Keywords

abstract	as	assert	async
await	base	break	case
catch	class	const	continue
covariant	default	deferred	do
dynamic	else	enum	export
extends	extension	external	factory
false	final	get	finally
for	Function	import	hide
if	implements	late	in
interface	is	null	library
mixin	new	part	of
on	operator	sealed	required
rethrow	return	super	set
show	static	throw	switch
sync	this	typedef	true
try	type	with	var
void	when		while
yield			

Naming Conventions

Classes	UpperCamelCase	MyClass
Enums	UpperCamelCase	MyEnum
Extensions	UpperCamelCase	MyExtension
Typedefs	UpperCamelCase	MyTypeDef
Fields	lowerCamelCase	myField
Private fields	_lowerCamelCase	_myField
Functions	lowerCamelCase	myFunction
Parameters	lowerCamelCase	myParameter
Variables	lowerCamelCase	myVariable
Constants	lowerCamelCase	myConstant
Directories, Source files, Packages & Imports	lowercase_with_underscores	my_package, main.dart

Operators

Arithmetic operators	+, -, *, /, ~/, %
Assignment operators	=, +=, -=, *=, /=, ~/=, %=, &=, |=, ^=, <<=, >>=, >>>=
Bitwise operators	&, |, ^, ~, <<, >>, >>>
Comparison operators	==, !=, <, >, <=, >=
Conditional operator	? :

Operators (cont)

Logical operators	`!, &&, ||`
Function application	`()`
Subscript access	`[]`
Conditional subscript access	`?[]`
Member access	`.`
Conditional member access	`?.`
Non-null assertion operator	`!`
Unary operators	`++, --`

Comments

```
// This is a single line comment.
```
```
/* This is a multi-line comment. */
```
```
/// This is a doc comment.
```

String Interpolation

```
$expr
${expr}
```

```
int a = 2;
int b = 2;
print("Sum of $a and $b is ${a + b}"); // Sum of 2 and 2 is 4
```

String Concatenation

```
expr1 + expr2 + ... + exprn
```

```
String str1 = "Hello";
String str2 = "World";
print(str1 + " " + str2);
```

Main Function

```
void main() {
    // Some code
}
```

Variables

```
T variableName = value;
```

```
int age = 21;
double height = 1.75;
var dateOfBirth = 2003;
String name = "Jack";
bool isHuman = true;
```

Imports

```
import library_name;
```

```
import 'dart:io';
import 'dart:math';
import 'dart:core';
```

Branches

```
if (condition) {
    ...
}
else if (condition) {
    ...
} else {
    ...
}
```

```
switch (value) {
case value1: ...
break;
case value2: ...
break;
...
default: ...
break;
}
```

Loops

```
for (...; condition; ...) {
    ...
}
```

```
while (condition) {
    ...
}
```

```
do {
    ...
} while (condition);
```

Functions

```
T functionName (parameter(s))
 {
    ...
}
```

```
static T functi-
onName (parameter(s))  {
    ...
}
```