〈 Libraries　　Built-in types 〉

Keywords

The following table lists the words that the Dart language treats specially.

abstract^{2}	else	import^{2}	show^{1}
as^{2}	enum	in	static^{2}
assert	export^{2}	interface^{2}	super
async^{1}	extends	is	switch
await^{3}	extension^{2}	late^{2}	sync^{1}
base^{2}	external^{2}	library^{2}	this
break	factory^{2}	mixin^{2}	throw
case	false	new	true
catch	final (variable)	null	try
class	final (class)^{2}	on^{1}	typedef^{2}
const	finally	operator^{2}	var
continue	for	part^{2}	void
covariant^{2}	Function^{2}	required^{2}	when
default	get^{2}	rethrow	while
deferred^{2}	hide^{1}	return	with
do	if	sealed^{2}	yield^{3}
dynamic^{2}	implements^{2}	set^{2}	

Avoid using these words as identifiers. However, if necessary, the keywords marked with superscripts can be identifiers:

- Words with the superscript **1** are **contextual keywords**, which have meaning only in specific places. They're valid identifiers everywhere.
- Words with the superscript **2** are **built-in identifiers**. These keywords are valid identifiers in most places, but they can't be used as class or type names, or as import prefixes.
- Words with the superscript **3** are limited reserved words related to asynchrony support. You can't use `await` or `yield` as an identifier in any function body marked with `async`, `async*`, or `sync*`.

All other words in the table are **reserved words**, which can't be identifiers.

〈 Libraries　　Built-in types 〉