**Lenguaje PHP**

**El código no se del lado del servidor porque php lo renderiza**

**Se puede meter todo el código html que se requiera en la pagina la etiqueta para declarar que vamos a usar php es**

**<?php y su cierre ?>**

**Declaramos variables**

**$variable(cualquier nombre no se admiten números de inicio)**

**Php es un lenguaje que se lee de arriba hacia abajo asi que una variable puede r cabiando su valor o funcionalidad uso**

**Tipode de datos admitidos por php**

**$variables = “contendio”;**

**$numero = 30; (números)**

**$nobrevariable = true; (trues o false admite boleanos)**

**Se concatena con . y un espacio seguido de las variables que queremos**

**Función eco para imprimir en pantalla**

**Tenemos las funciones matemáticas suma resta multiplicación y división**

**Comentarios en php // por línea y /\* \*/ para un comentario mas largos igual que en javascritp5**

**Existen también el sumar unoy restar uno con el ++ o –**

**Introducción libro resumen**

**Creado en 1995 por Rasmur Lerdorf ( personal home page tools)**

**Software libre**

**Dos tipos de paginas**

**1 páginas estáticas: una vez que sale a imprenta no es posible realizar cambios en su contenido**

**2 páginas dinámicas: puede mostrar información diferente que cada vez que**

**Es invocada.**

**Lenguaje interpretado por ello no necesita ser compilado**

1. **Aspectos básicos php** 
   1. **El escript se ejecuta en el servidor siempre debe empezar etiqueta de inicion <?php> etiqueta de cierre <?>**
   2. **Siempre que se acaba una sentencia o línea de código se pondrá “;”**
   3. **‘echo’ para mostrar texto por pantalla**
   4. **Las variables de declara con $ al inicio**
   5. **Texto plano: se puede escribir texto que no vaya a cambiar (plano) mezclando con etiqueta HTML**
   6. **Texto con etquetas HTML enriquece el código y será mas legible**
   7. **Echo para imprimir texto se usara de forma general ya que es un constructor del lenguaje**
   8. **Print para imprimir texto es una función asi que se ejecuta y devuelve resultado**

**2 Funciones**

**Sirven para hacer subrutinas predefinidas. Pueden recibir parámetros que se pueden trabajar como variables asi cada llamado puede dar un valor diferente.**

**Las funciones pueden devolver valor o no y se puede usar dicho valor en otras líneas de código**

**¿Llamado de una función <?php> llamandoFuncion($parametros); <?>**

**Para crea una función se debe utilizar una palabra llamada “funtion” seguido del nombre de la función y paréntesis con o sin argumentos parámetros**

**A continuación dentro de {} se pondrá lo que va a realizar la función**

**PARAMETROS POR DEFECTO**

**Se puede declarar parámetros que tengan una valor por defecto esto es optimo con los valores bolean true o false**

**Ejemplo: <?php funtion nombredelafuncion( $aargumento1= 1, $argumentos2= “hola”){**

**Echo ‘$argumento1. $argumento2;**

**}Nombredelafuncion(“otro valor”);**

**?>**

**PARAMETROS POR REFERECIA**

**Si el objetivos es que las variables se vean afectadas por fuera de l función tras modificarlas habrá que enviar los parámetros por referencia**

**Ejemplo: <?php**

**Funtion sumantorio(&$variable){**

**$variable ++;**

**$num = 1;**

**Sumatorio($num);**

**Echo $num; ?>**

**FUNCIONES VARIABLES**

**PHP permite llamar funciones tomando como nombre de la función valor de la variable**

**TIPO EN LOS PARAMETROS**

**Los parametros podemos obligarlos ejemplo con un bolenao dándole valor al paramtro y puediendolo cambiar al llamar la función**

**RETORNO DE VALORES**

**Las funciones como funcionalidad nos devuelven un resultado que lo captaremos con la palabra return sino tiene resultado nos devovlera null**

**VARIABLES**

**Contenedores para guardar números strings boleanos etc tienen caso sensitivo se tiene siempre que escribir igual**

**Las variables se asignan por valor esto quire decir que si se usan dos vaiables y se asigan a en b si se modifica a b nosufre cambio alguno**

**Se puede apuntar por referencia y para ello se debe usar el &**

**Se pueden unir una o varia variables con el “.”**

**Las variables se asignan por el propio lenguaje en el momento de ejecución se uede obligar a una variable que tenga un tipo concreto usando settype()**

**TIPOS DE DATOS**

**1 String**

**2 integer**

**3 float**

**4 bool**

**5 array**

**6 object**

**7 Null**

**8 resource**

* **String: cadena de caracteres cada cracter es guardado en bytes tiene como longitud máxima 256 caracteres y un peso de 2gigas** 
  + **Entre comillas simples ‘’ forma eficiente de escribir texto**
  + **Entre comillas dobles “” el texto se revisara ya que tenemos mas funciones como saltos de línea, retonrno de carro tabulación además de también la posibilidad de incrustar varibles que podrán ser leidas**
  + **Heredoc para escribir texto no aceptara varibles**
  + **Nowdoc para escribir texto e identificarlo aceptara vaiables con sus valores**
  + **Integer: números enteros o naturales**
  + **Float: numero decimal**
  + **Bool : expresiones boleanas verdarero o falso**
  + **Array: los arreglos son mapas de valores ordenados que pueden ser listas , pilas, colas etc**
  + **Object: para usar un objeto los primero es crearlo con el neww object**
  + **Null las variables pueden se de tipo null si al construirse no se les asigna un valor**
  + **Resourse: permite saber si existen recursos y si se pueen usar**

**ALCANSE DE LAS VARIABLES**

**Si se declara una variable global no se podrá usar dentro de la función nos dira que dich variable no esta definida en la función**

**Las variables se pueden pasar por referencia se pueden usar variables globales dentro de una función con [“$nombredelavariable”]**

**VARIABLES ESTATICAS**

**Se pueden declarar variables estaticas para un uso propio con la palabra static antes del nombre de la variable**

**MANEJO DE VARIABLES**

**Para comprobar que tipo de datos es el que estamos trabajado con la variable tenemos**

**varias funciones predefinidas como**

* **Empty devuelve cero si el valor introducido es o o o.o $bol =empty($variable)**
* **Boolval: devuelve un boleano $bool = bollval($variable)**
* **Intval: deuvle un int**
* **Floatval : devuelve los decimales**
* **In\_null: comprueba si es nulo o no**
* **Is\_bool: comprueba si es buelano**
* **Is\_array: comprueba si es de tipo array**

**ARRAYS**

**Para trabajar con variables mas complejas usamos los arrays seria contenedores con asociados calve valor**

**Las claves pueden ser números enteros o cadenas de caracteres si se pone un decimal lo convierte a entero los valores boleanos se covierten en 1 para verdadero y 0 para false**

**Los valores pueden ser números o cadenas de caracteres**

**ARRAY INDEXADOS**

**Sirven para guargar varios valores de forma simple podemos acceder a ello mediante la clave también debemos tener en cuenta que los array empiezan en la posición 0. Se pueden crear arrays vacios para posteriormente manipularlos pudiendo agregar nuevos valores si se agrega un valor sin el índice este se posicionara después del ultimo dato en cambio si se agrega con un índice que aya existe este se sobreescribira**

**Se puede acceder a alas variables como arrays usando los corchetes podríamos modificar una letra pasando dicha letra**

**$nombre = “Carlos”;**

**$nombre[4] = “a”; 🡪queraria como carlas**

**ARRAY ASOCIATIVOS**

**Sirve para referenciar datos concretos como el saliro de los empleados y mediante su clave podemos hacer cambios**

**OPERACIONES CON ARRAYS**

**Para comparar arrays debemos comprobar su clave, valor or den y tipos**

* **$a +$b = unir a con b**
* **$a === b = comprueba si a y b contienen las mismas claves y valores**
* **A != b = comprueba que a es distito que b**
* **A <>b = comprueba que a y b no tienen las mismas claves ni valores**
* **A ===b = comprueba que a y b tienen las mismas claves los mismos tipos y orden y valores**
* **A ¡= b comprueba que a y b no tienen las mismas claves y valores distintos orden o tipo**

**MANEJO DE ARRAYS**

**Funciones para el uso de arrays**

* **Count sirve para saber cuantos elementos contiene el array formula $c = count($array)**
* **In\_array permite saber si existe el valor introducido formula $encontrado = in\_array (“valor”, $array) nos devolvería true o false**
* **Key\_exist devuelve verdadero si ecuentra la clave o índice enviado formula $encontrado = key\_exist(“clave”, $array)**
* **Array\_ values devuelve una array con todos los valores del array introducidos formula $valores = array\_values ($array)**
* **Array\_keys devuelve un array con todas las claves del Aray formula $claves = array\_keys($array)**
* **Array\_combine devuelv un array usando un array para las clves y otro para los valores**
* **Aa\_merge combina dos o mas arrays y unifica los array introducidos y los devuelve como un array los valores se anidad uno detrás de otr**
* **Array \_unique devuelve un array son valores duplicados el primer parametr de la fucion será el array a eliminar el segundo parámetro es opcional no indic que compracion es tenemos varias que son** 
  + **SORT\_REGULAR compara ítems sin modificar los tipos**
  + **SORT\_ NUMERIC compara numéricamente**
  + **SORT\_STRING compra con cadenas de caracteres**
  + **SORT\_LOCALE\_STRING compra con cadena de caracteres usando ña codiguracion regional actual**
* **Sort devuelve un array ordenado se le pueden pasar varios flagts seria** 
  + **SORT\_NATURAL para ordenar de forma natural**
  + **SORT\_FLG\_CASE para ordear caracteres de forma sensible a las mayúsculas y minúsculas**
* **Suffle baraja un array devuelve verddero si lo pudo hacer**
* **Rango crea un array desde el valor inicial hasta el final se le puede dar un incremento secuancial**
* **Arry\_shift elimina el primer elemento del array y lo devuelve como valor si el array esta vacio devuelve null**
* **Array pop elimina el ultimo elemnto del array y lo deuvle como valor**
* **Array\_replace dado un primer array ira cambiando todas las claves valor**
* **Array\_slice puede recortar un array sin modificarlo y devuelve el array modificado**

**CONSTANTES**

**Valores únicos que no se van a poder modificar formula**

**Define(nombre de la constante, valor, sensitiva) en la nuevas versione se usa el CONST para crear las constantes en vez del define**

**Recomendable escribir las constantes en mayúsculas y no dejar acivo el sensitivo**

**Se puede acceder a ella de manera global**

**Define() saber si una constante existe**

**Se pueden crear constante de tipos como: int, float, bool o string**

**CONSTANTES PREDEFINIDAS**

* **PHP\_VERSION nos da la versión php**
* **E\_ERROR Constante de error**
* **NULL VARIABLE NULA**

**ETC PGINA 42 PHP BASICO**

**CONSTANTES MAGICAS**

**Existen constantes cuyo valo se modifica en función de donde se est ejecutando ejemplo**

* **\_LINE\_ El numero de la línea actual del fichero**
* **\_FILE\_ ruta completa y nombre de archivo**
* **\_DIR\_directorio del fichero**
* **\_FUNTION\_ nombre de la función**
* **\_CLASS\_C NOMBRE DE LA CLASE**
* **PG 43 PHP BASICO**

**OPERADORES**

**Estos pueden recibir uno o mas valores para trabajar con ellos y nos devuelven un resultado estos pueden ser almacenados en variables o escitos en el código**

**Operadores de asigancion los mas usados asignan un valor a una variable**

**Operadores aritméticos suma resta multiplicación división exponencial etc**

**Pagina 45 php básico todos los operadores**

**Asignacion con operadores trabajar con un variable a que se pueda calcular aritméticamente y el valor se guardar en a a diferencia de otros lenguajes que usaría ua tercera variable para el resultado php pemite guardarlo en la primera variable**

**COMPRACION**

**Para comparar nos devuelve true en caso de que se cumpla la compracion pg 45**