



# Reskilling4Employment Software Developer

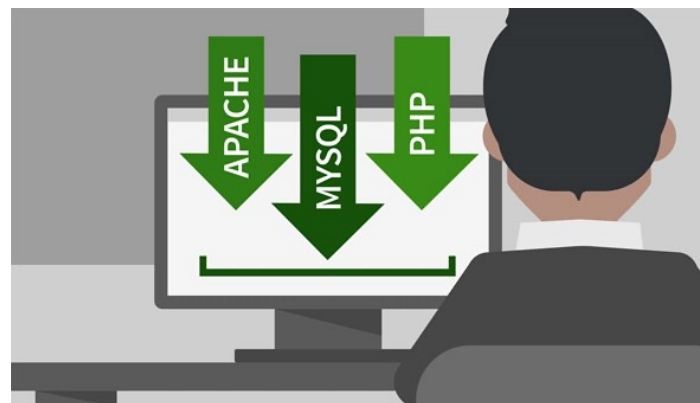
Programação para a WEB - cliente (client-side)

Bruno Santos

[bruno.santos@cesae.pt](mailto:bruno.santos@cesae.pt)

# AMP Stack

- Para utilização do PHP no Windows é necessária a utilização de uma aplicação com AMP STACK
  - Apache
  - MySQL
  - PHP
- Exemplos:
  - AMPPS;
  - WAMP;
  - XAMP;
  - Laragon;
  - ...



# PHP

- PHP é uma linguagem de programação versátil e muito familiar à linguagem HTML.
- O código PHP é escrito entre as tags: `<?php` e `?>`
- Com PHP é possível criar páginas com conteúdo dinâmico, por exemplo, a página do Facebook irá variar consoante o utilizador que se autenticar.
- O código PHP funciona do lado do servidor, assim não é visível para o utilizador.
- Documentação oficial disponível em:  
<https://www.php.net/manual/en/>

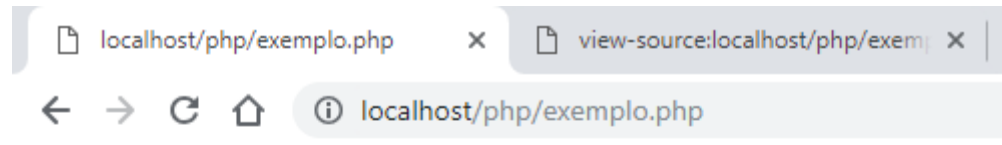
# PHP

- Todos os ficheiros PHP têm a extensão .php
- Mesmo em páginas apenas com conteúdo HTML deve ser utilizada a extensão .php
- Os projetos devem ser guardados na pasta específica do software instalado: htdocs, www ou wwwroot. Outros software AMP Stack podem utilizar uma pasta diferente.
- Para apresentar a página deve ser acedido o link <http://localhost>

# Php – exemplo

```
<!DOCTYPE html>
<html>
  <body>
    <h1>Exemplo de utilização PHP!</h1>
    <?php
      echo "Mensagem em PHP";
    ?>
  </body>
</html>
```

# Php – exemplo



## Exemplo de utilização PHP!

Mensagem em PHP

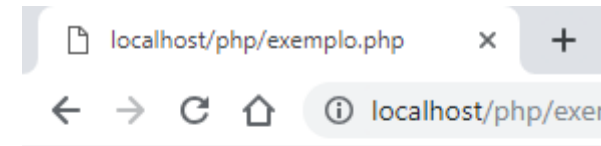


# Comentários

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      echo "<p>Olá, eu serei visto no ecrã</p>";
      // Eu não! Sou apenas um comentário.

      echo "<p>Olá, eu também serei visto</p>";
      # Já eu não serei!

      echo "<p>E eu aqui novamente no ecrã</p>";
      /* Eu não aparecerei no ecrã novamente
      pois sou um comentário */
    ?>
  </body>
</html>
```



Olá, eu serei visto no ecrã

Olá, eu também serei visto

E eu aqui novamente no ecrã

# Variáveis

- Todas as variáveis iniciam com \$
- Podem ser seguidas de letras ou \_
- Não podem conter caracteres especiais ou espaços
- São case sensitive (\$nome é diferente de \$Nome)



# Variáveis

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      $name = "Guilherme";
      $age = 20;

      echo $name; // Guilherme
      echo "</br>";
      echo $age; // 20
    ?>

  </body>
</html>
```

Guilherme  
20

# Variáveis

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      $a = "mundo!";
      echo "Olá, $a <br>"; // Olá,
      mundo!
      echo 'Olá, $a <br>'; // Olá, $a
    ?>
  </body>
</html>
```

Olá, mundo!  
Olá, \$a

# Variáveis

```
<?php
    $nome="José";
?>

<!DOCTYPE html>
<html>
    <body>
        <?php
            echo $nome;
        ?>
    </body>
</html>
```

José

# Super globais

- São tipos de variáveis especiais em PHP que significam que estão sempre acessíveis, independentemente do scope do script, função, class ou ficheiro.
- Exemplos:
  - `$GLOBALS`
  - `$_SERVER`
  - `$_REQUEST`
  - `$_POST`
  - `$_GET`
  - `$_FILES`
  - `$_ENV`
  - `$_COOKIE`
  - `$_SESSION`

# Super globais

- A variável `$GLOBALS` em php é uma superglobal que é usada para aceder a todas as variáveis globais.
- PHP armazena todas as variáveis globais num array `$GLOBALS[index]`. O index será o nome da variável.
- `print_r($GLOBALS);` //apresenta todas as superglobals

# Super globais

- A variável `$_SERVER` em php é uma superglobal que é usada para aceder a informações sobre headers, paths e localizações de scripts

# Super globais

- A variável `$_REQUEST` em php é uma superglobal que é usada para aceder a informações depois de submeter um formulário.
- Funciona para os 2 métodos GET e POST.

# Super globais

- A variável `$_POST` em php é uma superglobal que é usada para aceder a informações depois de submeter um formulário.
- Funciona apenas para o método POST dos formulários



# Super globais

- A variável `$_GET` em php é uma superglobal que é usada para aceder a informações depois de submeter um formulário.
- Funciona apenas para o método GET dos formulários.
- Pode ser usada para recolher informação de um URL

# Super globais

- A variável `$_SESSION` em php é uma superglobal que é usada para aceder armazenar informação entre várias páginas.
- Ao contrário de cookie, a informação não fica armazenada no computador cliente.
- Por defeito, duram até o utilizador fechar o browser.

# Super globais

- Um sessão começa com a função `session_start()`;
- Variáveis de sessão são superglobals de php : `$_SESSION`
- Exemplo:

```
session_start();  
$_SESSION['login'] = true;
```

# Super globais

- Para remover todas as variáveis de sessão `session_unset();`
- Para destruir a sessão : `session_destroy();`

# Super globais

- A variável `$_COOKIE` em php é uma superglobal que é usada para identificar um utilizador
- Um cookie é um pequeno ficheiro que o servidor embebe no computador do utilizador.
- De cada vez que o mesmo computador faz um request da página via browser, também vai enviar uma cookie.

# Super globais

- Um cookie é criado com a função `setcookie()`
- Sintaxe
  - `setcookie(name, value, expire, path, domain, secure, httponly);`
- Apenas o parâmetro do nome é necessário, os restantes parâmetros são opcionais.

# Constantes

- As constantes são em tudo semelhantes às variáveis, com a particularidade de não ser possível alterar o seu valor após serem definidas.

```
define("nome_da_constante", "valor");  
  
define("preco", 10);  
define("nome", "Manuel");
```

# Nomes reservados

- Alguns nomes são reservados pela própria linguagem não sendo possível a sua utilização para variáveis definidas pelo utilizador.
- Lista de variáveis:

<https://www.php.net/manual/en/reserved.keywords.php>



# Strings

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      echo "Olá," . " mundo!";
    ?>
  </body>
</html>
```

Olá, mundo!

# Strings

```
<?php
    $nome = "José";
    $sobrenome = "Silva";
?>

<!DOCTYPE html>
<html>
    <body>
        <?php
            echo $nome . " " . $sobrenome;
        ?>
    </body>
</html>
```

José Silva

# Funções de strings

```
<?php
    $nome = "Teste de PHP";
?>

<!DOCTYPE html>
<html>
    <body>
        <?php
            // strlen - Retorna nº de caracteres
            echo strlen($nome) . "<br>";

            // stripos - Retorna primeira ocorrência de um carater
            echo stripos($nome,"e") . "<br>";

            // strrpos - Retorna última ocorrência de um carater
            echo strrpos($nome,"e") . "<br>";

            // strtolower - converte para letras min.
            echo strtolower($nome) . "<br>";

            // strtoupper - converte para letras min.
            echo strtoupper($nome) . "<br>";
        ?>
    </body>
</html>
```

12  
1  
7  
teste de php  
TESTE DE PHP

# Funções de strings

```
<?php
    $nome = "Teste DE PHP";
?>

<!DOCTYPE html>
<html>
    <body>
        <?php
            // SUBSTR_COUNT - Conta quantas ocorreram
            // de um texto ou string
            // Faz diferença Maiúsculas e minúsculas

            echo substr_count($nome, "e") . "<br>";
            echo substr_count($nome, "E") . "<br>";
        ?>
    </body>
</html>
```

2  
1

# Operações

```
<?php
    $num1 = 10;
    $num2 = 4;
?>

<!DOCTYPE html>
<html>
    <body>
        <?php
            // Multiplicacao e Divisao
            echo $num1 * $num2 . "<br>";
            echo $num1 / $num2 . "<br>";
        ?>
    </body>
</html>
```

40  
2.5

# Funções de números

```
<?php
    $num1 = 10;
    $num2 = 4;
?>

<!DOCTYPE html>
<html>
    <body>
        <?php
            // Exponencial
            echo pow(8,2) . "<br>";

            // Raiz Quadrada
            echo sqrt(25) . "<br>";
        ?>
    </body>
</html>
```

64  
5

# Funções de números

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      // Valor absoluto
      echo abs(-10);
    ?>
  </body>
</html>
```

10

# Random

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      // Random
      echo rand() . "<br>";

      // Random num intervalo
      echo rand(1,10) . "<br>";
    ?>
  </body>
</html>
```

1989777520  
2



# Is\_numeric

```
<?php
    $n1 = 10;
    $n2 = 1.2;
    $n3 = "Olá";
?>
<!DOCTYPE html>
<html>
    <body>
        <?php
            echo "N1: ". is_numeric($n1) . "<br>";
            echo "N2: ". is_numeric($n2) . "<br>";
            echo "N3: ". is_numeric($n3) . "<br>";
        ?>
    </body>
</html>
```

N1: 1

N2: 1

N3:

# Is\_int

```
<?php
    $n1 = 10;
    $n2 = 1.2;
    $n3 = "Olá";
?>
<!DOCTYPE html>
<html>
    <body>
        <?php
            echo "N1: ". is_int($n1) . "<br>";
            echo "N2: ". is_int($n2) . "<br>";
        ?>
    </body>
</html>
```

N1: 1

N2:

# Is\_float

```
<?php
    $n1 = 10;
    $n2 = 1.2;
    $n3 = "Olá";
?>
<!DOCTYPE html>
<html>
    <body>
        <?php
            echo "N1: ". is_float($n1) . "<br>";
            echo "N2: ". is_float($n2) . "<br>";
        ?>
    </body>
</html>
```

N1:  
N2: 1

# Round

```
<?php
    $n1 = 1.2;
    $n2 = 1.7;
?>
<!DOCTYPE html>
<html>
    <body>
        <?php
            echo round($n1) . "<br>";
            echo round($n2) . "<br>";

            //arredonda sempre para baixo
            echo floor($n1) . "<br>";
            echo floor($n2) . "<br>";

            //arredonda sempre para cima
            echo ceil($n1) . "<br>";
            echo ceil($n2) . "<br>";
        ?>
    </body>
</html>
```

1  
2  
1  
1  
2  
2

# Conversões de tipos

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      $var = 100;
      $type_casting = (bool) $var;      // torna-se booleano
      $type_casting = (int) $var;       // torna-se inteiro
      $type_casting = (float) $var;     // torna-se float
      $type_casting = (string) $var;    // torna-se string
      $type_casting = (array) $var;     // torna-se array
      echo $type_casting = (bool)$var; // 1
    ?>
  </body>
</html>
```

# Cálculos

```
$v1=10;  
$v2=3;  
  
echo $v1 + $v2 . "<br>;  
echo $v1 - $v2 . "<br>;  
echo $v1 * $v2 . "<br>;  
echo $v1 / $v2 . "<br>;  
echo $v1 % $v2 . "<br>;
```

```
13  
7  
30  
3.33333333333333  
1
```

# Cálculos

```
$v1=10;  
$v2=3;  
  
$v1 += $v2;      echo $v1 . "<br>";  
$v1 -= $v2;      echo $v1 . "<br>";  
$v1 *= $v2;      echo $v1 . "<br>";  
$v1 /= $v2;      echo $v1 . "<br>";  
$v1 %= $v2;      echo $v1 . "<br>";
```

13  
10  
30  
10  
1

# Estruturas de decisão

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      $idade = 17;

      if($idade < 18) {
        echo "Não pode entrar aqui!";
      } else {
        echo "Bem-vindo!";
      }
    ?>
  </body>
</html>
```

Não pode entrar aqui!



# Estruturas de decisão

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      $idade = 21;
      $identidade = true;

      if($idade > 18 && $identidade == true)
      {
        echo "Bem-vindo!";
      }
    ?>
  </body>
</html>
```

Bem-vindo!

# Estruturas de decisão

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      $nome = "Till Lindemann";

      if($nome == "Richard Kruspe") {
        echo "Olá Richard Kruspe!";
      } elseif ($nome == "Oliver Riedel") {
        echo "Olá Oliver Riedel!";
      } elseif ($nome == "Till Lindemann") {
        echo "Olá Till Lindemann!";
      } else {
        echo "Olá $nome!";
      }
    ?>
  </body>
</html>
```

Olá Till Lindemann!

# Estruturas de decisão

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      $nome = "Fulano";

      switch($nome) {
        case "Fulano":
          echo "Olá Fulano!";
          break;

        case "Sicrano":
          echo "Olá Sicrano!";
          break;

        case "Beltrano":
          echo "Olá Beltrano!";
          break;

        default:
          echo "Qual é o seu nome?";
          break;
      }
    ?>
  </body>
</html>
```

Olá Fulano!

# Operador Ternário

- O operador ternário permite apresentar condições de forma mais simplificada

```
<?php
$preco = 10;
$resultado = ($preco > 0) ? 'Custa '.$preco : 'Gratuito';
echo $resultado;
?>
```

# Ciclos

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      $num = 0;

      while($num < 10 ) {
        echo $num++;
      }
    ?>
  </body>
</html>
```

0123456789

# Ciclos

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      for($a = 1; $a <= 10; $a++){
        $cubo = $a * $a * $a;
        echo "O cubo de $a é $cubo<br>";
      }
    ?>
  </body>
</html>
```

O cubo de 1 é 1  
O cubo de 2 é 8  
O cubo de 3 é 27  
O cubo de 4 é 64  
O cubo de 5 é 125  
O cubo de 6 é 216  
O cubo de 7 é 343  
O cubo de 8 é 512  
O cubo de 9 é 729  
O cubo de 10 é 1000

# Ciclos

```
<ul>
<?php
$nomes = ['Ana', 'Maria', 'Miguel', 'José'];

foreach($nomes as $nome) {
    echo "<li>".$nome."</li>";
}

?>

</ul>
```

- Ana
- Maria
- Miguel
- José

# Funções PHP

- `phpversion();` → mostra a versão do PHP
- `strtolower($var);` → transforma string para minúsculas
- `ucfirst($var);` → transforma o 1º char em maiúsculas
- `strip_tags($input);` → retira as tags html de uma string opcional  
incluir string com tags que não retira.



# Funções

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      function calcularSalario(){
        return 100*3;
      }

      echo calcularSalario();
    ?>
  </body>
</html>
```

300

# Funções

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      function calcularSalario($salario,$ndias){
        return $salario * $ndias;
      }

      echo calcularSalario(100,30);
    ?>
  </body>
</html>
```

3000

# Funções

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      function calcularSalario($salario,$ndias){
        $euro = $salario * $ndias;
        $dolar = $salario * $ndias * 1.2;

        return array($euro,$dolar);
      }

      $salarioFinal = calcularSalario(100,30);

      echo $salarioFinal[0] . " em euros <br>";
      echo $salarioFinal[1] . " em dolares <br>";
    ?>
  </body>
</html>
```

3000 em euros  
3600 em dolares

# datas

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      date_default_timezone_set("Europe/Lisbon");
      print_r(getdate());
    ?>
  </body>
</html>
```

```
Array ( [seconds] => 49 [minutes] => 10 [hours] => 19
[mday] => 1 [wday] => 3 [mon] => 5 [year] => 2019 [yday]
=> 120 [weekday] => Wednesday [month] => May [0] =>
1556734249 )
```

# Datas

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      date_default_timezone_set("Europe/Lisbon");

      $agora = getdate();
      $ano = $agora["year"];
      $mes = $agora["mon"];
      $dia = $agora["mday"];

      echo $dia."/".$mes."/".$ano;|
    ?>
  </body>
</html>
```

1/5/2019

Mais info em: [https://www.php.net/manual/pt\\_BR/function.getdate.php](https://www.php.net/manual/pt_BR/function.getdate.php)

# Datas

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      date_default_timezone_set("Europe/Lisbon");
      setlocale(LC_TIME, "pt-PT");

      $ano = strftime('%G');
      $mes = strftime('%B');
      $dia = strftime('%d');

      echo $dia."/".$mes."/".$ano;
    ?>
  </body>
</html>
```

01/maio/2019

Mais info em: [https://www.php.net/manual/pt\\_BR/function.strftime.php](https://www.php.net/manual/pt_BR/function.strftime.php)

# Datas

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      $data1 = new DateTime('2019-02-10');
      $data2 = new DateTime('2019-02-14');

      $diferenca = $data2->diff($data1);
    ?>

    <pre>
      <?php print_r($diferenca); ?>
    </pre>
  </body>
</html>
```

```
        DateInterval Object
(
    [y] => 0
    [m] => 0
    [d] => 4
    [h] => 0
    [i] => 0
    [s] => 0
    [f] => 0
    [weekday] => 0
    [weekday_behavior] => 0
    [first_last_day_of] => 0
    [invert] => 1
    [days] => 4
    [special_type] => 0
    [special_amount] => 0
    [have_weekday_relative] => 0
    [have_special_relative] => 0
)
```

# Links dinâmicos

- Pretendemos agora criar uma página com uma lista de artigos. Ao clicar num artigo devemos ser redirecionados para a página com os detalhes do artigo. Isto obrigar-nos-ia a criar uma página para cada artigo... mas com PHP podemos usar apenas 1 página e preenche-la consoante o artigo clicado.



# Links dinâmicos

exemplo.php

```
<!DOCTYPE html>
<html>
  <body>
    <ul>
      <li>
        <a href="artigo.php?cod=1">Computador</a>
      </li>
      <li>
        <a href="artigo.php?cod=2">Smartphone</a>
      </li>
      <li>
        <a href="artigo.php?cod=3">Caderno</a>
      </li>
    </ul>
  </body>
</html>
```

- [Computador](#)
- [Smartphone](#)
- [Caderno](#)

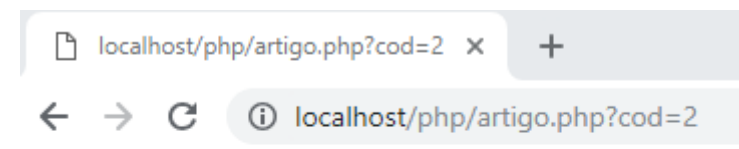
# Links dinâmicos

artigo.php

```
<!DOCTYPE html>
<html>
  <body>

  <?php
    $artigos = array("Computador", "Smartphone", "Caderno");
    $codigo = $_GET["cod"];
    echo $artigos[$codigo];
  ?>

  </body>
</html>
```



Caderno

# Arrays

```
<?php
    $salada=array("Alface", "Cenoura", "Cebola");
?>

<!DOCTYPE html>
<html>
    <body>
        <?php
            echo $salada[0] . "<br>";
            echo $salada[2] . "<br>";
        ?>
    </body>
</html>
```

Alface  
Cebola

# Arrays

```
<?php
    $salada=array("Alface", "Cenoura", "Cebola");
?>

<!DOCTYPE html>
<html>
    <body>
        <?php
            echo $salada[0] . "<br>";
            echo $salada[2] . "<br>";

            echo count($salada) . "<br>";

            $salada[]=$salada[0];

            echo count($salada) . "<br>";

            echo $salada[3] . "<br>";
        ?>
    </body>
</html>
```

Alface  
Cebola  
3  
4  
Rúcula

# Arrays

```
<?php
    $salada=array("Alface", "Cenoura", "Cebola");
?>

<!DOCTYPE html>
<html>
    <body>
        <pre>
            <?php
                print_r($salada);
            ?>
        </pre>
    </body>
</html>
```

Array

(

[0] => Alface

[1] => Cenoura

[2] => Cebola

)

# Arrays associativos

```
<?php
    $agenda=array(
        "nome" => "José",
        "sobrenome" => "Silva",
        "telefone" => 222333444
    );
?>

<!DOCTYPE html>
<html>
    <body>
        <pre>
            <?php
                print_r($agenda);
            ?>
        </pre>
    </body>
</html>
```

Array

```
(
    [nome] => José
    [sobrenome] => Silva
    [telefone] => 222333444
)
```

# Arrays associativos

```
<?php
    $agenda=array(
        "nome" => "José",
        "sobrenome" => "Silva",
        "telefone" => 222333444
    );
?>

<!DOCTYPE html>
<html>
    <body>
        <?php
            echo $agenda["nome"];|
        ?>
    </body>
</html>
```

José

# Funções de arrays

```
<?php
    $nums=array(10,5,3,44,5,88,7);
?>

<!DOCTYPE html>
<html>
    <body>
        <?php
            echo min($nums) . "<br>";
            echo max($nums) . "<br>";
            echo array_sum($nums) . "<br>";
        ?>
    </body>
</html>
```

3  
88  
162



# Funções de arrays

```
<?php
    $nums=array(10,5,3,44,5,88,7);
?>

<!DOCTYPE html>
<html>
    <body>
        <?php
            sort($nums);
        ?>

        <pre>
            <?php
                print_r($nums)
            ?>
        </pre>
    </body>
</html>
```

Array

(

[0]	=>	3
[1]	=>	5
[2]	=>	5
[3]	=>	7
[4]	=>	10
[5]	=>	44
[6]	=>	88

)

# Funções de arrays

```
<?php
    $nums=array(10,5,3,44,5,88,7);
?>

<!DOCTYPE html>
<html>
    <body>
        <?php
            rsort($nums);
        ?>

        <pre>
            <?php
                print_r($nums)
            ?>
        </pre>
    </body>
</html>
```

Array

(

[0]	=>	88
[1]	=>	44
[2]	=>	10
[3]	=>	7
[4]	=>	5
[5]	=>	5
[6]	=>	3

)

# Funções de arrays

```
<?php
    $nums=array(10,5,3,44,5,88,7);
?>

<!DOCTYPE html>
<html>
    <body>
        <?php
            shuffle($nums);
        ?>

        <pre>
            <?php
                print_r($nums)
            ?>
        </pre>
    </body>
</html>
```

Array

(

[0]	=>	5
[1]	=>	10
[2]	=>	44
[3]	=>	7
[4]	=>	88
[5]	=>	3
[6]	=>	5

)

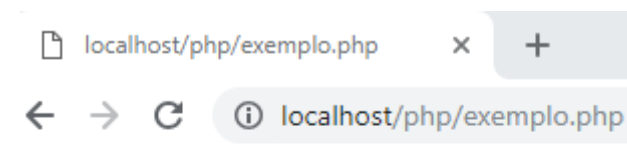
# GET/POST

- As variáveis GET e POST são consideradas superglobais, ou seja funcionam entre páginas permitindo a passagem de valores entre elas.
- No caso do GET as variáveis são enviadas no URL de destino.
- No caso do POST as variáveis não são visíveis.

Mais info em: [https://www.php.net/manual/pt\\_BR/language.variables.superglobals.php](https://www.php.net/manual/pt_BR/language.variables.superglobals.php)

# GET

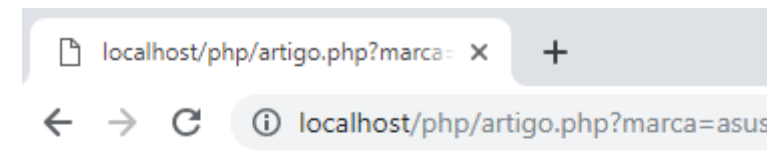
```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>
    <ul>
      <li>
        <a href="artigo.php?marca=asus">Asus</a>
      </li>
      <li>
        <a href="artigo.php?marca=hp">HP</a>
      </li>
    </ul>
  </body>
</html>
```



- [Asus](#)
- [HP](#)

# GET

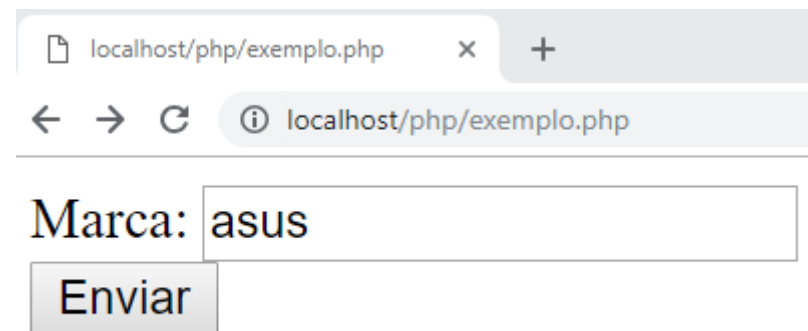
```
<!DOCTYPE html>
<html>
  <body>
    <pre>
      <?php
        print_r($_GET);
      ?>
    </pre>
  </body>
</html>
```



```
Array
(
    [marca] => asus
)
```

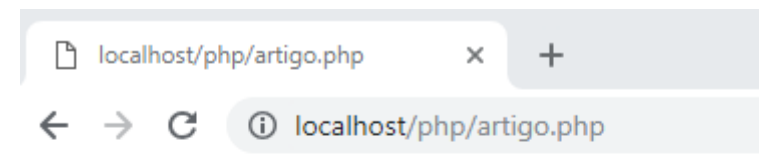
# POST

```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>
    <form action="artigo.php" method="post">
      <label for="marca">Marca: </label>
      <input type="text" name="marca" id="marca">
      <br>
      <input type="submit" value="Enviar">
    </form>
  </body>
</html>
```

A screenshot of a web browser window. The address bar shows 'localhost/php/exemplo.php'. The page content displays a form with the label 'Marca:' followed by a text input field containing the text 'asus'. Below the input field is a button with the text 'Enviar'.

# POST

```
<!DOCTYPE html>
<html>
  <body>
    <pre>
      <?php
        print_r($_POST);
      ?>
    </pre>
  </body>
</html>
```

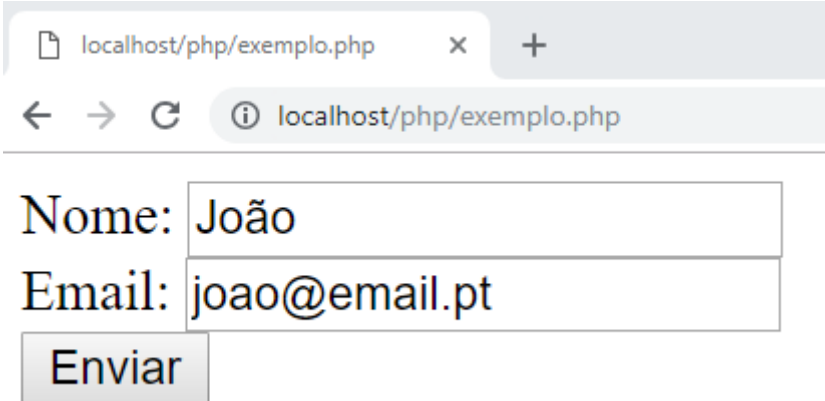


```
Array
(
    [marca] => asus
)
```



# Form

```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>
    <form action="artigo.php" method="post">
      <label for="marca">Nome: </label>
      <input type="text" name="nome" id="nome">
      <br>
      <label for="marca">Email: </label>
      <input type="text" name="email" id="email">
      <br>
      <input type="submit" value="Enviar">
    </form>
  </body>
</html>
```



localhost/php/exemplo.php x +

← → ↻ ⓘ localhost/php/exemplo.php

Nome: João

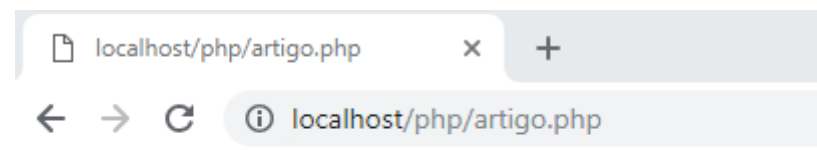
Email: joao@email.pt

Enviar

# Form

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      $nome=$_POST["nome"];
      $email=$_POST["email"];

      echo "Bem-vindo ".$nome." - ".$email;
    ?>
  </body>
</html>
```



Bem-vindo João - joao@email.pt

# Server-Side Includes

- Vantagens:
  - Ficheiros são unidos pelo lado Servidor
  - Ficheiros externos podem conter PHP, HTML ou texto
  - Reutilização de Código
  - Elementos comuns de uma página são atualizados apenas editando um ficheiro

# Server-Side Includes

- Comandos:
  - include
  - include\_once
  - require
  - require\_once

# Server-Side Includes

- include – tenta continuar o processamento da página mesmo se faltar o ficheiro
- require – para o processamento se o ficheiro externo não está disponível
- once – previne que o ficheiro seja incluído mais que uma vez

# Composer

- Gestor de dependências de PHP.
- Disponível em: <https://getcomposer.org/>

# Composer




A Dependency Manager for PHP

Latest: 1.9.3

[Getting Started](#)

[Download](#)

# Composer

 [Home](#) | [Getting Started](#) | [Download](#) | [Documentation](#) | [Browse Packages](#)

## Download Composer Latest: v1.9.3

### Windows Installer

The installer will download composer for you and set up your PATH environment variable so you can simply call `composer` from any directory.

Download and run [Composer-Setup.exe](#) - it will install the latest composer version whenever it is executed.

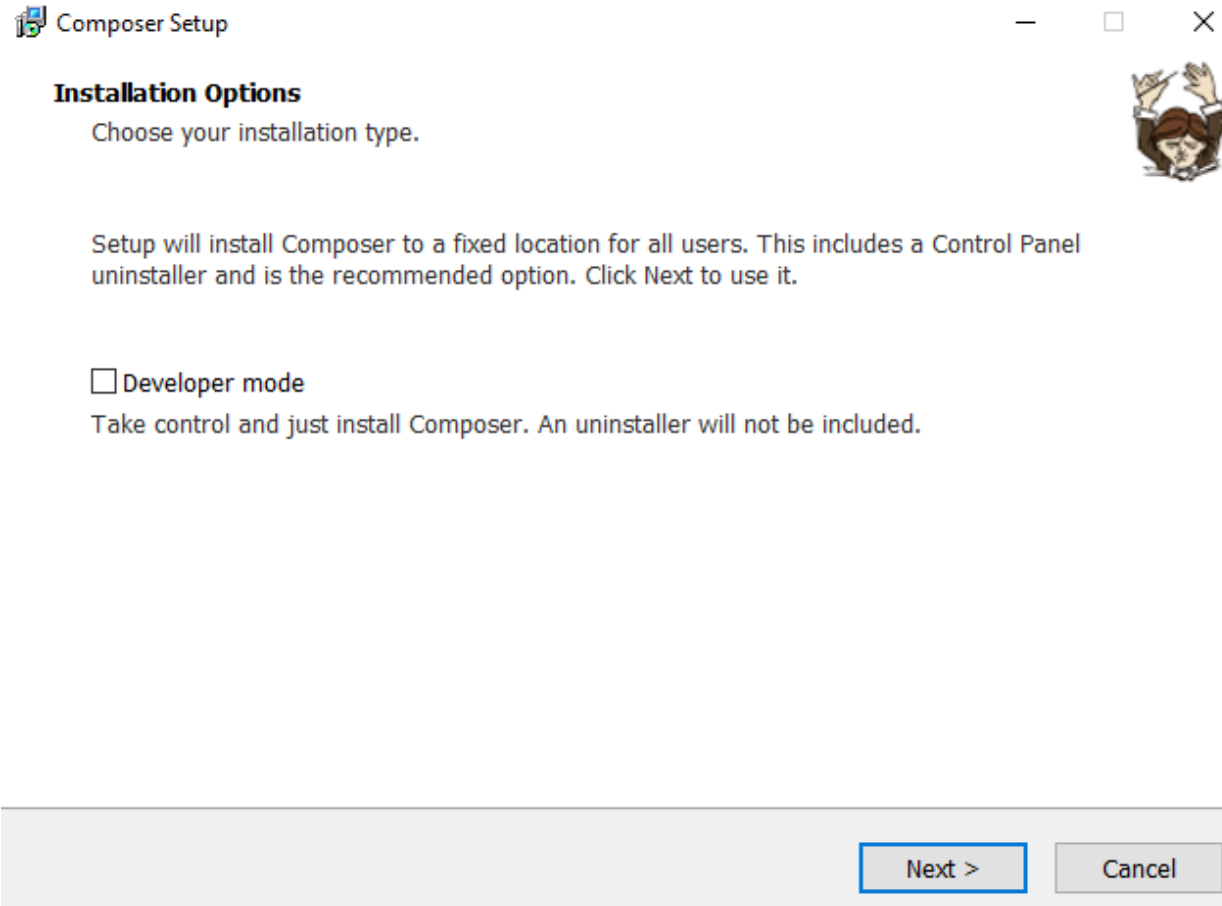
### Command-line installation

To quickly install Composer in the current directory, run the following script in your terminal. To automate the installation, use [the guide on installing Composer programmatically](#).

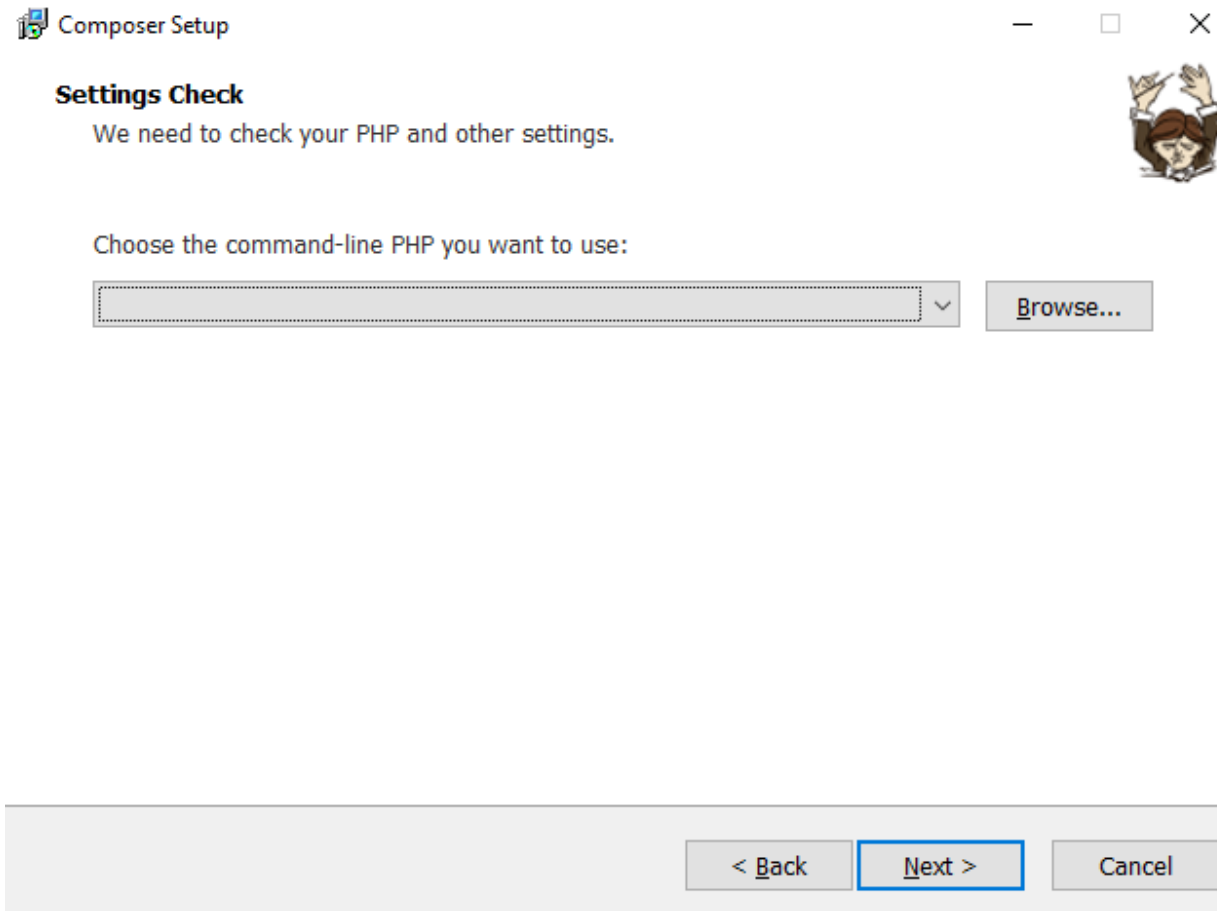
```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php -r "if (hash_file('sha384', 'composer-setup.php') === 'e0012edf3e80b6978849f5eff0d4b4e4c79ff1609dd1e613307e'
php composer-setup.php
php -r "unlink('composer-setup.php');"
```



# Composer

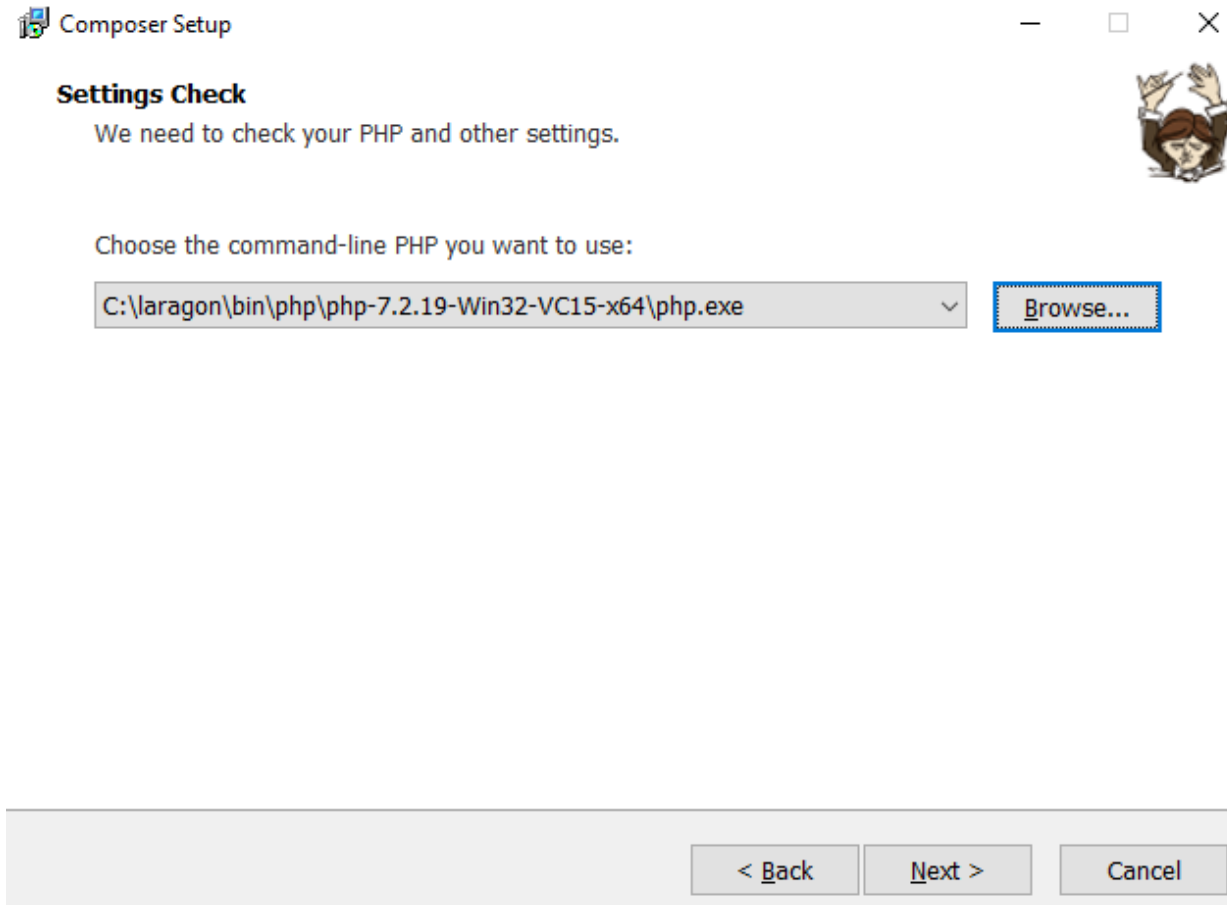


# Composer



C:\laragon\bin\php\XXX\php.exe (XXX – pasta com a versão pretendida)

# Composer



# Composer

Composer Setup

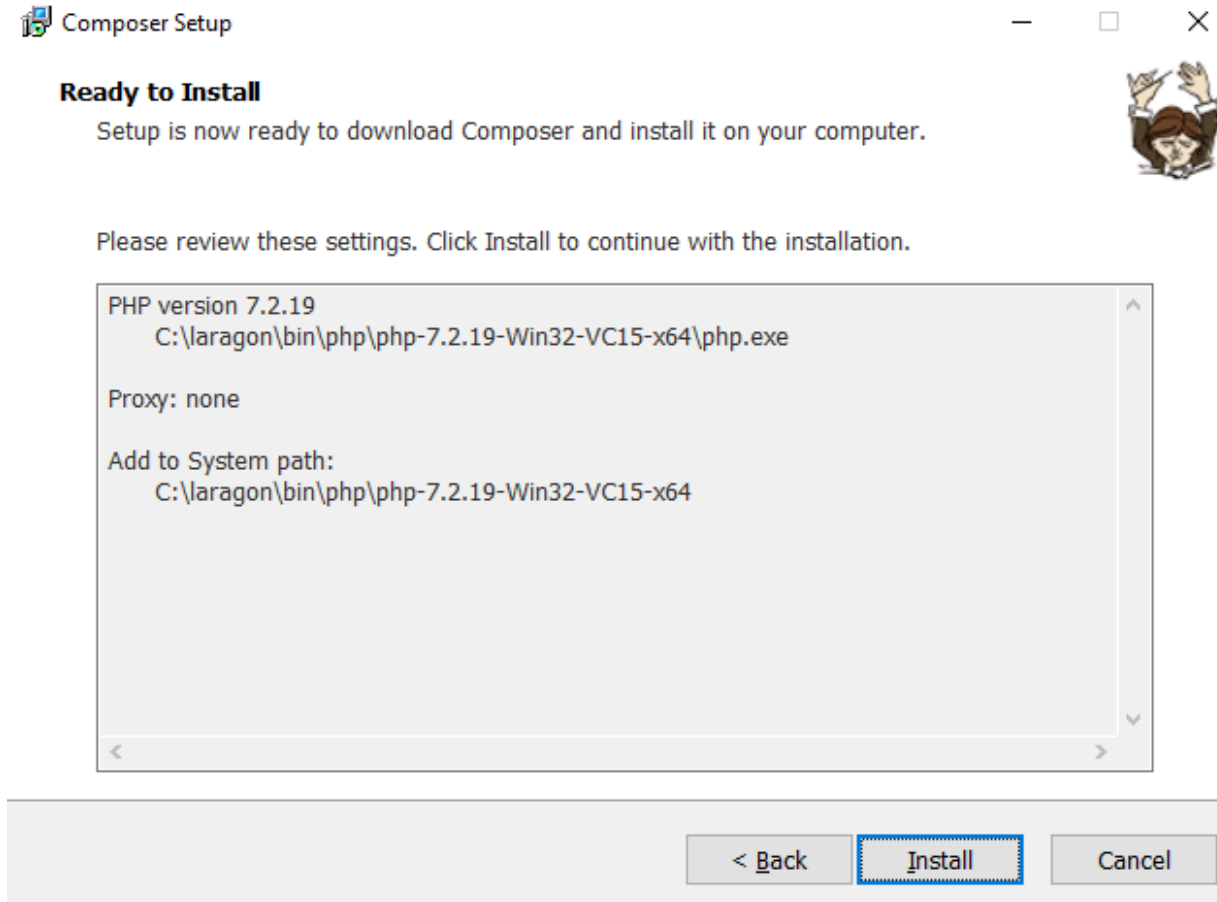
**Proxy Settings**  
Choose if you need to use a proxy.

☐ Use a proxy server to connect to internet

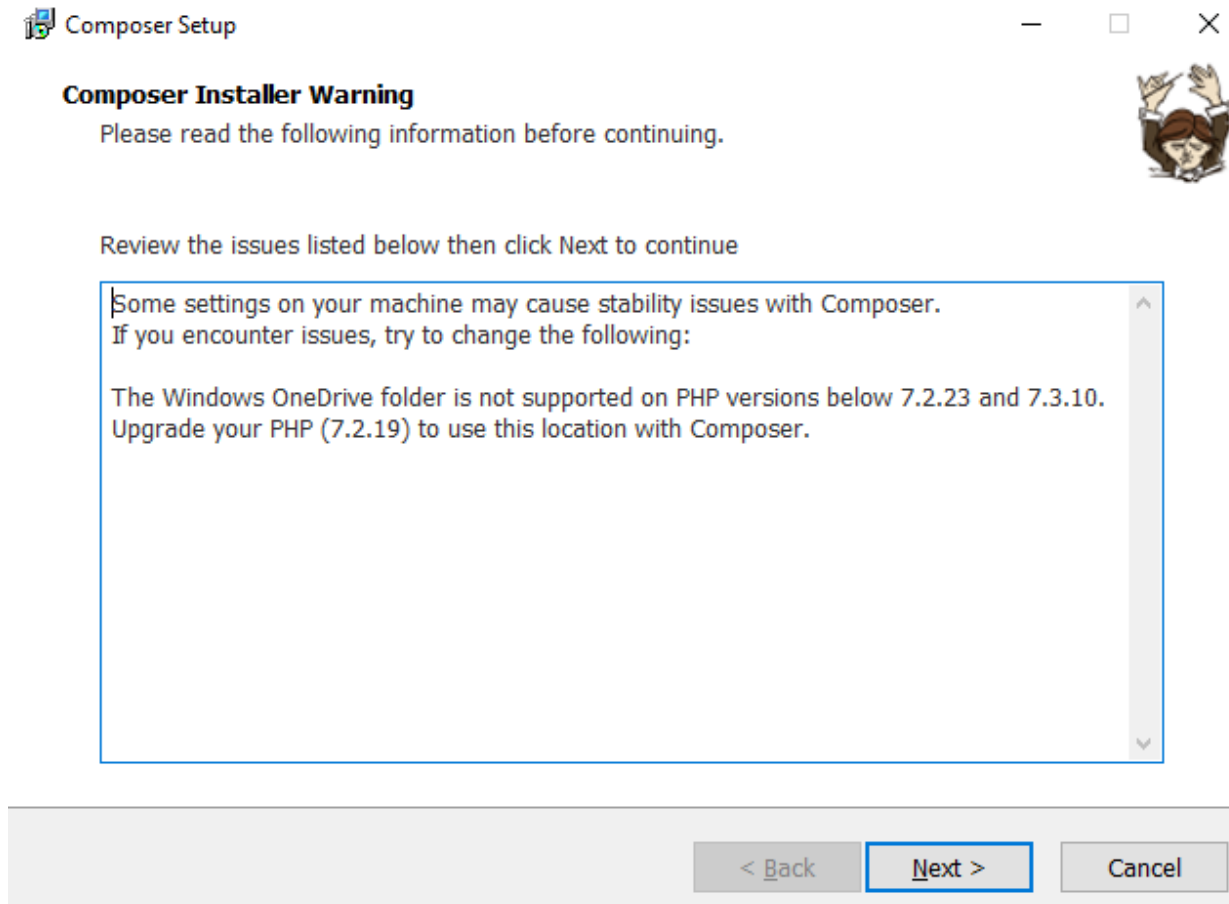
Enter proxy url

< Back Next > Cancel

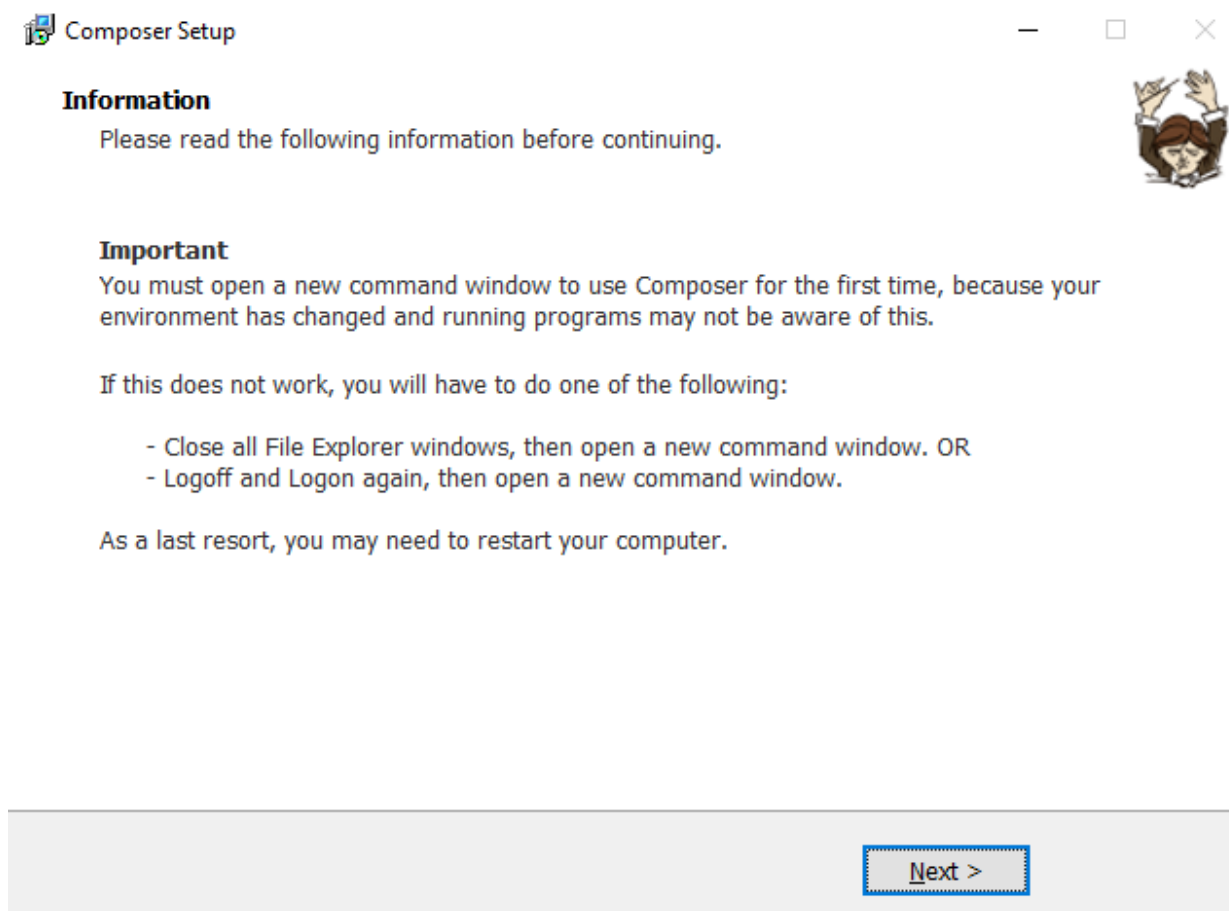
# Composer



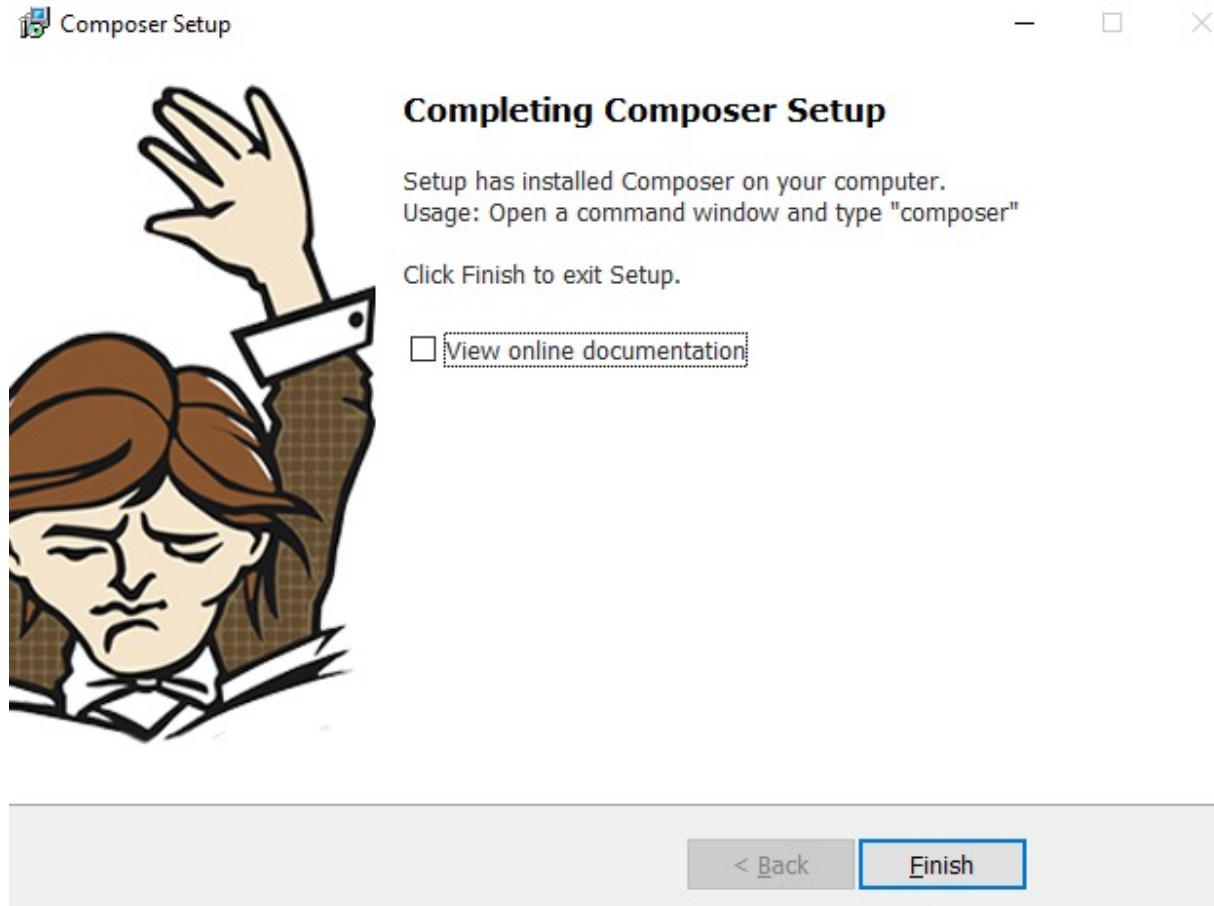
# Composer



# Composer

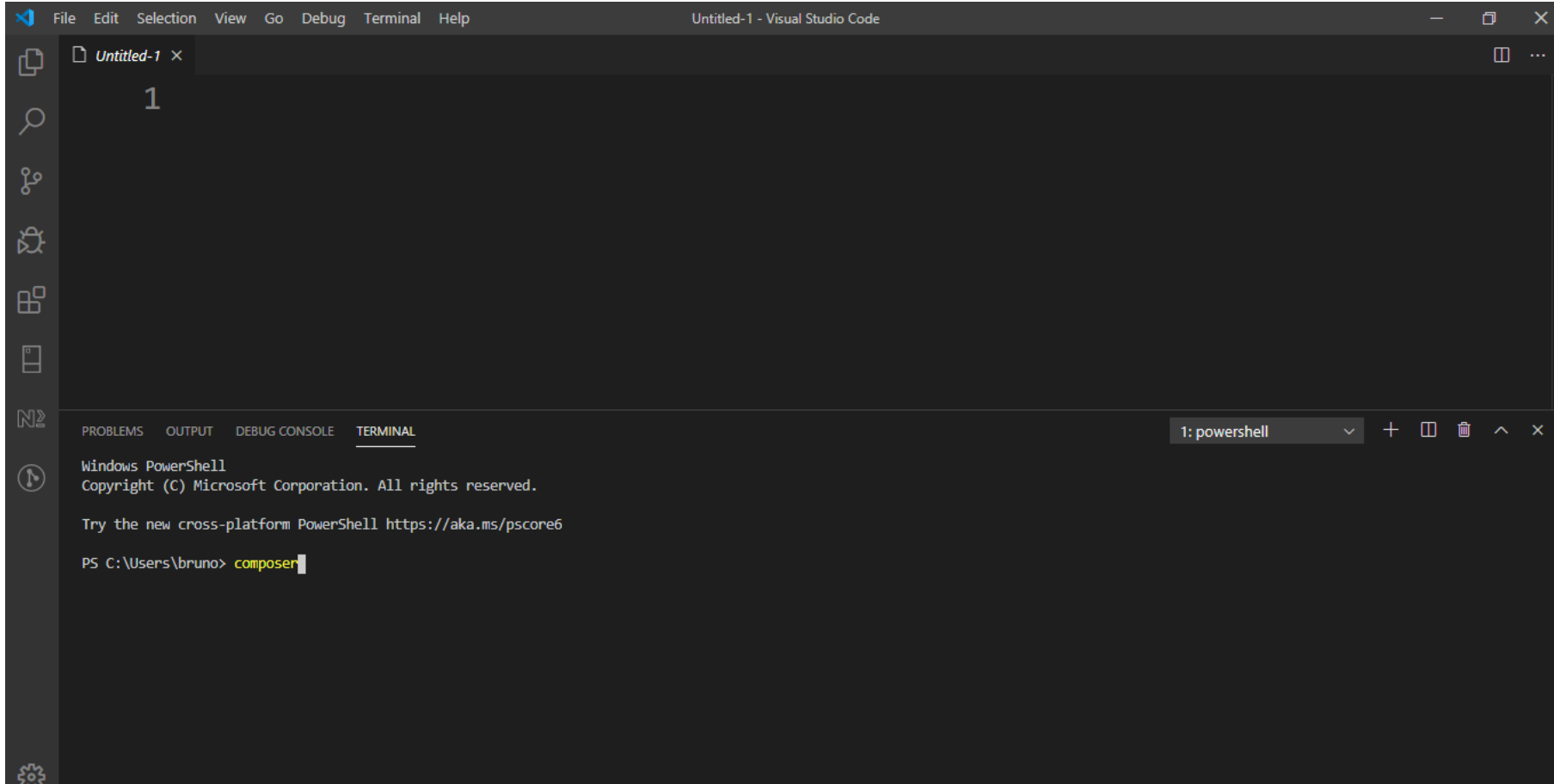


# Composer





# Composer



The screenshot shows the Visual Studio Code editor with a dark theme. The main editor area is empty, showing a single line with the number '1'. The bottom panel contains the 'TERMINAL' tab, which is active. The terminal output shows the Windows PowerShell prompt, copyright information, and a message about the new cross-platform PowerShell. The current command prompt is 'PS C:\Users\bruno> composer'.

```
File Edit Selection View Go Debug Terminal Help
Untitled-1 - Visual Studio Code

Untitled-1 x
1

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\bruno> composer
```

Abrir terminal no VS Code e escrever composer, caso sejam apresentados os comandos do composer significa que o mesmo está corretamente instalado

# PHPMailer

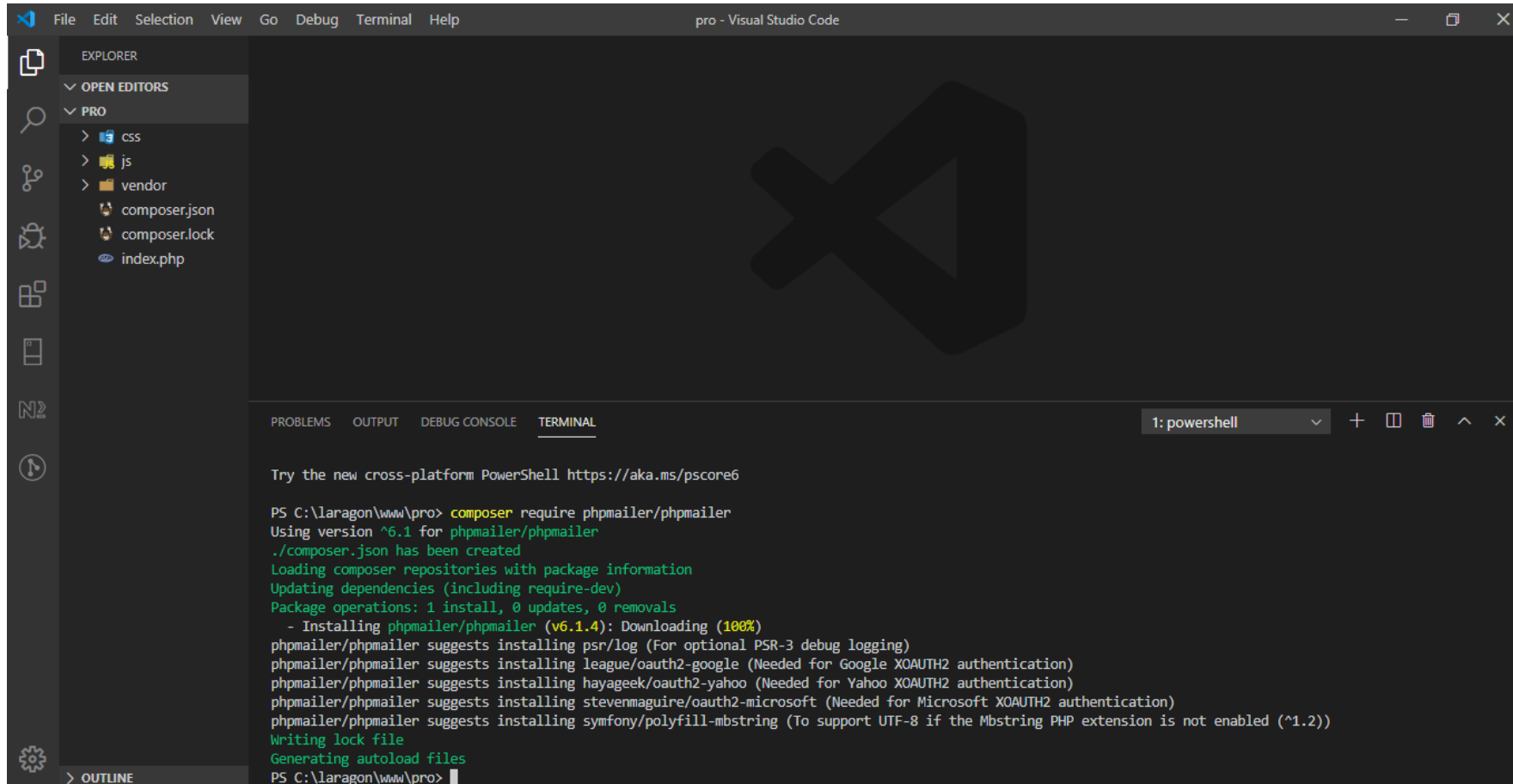
- Pacote de PHP que acrescenta a possibilidade de enviar emails a partir de um formulário PHP

<https://github.com/PHPMailer/PHPMailer>

- Abrindo a pasta do nosso projeto devemos agora no terminal executar o seguinte comando para instalar o PHPMailer:

```
composer require phpmailer/phpmailer
```

# PHPMailer



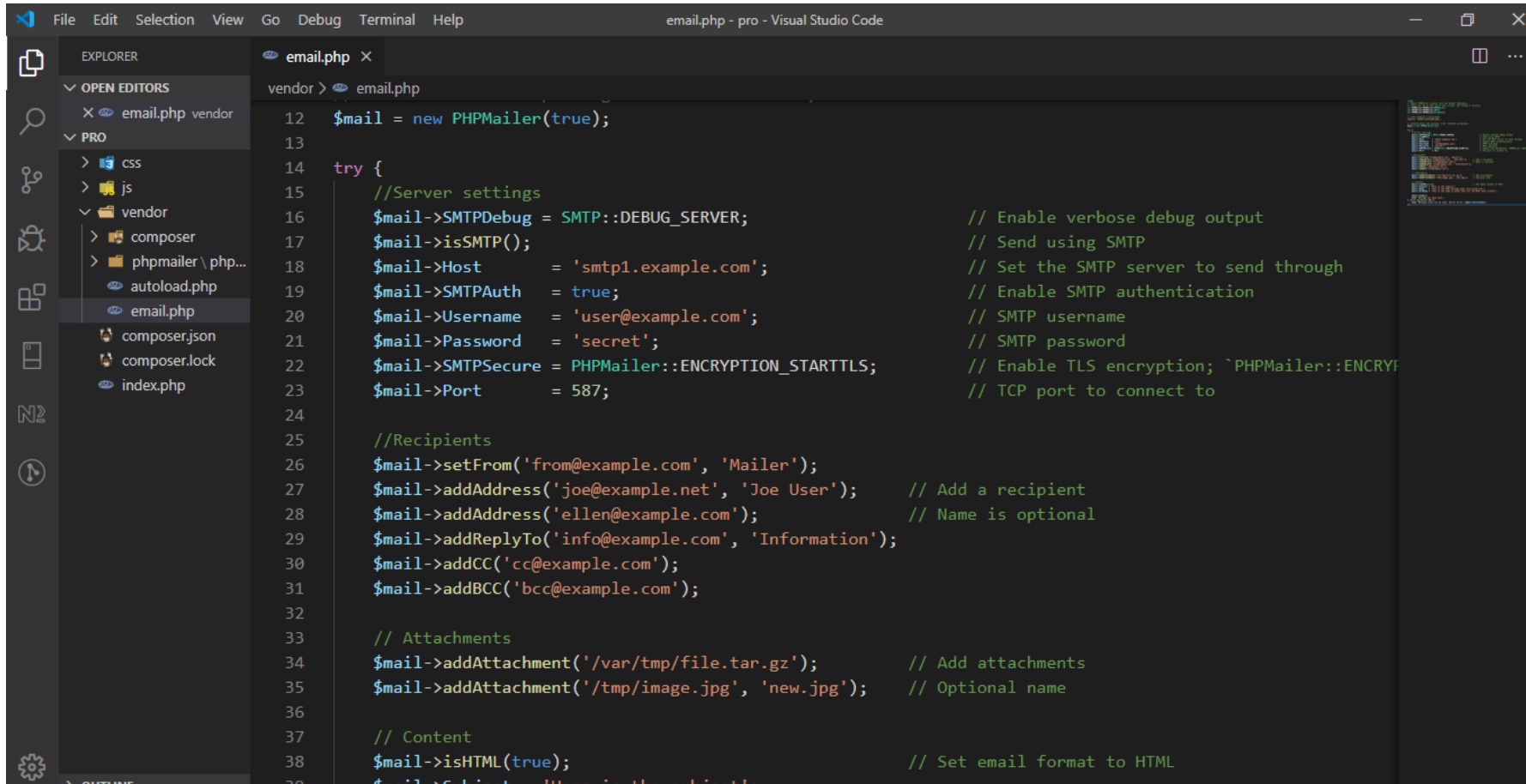
The screenshot shows the Visual Studio Code interface with a file explorer on the left showing a project structure with folders for 'css', 'js', and 'vendor', and files for 'composer.json', 'composer.lock', and 'index.php'. The main editor area is empty, displaying the Visual Studio Code logo. At the bottom, a terminal window is open, showing the execution of the following commands and output:

```
PS C:\laragon\www\pro> composer require phpmailer/phpmailer
Using version ^6.1 for phpmailer/phpmailer
./composer.json has been created
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
- Installing phpmailer/phpmailer (v6.1.4): Downloading (100%)
phpmailer/phpmailer suggests installing psr/log (For optional PSR-3 debug logging)
phpmailer/phpmailer suggests installing league/oauth2-google (Needed for Google XOAUTH2 authentication)
phpmailer/phpmailer suggests installing hayageek/oauth2-yahoo (Needed for Yahoo XOAUTH2 authentication)
phpmailer/phpmailer suggests installing stevenmaguire/oauth2-microsoft (Needed for Microsoft XOAUTH2 authentication)
phpmailer/phpmailer suggests installing symfony/polyfill-mbstring (To support UTF-8 if the Mbstring PHP extension is not enabled (^1.2))
Writing lock file
Generating autoload files
PS C:\laragon\www\pro>
```

# PHPMailer

- Dentro da pasta vendor vamos ter acesso à pasta com o PHP Mailer.
- Vamos agora voltar ao repositório do Github do PHP Mailer e copiar para um ficheiro o conteúdo de “A Simple Example”.

# PHPMailer



```
12 $mail = new PHPMailer(true);
13
14 try {
15     //Server settings
16     $mail->SMTPDebug = SMTP::DEBUG_SERVER;           // Enable verbose debug output
17     $mail->isSMTP();                                   // Send using SMTP
18     $mail->Host       = 'smtp1.example.com';          // Set the SMTP server to send through
19     $mail->SMTPAuth    = true;                         // Enable SMTP authentication
20     $mail->Username    = 'user@example.com';          // SMTP username
21     $mail->Password    = 'secret';                    // SMTP password
22     $mail->SMTPSecure  = PHPMailer::ENCRYPTION_STARTTLS; // Enable TLS encryption; `PHPMailer::ENCRYPTION_STARTTLS` will
23     $mail->Port        = 587;                          // TCP port to connect to
24
25     //Recipients
26     $mail->setFrom('from@example.com', 'Mailer');
27     $mail->addAddress('joe@example.net', 'Joe User'); // Add a recipient
28     $mail->addAddress('ellen@example.com');           // Name is optional
29     $mail->addReplyTo('info@example.com', 'Information');
30     $mail->addCC('cc@example.com');
31     $mail->addBCC('bcc@example.com');
32
33     // Attachments
34     $mail->addAttachment('/var/tmp/file.tar.gz');     // Add attachments
35     $mail->addAttachment('/tmp/image.jpg', 'new.jpg'); // Optional name
36
37     // Content
38     $mail->isHTML(true);                             // Set email format to HTML
39     $mail->Subject = 'Here is the subject';
```

# PHPMailer

- Preenchemos agora os campos “host”, “username” e “password” para configuração da conta.
- Preencher os campos da zona “Receipts” (basta setFrom e setAddress)
- Neste exemplo vamos apagar a parte dos “Attachments” uma vez que não vamos anexar qualquer ficheiro ao email
- E finalmente alterar o “Content” do email (podemos retirar o AltBody).

# PHPMailer

- Tomando em consideração que vamos usar uma conta Gmail temos de:
  - Ativar o IMAP
  - Permitir o acesso a aplicações externas não verificadas

# PHPMailer

## Definições



Geral Marcadores Caixa de entrada Contas e importação Filtros e endereços bloqueados Encaminhamento e POP/IMAP Suplementos

Chat Avançadas Offline Temas

### Encaminhamento:

[Saiba mais](#)

Adicionar um endereço de encaminhamento

Sugestão: Pode também encaminhar apenas algumas das suas mensagens ao [criar um filtro!](#)

### Transferência via POP:

[Saiba mais](#)

1. Estado: O **POP foi ativado** para todo o correio que chegou desde 14/02/08
- ☐ Ativar POP para **todo o correio** (mesmo correio que já tenha sido transferido)
  - ☐ A partir de agora, ativar o POP para **correio a receber**
  - ☐ **Desativar POP**

2. Quando as mensagens são acedidas com POP

3. Configure o seu cliente de email (por exemplo, Outlook, Eudora, Netscape Mail)

[Instruções de configuração](#)

### Acesso IMAP:

(acesso a Gmail por parte de outros clientes que utilizem o IMAP)

Estado: **O IMAP está ativado**

- ☒ Ativar IMAP
- ☐ Desativar IMAP



# PHPMailer

## Acesso a aplicações menos seguras

Para proteger a sua conta, as aplicações e os dispositivos que utilizam tecnologia de início de sessão menos segura estão bloqueados. Para manter a sua conta segura, a Google irá desativar automaticamente esta definição se não estiver a ser utilizada. [Saiba mais](#)

— Desativado

[Ativar o acesso \(não recomendado\)](#)



<https://myaccount.google.com/security>

# PHPMailer

```
$mail->SMTPDebug = SMTP::DEBUG_SERVER;
$mail->isSMTP();
$mail->Host      = 'smtp.gmail.com';
$mail->SMTPAuth  = true;
$mail->Username  = 'brunofrs7@gmail.com';
$mail->Password  = '          ';
$mail->SMTPSecure = PHPMailer::ENCRYPTION_STARTTLS;
$mail->Port      = 587;

//Recipients
$mail->setFrom('brunofrs7@gmail.com', 'Bruno');
$mail->addAddress('brunofrs7@gmail.com', 'Teste');

// Content
$mail->isHTML(true);
$mail->Subject = 'Assunto do Email';
$mail->Body    = 'Corpo do email';
```

# PHPMailer

```
2020-02-16 18:09:00 SERVER -> CLIENT: 220 smtp.gmail.com ESMTP s139sm17624839wme.35 - gsmt
2020-02-16 18:09:00 CLIENT -> SERVER: EHLO localhost
2020-02-16 18:09:00 SERVER -> CLIENT: 250-smtp.gmail.com at your service, [176.79.185.60]250-SIZE 35882577250-8BITM
2020-02-16 18:09:00 CLIENT -> SERVER: STARTTLS
2020-02-16 18:09:00 SERVER -> CLIENT: 220 2.0.0 Ready to start TLS
2020-02-16 18:09:00 CLIENT -> SERVER: EHLO localhost
2020-02-16 18:09:00 SERVER -> CLIENT: 250-smtp.gmail.com at your service, [176.79.185.60]250-SIZE 35882577250-8BITM
CHUNKING250 SMTPUTF8
2020-02-16 18:09:00 CLIENT -> SERVER: AUTH LOGIN
2020-02-16 18:09:00 SERVER -> CLIENT: 334 VXNlcm5hbWU6
2020-02-16 18:09:00 CLIENT -> SERVER: [credentials hidden]
2020-02-16 18:09:00 SERVER -> CLIENT: 334 UGFzc3dvcmQ6
2020-02-16 18:09:00 CLIENT -> SERVER: [credentials hidden]
2020-02-16 18:09:01 SERVER -> CLIENT: 235 2.7.0 Accepted
2020-02-16 18:09:01 CLIENT -> SERVER: MAIL FROM:<brunofrs7@gmail.com>
2020-02-16 18:09:01 SERVER -> CLIENT: 250 2.1.0 OK s139sm17624839wme.35 - gsmt
2020-02-16 18:09:01 CLIENT -> SERVER: RCPT TO:<brunofrs7@gmail.com>
2020-02-16 18:09:01 SERVER -> CLIENT: 250 2.1.5 OK s139sm17624839wme.35 - gsmt
2020-02-16 18:09:01 CLIENT -> SERVER: DATA
2020-02-16 18:09:01 SERVER -> CLIENT: 354 Go ahead s139sm17624839wme.35 - gsmt
2020-02-16 18:09:01 CLIENT -> SERVER: Date: Sun, 16 Feb 2020 18:09:00 +0000
2020-02-16 18:09:01 CLIENT -> SERVER: To: Teste <brunofrs7@gmail.com>
2020-02-16 18:09:01 CLIENT -> SERVER: From: Bruno <brunofrs7@gmail.com>
2020-02-16 18:09:01 CLIENT -> SERVER: Subject: Assunto do Email
2020-02-16 18:09:01 CLIENT -> SERVER: Message-ID: <W5Jf0xli0QpT6QZ6Vj21XuTMVZ9JUlhxRFsief0ac@localhost>
2020-02-16 18:09:01 CLIENT -> SERVER: X-Mailer: PHPMailer 6.1.4 (https://github.com/PHPMailer/PHPMailer)
2020-02-16 18:09:01 CLIENT -> SERVER: MIME-Version: 1.0
2020-02-16 18:09:01 CLIENT -> SERVER: Content-Type: text/html; charset=iso-8859-1
2020-02-16 18:09:01 CLIENT -> SERVER:
2020-02-16 18:09:01 CLIENT -> SERVER: Corpo do email
2020-02-16 18:09:01 CLIENT -> SERVER:
2020-02-16 18:09:01 CLIENT -> SERVER:
2020-02-16 18:09:02 SERVER -> CLIENT: 250 2.0.0 OK 1581876541 s139sm17624839wme.35 - gsmt
2020-02-16 18:09:02 CLIENT -> SERVER: QUIT
2020-02-16 18:09:02 SERVER -> CLIENT: 221 2.0.0 closing connection s139sm17624839wme.35 - gsmt
Message has been sent
```

# PHPMailer

Assunto do Email ➤ Caixa de entrada x



**Bruno** <brunofrs7@gmail.com>  
para mim ▼

Corpo do email

# PHP + MySQL

- Tipos de ligação à base de dados:
  - MySQLi
  - PDO (PHP Data Objects)
- MySQLi funciona só com bases de dados MySQL
- PDO funciona com 12 tipos de bases de dados diferentes assim a migração entre bases de dados é mais simples e há maior reaproveitamento de código.

# Ligação à bd (PDO)

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

try {
    $conn = new PDO("mysql:host=$servername;dbname=myDB", $username, $password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    echo "Connected successfully";
} catch(PDOException $e) {
    echo "Connection failed: " . $e->getMessage();
}
?>
```

[https://www.w3schools.com/php/php\\_mysql\\_connect.asp](https://www.w3schools.com/php/php_mysql_connect.asp)

# Fechar Ligação (PDO)

```
$conn = null;
```

[https://www.w3schools.com/php/php\\_mysql\\_connect.asp](https://www.w3schools.com/php/php_mysql_connect.asp)

# Criar BD (PDO)

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

try {
    $conn = new PDO("mysql:host=$servername", $username, $password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $sql = "CREATE DATABASE myDBPDO";
    // use exec() because no results are returned
    $conn->exec($sql);
    echo "Database created successfully<br>";
} catch(PDOException $e) {
    echo $sql . "<br>" . $e->getMessage();
}

$conn = null;
?>
```

[https://www.w3schools.com/php/php\\_mysql\\_create.asp](https://www.w3schools.com/php/php_mysql_create.asp)



# Criar tabela (PDO)

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDBPDO";

try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    // sql to create table
    $sql = "CREATE TABLE MyGuests (
    id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    firstname VARCHAR(30) NOT NULL,
    lastname VARCHAR(30) NOT NULL,
    email VARCHAR(50),
    reg_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
    )";

    // use exec() because no results are returned
    $conn->exec($sql);
    echo "Table MyGuests created successfully";
} catch(PDOException $e) {
    echo $sql . "<br>" . $e->getMessage();
}

$conn = null;
?>
```

[https://www.w3schools.com/php/php\\_mysql\\_create\\_table.asp](https://www.w3schools.com/php/php_mysql_create_table.asp)

# Inserir dados (PDO)

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDBPDO";

try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com')";
    // use exec() because no results are returned
    $conn->exec($sql);
    echo "New record created successfully";
} catch(PDOException $e) {
    echo $sql . "<br>" . $e->getMessage();
}

$conn = null;
?>
```

[https://www.w3schools.com/php/php\\_mysql\\_insert.asp](https://www.w3schools.com/php/php_mysql_insert.asp)

# Inserir e retornar id (PDO)

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDBPDO";

try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com')";
    // use exec() because no results are returned
    $conn->exec($sql);
    $last_id = $conn->lastInsertId();
    echo "New record created successfully. Last inserted ID is: " . $last_id;
} catch(PDOException $e) {
    echo $sql . "<br>" . $e->getMessage();
}

$conn = null;
?>
```

[https://www.w3schools.com/php/php\\_mysql\\_insert\\_lastid.asp](https://www.w3schools.com/php/php_mysql_insert_lastid.asp)

# Inserir vários registos (PDO)

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDBPDO";

try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    // begin the transaction
    $conn->beginTransaction();
    // our SQL statements
    $conn->exec("INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com')");
    $conn->exec("INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('Mary', 'Moe', 'mary@example.com')");
    $conn->exec("INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('Julie', 'Dooley', 'julie@example.com')");

    // commit the transaction
    $conn->commit();
    echo "New records created successfully";
} catch(PDOException $e) {
    // roll back the transaction if something failed
    $conn->rollback();
    echo "Error: " . $e->getMessage();
}

$conn = null;
?>
```

[https://www.w3schools.com/php/php\\_mysql\\_insert\\_multiple.asp](https://www.w3schools.com/php/php_mysql_insert_multiple.asp)

# Prepared statements (PDO)

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDBPDO";

try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    // prepare sql and bind parameters
    $stmt = $conn->prepare("INSERT INTO MyGuests (firstname, lastname, email)
VALUES (:firstname, :lastname, :email)");
    $stmt->bindParam(':firstname', $firstname);
    $stmt->bindParam(':lastname', $lastname);
    $stmt->bindParam(':email', $email);
```

```
    // insert a row
    $firstname = "John";
    $lastname = "Doe";
    $email = "john@example.com";
    $stmt->execute();

    // insert another row
    $firstname = "Mary";
    $lastname = "Moe";
    $email = "mary@example.com";
    $stmt->execute();

    // insert another row
    $firstname = "Julie";
    $lastname = "Dooley";
    $email = "julie@example.com";
    $stmt->execute();

    echo "New records created successfully";
} catch(PDOException $e) {
    echo "Error: " . $e->getMessage();
}
$conn = null;
?>
```

[https://www.w3schools.com/php/php\\_mysql\\_prepared\\_statements.asp](https://www.w3schools.com/php/php_mysql_prepared_statements.asp)

# SELECT (PDO)

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    echo "<table><tr><th>ID</th><th>Name</th></tr>";
    // output data of each row
    while($row = $result->fetch_assoc()) {
        echo "<tr><td>".$row["id"]."</td><td>".$row["firstname"]." ".$row["lastname"]."</td></tr>";
    }
    echo "</table>";
} else {
    echo "0 results";
}
$conn->close();
?>
```

[https://www.w3schools.com/php/php\\_mysql\\_select.asp](https://www.w3schools.com/php/php_mysql_select.asp)

# SELECT where (PDO)

```
<?php
echo "<table style='border: solid 1px black;'>";
echo "<tr><th>Id</th><th>Firstname</th><th>Lastname</th></tr>";

class TableRows extends RecursiveIteratorIterator {
    function __construct($it) {
        parent::__construct($it, self::LEAVES_ONLY);
    }

    function current() {
        return "<td style='width:150px;border:1px solid black;'>" . parent::current(). "</td>";
    }

    function beginChildren() {
        echo "<tr>";
    }

    function endChildren() {
        echo "</tr>" . "\n";
    }
}
```

```
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDBPDO";

try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $stmt = $conn->prepare("SELECT id, firstname, lastname FROM MyGuests WHERE lastname='Doe'");
    $stmt->execute();

    // set the resulting array to associative
    $result = $stmt->setFetchMode(PDO::FETCH_ASSOC);
    foreach(new TableRows(new RecursiveArrayIterator($stmt->fetchAll())) as $k=>$v) {
        echo $v;
    }
} catch(PDOException $e) {
    echo "Error: " . $e->getMessage();
}
$conn = null;
echo "</table>";
?>
```

[https://www.w3schools.com/php/php\\_mysql\\_select\\_where.asp](https://www.w3schools.com/php/php_mysql_select_where.asp)

# SELECT order by (PDO)

```
<?php
echo "<table style='border: solid 1px black;'>";
echo "<tr><th>Id</th><th>Firstname</th><th>Lastname</th></tr>";
```

```
class TableRows extends RecursiveIteratorIterator {
    function __construct($it) {
        parent::__construct($it, self::LEAVES_ONLY);
    }

    function current() {
        return "<td style='width:150px;border:1px solid black;'>" . parent::current(). "</td>";
    }

    function beginChildren() {
        echo "<tr>";
    }

    function endChildren() {
        echo "</tr>" . "\n";
    }
}
```

```
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDBPDO";
```

```
try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $stmt = $conn->prepare("SELECT id, firstname, lastname FROM MyGuests ORDER BY lastname");
    $stmt->execute();

    // set the resulting array to associative
    $result = $stmt->setFetchMode(PDO::FETCH_ASSOC);
    foreach(new TableRows(new RecursiveArrayIterator($stmt->fetchAll())) as $k=>$v) {
        echo $v;
    }
} catch(PDOException $e) {
    echo "Error: " . $e->getMessage();
}
$conn = null;
echo "</table>";
?>
```

[https://www.w3schools.com/php/php\\_mysql\\_select\\_orderby.asp](https://www.w3schools.com/php/php_mysql_select_orderby.asp)



# SELECT Delete (PDO)

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDBPDO";

try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    // sql to delete a record
    $sql = "DELETE FROM MyGuests WHERE id=3";

    // use exec() because no results are returned
    $conn->exec($sql);
    echo "Record deleted successfully";
} catch(PDOException $e) {
    echo $sql . "<br>" . $e->getMessage();
}

$conn = null;
?>
```

[https://www.w3schools.com/php/php\\_mysql\\_delete.asp](https://www.w3schools.com/php/php_mysql_delete.asp)

# SELECT Update (PDO)

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDBPDO";

try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    $sql = "UPDATE MyGuests SET lastname='Doe' WHERE id=2";

    // Prepare statement
    $stmt = $conn->prepare($sql);

    // execute the query
    $stmt->execute();

    // echo a message to say the UPDATE succeeded
    echo $stmt->rowCount() . " records UPDATED successfully";
} catch(PDOException $e) {
    echo $sql . "<br>" . $e->getMessage();
}

$conn = null;
?>
```

[https://www.w3schools.com/php/php\\_mysql\\_update.asp](https://www.w3schools.com/php/php_mysql_update.asp)