Dart:

// Es de tipado estático

// Tipado explicito – creo (no sé) es que las variables es como tipado estático.

//Declaración de Constante con tipo explícito.

//Tiempo de compilación 'const'.

//Tiempo de ejecución 'final', 'runtime'.

.isEven //Muestra si es par

IsOdd ////Muestra si es impar

isNegative //Creo que devuelve true or false si algo es o no negativo.

print(b.floor()); //Muestra piso-Redondea hasta abajo

print(b.ceil()); //Muestra cielo -Redondea hasta arriba

print(b.round()); //Redondea dependiendo del decimal, arriba o abajo

print(b.truncate()); //Trunca los decimales

print(b.isNaN); //Mustra si no es numero con false or true

//   contador = contador-1;

//   print(contador);

//   contador -=1;

//   print(contador);

//   contador--;

//   print(contador);

//   --contador;

//   print(contador);

//   var c = 10;

//   print(++c); //11 con ++

//   c = 10;

//   print(c++); //10

//   print(c); //Ya imprime el 11 con ++

 print(numero.toStringAsFixed(3)); //Muestra el numero de decimales que le pongas

**Orientado a objetos:**

Paradigma de dart, orientado a objetos.

UML = unified modeling language = la tabla donde va:

-Nombre –Clase-

-Características -Propiedades- = variables

-Métodos = funciones

Behavors = comportamiento = métodos = Lo que hace el objeto.

Instancia = copia de la clase = int n = instancia de ‘int’ en ‘n’.

//Una sola linea

/\*Comentarios

en

bloque

\*/

//El encapsulamiento esconde las propiedades del exterior, que no se puedan visualizar

//setter: Es un método que establece el valor de una propiedad, no más

//getter: Es un método que obtiene el valor de una propiedad, no más

//set y ger es para variables de instancia

// //Los constructores son la forma en que una instancia se inicializa

Siempre se impime lo que hay dentro de un constructor.

//constructor basico, debe llevar el nombre de la clase

/// Clase que reepresenta un usuario

Las 3 / ayuda en el código para remarcarte un mensaje de ¿ayuda? A la hora de entender que es lo que estás usando.

La \_ en una variable como \_marca es para privar la variable.

*User* usuario = *User*.nombre("Amhec"); //Usuario es una instancia

/// Clase que reepresenta un usuario

class *User*{

  /// Propiedad nobre de tipo String

*String*? nombre; //No tiene sentido asignarle un valor por defecto 'EJEMPLO'

  //el signo ? es para un dato con valor null

//metodos setter y getter

*void* set setNombre(*String* nombre){

    nombre = nombre;

  }

*void* set setEdad(*int* edad) => edad = edad;

*String* get getNombre{

    return nombre!; //El ! es para obligarlo a regresar el valor

  }

*int* get getEdad => edad!;

}

*User*(*String* nombre, *int* edad){ //constructor basico, debe llevar el nombre de la clase

    print("Este es el constructor de User"); //siempre imprime lo que hay aquí

    this.\_nombre = nombre;

    this.\_edad = edad; //nis permite

  }

**//Declaración con var 'Interferencia de tipo'**

var numero =100;

var numero1 =9.81;

var nombre = "Hector";

print(numero.runtimeType);

print(numero1.runtimeType);

print(nombre.runtimeType);

var resultado;

resultado= numero + numero1;

**//Concatenación de Cadenas.**

String nombreCurso = "Programacion Funcional";

String carrera = "ICI";

print(carrera + "" + nombreCurso);

**//Interpolación de Cadenas.**

print("$carrera $nombreCurso");

//Operaciones,llamar funciones etc '{}'.

String getCarrera(){

return "ICI";

}

print("${getCarrera()} ${nombreCurso}");

var nume = 4;

print("El cuadrado de $nume es ${nume \* nume}");

//Por ser final no va a cambiar.

**//Listas**

final calificaciones = [];//Vacía.

final calificaciones1 = [10,6,9,8,10,8];

print(calificaciones1);

**//Los elementos de las listas deben**

**//ser del mismo tipo.**

calificaciones1.add(2);

print(calificaciones1);

for (var i=0; i<calificaciones1.length; i++){

print(calificaciones1[i]);

}

**//Condicionales**

int n8 = 5, n9 = 3;

if (n8 >n9){

print("$n8 > $n9");

} else if (n8 == n9){

print("$n8 = $n9");

} else {

print("$n8 < $n9");

}

**//Operador Ternario**

int n7 = 9, n6= 4;

int mayor;

if (n7 > n6){

mayor = n7;

} else {

mayor = n6;

}

print("El mayor es $mayor");

**//Declarativo 'Funcional'**

menor =n10 < n11 ? n10 : n11;

print("El Menor es: $menor");

**//Sentencia Switch case**

switch (dia){

case "Lunes":

print("Hoy es lunes");

break;

case "Martes":

print("Hoy es Martes");

break;

default:

print("Dia no conocido");

}

**//Ciclos**

for (var i=1; i<=5; i++){

print("$i");

}

var n = ["1","2",3.1416,true,5];

for (var e in n){

print("$e");

}

n.forEach((e){

print("$e");

});