PHP 1

Vamos el dia de hoy a hablar sobre PHP y MySQL.

Cual es la diferencia entre PHP y otros lenguajes de programación?

Algunas personas dicen que las siglas PHP no corresponden a nada y otras personas argumentan que se significa Hypertext Preprocessor, es un lenguaje del lado del servidor de scripts. Lo podríamos comparar con lenguajes como ASP o .NET o Perl. Se ejecuta en el servidor y luego crea código HTML que sera desplegado en el explorador. La razón para la que necesitamos PHP es para hacer ciertas cosas que no podríamos hacer en el lado del cliente con HTML y JavaScript. Por ejemplo comunicarnos con una base de datos o crear un archivo de texto a partir de inputs que el usuario genera. Manejar juegos multiplayer o experiencias multiusuarios puede ser realizado con PHP. Mientras PHP es el lenguaje de programación que usaremos MySQL es el nombre de la base de datos. Es un sistema de manejo de datos relacional. Básicamente nos permite crear relaciones entre elementos en una base de datos, para no repetir datos.

Lo primero que necesitamos para generar PHP y base de datos en un editor de texto.

http://www.wampserver.com/ (windows)

https://www.mamp.info/en/ (MAC)

Normalmente con HTML y JavaScript podemos probar nuestros códigos directamente en el explorador pero en el caso de PHP y base de datos necesitamos probar nuestro código en un servidor. Es por eso que descargamos MAMP.

Una vez activado nuestro servidor local accederemos a la dirección localhost:8888. Local Host simplemente nos esta diciendo que nuestro servidor esta corriendo.

Donde ponemos los archivos que se desplegaran en el explorador? Nos dirigimos a nuestra ventana principal de MAMP folder, después buscamos la carpeta htdocs en windows se llama www.

Intentemos meter un archivo simple HTML para ver que pasa antes de que llegue a la pantalla del explorador.

Este software nos permite que en el caso de PHP el codigo se corra desde el servidor.

Vamos a crear nuestro primer programa PHP.

```
<html>
	<head>
		<title>First PHP Document</title>
	</head>
	<body>
		<?php
			echo(“This is my first PHP document”);

		?>
	</body>
</html>
```

Lo primero que deben saber para generar un documento PHP es saber como cerrar y abrir una sección de nuestro documento. La sección de PHP se procesara diferente que la de HTML. Salvar el documento como .php

guardemoslo como firt_document

echo es un comando en PHP que presenta contenido al explorador. No necesariamente necesita los paréntesis es una excepción.

Cada linea de programación termina con un punto y como (;)

```
<html>
	<head>
		<title>Mi primer php documento</title>
	</head>
	<body>

		<?php
			echo ("<p><strong>Este es nuestro primer archivo PHP</strong></p>");
		?>
	</body>
</html>
```

A scripting language or script language is a programming language that supports scripts, programs written for a special run-time environment that can interpret (rather than compile) and automate the execution of tasks that could alternatively be executed one-by-one by a human operator.

PHP 2

```
		<?php
			echo ("<p><strong>Este es nuestro primer archivo PHP</strong></p>");
		?>
```

No es necesario poner la estructura del HTML, aunque no es recomendable ya que no estamos dando ninguna tipo de estructura al explorador.

---

Salvar como output.php

Algo que tenemos que tomar en cuenta es el manual de PHP http://php.net/manual/en/index.php

Print function, es una función que puede funcionar como echo

```
html>
  <head>
    <title>Output</title>
  </head>
  <body>
    <?php
    echo "This is echoed out.";
```

```
        echo ("<br/>This is also echoed out.");
        print("<br/>Look ma! I didn't use echo, I used print!");
        print "<br/>Print also works without the aid of ().";
        printf("<br/>Mark is %d years old.", 37);
        printf("<br/>The interest rate is %0.2f", 8.993124645);
        ?>
    </body
</html>
```

Existen 3 formas de presentar contenido en PHP:
echo
print
prinf que quiere decir print with format y sirve para formatear información y hacerla mas “accesible al usuario”.
signifiers—

PHP2

Variables

Esta clase hablaremos sobre variables, las variables básicamente son usadas para guarda informació y podemos guardar 2 tipos de información Strings o valores.

Vamos generar un nuevo documento con una estructura básica HTML, después abriremos PHP, comenzamos a escribir nuestra primera variable, en PHP las variables comienzan con un signo de dólar y después el nombre de la variable.

Value or Numerical Variable
String

Los comentarios en PHP se realizan con //

En php para poder ligar las variables usamos un operador llamado concatenation y es un (.)

Con en nombre de variables únicamente usaremos letras y _

Se darán cuenta que en PHP no mencionamos el tipo de la variable como en otros lenguajes de programación.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd"
    >
<html lang="en">
<head>
    <title>Variables</title>
</head>
<body>
    <?php
        $age = 37;                  //Value or Numerical Variable
        $name = "Mark Lassoff";     //String Variable
        $tax_rate = .06;            //Floating point number

        print($name);
        print("<br/>");
        print($age);
        print("<br/>");
```

```
        print($name . " is " . $age . " years old");      //Concatenation
        print("<br/>");
        printf("The tax rate in Connecticut is %0.2f", $tax_rate);

        $value = 37*3/14+25-6+.003/6;
        print("<br/>");
        print("Value: " . $value);

    ?>
</body>
</html>
```

Operadores

```
<html>
    <head>
        <title>Operator</title>
    </head>
    <body>
        <?php

            $x = 12.55;
            $y = 13;

            $z = $x + $y;

            echo($x . " + "  . $y . " = " . ($x + $y));
            echo("<br/>");
            echo($x . " - "  . $y . " = " . ($x - $y));
            echo("<br/>");
            echo($x . " * "  . $y . " = " . ($x * $y));
            echo("<br/>");
            echo($x . " / "  . $y . " = " . ($x / $y));
            echo("<br/>");
            echo(9%3);
            echo("<br/>");
            echo(10%3);
            echo("<br/>");
            echo(11%3);

            $x++;
            $y--;
            echo("<br/>");
            echo("x: "  . $x . " y: " . $y);

            $poem = "Roses are red, voilets are blue";
            $poem .= "<br/>PHP is fun and so are you";
            echo("<br/>");
            echo $poem;

            $operand1 = 17;
            $operand2 = 13;

            $operand2 -=$operand1;
            echo("<br/>");
            echo($operand2);
```

```
        ?>
    </body>
</html>
```

modulus %

Hicimos suma, multiplicación, división, modulus, que es el restante de la division, hicimos increment and decrement y realizamos operadores de aislamiento combinado

_____

PHP 3

Arrays

Los arreys son cierto tipo de variable que pueden contener mas de un valor. Son bueno por ejemplo para generar una lista.

arrey_simple.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd"
    >
<html lang="en">
<head>
    <title>Simple Arrays</title>
</head>
<body>
    <?php
        $family = array("Mark", "Joan", "Rick", "Brett", "Kerry", "Kevin");

        $vehicles[0] = "car";
        $vehicles[1] = "airplane";
        $vehicles[2] = "cruise ship";
        $vehicles[3] = "train";


        print($family[0] . " just purchased a new ". $vehicles[0]);
        print("<br/>");
        print($family[3] . " vacations on a " . $vehicles[2]);

    ?>
</body>
</html>
```

___

Los Arreys asociativos nos permiten asociar un indice de texto, en lugar de un indice numérico.

arrey_assoc.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
   "http://www.w3.org/TR/html4/strict.dtd"
   >
<html lang="en">
<head>
   <title>Associative Arrays</title>
</head>
<body>
   <?php
      //Declare the Array
      $gpas = array(
               "Smith" => 3.25,
               "Johnson" => 3.1,
               "Blackstone" => 2.5,
               "Craigson" => 1.77,
               "Garcia" => 4.0,
               "Wendell" =>3.11

            );

      print("Eric Smith has a gpa of " . $gpas["Smith"] . "<br/>");
      print("Fred Garcia has a gpa of " . $gpas["Garcia"]);
   ?>
</body>
</html>
```

_____

Los Arreys multidimensionales son basicamente arreglos de arreys. Son muy útiles si tenemos datos mas complejos.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
   "http://www.w3.org/TR/html4/strict.dtd"
   >
<html lang="en">
<head>
   <title>Multi Dimensional</title>
</head>
<body>
   <?php
      $teams = array
      (
         "Yankees" => array
            (
               "Rivera",
               "Jeter",
               "Granderson",
               "Sabathia",
               "Gardner"
            ),
         "Mets" => array
            (
               "Dickey",
               "Acosta",
               "Pelfrey"
```

```
            ),
          "Red Sox" => array
            (
               "Ortiz",
               "Bard",
               "Buckholz",
               "Beckett"
            )
       );

       echo($teams['Red Sox'][0]);
       echo("<br/>");
       echo($teams['Yankees'][3]);

   ?>
</body>
</html>
```

_____

Super Global Arreys

Esta parte del PHP es de mis favoritas ya que podemos ver mas posibilidades de lo que el pHP puede hacer.

form.html

En HTML sabemos que podemos generar un formulario y luego pasar ese formulario al servidor para manejar la información y hacer algo de computo.

Calculadora de años de perros

Lo que vamos a hacer es desplegar un formulario muy corto al que el usuario responderá  y el formulario mandara la información a una pagina que llamaremos Audit page que tomara los elementos del formulario para manejarlos.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
   "http://www.w3.org/TR/html4/strict.dtd"
   >
<html lang="en">
<head>
   <title>Dog Years Calculator</title>
</head>
<body>
   <h1>Please provide the following information</h1>
   <form action="formProcess.php" method="get">
      <p>What is your dog's name?
      <input name="dogName" maxlength="30" />
      </p>

      <p>How old is your dog in human years?
      <input name="dogAge" maxlength="3" />
      </p>
```

```
        <input type="submit" value="Complete"/>
    </form>
</body>
</html>


<?php
    $name = $_REQUEST['dogName'];
    $age = $_REQUEST['dogAge'];

    echo("Your dog is named " . $name . " and your dog is " . $age . " human years old.");

    $dogAge = $age * 7;

    echo("<br/>Your dog is " . $dogAge . " in dog years.");
?>
```

---

PHP4

Condicionales

abecés en sus proyectos requerirán el uso de condicionales para que el programa efectúe diferentes acciones dependiendo de decisiones.

conditionals.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd"
    >
<html lang="en">
<head>
    <title>Conditional</title>
</head>
<body>
    <?php
        $age = 19;
        $citizen = false;        //Boolean
        /*
            Comparison Operators

            ==      Equivlency
            >       Greater Than
            <       Less Than
            >=      Greater than or Equal To
            <=      Less Than or Equal To
            !=      Not Equal To

            Compound Conditionals

            &&      and
            ||      Or
```

```
        */
        if($age >= 18  ll $citizen == true)
        {
            print("<strong>You are eligible to vote!</strong>");
        }
    ?>

</body>
</html>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd"
    >
<html lang="en">
<head>
    <title>Conditional</title>
</head>
<body>
    <form action="cond_audit.php" method="post">
        <p>What is your age?<br/>
        <input type="text" name="age"/></p>
        <p>Are you a citizen?<br/>
        <input type="radio" name="citizen" value="true" />Yes<br/>
        <input type="radio" name="citizen" value="false" />No<br/>
        <input type="submit" />
    </form>
</body>
</html>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd"
    >
<html lang="en">
<head>
    <title>Checking Your age</title>
</head>
<body>
    <?php
        $age = $_REQUEST['age'];
        $citizen = $_REQUEST['citizen'];

        if($age >= 18 && $citizen=="true")
        {
            print("<strong>You are eligible to vote</strong>");

        }
    ?>
</body>
</html>
```

---

Condicionales complejas

básicamente estas son condicionales que evalúan mas de una condición, o pueden excluir alguna situación las condicionales son ejecutadas.

complex_conditionals.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
   "http://www.w3.org/TR/html4/strict.dtd"
   >
<html lang="en">
<head>
   <title>Complex Conditionals</title>
</head>
<body>
   <?php
      $age = 37;
      $citizen = false;

      if($age >= 18 && $citizen)
      {
         print("<strong>You are eligible to vote</strong>");
      } else
      {
         print("<strong>You are not eligible to vote.</strong>");
      }


   ?>
</body>
</html>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
   "http://www.w3.org/TR/html4/strict.dtd"
   >
<html lang="en">
<head>
   <title>More Complex</title>
</head>
<body>
   <?php
      $gpa = .2;

      if($gpa == 4.0)
      {
         print("You have a perfect GPA");
      } elseif ($gpa > 3.0)
      {
         print("You have a very good GPA");
      } elseif($gpa > 2.0)
      {
         print("Your GPA is about average.");
      } elseif($gpa > 1.0)
      {
         print("Your GPA is poor.  Time to hit the books");
```

```
        } else
        {
            print("Why are you even in school?");
        }

    ?>
</body>
</html>
```

---

Switch, case and break

Son una gran forma de probar equivalencias en un código PHP. Son una de las mas viejas estructuras condicionales que existen en la mayoría de lenguajes de programación.

switch.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd"
    >
<html lang="en">
<head>
    <title>Switch...Case...Break</title>
</head>
<body>
    <?php
        $grade = 'b';

        switch($grade)
        {
            case 'A':
            case 'a':
                print("That's a great grade!");
                break;

            case 'B':
            case 'b':
                print("B is above average");
                break;

            case 'C':
            case 'c':
                print("C is an average grade");
                break;

            case 'D':
            case 'd':
                print("D is a low passing grade.");
                break;

            case 'F':
            case 'f':
                print("F is a failing grade.  You must retake the course");
                break;
```

```
            default:
                print("Letter grade not recognized.");
        }

    ?>
</body>
</html>
```

---

Ternary Operator
Es una declaración (statement) compacta de IF y Else, básicamente nos permite hacer un if y else sin escribir toda la estructura. Preferimos evitarlos por la complejidad del codigo

ternary.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd"
    >
<html lang="en">
<head>
    <title>Ternary</title>
</head>
<body>
    <?php
        $age = 5;

        $voteString = ($age >= 18) ? "can vote" : "can't vote";

        print ($voteString);

        $testScore = 40;

        $pass = ($testScore >= 60) ? true : false;

        print ("<br/>");
        print ($pass);


    ?>
</body>
</html>
```

PHP5

Loops

Los loops son un concepto muy importante de  las ciencias de la computación. Nos permiten repetir una sección del código cierto numero de veces.

loops.php

El día de hoy vamos a hablar de dos clases de loops: while loops / do while loops

Comenzaremos nuestro loop creando una variable llamada: counter variable, esta nos permite saber en donde nos encontramos en nuestro loop.

Cada vez que damos un vuelta en loop se llama iteración.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
   "http://www.w3.org/TR/html4/strict.dtd"
   >
<html lang="en">
<head>
   <title>Loops</title>
</head>
<body>
   <?php

      $i = 105;
      while($i < 101)
      {
         print($i);
         print("<br/>");
        $i +=10;
      }

      /*
      $airlines = array("American", "Southwest", "Delta", "US Airways", "United", "jetBlue",
"Frontier");



      $x = 0;
      while($x< count($airlines))
      {
         print($airlines[$x]);
         print("<br/>");
         $x++;
      }
      */

      $y=100;
      do
      {
         print($y);
         print("<br/>");
         $y++;
```

```
        } while ($y < 35);



    ?>
</body>
</html>
```

Si nuestra condicional es falsa, do while loop iterara por lo menos una vez en el caso de while loops no habra ninguna iteración.
Si queremos que la iteración se realice por lo menos una vez escogemos el while loop.
Es una pequeña diferencia pero nos permitirá generar distintos programas en PHP.

---

For Loops

vamos a ver otra clase de loops

forLoop.php

Las tre partes de un loop son:

Initialization
continuation condition
counter

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
   "http://www.w3.org/TR/html4/strict.dtd"
   >
<html lang="en">
<head>
   <title>For Loop</title>
</head>
<body>
   <?php
      // Initialization... Continuation Condition... Counter

      //for($i=0; $i< 100; $i++)
      for($i=0; $i< 200; $i=$i+5)
      {
         print($i . "<br/>");
      }


      for($x=400; $x> -1; $x = $x -21.67)
      {
         print($x . "<br/>");
      }
   ?>
</body>
</html>
```

---

Foreach Loops

forEach.php

Son algo diferente de los loops que hemos visto hasta el momento. Nos permitir iterar por un array incluso un array complejo.

Vamos a definir un array multidimensional. Este tipo de loop tiene una sintaxis que puede ser compleja si no hemos trabajado con estos loops antes.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
   "http://www.w3.org/TR/html4/strict.dtd"
   >
<html lang="en">
<head>
   <title>For Each Loops</title>
</head>
<body>
   <?php
      $gpas;
      $gpas["Fred"] = 3.62;
      $gpas["Mary"] = 2.75;
      $gpas["Jan"] = 4.00;
      $gpas["Jason"] = 2.10;
      $gpas["Mark"] = 2.87;

      $totalGpa = 0;

      foreach ($gpas as $key => $value)
      {
         print("Name: " . $key);
         print("<br/>GPA: " . $value);
         print("<br/>");
         $totalGpa += $value;
      }

      print("Total GPA: " . $totalGpa);
      $average = $totalGpa / count($gpas);
      print("<br/>Average GPA: " . $average);

   ?>
</body>
</html>
```

---

functions

Esta parte del curso es sobre todo sobre funciones, pero para usar las funciones efectivamente en PHP tenemos que entender primero la idea de “included code” código incluido.

includeMe.php

```
<?php
   print("This output has been included from the file includeMe.php");
?>
```

container.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
   "http://www.w3.org/TR/html4/strict.dtd"
   >
<html lang="en">
<head>
   <title>Display included data</title>
</head>
<body>
   <?php
      print("Printed from container.php");
      print("<br/>");
      include_once("includeMe.php");
      include("includeMe.php");
      include("includeMe.php");
      include("includeMe.php");

   ?>
</body>
</html>
```

podemos usar include o require.

---

Vamos a empezar generando una función básica en PHP.

El primer archivo que vamos a crear es para nuestras funciones.

functions.php

```
<?php

   function greetings()
   {
      print("Hello! How are you");
   }
```

```
    function repeatMyself()
    {
        for($i =0; $i<10; $i++)
        {
            print("Hi! This is repeat number " . ($i+1));
            print("<br/>");

        }
    }

?>
```

La funciones son partes de nuestro programa que tal vez queremos usar una y otra vez.

functionCalls.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd"
    >
<html lang="en">
<head>
    <title>Function Calls</title>
    <?php require_once("functions.php"); ?>
</head>
<body>
    <?php
        greetings();
        repeatMyself();
    ?>
</body>
</html>
```

---

Function Arguments

Aquí hablaremos sobre las funciones a las cueles les podemos pasar información, para que las funciones la procesen.

functionArgs.php

```
<?php

    function dogAgeCalc($age)
    {
        $age = $age *7;
        print("The age: " . $age);

    }

    function greetWho($name)
    {
        print("Greetings, " . $name);
    }
```

```
    function addThem($x, $y)
    {
        print("The result is " . ($x + $y));
    }
?>
```

function dogAgeCalc(Argumento)

dogYear.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd"
    >
<html lang="en">
<head>
    <title>Dog Age Calce</title>
    <?php include_once("functionArgs.php"); ?>
</head>
<body>
    <?php
        //$ageToCalc = 4;
        //dogAgeCalc($ageToCalc);

        //greetWho("Fred");
        //dogAgeCalc("Stepney. Connecticut");


        addThem(19,45);

    ?>
</body>
</html>
```

multiples argumentos

---

Vimos como las funciones pueden tomar argumentos, ademas de tomar argumentos estas nos pueden regresar valores.

return.php

Existen distintas funciones predeterminadas en PHP, estas las podemos encontrar en el manual.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd"
    >
<html lang="en">
<head>
    <?php
        //Functions Section
        function double($value)
        {
            $value = $value * $value;
            return ($value);
```

```
    }
  ?>
  <title><!-- Insert your title here --></title>
</head>
<body>
  <?php
    //Function Calls

    print("The value of 25.55 doubled is " . double(25.55));

    $result = double(37);
    print("<br/>The result is " . $result);

    $a = 45;
    $b = 57.25;
    $c = 18;

    $result2 = double($a+$b)/ $c * double($a/$b/$c) + double(double($a));
    print("<br/>Result 2 is: " . $result2);

    print("<br/>" . strtoupper("this is in lowercase"));

  ?>
</body>
</html>
```

---

FILE I/O (Input/output)

File i/o en PHP puede ser algo diferente si vienen de otros lenguajes de programación.

Los exploradores por cuestiones de seguridad no pueden acceder a archivos locales en nuestra computadora, ya que si esto fuera posible seria muy fácil escribir un program que causara daños en nuestra computadora. Por eso en PHP podemos escribir archivos al servidor.

fileio.php

File pointer, esto es una variable que hace referencia al archivo en si.

file name = $fileName = “name.txt”; //el nombre del archivo que vamos a usar
file pointer = $fp = fopen($fileName, ‘w’);  //file pointer tiene dos argumentos, el primero es el archivo que vamos a abrir y el segundo es llamado (resource contex) o mode, existen distintos modos.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
   "http://www.w3.org/TR/html4/strict.dtd"
   >
<html lang="en">
<head>
</head>
<body>
   <?php
     $fileName ="names.txt";
     $fp = fopen($fileName, 'w') or die ("Can't open the file");
     $name = array("Jorge","Monica", "Gaby", "Rafael", "Miguel", "Lucia", "Rosa");
     for($i=0; $i< count($name); $i++)
     {
          fwrite($fp, $name[$i] . "\n");
     }
     fwrite($fp , "####################");
     fclose($fp);
     print("salvados");
   ?>
</body>
</html>
```

---

Ya que aprendimos a generar archivos en nuestro servidor aprenderemos a leer estos archivos.

Reading Files on the server

primero vamos a generar un file pointer.

fileio02.php

```
<html>
<head>
</head>
<body>
   <?php
      $file = "poema.txt";
      $fp = fopen($file, 'r');
      $string = fread($fp, 20);
      fclose($fp);

      echo($string);

   ?>
</body>
```

El 20 representa los caracteres que podemos leer, esto si sabemos cuantos caracteres queremos leer ahora le daremos la vuelta. Probablemente queremos leer todo el poema pero no tenemos forma de saber cuantos caracteres tenemos, lo que vamos a hacer es ir por todo el poema. La forma mas fácil de hacer esto es con filesize comando

```
$string = fread($fp, filesize($file));
```

Hay una mejor forma de realizar esta tarea que tiene un mejor desempeño en los servidores.

```
 /*
      $file = "poema.txt";
      $fp = fopen($file, 'r');
      $string = fread($fp, filesize($file));
      fclose($fp);

      echo($string);

   */
      $file = "poema.txt";
      $string = file_get_contents($file, true);
      echo ($string);
```

este método esta recomendado por el manual de php como una manera mas rápida de obtener los datos.

A veces necesitamos linea por linea., aquí obtenemos la primera linea, para obtener todas las lineas tendremos que generar un loop.

```
      $file = "poema.txt";
      $fp = fopen($file, 'r');
      /*
      $string = fread($fp, filesize($file));
      fclose($fp);

      echo($string);


      $file = "poema.txt";
      $string = file_get_contents($file, true);
      */
      $string =fgets($fp);
      echo ($string);
```

Para loo-pear haríamos esto:

```
<html>
<head>
</head>
<body>
   <?php


      $file = "poema.txt";
      $fp = fopen($file, 'r');
      /*
      $string = fread($fp, filesize($file));
      fclose($fp);
```

```
        echo($string);


        $file = "poema.txt";
        $string = file_get_contents($file, true);

        $string =fgets($fp);
        echo ($string);
        */
        $string = "";

        while (!feof($fp)) {
          $string .= fgets($fp, 256);
          $string .= "<br/>";

        }
        print($string);

    ?>
</body>
```

---

Append & delete

airlines.php

```
<html>
<head>
</head>
<body>
    <?php
        $file = "aerolineas.txt";
        $fp = fopen($file, 'a') or die ("No se puede abrir el archivo");

        $airline1 = "Aerolinea1";
        $airline2 = "Aerolinea2";

        fwrite($fp, $airline1. "\n");
        fwrite($fp, $airline2. "\n");
    ?>
</body>
```

Que estamos escribiendo en nuestro documento. append - anexar

Cuando vamos a borrar con estos métodos tenemos que tener mucho cuidado para no borrar archivos importantes.

Tengo un archivo muestra que se llama example.txt

cuando borramos files en php lo que haces es desligarlos para que el servidor se olvide de ellos.

```
<html>
<head>
</head>
<body>
   <?php
      $file = "aerolineas.txt";
      $fp = fopen($file, 'a') or die ("No se puede abrir el archivo");

      $airline1 = "Aerolinea1";
      $airline2 = "Aerolinea2";

      fwrite($fp, $airline1. "\n");
      fwrite($fp, $airline2. "\n");

      unlink("example.txt");
      print("Aerolineas file append y ejemplo de file borrado");
   ?>
</body>
```

---

CSV FILES

Comon separeted value form

Algo util de PHP es el manejo de archivos csv.

parts.csv

```
2342, Producto1, $15.40 ,25
4221, Producto2, $9.97 ,14
5554, Producto3, $11.37 ,41
4342, Producto4, $41.11 ,2
```

readCsv.php

```
<html>
<head></head>
<body>
        <?php
        $file = "parts.csv";
        $fp = fopen($file, "r");

        $output ="";
        while(!feof($fp))
        {
                $inventory = fgetcsv($fp, 1024);
                $line = "";
                $line .= "<tr>";
                $line .= "<td>" . $inventory[0] . "</td>";
                $line .= "<td>" . $inventory[1] . "</td>";
                $line .= "<td>" . $inventory[2] . "</td>";
                $line .= "<td>" . $inventory[3] . "</td>";
```

```
                $line .= "</tr>";
                $output .= $line;

        }

print("<table border='1'>");
print($output);
print("</table>");

?>
</body>
</html>
```

---

SEnding Email with PHP

sending text email

mailSend.php

Mandar mails desde el servidor es muy fácil con PHP.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
   "http://www.w3.org/TR/html4/strict.dtd"
   >
<html lang="en">
<head>
   <title>Send Some Mail</title>
</head>
<body>
   <?php
      $to = "mark@learntoprogram.tv";
      $subject = "Feedback for you";
      $message = "This is a test email from the PHP class";
      $message .= "from LearnToProgram.tv";
      $from = "ernie@learntoprogram.tv";
      $headers = "From:" . $from;
      mail($to, $subject, $message, $headers);
      echo("Mail has been sent.");


   ?>
</body>
</html>
```

---

Sending HTML Email

En el ejercicio pasado mandamos un mail simple por PHP, ahora daremos un paso mas adelante y mandaremos un mail HTML.

htmlEmailSend.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
   "http://www.w3.org/TR/html4/strict.dtd"
   >
<html lang="en">
<head>
   <title>HTML Email</title>
</head>
<body>
   <?php
     $to = "mark@learntoprogram.tv";
     $from = "ernie@learntoprogram.tv";
     $subject = "HTML Test Email";

     $headers = "From: " . $from . "\r\n";
     $headers = "Repy-To " . $from . "\r\n";
     $headers = "MIME-Version : 1.0\r\n";
     $headers = "Content-Type: text/html; charset=ISO-8859-1\r\n";

     $message = "<html><body>";
     $message = "<h1>Information From Me</h1>";
     $message = "<p><strong>It's time to Act!</strong></p>";

     mail($to, $subject, $message, $headers);
   ?>
</body>
</html>
```

---

Setting Up the Database

```
cd /Applications/MAMP/Library/bin
```

```
Vamos a entrar al mamp al php admin y crear una nueva base de datos:

employees y voy a dar en create.

Vamos a crear 2 tablas en employees

1 employee # 6

El primer field es empId , type int, 11, index PRIMARY, A_I x (auto
increment) , (este es el campo que tiene que garantizar sea unico
durante la base de datos)

lastName, VARCHAR, 40

firstName, VARCHAR, 20

department, INT, 2

position, VARCHAR, 20
```

salary, INT, 10

despues vamos a crear una segunda tabla

departments 2

deptId, INT, 11, Index Primary, a_i,

departmentName, 40,

Ahora vamos a meter algunos datos,

departmentName, value Accounting

ignore
departmentName, value Sales

insert another new row:

departmentName, value Information Technology

ignore
departmentName, value Management

_____

employees

Insert
lastName,  value Martinez
firstName, value Alfredo
3
Programer
75000

Smith
Fred
1
Assistant
34000

db_connect.php

```
<?php
    $db_name = "employees";
    $un = "root";
    $pw = "root";
    $host = "localhost";

    mysql_connect($host, $un, $pw) or die (mysql_error());
    //echo("Connected to mySQL Database");

    mysql_select_db($db_name) or die(mysql_error());
    //echo ("Connected to employees");


?>
```

index.php

```
<?php

require_once("db_connect.php");

$sql = "SELECT * FROM employee";
$result = mysql_query($sql) or die(mysql_error());



//$row = mysql_fetch_array($result) or die(mysql_error());
echo("<table>");
while ($row = mysql_fetch_array($result)) {
    # code...

//echo $row['empId'] . ' ' . $row['lastName'] . ' ' .
$row['firstName'] . ' ' . $row['department'] . ' ' . $row['position'] .
' ' . $row['salary'];
echo("<tr>");

echo "<td>" . $row['empId'] . "</td>"
    . "<td>" . $row['lastName'] . "</td>"
     . "<td>" . $row['firstName'] . "</td>"
      .  "<td>" . $row['department'] . "</td>"
       . "<td>" . $row['position'] . "</td>"
        . "<td>" . $row['salary'] . "</td>";
echo("</tr>");


}
echo("<table>");
?>
```

Vamos insertar datos a nuestra base de datos, para esto necesitamos un formularios que el usuario rellenara.

insert.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd"
    >
<html lang="en">
<head>
    <title>Insert Employee</title>
</head>
<body>
    <form action="insert_process.php" method="post">
        <p>Last Name<br/>
        <input type="text" name="last"/></p>

        <p>First Name<br/>
        <input type="text" name="first"/></p>

        <p>Department<br/>
        <select name="department">
            <option value="1">Accounting</option>
            <option value="2">Sales</option>
            <option value="3">Information Technology</option>
            <option value="4">Management</option>
        </select>

        <p>Position<br/>
        <input type="text" name="position"/></p>

        <p>Salary<br/>
        <input type="text" name="salary"/></p>

        <input type="submit" value="Save Information" />
    </form>
</body>
</html>
```

insert_process.php

```
<?php

    require_once("db_connect.php");

    $last = $_REQUEST['last'];
    $first= $_REQUEST['first'];
    $department= $_REQUEST['department'];
    $position= $_REQUEST['position'];
    $salary= $_REQUEST['salary'];

    //echo($last . ' ' . $first . ' ' . $department . ' ' . $position .
' ' . $salary);

    $sql= "INSERT INTO employee VALUES (''," .
                                        "'" . $last . "'," .
                                        "'" . $first . "'," .
                                        "'" . $department . "'," .
                                        "'" . $position . "'," .
                                        "'" . $salary . "')";

    mysql_query($sql);
    mysql_close($conn);

    echo($last . " successfully added to the database.");
    echo("<br/>Go back to <a href='index.php'>main page.</a>")

    //echo($sql);


?>
```