Dart Programming - Numbers

Dart numbers can be classified as −

- **int** − Integer of arbitrary size. The **int** data type is used to represent whole numbers.
- **double** − 64-bit (double-precision) floating-point numbers, as specified by the IEEE 754 standard. The **double** data type is used to represent fractional numbers

The **num** type is inherited by the **int** and **double** types. The **dart core library** allows numerous operations on numeric values.

The syntax for declaring a number is as given below −

```
int var_name;      // declares an integer variable
double var_name;   // declares a double variable
```

Example

```
void main() {
   // declare an integer
   int num1 = 10;

   // declare a double value
   double num2 = 10.50;

   // print the values
   print(num1);
   print(num2);
}
```

It will produce the following output −

```
10
10.5
```

Note − The **Dart VM** will throw an exception if fractional values are assigned to integer variables.

Parsing

The **parse()** static function allows parsing a string containing numeric literal into a number. The following illustration demonstrates the same −

Live Demo

```
void main() {
   print(num.parse('12'));
   print(num.parse('10.91'));
}
```

The above code will result in the following output −

```
12
10.91
```

The parse function throws a **FormatException** if it is passed any value other than numerals. The following code shows how to pass an alpha-numeric value to the **parse()** function.

Example

Live Demo

```
void main() {
   print(num.parse('12A'));
   print(num.parse('AAAA'));
}
```

The above code will result in the following output −

```
Unhandled exception:
FormatException: 12A
#0 num.parse (dart:core/num.dart:446)
#1 main (file:///D:/Demos/numbers.dart:4:13)
#2 _startIsolate.<anonymous closure> (dart:isolatepatch/isolate_patch.dart:20
#3 _RawReceivePortImpl._handleMessage (dart:isolatepatch/isolate_patch.dart:1
```

Number Properties

The following table lists the properties supported by Dart numbers.

Sr.No	Property & Description
1	hashcode Returns a hash code for a numerical value.
2	isFinite True if the number is finite; otherwise, false.
3	isInfinite True if the number is positive infinity or negative infinity; otherwise, false.
4	**isNan** True if the number is the double Not-a-Number value; otherwise, false.
5	isNegative True if the number is negative; otherwise, false.
6	sign Returns minus one, zero or plus one depending on the sign and numerical value of the number.
7	isEven Returns true if the number is an even number.
8	isOdd Returns true if the number is an odd number.

Number Methods

Given below are a list of commonly used methods supported by numbers −

Sr.No	Method & Description
1	abs Returns the absolute value of the number.
2	ceil Returns the least integer no smaller than the number.
3	compareTo Compares this to other number.
4	Floor Returns the greatest integer not greater than the current number.
5	remainder Returns the truncated remainder after dividing the two numbers.
6	Round Returns the integer closest to the current numbers.
7	toDouble Returns the double equivalent of the number.
8	toInt Returns the integer equivalent of the number.
9	toString Returns the string equivalent representation of the number.
10	truncate Returns an integer after discarding any fractional digits.