PHP Conditional Statements

Assist prof Dr. huda Abdulaali

- Conditional statements are used to perform different actions based on different conditions.
- Very often when you write code, you want to perform different actions for different conditions. You can use conditional statements in your code to do this.
- In PHP we have the following conditional statements:
- if statement - executes some code if one condition is true
- if...else statement - executes some code if a condition is true and another code if that condition is false
- if...elseif...else statement - executes different codes for more than two conditions
- switch statement - selects one of many blocks of code to be executed

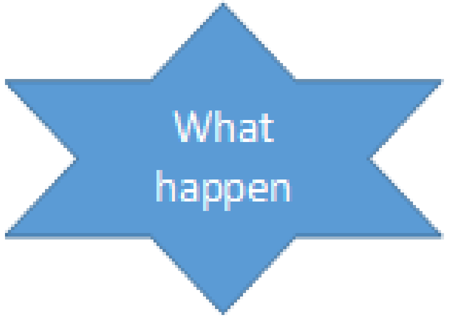
The if Statement

- The if statement executes some code if one condition is true.
- Syntax

```
 if (condition) {
code to be executed if condition is true;
   }
```

- Example
- Output "Have a good day!" if the current time (HOUR) is less than 20:

```
 <?php
  $t = date("H");

  if ($t < "20") {
  echo "Have a good day!";
  }
  ?>
```



What happen

if…else Statement

- The if…else statement executes some code if a condition is true and another code if that condition is false.
- Syntax

```
if (condition) {
   code to be executed if condition is true;
  } else {
 code to be executed if condition is false;
 }
```

- **Example**

- Output "Have a good day!" if the current time is less than 20, and "Have a good night!" otherwise: Have a good night

```
<?php
$t = date("H");

if ($t < "20") {
  echo "Have a good day";
} else {
  echo "Have a good night";
}
?>
```

RUN

if...elseif...else Statement

- The if...elseif...else statement executes different codes for more than two conditions.
- Syntax

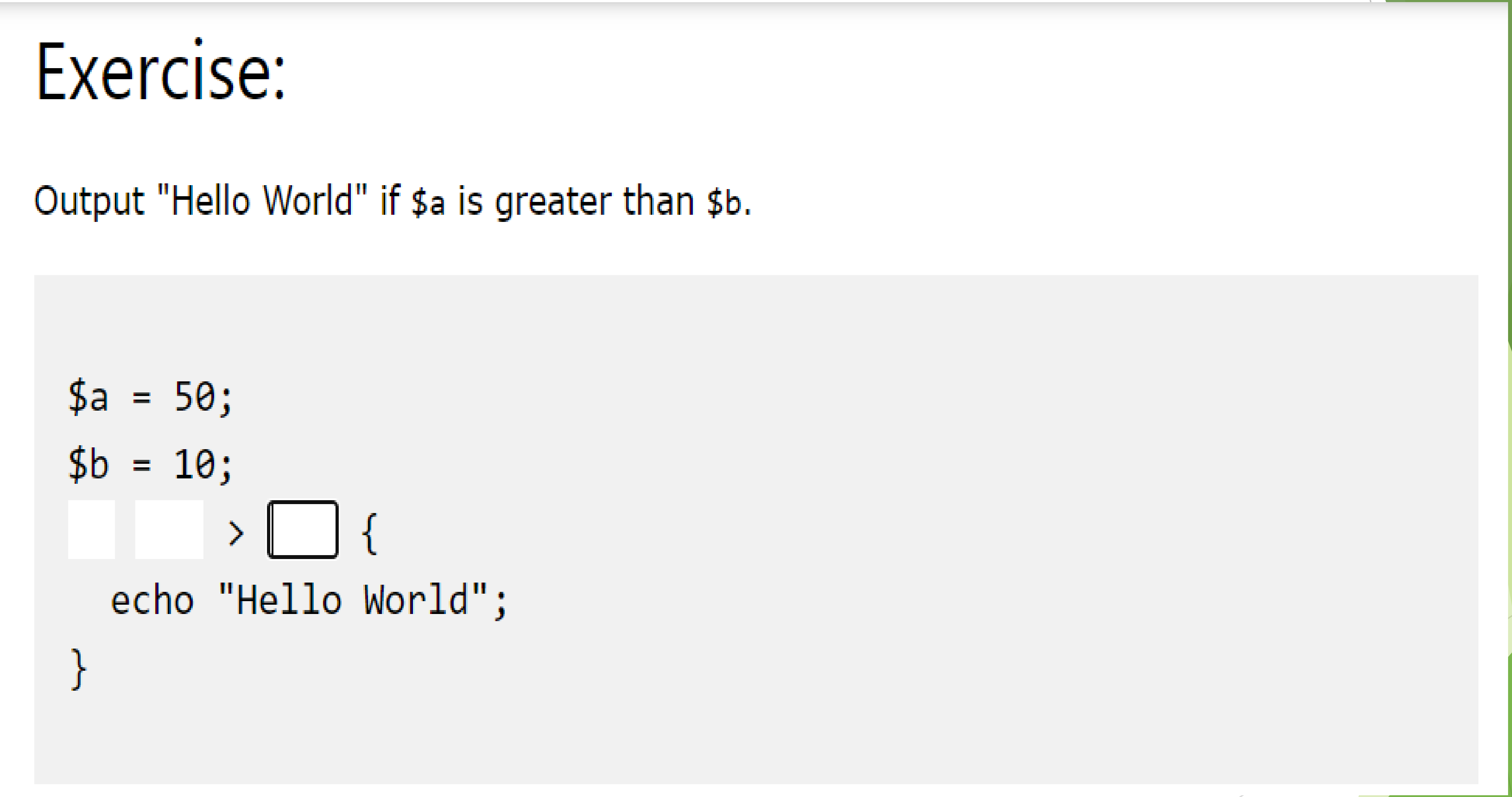
```
  if (condition) {
   code to be executed if this condition is true;
  } elseif (condition) {
```

- code to be executed if first condition is false and this condition is true;

```
} else {
  code to be executed if all conditions are false;
}
```

- **Example**
- Output "Have a good morning!" if the current time is less than 10, and "Have a good day!" if the current time is less than 20. Otherwise it will output "Have a good night!":

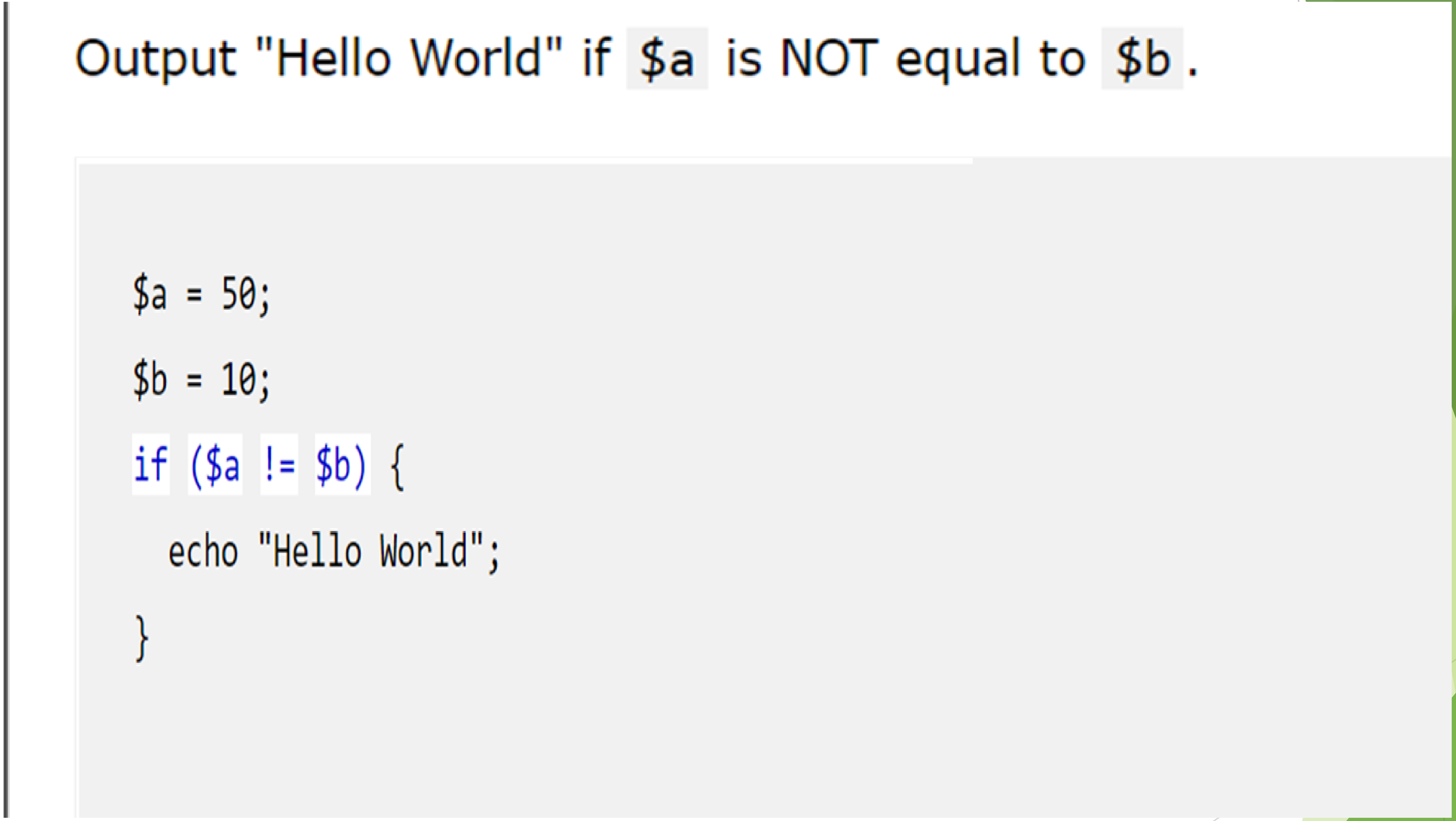
```
<?php
$t = date("H");
echo "<p>The hour (of the server) is " . $t;
echo ", and will give the following message:</p>";

if ($t < "10") {
  echo "Have a good morning!";
} elseif ($t < "20") {
  echo "Have a good day!";
} else {
  echo "Have a good night!";
}
?>
```



Exercise:

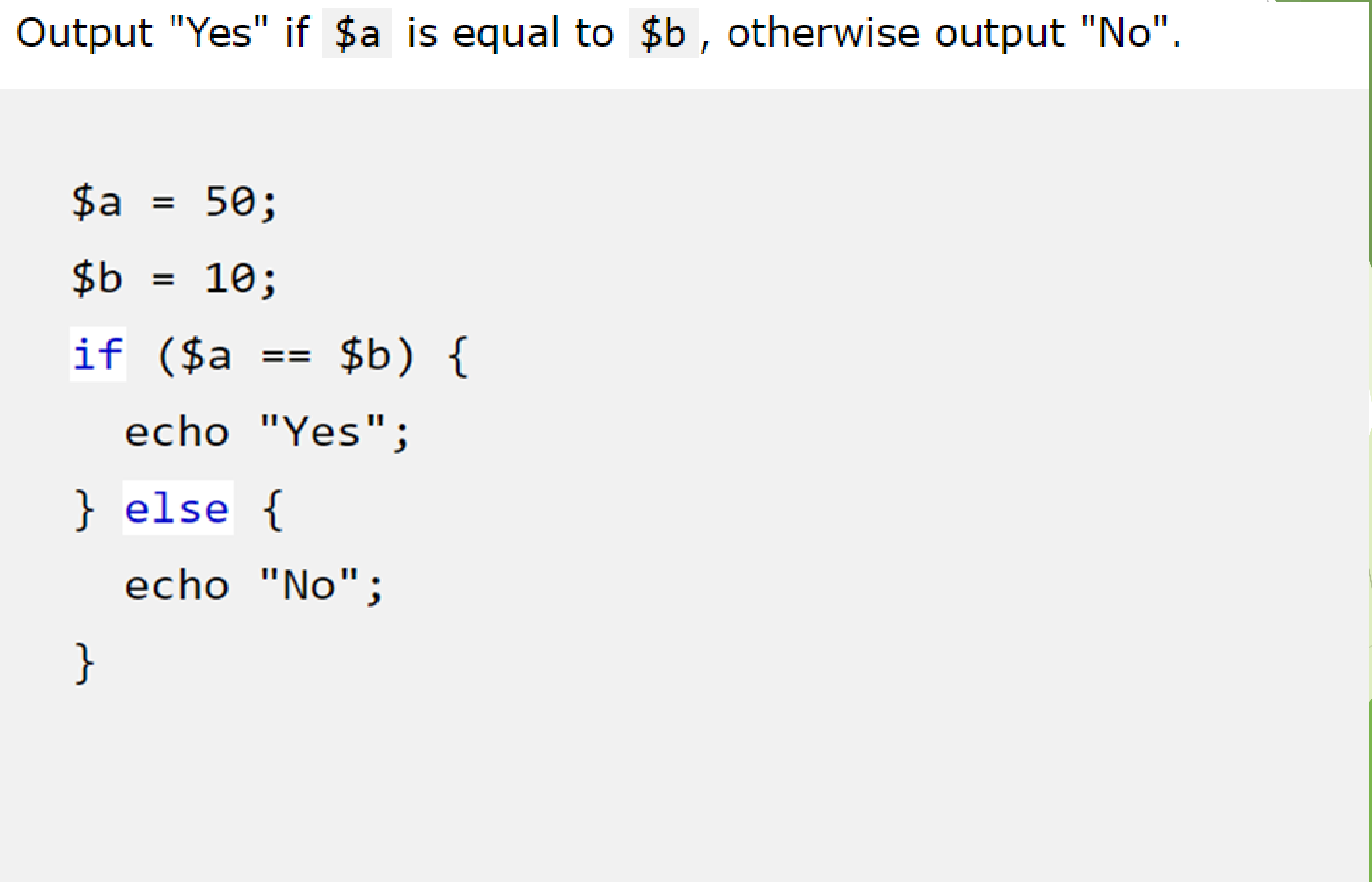
Output "Hello World" if `$a` is greater than `$b`.

```
$a = 50;
$b = 10;
□ □ > □ {
  echo "Hello World";
}
```



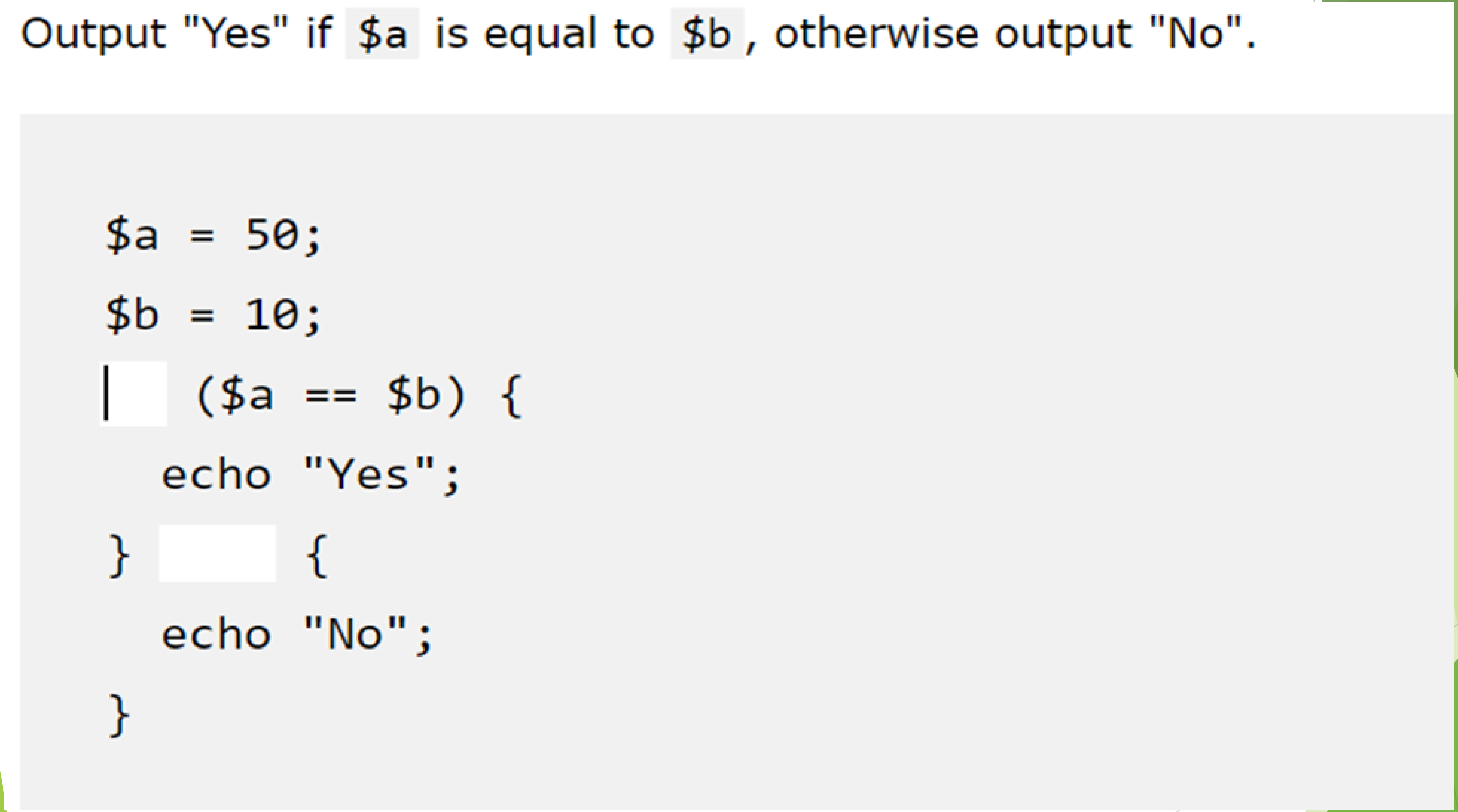
Output "Hello World" if `$a` is NOT equal to `$b`.

```
$a = 50;

$b = 10;

if ($a != $b) {

  echo "Hello World";

}
```



Output "Yes" if `$a` is equal to `$b`, otherwise output "No".

```
$a = 50;
$b = 10;
if ($a == $b) {
  echo "Yes";
} else {
  echo "No";
}
```



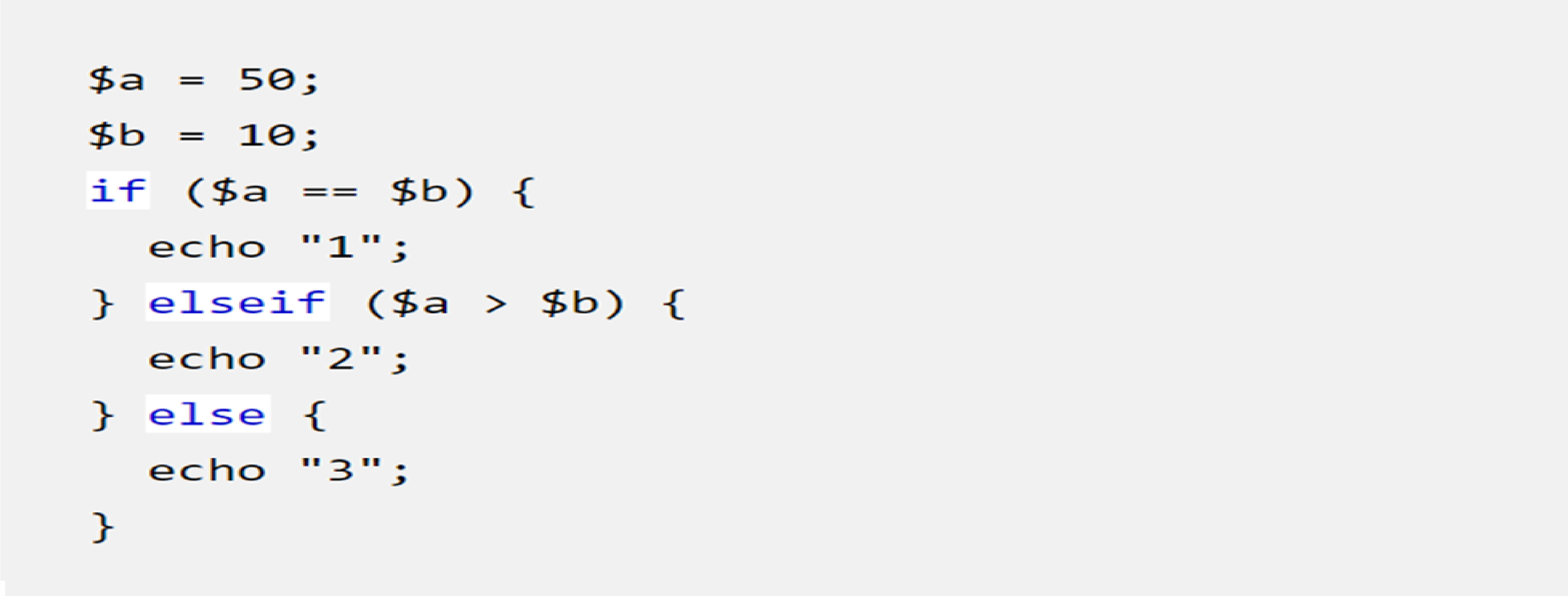
Output "Yes" if `$a` is equal to `$b`, otherwise output "No".

```
$a = 50;
$b = 10;
   ($a == $b) {
  echo "Yes";
}      {
  echo "No";
}
```

Exercise:

Print "1" if a is equal to b, print "2" if a is greater than b, otherwise print "3".

Output "1" if `$a` is equal to `$b`, print "2" if `$a` is greater than `$b`, otherwise output "No".



```
$a = 50;
$b = 10;
if ($a == $b) {
  echo "1";
} elseif ($a > $b) {
  echo "2";
} else {
  echo "3";
}
```

switch Statement

- The switch statement is used to perform different actions based on different conditions.
- Use the switch statement to select one of many blocks of code to be executed.
- Syntax

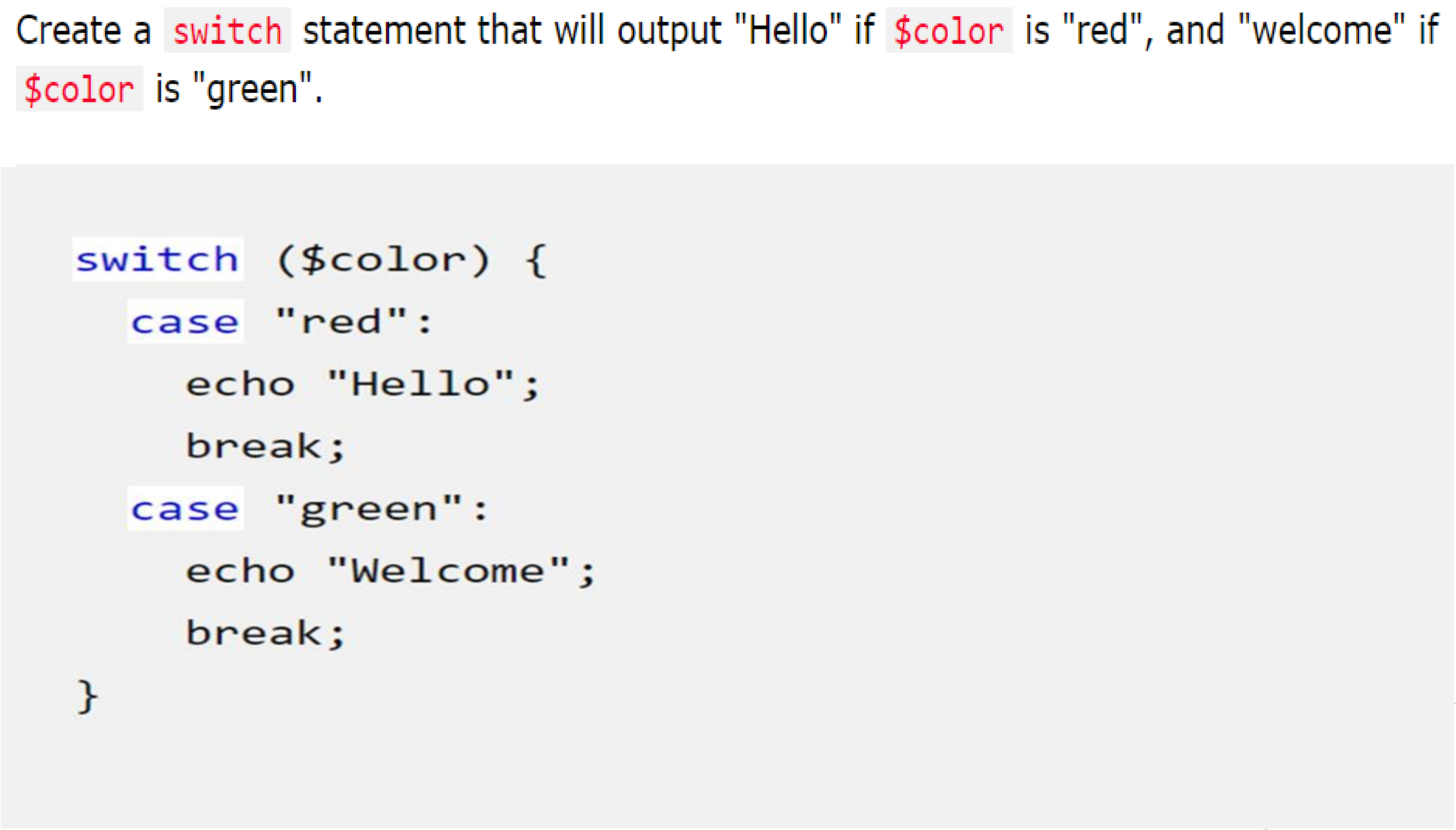
```
switch (n) {
   case label1:
  code to be executed if n=label1;
  break;
   case label2:
  code to be executed if n=label2;
  break;
  case label3:
  code to be executed if n=label3;
  break;
     ...
   default:
  code to be executed if n is different from all labels;
        }
```

- This is how it works: First we have a single expression n (most often a variable), that is evaluated once. The value of the expression is then compared with the values for each case in the structure. If there is a match, the block of code associated with that case is executed. Use break to prevent the code from running into the next case automatically. The default statement is used if no match is found.

- Example

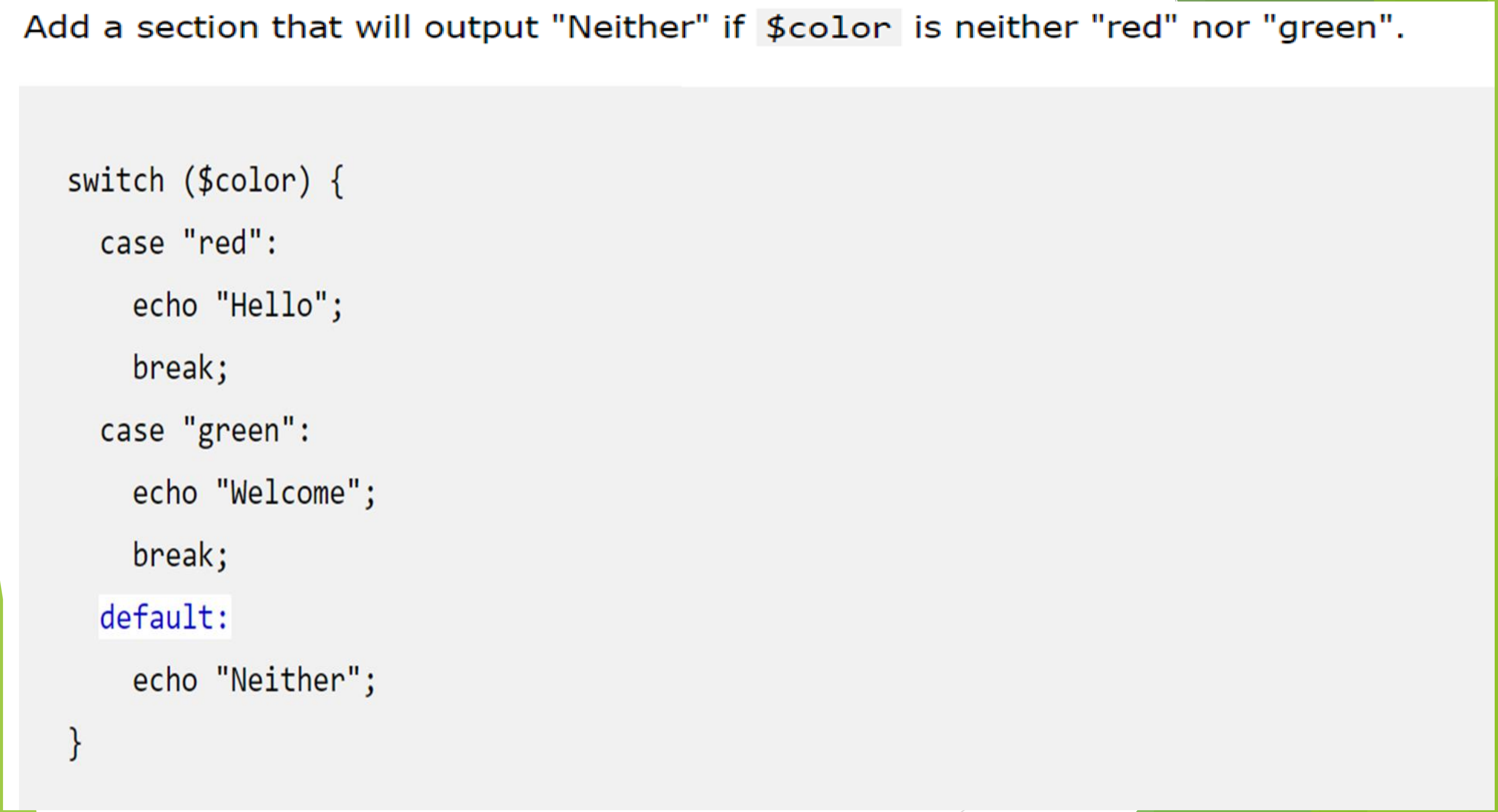
```
<?php
$favcolor = "red";

 switch ($favcolor) {
   case "red":
   echo "Your favorite color is red!";
    break;
    case "blue":
     echo "Your favorite color is blue!";
     break;
     case "green":
     echo "Your favorite color is green!";
      break;
      default:
      echo "Your favorite color is neither red, blue, nor green!";
     }
         ?>
```

Exercise:



Create a `switch` statement that will output "Hello" if `$color` is "red", and "welcome" if `$color` is "green".

```
switch ($color) {
  case "red":
    echo "Hello";
    break;
  case "green":
    echo "Welcome";
    break;
}
```



Add a section that will output "Neither" if `$color` is neither "red" nor "green".

```
switch ($color) {
  case "red":
    echo "Hello";
    break;
  case "green":
    echo "Welcome";
    break;
  default:
    echo "Neither";
}
```

Thank you

# Stay home