BREAK AND CONTINUE IN DART

Dart Break and Continue

In this tutorial, you will learn about the **break and continue** in dart. While working on loops, we need to skip some elements or terminate the loop immediately without checking the condition. In such a situation, you can use the break and continue statement.

Break Statement

Sometimes you will need to break out of the loop immediately without checking the condition. You can do this using break statement.

The break statement is used to exit a loop. It stops the loop immediately, and the program's control moves outside the loop. Here is syntax of break:

```
break;
```

Example 1: Break In Dart For Loop

Here, the loop condition is true until the value of i is less than or equal to 10. However, the break says to go outside the loop when the value of i becomes 5.

```
void main() {
  for (int i = 1; i <= 10; i++) {
    if (i == 5) {
      break;
    }
    print(i);
  }
}
```

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

› Show Output

Run Online

Example 2: Break In Dart Negative For Loop

Here, the loop condition is true until the value of i is more than or equal to 1. However, the break says to go outside the loop when the value of i becomes 7.

```
void main() {
  for (int i = 10; i >= 1; i--) {
    if (i == 7) {
      break;
    }
    print(i);
  }
}
```

› Show Output

Run Online

Example 3: Break In Dart While Loop

Here, this while loop condition is true until the value of i is less than or equal to 10. However, the break says to go outside the loop when the value of i becomes 5.

```
void main() {
 int i =1;
 while(i<=10){
  print(i);
   if (i == 5) {
      break;
    }
    i++;
 }
}
```

› Show Output

Run Online

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

Example 4: Break In Switch Case

As we already learn in dart switch case, it is important to add **break** keyword in switch statement. This example prints the month name based on the number of the month using a switch case.

```
void main() {
  var noOfMoneth = 5;
  switch (noOfMoneth) {
    case 1:
      print("Selected month is January.");
      break;
    case 2:
      print("Selected month is February.");
      break;
    case 3:
      print("Selected month is march.");
      break;
    case 4:
      print("Selected month is April.");
      break;
    case 5:
      print("Selected month is May.");
      break;
    case 6:
      print("Selected month is June.");
      break;
    case 7:
      print("Selected month is July.");
      break;
    case 8:
      print("Selected month is August.");
      break;
    case 9:
      print("Selected month is September.");
      break;
    case 10:
      print("Selected month is October.");
      break;
    case 11:
      print("Selected month is November.");
      break;
    case 12:
      print("Selected month is December.");
      break;
    default:
      print("Invalid month.");
      break;
  }
}
```

› Show Output

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

Run Online

Continue Statement

Sometimes you will need to skip an iteration for a specific condition. You can do this utilizing continue statement.

The continue statement skips the current iteration of a loop. It will bypass the statement of the loop. It does not terminate the loop but rather continues with the next iteration. Here is the syntax of continue statement:

```
continue;
```

Example 1: Continue In Dart

Here, the loop condition is true until the value of i is less than or equal to 10. However, the continue says to go to the next iteration of the loop when the value of i becomes 5.

```
void main() {
  for (int i = 1; i <= 10; i++) {
    if (i == 5) {
      continue;
    }
    print(i);
  }
}
```

› Show Output

Run Online

Example 2: Continue In For Loop Dart

Here, the loop condition is true until the value of i is more than or equal to 1. However, the continue says to go to the next iteration of the loop when the value of i becomes 4.

```
void main() {
  for (int i = 10; i >= 1; i--) {
```

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

```
    if (i == 4) {
      continue;
    }
    print(i);
  }
}
```

› Show Output

Run Online

Example 3: Continue In Dart While Loop

Here, this while loop condition is true until the value of i is less than or equal to 10. However, the continue says to go to the next iteration of the loop when the value of i becomes 5.

```
void main() {
  int i = 1;
  while (i <= 10) {
    if (i == 5) {
      i++;
      continue;
    }
    print(i);
    i++;
  }
}
```

› Show Output

Run Online

Video

Watch our video on break and continue in Dart.

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF