ÍNDICE

Ejericio 1.- Hello Wold

```
Run | Debug
1  void main() {
2    print("Hello, World!");
3  }
4
```

```
PS C:\Users\CESAR\Desktop\R1_U2_Ejercicios_Dart_Flutter\hello_word>  & 'C:\Users\CESA
ache\dart-sdk\bin\dart.exe' '--disable-dart-dev' '--enable-asserts' 'C:\Users\CESAR\
ercicios_Dart_Flutter\hello_word\bin\hello_world.dart'
c3c-9cc5-85d8675f846fHello, World!
```

Ejercicio 2.- Variables y constantes

```
Run | Debug
void main() {
  int httpStatusCode = 200;
  int response = httpStatusCode;
  print('httpStatusCode: $httpStatusCode,response: $response');

  httpStatusCode = 500;
  print('httpStatusCode: $httpStatusCode,response: $response');
}
```

```
PS C:\Users\CESAR\Desktop\R1_U2_Ejercicios_Dart_Flutter\variables_y_constantes>  &
lutter\bin\cache\dart-sdk\bin\dart.exe' '--disable-dart-dev' '--enable-asserts' 'C
top\R1_U2_Ejercicios_Dart_Flutter\variables_y_constantes\bin\variables_constantes.
httpStatusCode: 200,response: 200
httpStatusCode: 500,response: 200
```

```dart
Run | Debug
void main() {
  double distanceMile = 1;
  const toKm = 1.609;
  double distanceKm = distanceMile * toKm;

  print('$distanceMile Mile =  $distanceKm Km');
}
```

```
SAR\Desktop\R1_U2_Ejercicios_Dart_Flutte
he\dart-sdk\bin\dart.exe' '--disable-dar
cicios_Dart_Flutter\variables_y_constant
1.0 Mile =  1.609 Km
```

```dart
Run | Debug
void main() {
  final DateTime currentTime;
  currentTime = DateTime.now();

  print(currentTime);
}
```

```
SAR\Desktop\R1_U2_Ejercicios_Dart
he\dart-sdk\bin\dart.exe' '--disa
cicios_Dart_Flutter\variables_y_c
2024-08-07 11:56:54.451506
```

Ejercicio 3.- Funciones

```dart
greet(String name, String title) {
  return 'Hola $title $name!';
}

Run | Debug
void main() {
  print(greet('Torres', 'Profesor'));
}
```

```
nes\bin\funciones_greet.dart'
Hola Profesor Torres!
PS C:\Users\CESAR\Desktop\R1_U2
```

```dart
String greet(String name, [String title = '']) {
  if (title.isEmpty) {
    return 'Hello $name';
  }
  return 'Hello $title $name!';
}

Run | Debug
void main() {
  print(greet('John'));
  print(greet('Alice', 'Professor'));
}
```

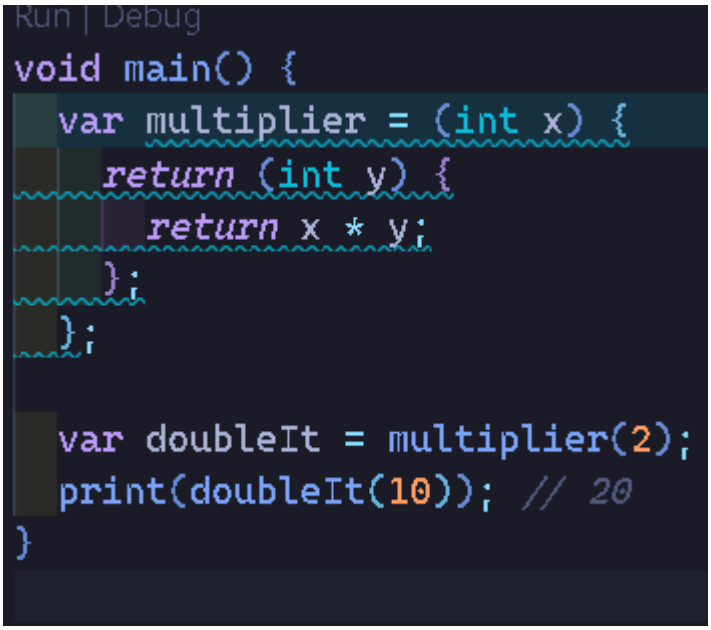
```
'--disable-dart-dev' '--enable-asser
nes\bin\parametros.dart'
Hello John
Hello Professor Alice!
```

```dart
void connect(String host,
    {int port = 3306, required String user, required String password}) {
  print('Connecting to $host on $port using $user/$password ... ');
}

Run | Debug
void main() {
  connect('localhost', user: 'root', password: 'secret');
}
```

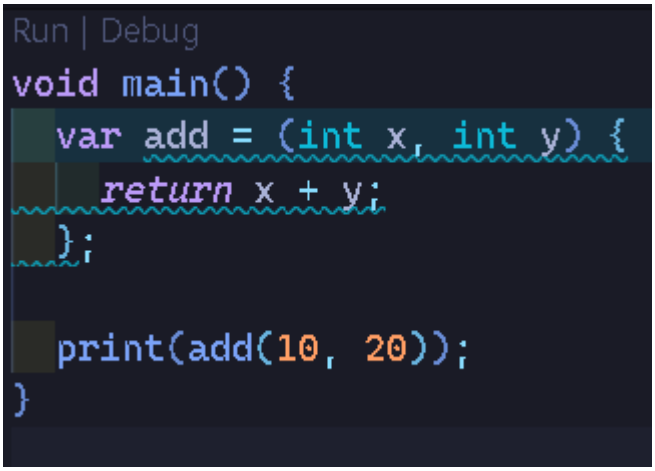
```
1_U2_Ejercicios_Dart_Flutter\funciones'; & 'C:\Users\CESAR\flutter\b
'--disable-dart-dev' '--enable-asserts' 'C:\Users\CESAR\Desktop\R1_U
nes\bin\parametros_con_nombre.dart'
Conectando al localhost en el puerto: 3306 usando root/secret...
```

```
add(int x, int y) {
  return x + y;
}

subtract(int x, int y) {
  return x - y;
}

Function calculation(String op) {
  if (op == '+') return add;
  if (op == '-') return subtract;
  return add;
}

Run | Debug
void main() {
  var fn = calculation('+');
  print(fn(10, 20));

  fn = calculation('-');
  print(fn(30, 20));
}
```

```
nes\bin\funciones_
30
10
```




```
nes\bin\fu
20
```




```
nes\bin
30
PS C:\U
```

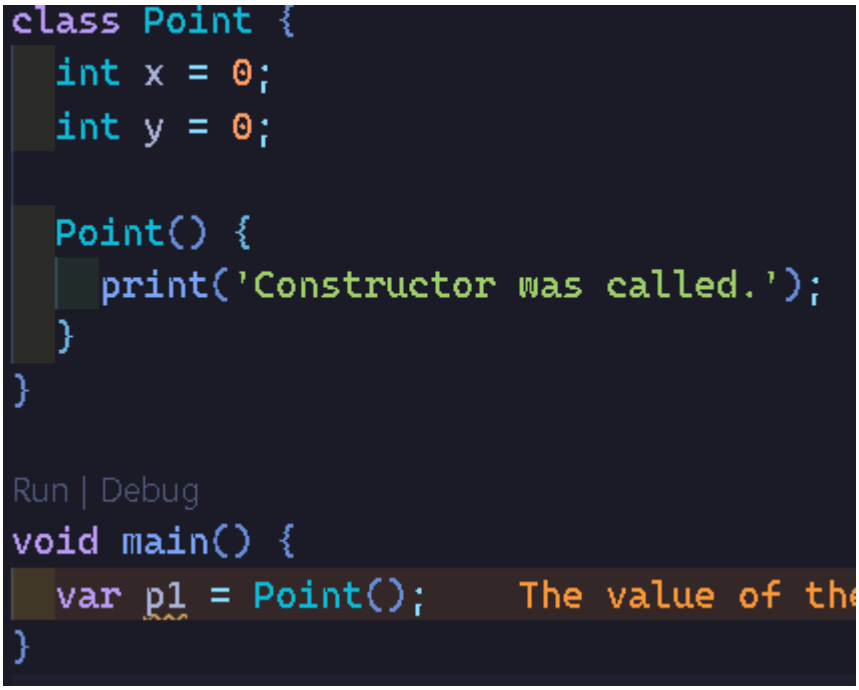
Ejercicio 4.- Clases

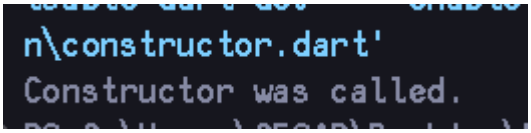
```dart
class Point {
  int x = 0;
  int y = 0;

  void move(int x1, int y1) {
    x = x1;
    y = y1;
  }

  void show() {
    print('Point($x,$y)');
  }
}

Run | Debug
void main() {
  var p1 = Point()
    ..x = 10
    ..y = 20;
  p1.move(100, 200);
  p1.show();
}
```

```
Point(100,200)
```

```dart
class Point {
  int x = 0;
  int y = 0;

  Point move(int x, int y) {
    this.x = x;
    this.y = y;
    return this;
  }

  Point reset() {
    x = 0;
    y = 0;
    return this;
  }

  Point show() {
    print('Point($x,$y)');
    return this;
  }
}

Run | Debug
void main() {
  var p1 = Point();
  p1.move(10, 20).show().reset();
}
```

```
n\this.dart'
Point(10,20)
```


class Point {
int x = 0;
int y = 0;
Point() {
print('Constructor was called.');
}
}
Run | Debug
void main() {
var p1 = Point();
The value of the
}


n\constructor.dart'
Constructor was called.

```dart
class Point {
  int _x = 0;
  int _y = 0;

  Point({int x = 0, int y = 0}) {
    this._x = x;    Unnecessary 'th
    this._y = y;    Unnecessary 'th
  }
  show() {
    print('Point(x=$_x,y=$_y)');
  }
}

Run | Debug
void main() {
  var p1 = Point(x: 10, y: 20);
  p1.show();
}
```

```
n\campos_privados.dart'
Point(x=10,y=20)
```

```dart
class Circle {
  double _radius = 0;

  Circle(double radius) {
    this.radius = radius;
  }

  set radius(double value) {
    if (value >= 0) {
      _radius = value;
    }
  }

  double get radius => _radius;

  get area => radius * radius * 3.14;
}
```

```dart
import 'getters_setters.dart';

Run | Debug
void main() {
  var circle = Circle(10);
  circle.radius = 100;
  print('El area es: ${circle.area}');
}
```

```
El area es: 31400.0
PS C:\Users\CESAR\Deskto
```

```dart
class Text {
  final String content;
  const Text({this.content = ''});
}

Run | Debug
void main() {
  var text1 = const Text(content: 'Hello');
  var text2 = const Text(content: 'Hello');
  print(identical(text1, text2)); // true
}
```

```
n\constructor_consta
true
```

```dart
class Point {
  int x;
  int y;
  Point({this.x = 0, this.y = 0}) {
    _counter++;
  }
  static int _counter = 0;
  static int get counter => _counter;
}

Run | Debug
void main() {
  var p1 = new Point(x: 10, y: 20);    Th
  var p2 = new Point(x: 100, y: 200);

  print(Point.counter); // 2
}
```

```
PS C:\Users\CE
2
2
```

Ejercicio 5.- Excepciones

```dart
Run | Debug
void main() {
  String message = "Hello";

  try {
    print("The character at the position 5 is ${message[5]}.");
  } on RangeError {
    print('The valid range for the index is [0..${message.length - 1}].');
  } catch (e) {
    print(e);
  }

  print('Bye!');
}
```

```
utter\excepciones\bin\try_catch.dart'
The valid range for the index is [0..4].
Bye!
```

```
RangeError (index): Invalid value: Not in inclusive range 0..4: 5
Bye!
```

```dart
class IndexOutOfRangeException implements Exception {
  String error;
  IndexOutOfRangeException(this.error);

  @override
  String toString() => error;
}

String getCharAt(String s, int index) {
  if (index < 0 || index > s.length) {
    throw IndexOutOfRangeException('Index is out of range [0..${s.length}]
  }
  return s[index];
}
```

```
Run | Debug
void main() {
  String message = 'Hello';
  try {
    String s = getCharAt(message, 10);
    print(s);
  } on IndexOutOfRangeException catch (e) {
    print(e);
  }
}
```

```
utter\exceptions\bin\throw.dart`
Index is out of range [0..5]
```