




COME TO
THE
DART
— SIDE

https://github.com/kevmoo/dart_side



COME ^{TO} THE
DART
— SIDE



Intro a dart para coders

Ivan Ramirez



@fluttermx



Que rollo con dart?

“Dart is a *client-optimized* language for fast apps on any platform”

Nacido en el 2011

Hecho para ‘competir a js’ compilando a JS

Hoy es usado principalmente para flutter

Se pueden crear ‘headless apps’ en dart





Que rollo con dart?

Dart es un lenguaje optimizado para desarrollo de clientes

Facil de aprender

JS
C
Java
Kotlin

Multiple OS

Windows
OSX
Linux
Fuschia



Hello world

```
// This is where the app starts executing.
```

```
main() {
```

```
// Declare and initialize a variable.
```

```
var world = "world";
```

```
// Print to console
```

```
print('Hello $world.');
```

```
}
```

var Declara variables sin tipo

print() Imprime en consola

\$variable o \${funcion}

Agrega el valor a un string
Esto se llama 'Interpolación'



Uso de funciones

// Define a function.

```
printInteger(int aNumber) {  
  print('The number is $aNumber.');
```

int tipo de variable (Otras: List, Bool)

print Imprime en consola

// This is where the app starts executing.

```
main() {  
  var number = 42; // Declare and initialize a variable.  
  printInteger(number); // Call a function.  
}
```

var Declaración variable



Estructuras

```
List<int> someList = List();
```

List<int> Arreglo de ints

```
List someOtherList = List();
```

List sin declaración es arreglo de variables

Map Declaración de un diccionario <K,V> Key-Value

```
Map<String,String> someMap = Map();
```

```
Map<String,dynamic> someOtherMap = Map();
```

```
// {"name" : "ivan"}
```

```
// {"items" : [1,2,3]}
```



Private, Protected, Public

Eso no existe

`_` declara la variables privada al objeto

```
var _someVar = "I am private"
```

```
String _hello(){  
    return 'world';  
}
```

`_hello` declara la function privada al objeto



'Type Safety'

Chida pero gacho cuando vienes de lenguajes viejitos...

```
bool yesNo;           // es null
```

```
//_crash por no checar el valor  
if (yesNo){}
```

```
// valido  
if (yesNo == null){}
```



'Type Safety'

Codigo listo para morir

```
bool yesNo = false;
```

```
// valido  
if (yesNo){}
```

Ya se vuelve válido

```
// valido  
if (yesNo == false){}
```

```
if (yesNo != null){  
    if (yesNo == false){  
    }  
}
```

Esto siempre se debe de hacer



'Operador ?'

Asquerosamente chido

```
bool someVar;
```

```
someVar ?? debugPrint("empty");
```

?? Checa si es null

```
someVar != null ? debugPrint("not empty") : debugPrint("empty");
```

El clásico <condición> ? si : no;

```
someVar != null ? (someVar == true ? debugPrint("true") : debugPrint("false")) : debugPrint("empty");
```

No hay parentesis, porque wtf

Regresa el valor de bool



Que bonito lo bonito

Notación cascada

```
class Mexico{  
  void norte() {debugPrint("Chihuahua");}  
  
  void sur() {debugPrint("Chiapas");}  
}
```

```
Mexico pais = Mexico();  
pais.sur();  
pais.norte();
```

Nada Especial

```
Mexico pais = Mexico()..sur()..norte();
```

Muy útil en clases



Closures

```
var talk = (s) => print(s);  
talk("Hi there"); // prints "Hi there"
```

```
Function say(String something) { // "Function" is an optional return type annotation  
  return () => print(something);  
}
```

```
void main() {  
  var talk = say("How are you?"); // Function say(...) closes over variable talk.  
  talk(); // prints "How are you?"  
}
```





Aprende dart experimentando

<https://dartpad.dev/>





Para mas informacion

Dart.dev

Checar los ejemplos

<https://www.reddit.com/r/dartlang/>

Usando Flutter



@fluttermx

