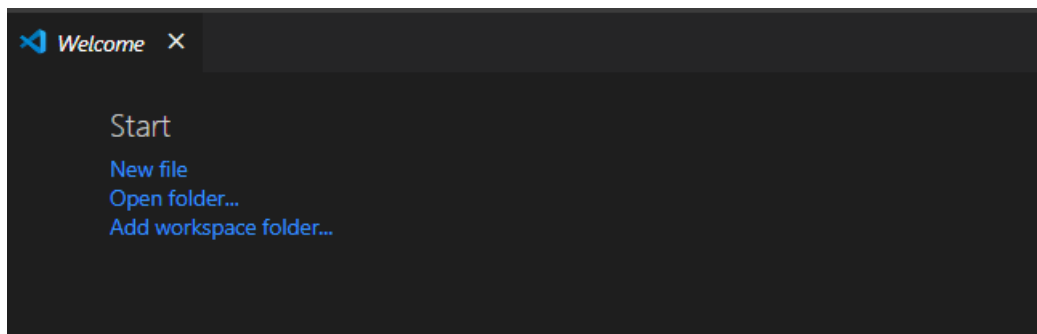
- String. "Cadenas de texto".
- int. "Números enteros".
- boolean. "Llaves de estado. Solo pueden estar en true o false".
- float. "Números que admiten decimales". (No lo usaremos por ahora)
- double. "Como los float pero permiten números mucho mayores".

Comenzar un programa.

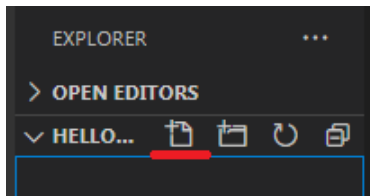
Para crear un programa en Java lo más cómodo será que hagamos una carpeta para cada nuevo programa. Esto ayudara a que no haya conflicto entre los archivos a la hora de programar. Vamos a hacer un programa de ejemplo:

(Leer todo una vez antes de empezar a seguir los pasos)

Abrimos VSC y si no tenemos ningún proyecto precargado, nos encontraremos con la siguiente ventana:

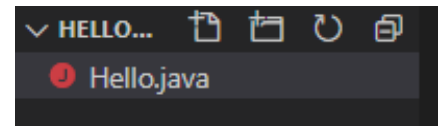


Seleccionamos “Open folder...” y desde el explorador de carpetas creamos la carpeta de nuestro proyecto donde queramos. Una vez hecha la seleccionamos. Ahora veremos que el nombre de nuestra carpeta aparecerá en la barra “Explorer” de la izquierda, debajo de “Open Editors”. Si esta barra no está, haz clic sobre el icono de los dos folios. Al pasar el ratón por encima del nombre de nuestra carpeta vemos lo siguiente:



El icono indicado nos permitirá crear un nuevo archivo, le damos y ponemos nombre.

Tan pronto escribamos “.java” el icono del archivo cambiara al que tiene ahora, indicando que ya es un archivo Java. Ahora sí, vamos a crear los dos pilares del proyecto, la clase y el main.



La clase ya hemos visto que es el contenedor primitivo, en ella se acumulan todas las funciones y variables que nuestro programa usa. No se puede escribir código fuera de una clase. Es el mantel sobre el que se pone la mesa.

El main es la función de arranque a la que nuestro programa se va a dirigir siempre que se inicie, desde aquí iniciaremos el programa y controlaremos su ritmo. No puede haber más de un main por clase o proyecto.

Para crear estos dos elementos, vamos a utilizar unos atajos, ya que no es importante conocerlos a fondo por el momento. Comenzaremos a escribir “class” y veremos que la barra de autocompletado (si no sale pulsar Ctrl+Space) nos recomienda dos palabras “class”. Seleccionamos la segunda, la que parece tener el icono de un cuadrado. Al hacer enter o clic sobre esta sugerencia se nos auto-completará la clase básica de nuestro programa. Ahora, justo donde nos ha dejado el cursor, podemos continuar y escribir “main”, de nuevo volverá a salirnos recomendaciones, en este caso seleccionamos la primera. De este modo ya hemos creado la clase y el main, nuestro programa está listo para ejecutarse. Si algo de esto no ha quedado muy claro, puedes consultar el video “Creación de archivo Java y main” que hay dentro de la carpeta “Videos Tutoriales” dentro de la carpeta donde estaba este archivo.

Hello World.

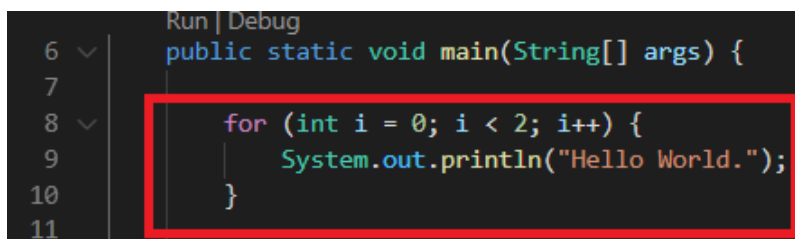
Como es tradición en el mundo informático/robótico/programador, todo comienzo se hace siempre con un “Hello World”, en este caso vamos a hacer que eso salga por pantalla, o más exactamente, por consola. Y al igual que antes, vamos a usar atajos. Si escribimos “sysout”, la sugerencia nos permitirá auto-completar un “System.out.println();”. Ahora dentro de los paréntesis, escribimos la cadena de texto “Hello World”. Recuerda que las cadenas de texto siempre han de ir entre comillas y al escribir la de apertura, automáticamente se escribirá la de cierre. Una vez escrito, pulsamos F5 y se nos abrirá en la parte inferior la consola de VSC y nos mostrará el resultado de nuestro programa.

for().

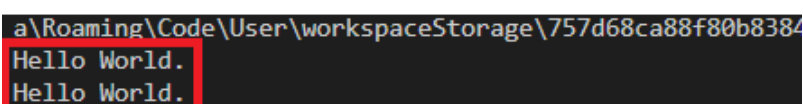
La palabra “For” nos permite crear bucles. La sintaxis completa de for, es la siguiente:

```
for (int i = 0; i < 2; i++) {  
    //Codigo  
}
```

Ahora bien, ¿qué significa cada cosa? “i” es la variable que usaremos como contador para indicar cuantas vueltas a dar el bucle. La declararemos/crearemos siempre al inicio del paréntesis y le asignaremos un valor base, en este caso 0. “i<2” es la condición para que el bucle siga ejecutándose. Es decir, mientras “i” sea menos que 2, sucederá lo que escribamos dentro de los paréntesis. Y finalmente “i++” que es la condición con la que marcamos el ritmo del bucle. Lo que quiere decir en este caso “i++” es que a cada vuelta que del bucle, a “i” se le sumara 1. “++” es la forma corta de escribir “i+1=i”, de esa manera se asignara automáticamente a “i” que su nuevo valor es el anterior más 1. Prueba a poner un saludo de Hello World dentro de un bucle como ese en el main. ¿Ves su funcionamiento?



```
Run | Debug  
6  public static void main(String[] args) {  
7  
8      for (int i = 0; i < 2; i++) {  
9          System.out.println("Hello World.");  
10     }  
11
```



```
a\Roaming\Code\User\workspaceStorage\757d68ca88f80b8384  
Hello World.  
Hello World.
```

If y else.

If y else son la forma más sencilla de comparación y negativa de la programación. A través de comparar con if podemos saber si una variable dada es igual o diferente a un valor u otra variable.

```
if (numero == 5) {  
    // Código  
} else {  
    // Código  
}
```

Si número es igual a 5, se ejecutara el primer “// Código” y el segundo será ignorado. Si por el contrario, número fuese cualquier valor diferente de 5, se ejecutaría el segundo “// Código”. Else no es la opción contraria a la comparación, si no que cualquier resultado diferente al de la comparación. En este caso cualquier número que no sea 5. Podemos comparar cualquier valor. Para la comparación usamos varios símbolos:

- ==, Si queremos saber si los valores son iguales.
- !=, Si queremos saber si son diferentes.
- > o <, Si queremos saber si el uno es menor o mayor que el otro.
- >= o <=, Si queremos saber si son iguales o mayores/menores que el otro valor.

Estos dos últimos apartados solo son aplicables a números. Podemos utilizar if para comparar strings pero ya veremos más adelante que hay otros métodos más cómodos. Para los booleanos podemos poner solo la variable dentro del if si queremos comparar si está a true o poniéndole un “!” delante si queremos saber si está a false.

Recoger información por teclado.

Lo último que necesitamos antes de ponernos manos a la obra es saber introducir nuestros propios valores en las variables a través del teclado mientras el ejercicio se ejecuta. Para utilizar el teclado en nuestro programa, primero debemos crear un “Scanner” que es la entrada por teclado. Para esto necesitamos

primero importar la clase escáner. Esto lo vamos a hacer mediante el único código que podemos poner fuera de una clase, “import”.

```
import java.util.*;
```

Lo que esto hace es traer a nuestro proyecto toda referencia a unas librerías pre-hechas de Java. Pero por ahora solo debemos recordar el hecho de hacer la importación y ya.

Lo siguiente es crear el “Scanner”, para esto hacemos una declaración de objeto.

```
Scanner sc = new Scanner(System.in);
```

Ahora ya podemos empezar a recoger datos y almacenarlos en variables, de las siguientes formas:

```
// String
texto = sc.next();
// int
numero = sc.nextInt();
// double
decimal_simple=sc.nextDouble();
// float
decimal = sc.nextFloat();
```

El único valor que no podemos introducir directamente desde teclado a una variable son los booleans.

Con todo esto podemos comenzar a hacer nuestros primeros ejercicios.

Practica.

Para mantener una mejor claridad de proyectos y que le cojas el punto a la creación de proyectos, habrá una carpeta “Practicas Java” dentro del repositorio y dentro de esta una carpeta por cada ejercicio, “Practica_X” es importante que el nombre de la carpeta sea una palabra completa, para poner más de una palabra en un nombre utilizaremos siempre barra baja (Y sin tildes, que pueden generar conflictos). De esta manera por cada ejercicio creas un proyecto como está escrito al principio y poniéndole en nombre y número de cada ejercicio.

1. **Un saludador.** Este ejercicio ya lo conoces y hasta tienes uno de ejemplo, pero como este lo vas a hacer desde cero, será más sencillo. Deberá preguntársele al usuario su nombre y después su edad. Una vez que tengamos ambos valores imprimiremos el siguiente mensaje por pan-

talla: "Hola X. Edad: X. Bienvenido/a.". Si quieres hacer esto un poco más complejo, haz que se pregunte el género. Para esto utiliza un 1 si es hombre y un 2 si es mujer. No es necesario usar booleanos. En función de lo que se haya seleccionado, se mostrara "Bienvenido" o "Bienvenida".

2. Queremos realizar dos contadores. Uno contara de 0 a 100, ambos incluidos y otro ira de 0 a -100, ambos incluidos. Deberá mostrarse por pantalla en cascada todos los números, primero los positivos y todos los negativos después. Si quieres darle un punto extra al ejercicio, añade un divisor y un punto de espera entre ambas series de números. Deberá pulsarse cualquier tecla para que se lance el segundo.
3. Calculadora. Vamos a hacer una calculadora sencilla que pueda sumar. Para esto se le pedirá al usuario que introduzca primero un número y luego otro, finalmente dibujamos un divisor con guiones en pantalla y debajo sacamos el resultado. Para hacerlo más completo, podemos añadir un menú y escoger primero cuál de las cuatro operaciones básicas deseamos hacer: suma, resta, multiplicación o división. Los símbolos que usa Java para las operaciones son +, -, *, /. Para el menú puedes tratar de hacer algo como esto.

```
Escoja una operacion:
```

```
-----  
1. Suma  
2. Resta  
3. Multiplicación  
4. División  
-----
```