

C++ helloworld	Dart helloworld																				
<pre>int main() { cout << "Hello World"; // prints Hello World return 0; }</pre>	<pre>main() or (void main()) { print("Hello World!"); }</pre>																				
C++ data types	dart data types																				
<table border="0"> <tr> <td>int</td> <td>float</td> </tr> <tr> <td>double</td> <td>char</td> </tr> <tr> <td>string</td> <td>boolean</td> </tr> </table>	int	float	double	char	string	boolean	<table border="0"> <tr> <td>Number</td> <td>strings</td> </tr> <tr> <td>boolean</td> <td>list</td> </tr> <tr> <td>map</td> <td></td> </tr> </table>	Number	strings	boolean	list	map									
int	float																				
double	char																				
string	boolean																				
Number	strings																				
boolean	list																				
map																					
<p>c++ is statically typed 'type variable_list;'</p>	<p>lists are equivalent of arrays(ordered lists) Maps represent a collection of objects, Map data type represents a set of va Dart is an optionally typed language. If the type of a variable is not expl type is dynamic. The dynamic keyword can also be used as a type ann</p>																				
C++ variables	Dart variables																				
<pre>type variable_name = value; int i, j, k; char c, ch; float f, salary; double d;</pre>	<pre>String name = 'Smith'; dynamic x = "tom"; final variable_name; const variable_name;</pre>																				
<p>lvalue – Expressions that refer to a memory location is called "lva- lue" expression.</p> <p>rvalue – The term rvalue refers to a data value that is stored at some address in memory.</p>	<p>-All uninitialized variables have an initial value of null. This is because Dart c -Variables declared without a static type are implicitly declared as dynamic using the dynamic keyword in place of the var keyword.</p> <p>-The final and const keyword are used to declare constants, These keyword: variable's data type or instead of the var keyword.</p> <p>-The const keyword is a compile time const.</p>																				
C++ operators	Dart operators																				
<table border="0"> <tr> <td>=</td> <td>==</td> </tr> <tr> <td>+</td> <td>-</td> </tr> <tr> <td>--</td> <td>++</td> </tr> <tr> <td>%</td> <td>*</td> </tr> <tr> <td>/</td> <td></td> </tr> </table>	=	==	+	-	--	++	%	*	/		<p>in addition to c style operators</p> <table border="0"> <tr> <td>-expr</td> <td>Unary minus, also known as negation (reverse the sign of the exp</td> </tr> <tr> <td>~/</td> <td>Divide, returning an integer result</td> </tr> <tr> <td>is</td> <td>it is</td> </tr> <tr> <td>is!</td> <td>it is not</td> </tr> <tr> <td>??=</td> <td>Assign the value only if the variable is null</td> </tr> </table>	-expr	Unary minus, also known as negation (reverse the sign of the exp	~/	Divide, returning an integer result	is	it is	is!	it is not	??=	Assign the value only if the variable is null
=	==																				
+	-																				
--	++																				
%	*																				
/																					
-expr	Unary minus, also known as negation (reverse the sign of the exp																				
~/	Divide, returning an integer result																				
is	it is																				
is!	it is not																				
??=	Assign the value only if the variable is null																				
Storage Classes in C++																					
<table border="0"> <tr> <td>auto</td> <td>default storage class for all local variables</td> </tr> <tr> <td>register</td> <td>used to define local variables that should be stored in a register instead of RAM.</td> </tr> <tr> <td>static</td> <td>keep a local variable in existence during the life-time of the program instead of creating and destroying it each time</td> </tr> <tr> <td>extern</td> <td>is used to give a reference of a global variable that is visible to ALL the program files.</td> </tr> <tr> <td>mutable</td> <td>It allows a member of an object to override const member function(applies only to class objects)</td> </tr> </table>	auto	default storage class for all local variables	register	used to define local variables that should be stored in a register instead of RAM.	static	keep a local variable in existence during the life-time of the program instead of creating and destroying it each time	extern	is used to give a reference of a global variable that is visible to ALL the program files.	mutable	It allows a member of an object to override const member function(applies only to class objects)											
auto	default storage class for all local variables																				
register	used to define local variables that should be stored in a register instead of RAM.																				
static	keep a local variable in existence during the life-time of the program instead of creating and destroying it each time																				
extern	is used to give a reference of a global variable that is visible to ALL the program files.																				
mutable	It allows a member of an object to override const member function(applies only to class objects)																				



Dart for in loop

```
for (variablename in object){
    statement or block to execute
}

void main() {
    var obj = [12,13,14];

    for (var prop in obj) {
        print(prop);
    }
}
```



By **Nima** (nimakarimian)
cheatography.com/nimakarimian/
www.nimakarimian.ir

Not published yet.
Last updated 2nd April, 2020.
Page 2 of 2.

Sponsored by **Readab**
Measure your website
<https://readable.com>