



Introduction to PHP Programming

Learning Objectives

- Describe what PHP is and how it works in web development
- Write basic PHP scripts using variables and output
- Use conditional statements and loops to control program flow
- Create simple PHP functions to organize reusable code



What is PHP?

- PHP PHP: Hypertext Preprocessor
- A popular server-side scripting language
- Used to create dynamic websites and web applications
- Embedded inside HTML and runs on a web server (like XAMPP)

Why Learn PHP?

- Powers popular platforms (WordPress, Facebook originally used PHP)
- Simple and beginner-friendly
- Works well with HTML and MySQL
- Prepares you for building real-world systems

Setting Up Your PHP Environment

- Download and install XAMPP
- Save .php files inside htdocs/ folder
- Start Apache to run your server or go to terminal `php filename.extension`

VARIABLES

```
main.php
1  <?php
2
3  |
4  $name = "Juan";           // String
5  $age = 20;                // Integer
6  $height = 5.8;            // Float (decimal number)
7  $grade = 'A';             // Character (also a string in PHP)
8  $price = 99.99;           // Decimal (also float)
9
10 ?>
```

VARIABLES

```
main.php
1  <?php
2
3  |
4  $name = "Juan";           // String
5  $age = 20;                // Integer
6  $height = 5.8;            // Float (decimal number)
7  $grade = 'A';             // Character (also a string in PHP)
8  $price = 99.99;           // Decimal (also float)
9
10 ?>
```

Input and Output in PHP

Output is how PHP displays data to the screen.

We use:

echo → for simple output

Examples:

```
echo "Hello, World!";
```


Input and Output in PHP

Form Input (Web Browser)

- Uses `$_POST` or `$_GET` from an HTML form.

Command Line Input (Terminal)

- Uses `fgets(STDIN)` or `readline()`
- Used when running PHP from the terminal.

Input and Output in PHP

fgets(STDIN)

```
index.php
2  echo "Enter your name: ";
3  $name = trim(fgets(STDIN));
4  echo "Hello, $name!";
5  ?>
6
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\Jerico PC\Documents\OOP_1ST-SEM-2025_2026> php index.php
Enter your name: Jerico
Hello, Jerico!
PS C:\Users\Jerico PC\Documents\OOP_1ST-SEM-2025_2026>
```

readline()

```
index.php
1  <?php
2  $name = readline("Enter your name: ");
3  echo "Hello, $name!";
4  ?>
5
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\Jerico PC\Documents\OOP_1ST-SEM-2025_2026> php index.php
Enter your name: Jerico
Hello, Jerico!
PS C:\Users\Jerico PC\Documents\OOP_1ST-SEM-2025_2026>
```

Operators

Arithmetic Operators

```
main.php X
main.php
1  <?php
2
3  $a = 10;
4  $b = 3;
5  $sum = $a + $b;
6  $difference = $a - $b;
7  $product = $a * $b;
8  $quotient = $a / $b;
9  $remainder = $a % $b;
10
11 echo $sum;
12 ?>
13
14
15
```

Operators

Assignment Operators

```
main.php X
main.php
1  <?php
2
3  $x = 5;
4  $x += 3; // $x is now 8
5  $x -= 2; // $x is now 6
6  $x *= 2; // $x is now 12
7  $x /= 3; // $x is now 4
8  $x .= " PHP"; // $x is now "4 PHP"
9
10 ?>
11
12
13
```

Operators

Comparison Operators

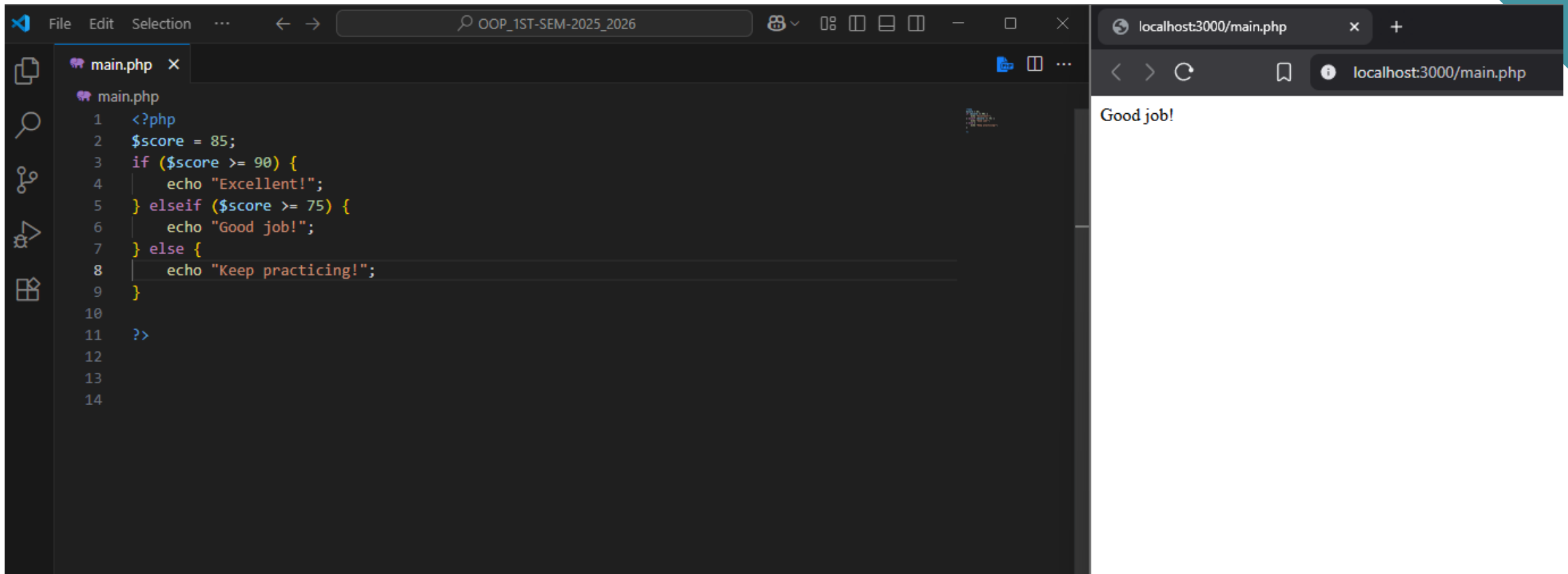
```
main.php X
main.php
1  <?php
2
3  $isEqual = (5 == 5);    // true
4  $isNotEqual = (5 != 3); // true
5  $isGreater = (5 > 3);   // true
6  $isLess = (5 < 3);      // false
7  $isGreaterOrEqual = (5 >= 5); // true
8  $isLessOrEqual = (5 <= 4); // false
9
10 ?>
11
12
13
```

Operators

Logical Operators

```
main.php X
main.php
1  <?php
2
3  $a = 5;
4  $b = 10;
5
6  $andCondition = ($a < 10 && $b > 5); // true
7  $orCondition = ($a < 3 || $b > 5);  // true
8  $notCondition = !($a > 10);          // true
9  ?>
10
11
12
```

Conditional Statements



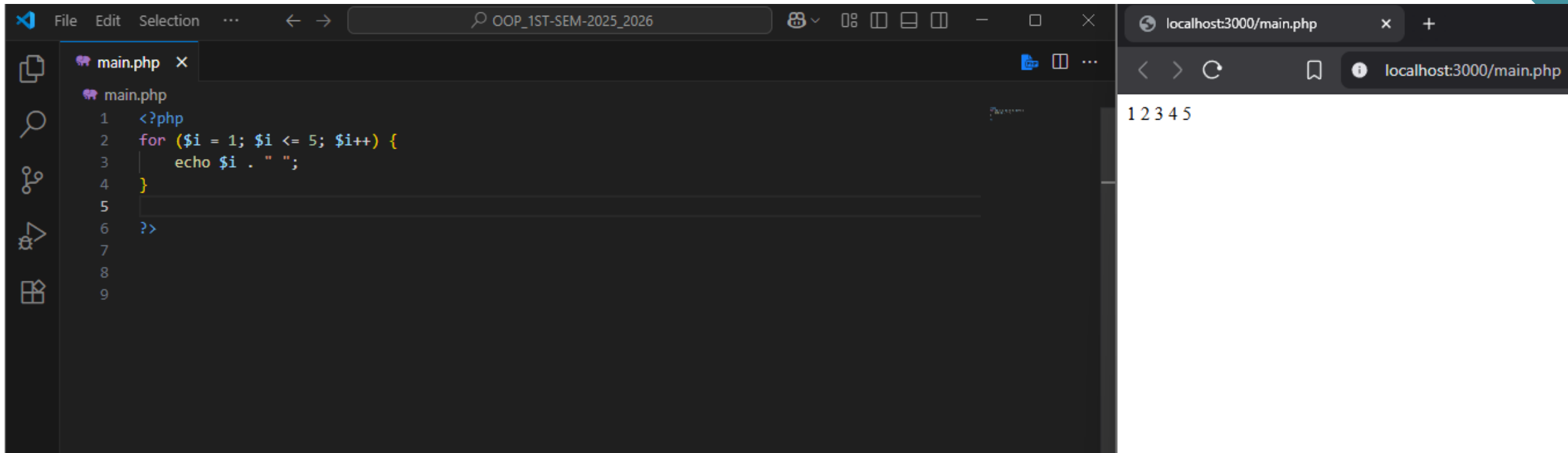
The image shows a code editor window on the left and a web browser window on the right. The code editor displays a PHP script named `main.php` with the following code:

```
1 <?php
2 $score = 85;
3 if ($score >= 90) {
4     echo "Excellent!";
5 } elseif ($score >= 75) {
6     echo "Good job!";
7 } else {
8     echo "Keep practicing!";
9 }
10
11 ?>
```

The web browser window shows the output of the script at `localhost:3000/main.php`, which is:

Good job!

Loops – For Loop

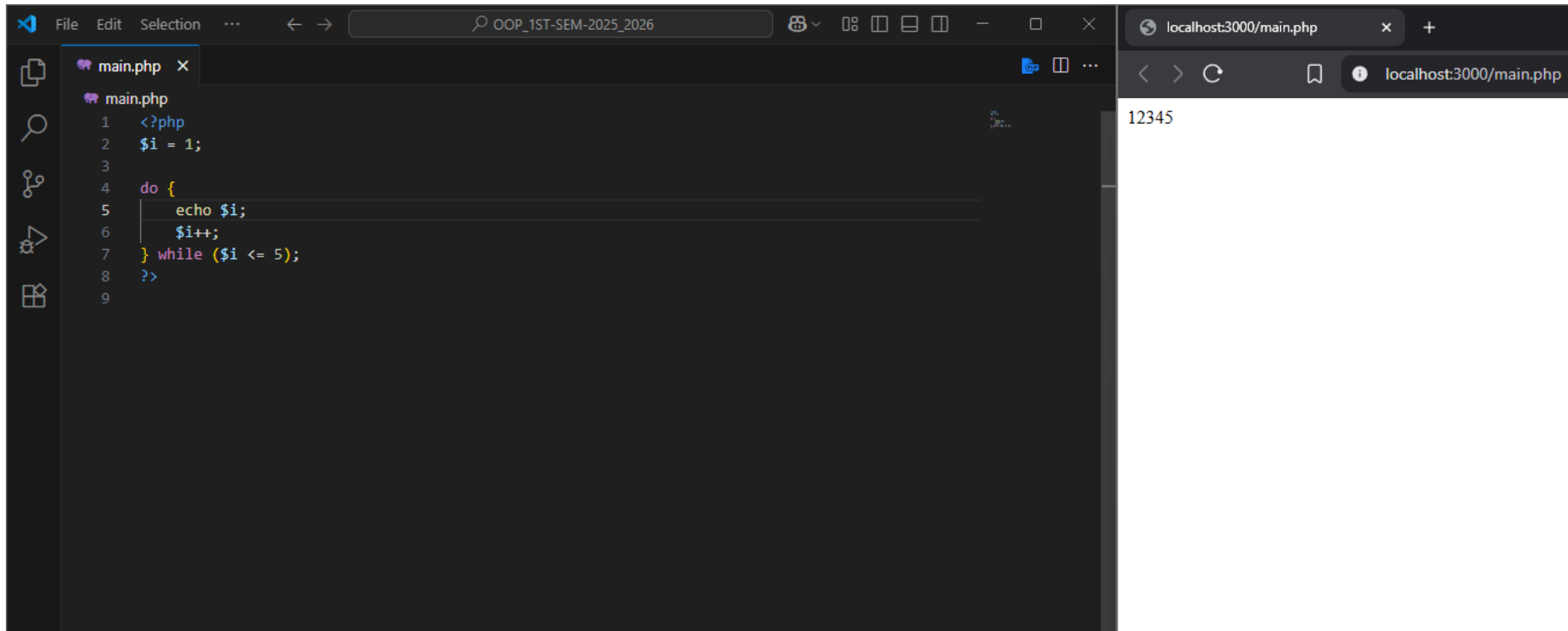


The image shows a code editor window on the left and a web browser window on the right. The code editor displays a PHP file named `main.php` with the following code:

```
1 <?php
2 for ($i = 1; $i <= 5; $i++) {
3     echo $i . " ";
4 }
5
6 ?>
```

The web browser window shows the output of the PHP script at `localhost:3000/main.php`, which is the sequence of numbers `1 2 3 4 5`.

Loops – Do while

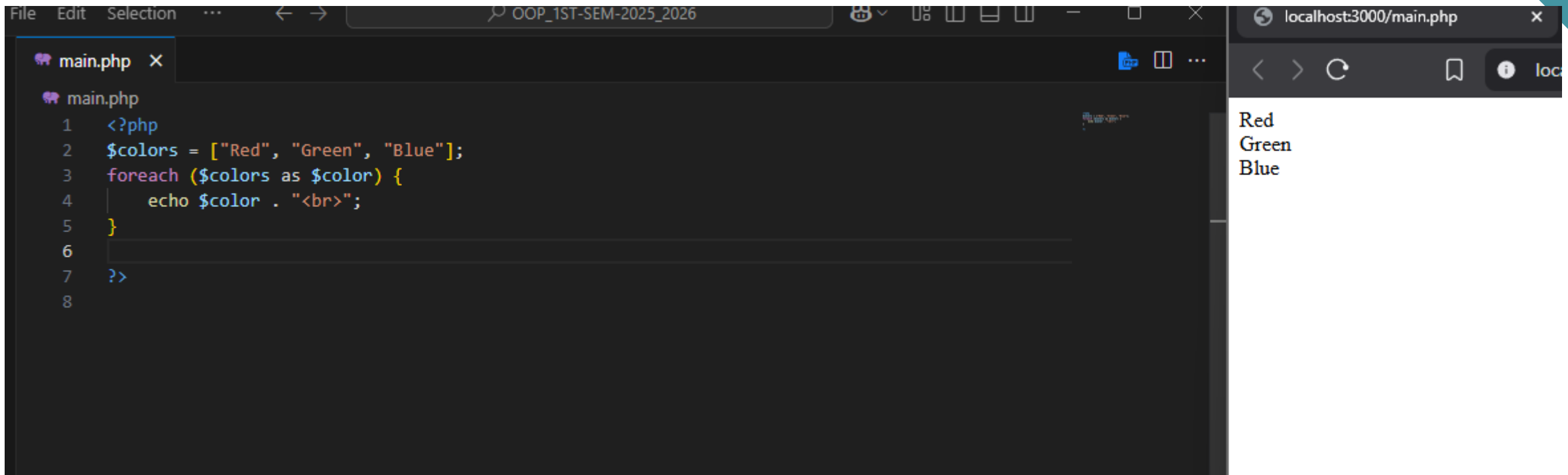


The image shows a code editor window on the left and a web browser window on the right. The code editor displays a PHP file named `main.php` with the following code:

```
1 <?php
2 $i = 1;
3
4 do {
5     echo $i;
6     $i++;
7 } while ($i <= 5);
8 ?>
9
```

The web browser window shows the output of the script at `localhost:3000/main.php`, displaying the number `12345`.

Loops – For Each



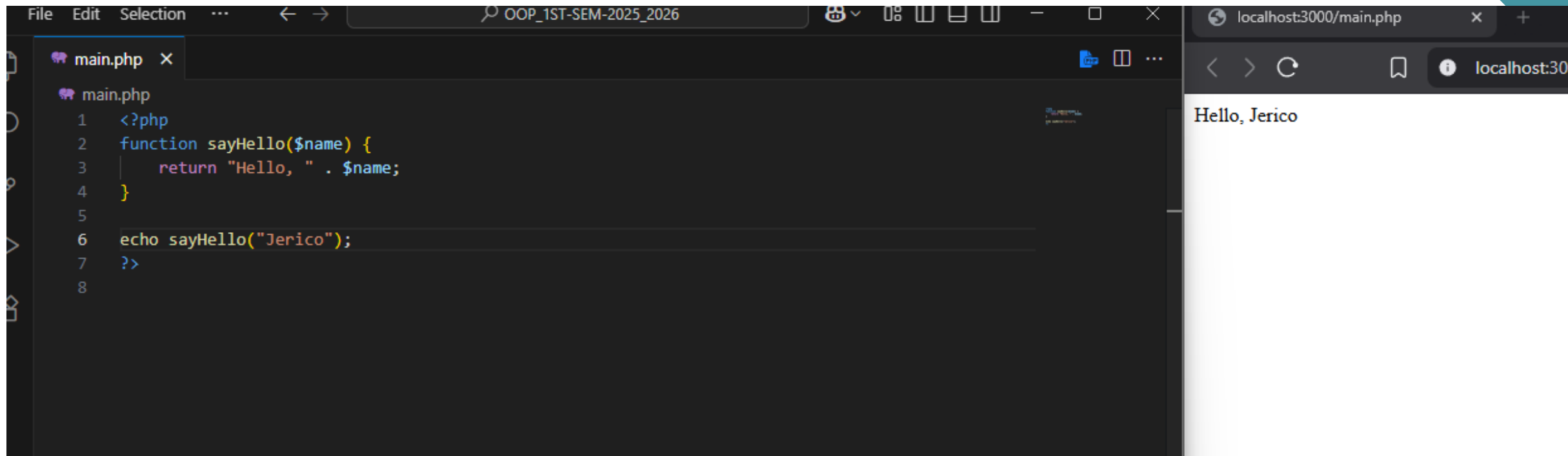
The image shows a code editor window with a file named `main.php` open. The code is as follows:

```
1 <?php
2 $colors = ["Red", "Green", "Blue"];
3 foreach ($colors as $color) {
4     echo $color . "<br>";
5 }
6
7 ?>
```

Below the code editor, a web browser window is open at `localhost:3000/main.php`. The browser displays the output of the PHP script:

```
Red
Green
Blue
```

Functions in PHP



The image shows a code editor window on the left and a web browser window on the right. The code editor displays a PHP file named `main.php` with the following code:

```
1 <?php
2 function sayHello($name) {
3     return "Hello, " . $name;
4 }
5
6 echo sayHello("Jerico");
7 ?>
8
```

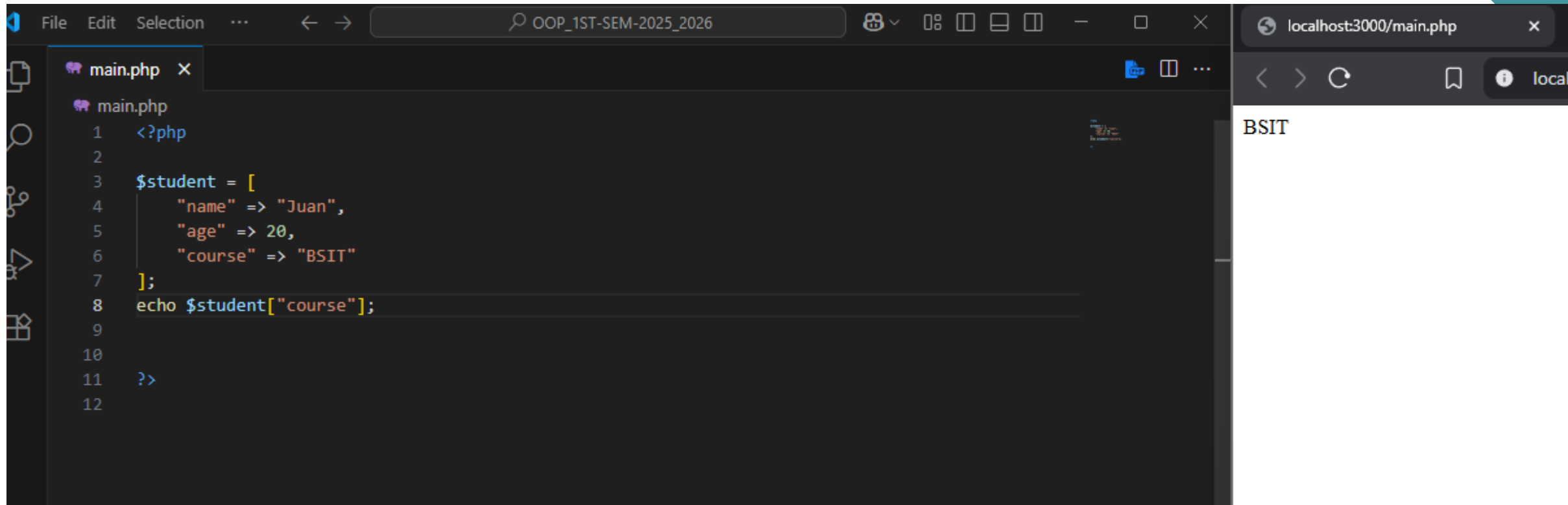
The web browser window shows the output of the PHP script, which is "Hello, Jerico". The browser's address bar indicates the file is being served from `localhost:3000/main.php`.

Arrays – Indexed Array

```
File Edit Selection ... OOP_1ST-SEM-2025_2026
main.php x
main.php
1 <?php
2 $fruits = ["Apple", "Banana", "Mango"];
3 echo $fruits[0];
4
5 ?>
6
```

localhost:3000/main.php x +
< > C localhost:3000/main
Apple

Arrays – Associative Array



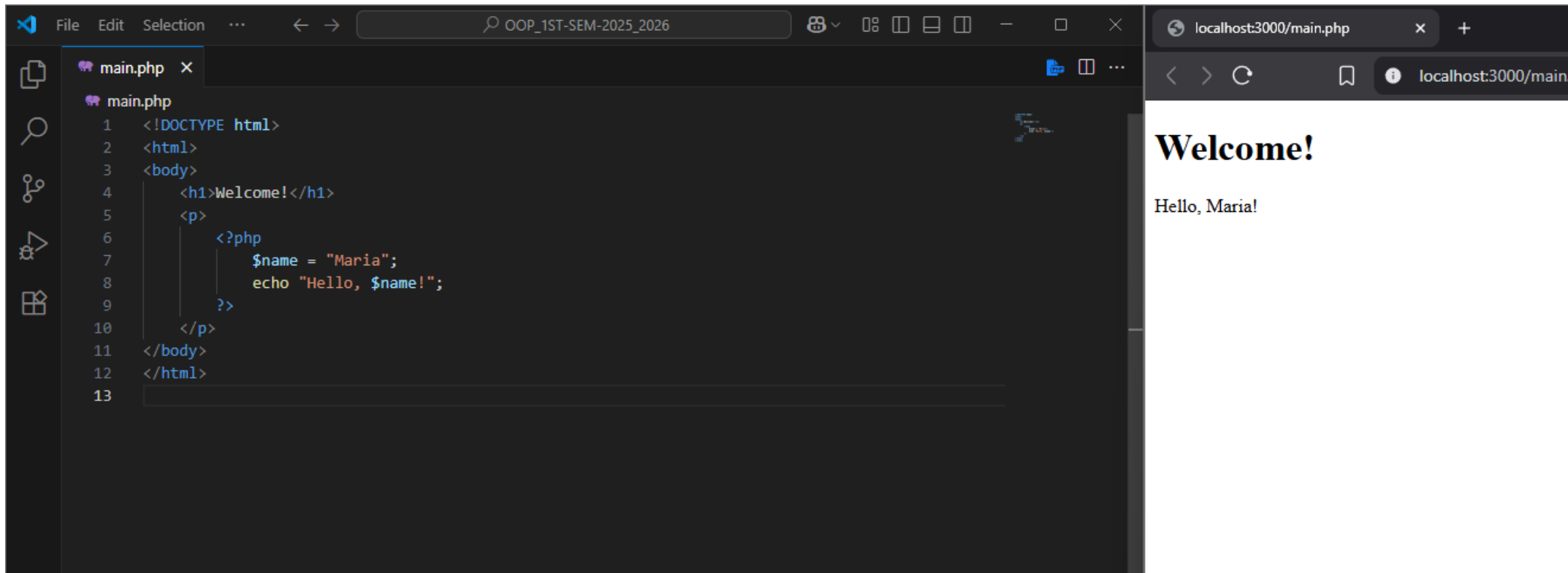
The image shows a code editor window with a file named `main.php` open. The code defines an associative array `$student` with three key-value pairs: `"name" => "Juan"`, `"age" => 20`, and `"course" => "BSIT"`. The code then echoes the value of `$student["course"]`. To the right, a web browser window shows the output of the script at `localhost:3000/main.php`, which is `BSIT`.

```
1 <?php
2
3 $student = [
4     "name" => "Juan",
5     "age" => 20,
6     "course" => "BSIT"
7 ];
8 echo $student["course"];
9
10
11 ?>
12
```

localhost:3000/main.php

BSIT

PHP Inside HTML



The image shows a code editor window on the left and a web browser window on the right. The code editor displays a file named `main.php` with the following content:

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4   <h1>Welcome!</h1>
5   <p>
6     <?php
7       $name = "Maria";
8       echo "Hello, $name!";
9     ?>
10  </p>
11 </body>
12 </html>
13
```

The web browser window shows the output of the PHP code at `localhost:3000/main.php`. The output is:

Welcome!

Hello, Maria!

Practice Activity

Print Even Numbers:

Use a for loop to print all even numbers from 1 to 20.

Sum of Positive Numbers:

Write a program using a while loop that asks the user to enter 5 numbers and calculates the sum of all positive numbers only.

Thank you

