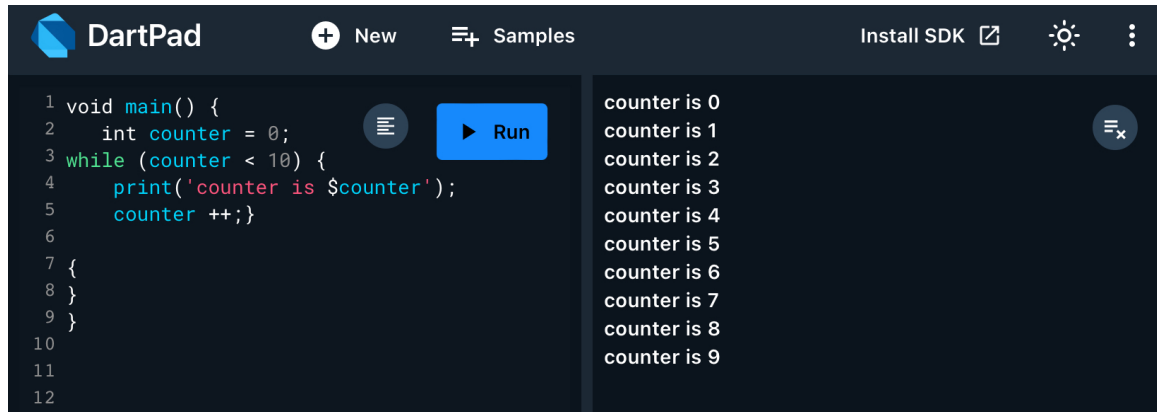


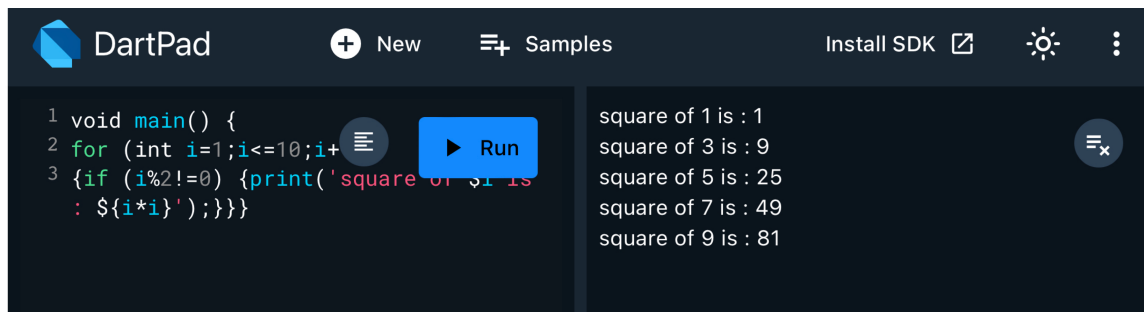
Ass 3



The screenshot shows the DartPad web interface. The code editor on the left contains a Dart program with a `while` loop that prints the value of a `counter` variable from 0 to 9. A blue `Run` button is visible. The output console on the right displays the results of the program execution.

```
1 void main() {  
2   int counter = 0;  
3   while (counter < 10) {  
4     print('counter is $counter');  
5     counter ++;  
6   }  
7 }  
8 }  
9 }  
10  
11  
12
```

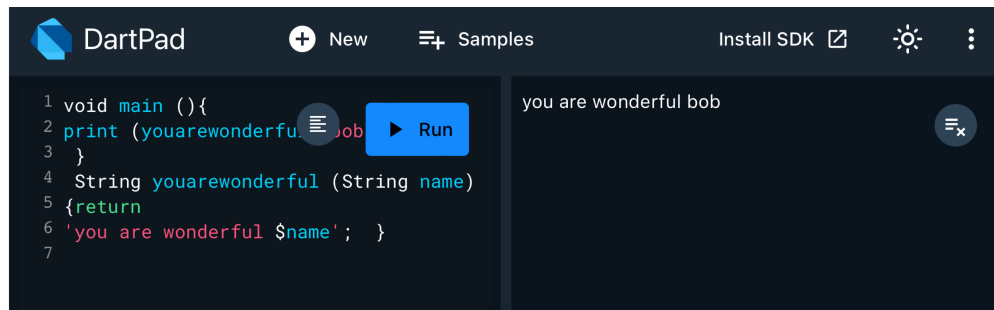
counter is 0
counter is 1
counter is 2
counter is 3
counter is 4
counter is 5
counter is 6
counter is 7
counter is 8
counter is 9



The screenshot shows the DartPad web interface. The code editor on the left contains a Dart program with a `for` loop that prints the square of numbers from 1 to 9. A blue `Run` button is visible. The output console on the right displays the results of the program execution.

```
1 void main() {  
2   for (int i=1;i<=10;i++)  
3   {if (i%2!=0) {print('square of $i is  
: ${i*i}');}}}
```

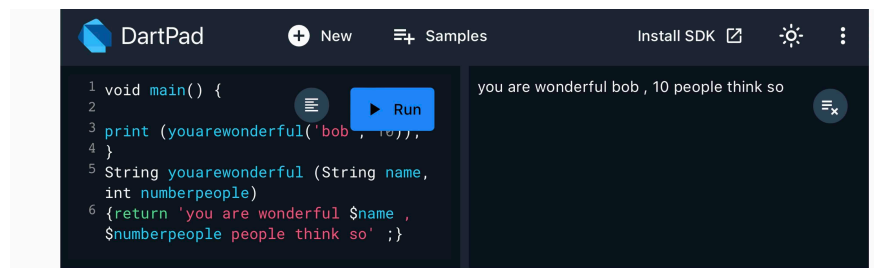
square of 1 is : 1
square of 3 is : 9
square of 5 is : 25
square of 7 is : 49
square of 9 is : 81



The screenshot shows the DartPad web interface. The code editor on the left contains a Dart program with a function `youarewonderful` that takes a `String` parameter and returns a string. A blue `Run` button is visible. The output console on the right displays the result of the program execution.

```
1 void main () {  
2   print (youarewonderful('bob'))  
3 }  
4 String youarewonderful (String name)  
5 {return  
6   'you are wonderful $name'; }  
7
```

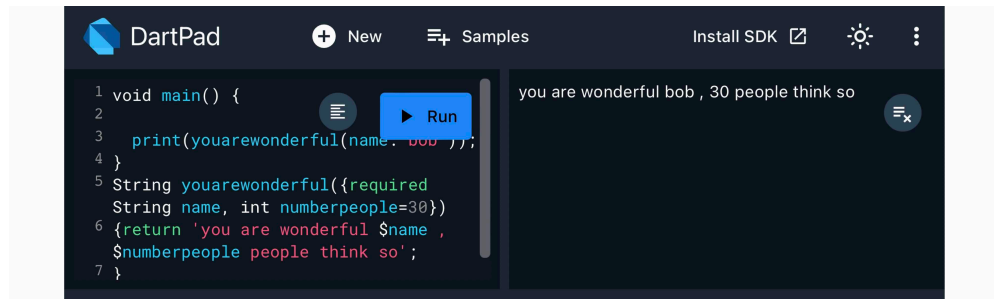
you are wonderful bob



The screenshot shows the DartPad web interface. The code editor on the left contains a Dart program with a function `youarewonderful` that takes a `String` and an `int` parameter and returns a string. A blue `Run` button is visible. The output console on the right displays the result of the program execution.

```
1 void main() {  
2  
3   print (youarewonderful('bob', 10));  
4 }  
5 String youarewonderful (String name,  
6   int numberpeople)  
7 {return 'you are wonderful $name ,  
   $numberpeople people think so' ;}
```

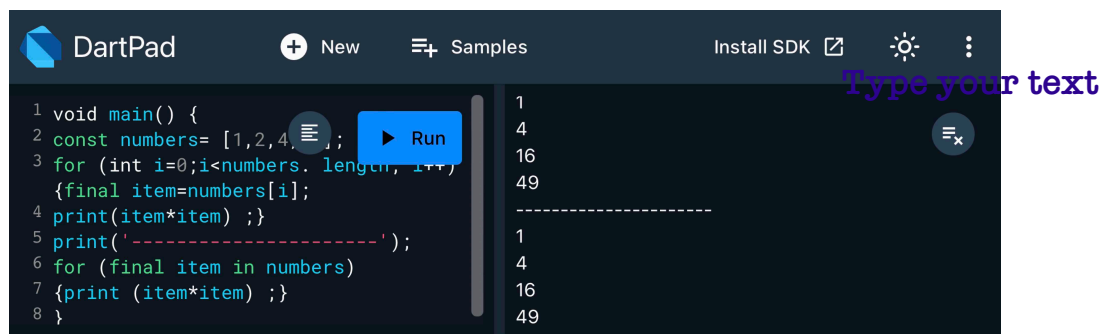
you are wonderful bob , 10 people think so



The screenshot shows the DartPad web interface. The code editor on the left contains a Dart program with a `main` function that calls `youarewonderful(name: 'bob')`. The function definition is also visible. A blue 'Run' button is highlighted. The output pane on the right displays the result: 'you are wonderful bob , 30 people think so'.

```
1 void main() {  
2  
3   print(youarewonderful(name: 'bob'));  
4 }  
5 String youarewonderful({required  
6   String name, int numberpeople=30})  
7 {return 'you are wonderful $name ,  
   $numberpeople people think so';  
}
```

you are wonderful bob , 30 people think so



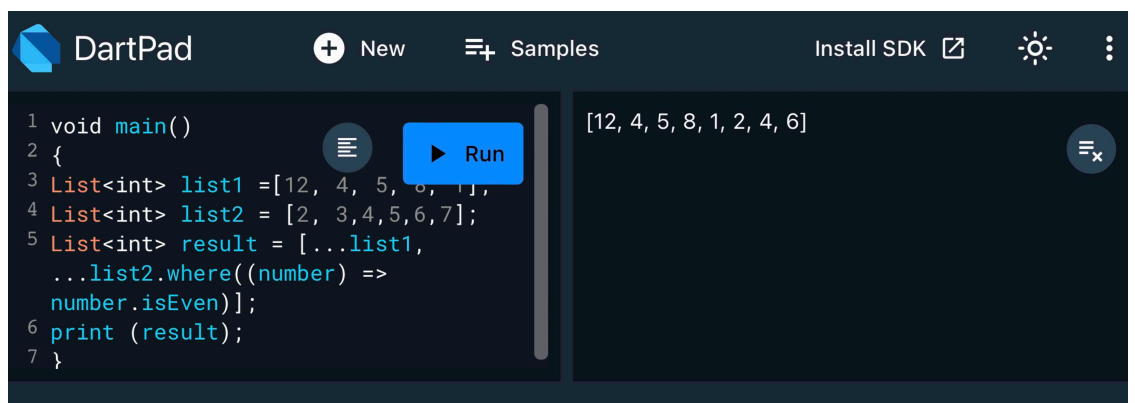
The screenshot shows the DartPad web interface. The code editor on the left contains a Dart program that iterates over a list of numbers and prints their squares. A purple text annotation 'Type your text' is overlaid on the right side of the image. The output pane on the right displays the results: 1, 4, 16, 49, followed by a separator line and then 1, 4, 16, 49.

```
1 void main() {  
2   const numbers= [1,2,4,];  
3   for (int i=0;i<numbers.length,i++)  
4     {final item=numbers[i];  
5     print(item*item) ;}  
6   print('-----');  
7   for (final item in numbers)  
8     {print (item*item) ;}  
9 }
```

1
4
16
49

1
4
16
49

Type your text



The screenshot shows the DartPad web interface. The code editor on the left contains a Dart program that creates two lists, `list1` and `list2`, and then creates a new list `result` by combining `list1` with elements from `list2` that are even. The output pane on the right displays the resulting list: [12, 4, 5, 8, 1, 2, 4, 6].

```
1 void main()  
2 {  
3   List<int> list1 =[12, 4, 5, 8, 1],  
4   List<int> list2 = [2, 3,4,5,6,7];  
5   List<int> result = [...list1,  
6     ...list2.where((number) =>  
7       number.isEven)];  
8   print (result);  
9 }
```

[12, 4, 5, 8, 1, 2, 4, 6]