WHILE LOOP IN DART

While Loop

In **while loop**, the loop's body will run until and unless the condition is true. You must write conditions first before statements. This loop checks conditions on every iteration. If the condition is true, the code inside {} is executed, if the condition is false, then the loop stops.

Syntax

```
while(condition){
      //statement(s);
      // Increment (++) or Decrement (--) Operation;
}
```

- A while loop evaluates the condition inside the parenthesis ().
- If the condition is true, the code inside {} is executed.
- The condition is re-checked until the condition is false.
- When the condition is false, the loop stops.

Example 1: To Print 1 To 10 Using While Loop

This program prints 1 to 10 using while loop.

```
void main() {
  int i = 1;
  while (i <= 10) {
    print(i);
    i++;
  }
}
```

❯ Show Output

Run Online

Example 2: To Print 10 To 1 Using While Loop

This program prints 10 to 1 using while loop.

```
void main() {
  int i = 10;
  while (i >= 1) {
    print(i);
    i--;
  }
}
```

> Show Output

Run Online

Example 3: Display Sum of n Natural Numbers Using While Loop

Here, the value of the total is 0 initially. Then, the while loop is iterated from **i = 1 to 100**. In each iteration, **i** is added to the total, and the value of **i** is increased by 1. Result is **1+2+3+....+99+100**.

```
void main(){

  int total = 0;
  int n = 100; // change as per required
  int i =1;

  while(i<=n){
    total = total + i;
    i++;
  }

  print("Total is $total");
```

```
    }
```

> Show Output

[Run Online]

Example 4: Display Even Numbers Between 50 to 100 Using While Loop

This program will print even numbers between 50 to 100 using while loop.

```
void main(){
  int i = 50;
  while(i<=100){
  if(i%2 == 0){
      print(i);
    }
    i++;
  }
}
```

> Show Output

[Run Online]

Video

Watch our video on while loop in Dart.