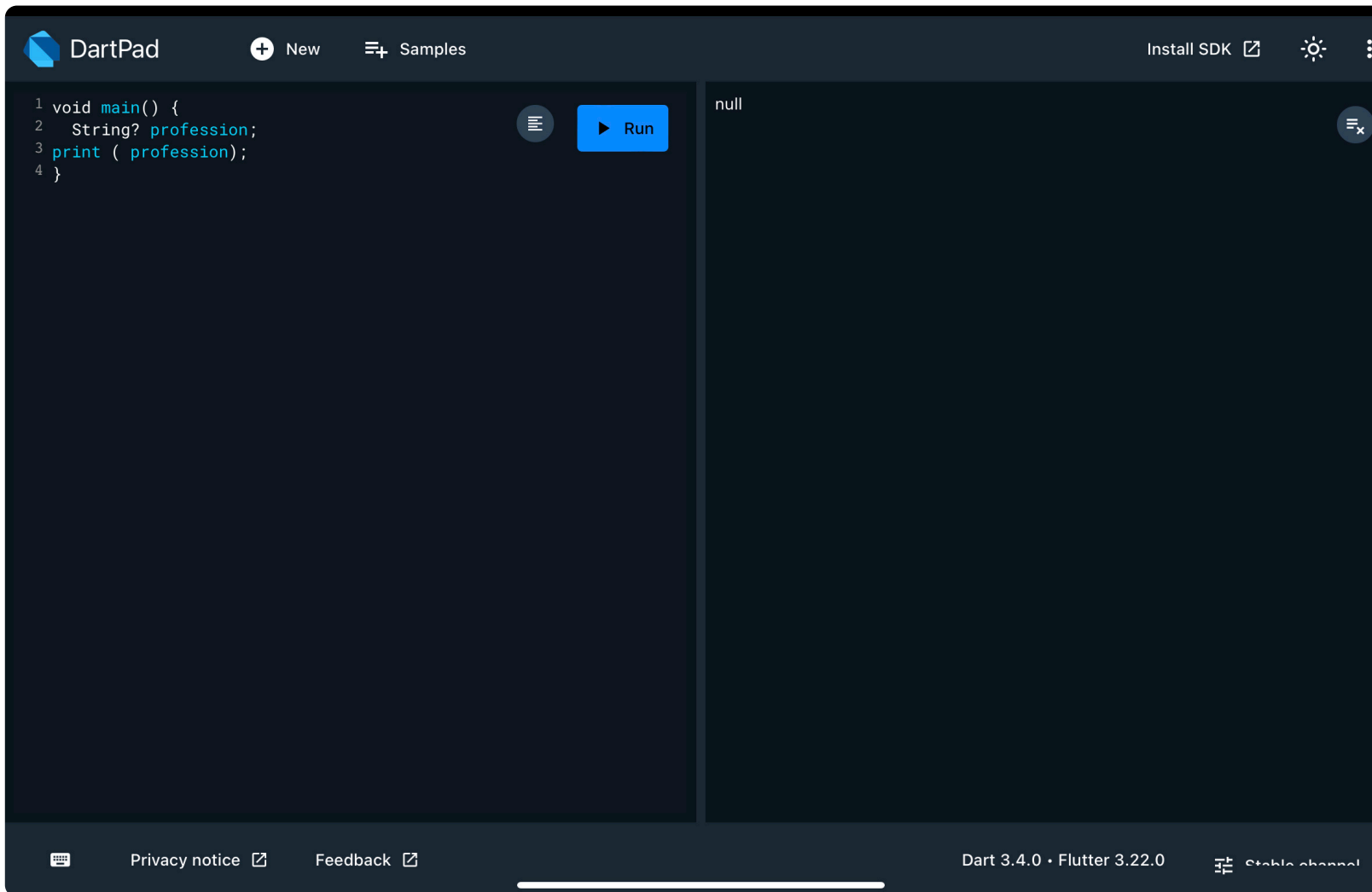


Exercise 1

1. Create a `String?` variable called `profession`, but don't give it a value.
2. Then you'll have `profession` `null`. Get it? Professional?
3. Print the output.
4. Comment on the output you have seen.

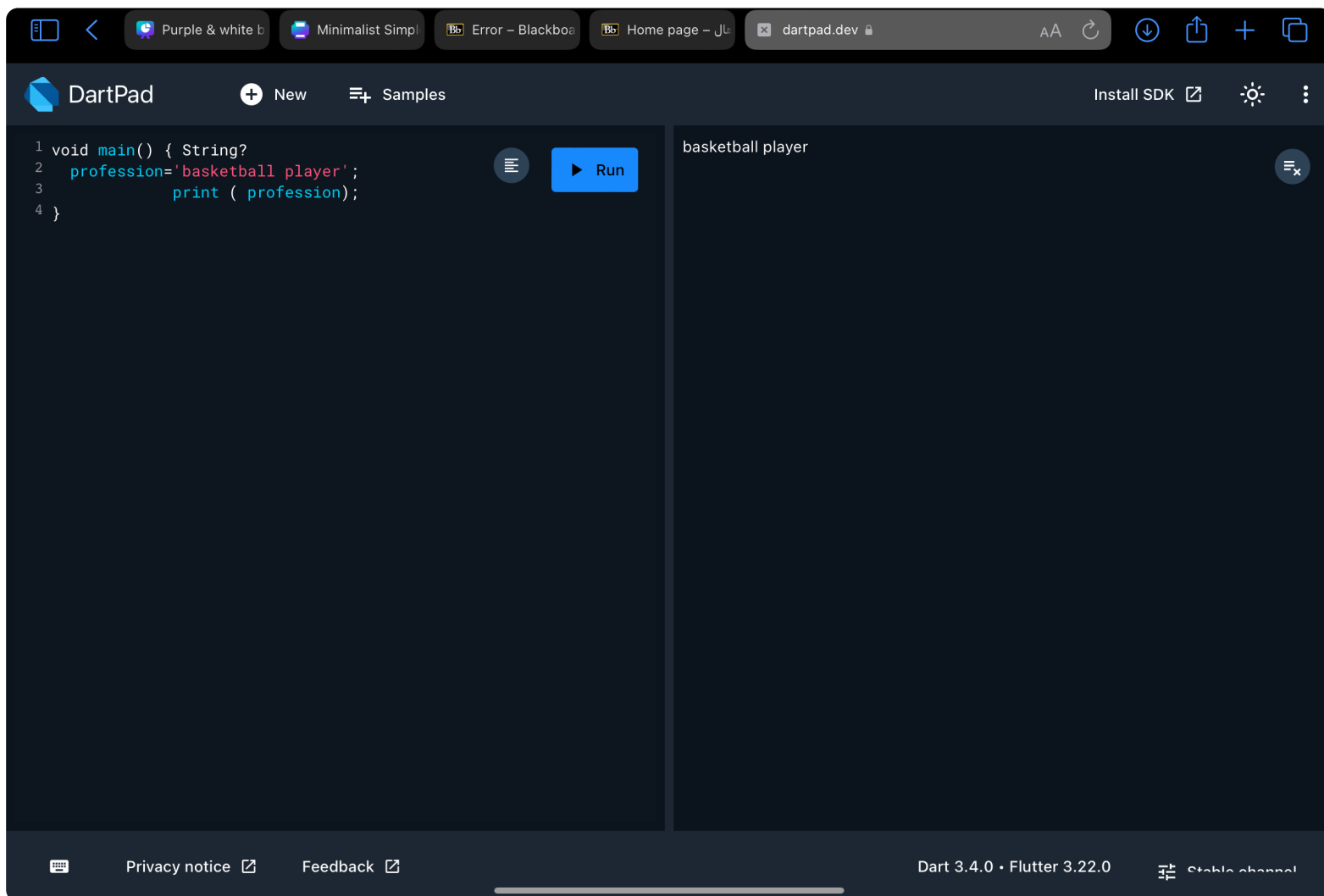
```
void main() {  
  String? profession;  
  print ( profession);  
  
}
```



Exercise 2

1. Give `profession` a value of "basketball player".
2. Print the output.
3. Comment on the output you have seen.

```
void main() {  
String? profession=  
  'basketball player';  
  print ( profession);  
  
}
```



Exercise 3

1. Write the following line and then hover your cursor over the variable name. What type does Dart infer `iLove` to be? `String` or `String?`?

```
const iLove = 'Dart';
```

Challenge 1: Naming Customs

People around the world have different customs for giving names to children. It would be difficult to create a data class to accurately represent them all, but try it like this:

1. Create a class called `Name` with `givenName` and `surname` properties.
2. Some people write the surname last and some write it first.
3. Add a Boolean property called `surnamesFirst` to keep track of this.
4. Not everyone in the world has a surname.
5. Add a `toString` method that prints the full name.

```
class Name{  
  String givenName;  
  String? surname;  
  bool surnamesFirst;  
  Name(this.givenName,this.surname,this.surnamesFirst);  
  @override  
  String toString (){  
    if (surname != null){  
      if(surnamesFirst){  
        return '$surname $givenName';  
      }  
      else{
```

```

        return '$givenName $surname';
    }
}
else{
    return givenName;
}
}
}
}

```

```

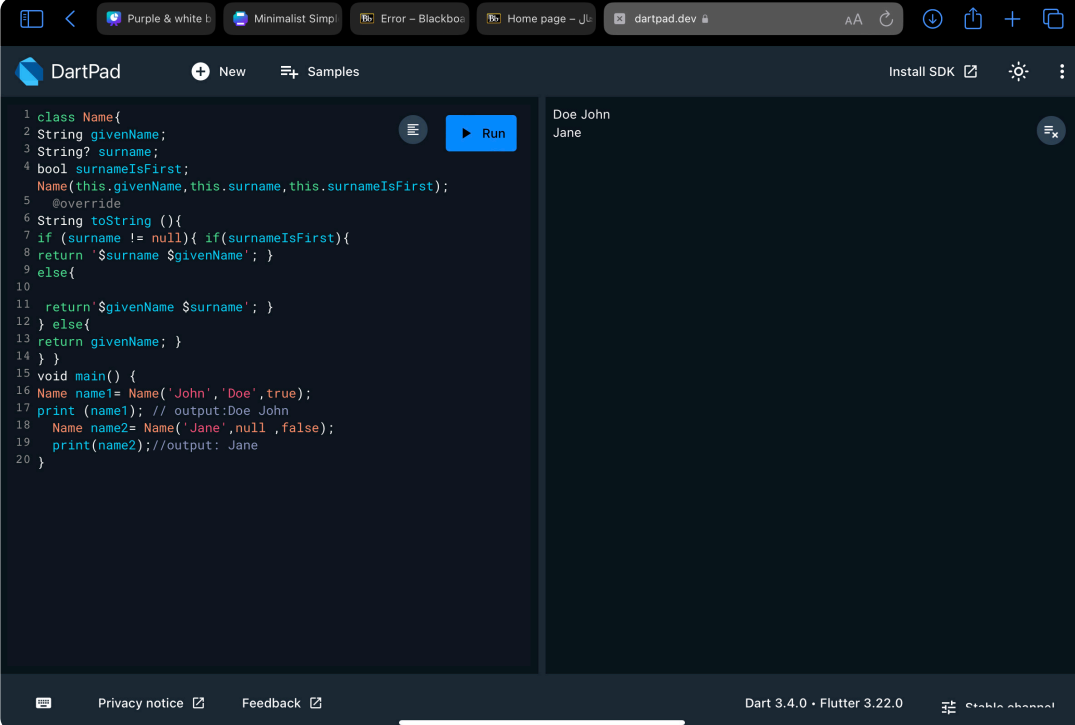
void main() {
Name name1= Name('John','Doe',true);

print (name1);// output:Doe John

Name name2= Name('Jane',null ,false);

print(name2);//output: Jane
}

```



The screenshot shows the DartPad web application interface. The left pane contains the following Dart code:

```

1 class Name{
2   String givenName;
3   String? surname;
4   bool surnameIsFirst;
5   Name(this.givenName,this.surname,this.surnameIsFirst);
6   @override
7   String toString (){
8     if (surname != null){ if(surnameIsFirst){
9       return '$surname $givenName'; }
10    } else{
11      return '$givenName $surname'; }
12    } else{
13      return givenName; }
14    } }
15 void main() {
16   Name name1= Name('John','Doe',true);
17   print (name1); // output:Doe John
18   Name name2= Name('Jane',null ,false);
19   print(name2);//output: Jane
20 }

```

The right pane displays the output of the code execution:

```

Doe John
Jane

```

The interface includes a top navigation bar with tabs for 'Purple & white b...', 'Minimalist Simp...', 'Error - Blackbo...', 'Home page - JL...', and 'dartpad.dev'. Below the tabs are buttons for 'New' and 'Samples', and a link to 'Install SDK'. The bottom status bar shows 'Dart 3.4.0 • Flutter 3.22.0' and a link to the 'Stable channel'.