IIC2413

AYUDANTÍA 2

¿QUÉ VEREMOS?

P

H

P

¿QUÉ ES PHP?

- PHP (Hypertext Preprocessor) es un lenguaje de programación del lado del servidor diseñado para el desarrollo web.
- Es utilizado para crear páginas dinámicas e interactivas.
- Se ejecuta en el servidor y genera HTML para ser enviado al navegador.
- Facebook, Yahoo, Wikipedia, Spotify, etc.

PRIMER ARCHIVO

- Para cada archivo de php se necesita incluir “<?php" al inicio y “?>" al final
- Para cada línea de código se tiene que agregar “;"
- Para mostrar texto por consola se utiliza el método “echo"
- Al imprimir por consola no se hacen saltos de linea
- Para pedir un input por consola se utiliza el comando “readline()"

¿CÓMO EJECUTARLO?

- Tener instalado php en el dispositivo.
- Para ejecutar archivos se necesita ejecutar se ingrese por consola “php" junto al nombre del archivo, ejemplo “php archivo.php"

¿CÓMO EJECUTARLO?

```php
<?php
echo "hola mundo\n";
echo "Escribe tu nombre: ";
echo readline();
?>
```

```
php archivo.php
hola mundo
Escribe tu nombre: mi_nombre
mi_nombre%
```

VARIABLES Y TIPO DE DATOS

Declaración de variables ($variable).

Tipos de datos:

- String ($nombre = "Juan";)
- Integer ($edad = 25;)
- Float ($precio = 10.5;)
- Boolean ($es_valido = true;)
- Array ($frutas = ["Manzana", "Banana", "Naranja"];)

Uso de var_dump() para inspeccionar variables.

VARIABLES Y TIPO DE DATOS

Archivo.php

```php
<?php
$nombre = "Juan";
$edad = 25;
$precio = 10.5;
$es_valido = true;
$frutas = ["Manzana", "Banana", "Naranja"];

echo $nombre, "\n";
echo $edad, "\n";
echo $precio, "\n";
echo $es_valido, "\n";
print_r($frutas);
?>
```

Terminal

```
php ayudantia_2.php
Juan
25
10.5
1
Array
(
    [0] => Manzana
    [1] => Banana
    [2] => Naranja
)
```

OPERADORES LÓGICOS

Tipos de Operaciones Lógicas:

- Verdadero: “true"
- Falso: “false"
- Igual: “x == y"
- Distinto: “x != y"
- Menor que / Mayor que: “ x < y"
- Menor o igual que / Mayor o igual que: “x >= y"
- Negación: “!x"
- Conjunción(Y): “&&"
- Disjunción(O): “||"

VARIABLES Y TIPO DE DATOS

php

```php
<?php
$nombre = "Juan";
$edad = 25;
$precio = 10.5;
$es_valido = true;
$frutas = ["Manzana", "Banana", "Naranja"];

echo $nombre, "\n";
echo $edad, "\n";
echo $precio, "\n";
echo $es_valido, "\n";
print_r($frutas);
?>
```

python

```python
nombre = "Juan"
edad = 25
precio = 10.5
es_valido = True
frutas = ["Manzana", "Banana", "Naranja"]

print("Esto es un string: " + nombre)
print(edad)
print(precio)
print(es_valido)
print(frutas)
```

OPERADORES LÓGICOS

php

```php
<?php
$verdadero = true;
$falso = false;

$cond1 = 1 == 2;
$cond2 = "b" != "a";
$cond3 = 10 < 11;
$cond4 = 10 >= 11;
$cond5 = !true;
$cond6 = $verdadero && $falso;
$cond7 = $verdadero || $falso;
?>
```

python

```python
verdadero = True
falso = False

cond1 = 1 == 2
cond2 = "b" != "a"
cond3 = 10 < 11
cond4 = 10 >= 11
cond5 = not True
cond6 = verdadero and falso
cond7 = verdadero or falso
```

ITERABLES

Tipos de iterables:

- While: “while (condición) {bucle};"
- For: “for (variable; condición; aumento_variable) {iteración}"
- Do-While: “do {bucle} while (condición);"

```php
<?php
$i = 0;
while ($i < 10) {
    echo $i;
    $i++;
}
?>
```

While

```php
<?php
for ($j = 0; $j < 10; $j++) {
    echo $j;
}
?>
```

For

```php
<?php
$i = 10;
do {
    echo $i;
    $i += 1;
} while ($i < 10);
?>
```

Do-While

ITERABLES

php

```php
<?php
$i = 0;
while ($i < 10) {
    echo $i;
    $i++;
}
?>
```

```php
<?php
for ($j = 0; $j < 10; $j++) {
    echo $j;
}
?>
```

python

```python
i = 0
while i < 10:
    print(i)
    i += 1
```

```python
for j in range(10):
    print(j)
```

LISTAS

Se pueden instanciar listas de 2 formas:

- $lista_con_array = array('a', 'b', 'c', 'd', 'e');
- $lista_con_corchetes = ['a', 'b', 'c', 'd', 'e'];

Obtener elemento: $lista[índice]

Modificar elemento lista: $lista[índice] = $elemento;

Nuevo elemento: $lista[] = $new_elem // arraypush($lista, $new_elem)

Largo: $largo = count($lista)

Pop: $ultimo_elemento = array_pop($lista)

Eliminar elemento: unset($lista[índice])

Recorrer: foreach($lista as $elemento){iteración}

Aclaración: unset() elimina el elemento pero no reordena los índices, ej, si se elimina el elemento con índice 2, los índices de la lista serán 0, 1, 3, ... , etc.

LISTAS

php

```php
<?php
//Definir una lista
$lista_con_array = array('a', 'b', 'c', 'd', 'e');
$lista_con_corchetes = ['a', 'b', 'c', 'd', 'e'];

//Obtener elemento
$elemento = $lista_con_array[0];

//Modificar elemento
$lista_con_array[0] = 'z';

//Nuevo elemento
$lista_con_array[] = 'final';
array_push($lista_con_array, "final2");
```

python

```python
# Definir una lista
lista_con_array = ['a', 'b', 'c', 'd', 'e']

# Obtener elemento
elemento = lista_con_array[0]

# Modificar elemento
lista_con_array[0] = 'z'

# Nuevo elemento
lista_con_array.append('final')
lista_con_array.append('final2')
```

LISTAS

php

```php
//Largo lista
$largo = count($lista_con_array);

//Recorrer lista
foreach ($lista_con_array as $dato){
    echo $dato, "\n";}

// Método pop
$ultimo_elemento = array_pop($lista_con_array);
echo "Pop: ", $ultimo_elemento, "\n";

// Eliminar elemento en específico
unset($lista_con_array[2]);
```

python

```python
# Largo lista
largo = len(lista_con_array)

# Recorrer lista
for dato in lista_con_array:
    print(dato)

# Método pop
ultimo_elemento = lista_con_array.pop()
print("Pop:", ultimo_elemento)

# Eliminar elemento en específico
lista_con_array.pop(2)
```

LISTAS

php

```php
// Listas de listas (matrices)
$matriz1 = array(
    array(1, 2, 3),
    array(4, 5, 6),
    array(7, 8, 9)
);
$matriz2 = [
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
];
```

python

```python
# Listas de listas (matrices)
matriz1 = [
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
]
matriz2 = [
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
]
```

STRINGS

Definir string: $string = 'Hola Mundo';
Obtener carácter según índice: $string[índice]
Concatenar strings: $string . ' :D', "\n" (puede ser con “.” o “,”)
Largo string: $largo = strlen($string)
Mayúsculas: strtoupper($string)
Minúsculas: strtolower($string)
Reemplazar: str_replace($fragmento_a_reemplazar, $texto_reemplazante, $string);
Método split (convertir a lista): explode(",", $string);

STRINGS

```php
<?php
$string = 'Hola Mundo';
echo $string[6], "\n";          // u (obtiene el caracter en la posición 6)
echo $string . ' :D', "\n";     // Hola Mundo :D (concatenación)
echo strlen($string), "\n";     // 10 (calcula la longitud del string)
echo strpos($string, 'Mundo'), "\n"; // 5 (busca el índice de la palabra 'Mundo')
echo strtoupper($string), "\n"; // HOLA MUNDO (convierte a mayúsculas)
echo strtolower($string), "\n"; // hola mundo (convierte a minúsculas)
echo str_replace('Mundo', 'Planeta', $string), "\n"; // Hola Planeta (reemplaza 'Mundo' por 'Planeta')
$string = "Hola,Mundo";
$spliteado = explode(",", $string); // ['Hola', 'Mundo']
?>
```

STRINGS

php

```php
<?php
$string = 'Hola Mundo';
echo $string[6], "\n";
echo $string . ' :D', "\n";
echo strlen($string), "\n";
echo strpos($string, 'Mundo'), "\n";
echo strtoupper($string), "\n";
echo strtolower($string), "\n";
echo str_replace('Mundo', 'Planeta', $string), "\n";
$string = "Hola,Mundo";
$spliteado = explode(",", $string);
?>
```

python

```python
string = 'Hola Mundo'
print(string[6])
print(string + ' :D')
print(len(string))
print(string.find('Mundo'))
print(string.upper())
print(string.lower())
print(string.replace('Mundo', 'Planeta'))
string = "Hola,Mundo"
spliteado = string.split(",")
```

ARCHIVOS

Abrir archivos

- Modo lectura: $archivo = fopen("archivo.txt", "r");
- Modo escritura (de cero): $archivo = fopen("archivo.txt", "w");
- Modo agregar líneas: $archivo = fopen("archivo.txt", "a");

Cerrar archivos: fclose($archivo);

Escribir en archivos: fwrite($archivo, $string);

Obtener líneas: fgets($archivo);

Verificar final: feof($archivo);

Aclaración: “fgets()" genera las líneas del archivo, por lo que al llamar a la función va a entregar la primera línea, luego la segunda, y así sucesivamente. La función “feof()" entrega un booleano dependiendo si se llegó al final del archivo (no puede generar más líneas con “fgets")

ARCHIVOS

Leer archivos

php

```php
<?php
$archivo1 = fopen("archivo1.txt", "r"); //Abrir archivo en modo lectura

while (!feof($archivo1)) {      //feof es final del archivo
    $linea = fgets($archivo1);
    echo $linea;
}
fclose($archivo1); //Cerrar archivo
```

python

```python
archivo1 = open("archivo1.txt", "r")  # Abrir archivo en modo lectura

for linea in archivo1:
    print(linea)

archivo1.close()  # Cerrar archivo
```

ARCHIVOS

Escribir de cero

php

```php
$archivo2 = fopen("archivo2.txt", "w"); //Abrir archivo en modo escritura
fwrite($archivo2, 'sobreescribí!');
fclose($archivo2); //Cerrar archivo
```

python

```python
archivo2 = open("archivo2.txt", "w")  # Abrir archivo en modo escritura
archivo2.write('sobreescribí!')
archivo2.close()  # Cerrar archivo
```

Agregar contenido

php

```php
$archivo3 = fopen("archivo3.txt", "a"); //Abrir archivo en modo escritura al final del archivo
fwrite($archivo3, "\nme agregué!");
fclose($archivo3); //Cerrar archivo
```

python

```python
archivo3 = open("archivo3.txt", "a")  # Abrir archivo en modo escritura al final del archivo
archivo3.write("\nme agregué!")
archivo3.close()  # Cerrar archivo
```

FUNCIONES

Crear función:

```
function nombre_funcion($variable) {
    ...
    return $resultado;
}
```

Usar una función: nombre_funcion($variable);

Para funciones recursivas se llama a la función dentro de la misma

FUNCIONES

php

```php
<?php
function contar_caracteres($texto) {
    // Código de la función
    $largo = strlen($texto);
    return $largo;
}
$texto = "Hola mundo!";
$largo = contar_caracteres($texto);
echo $largo;
?>
```

python

```python
def contar_caracteres(texto):
    # Código de la función
    largo = len(texto)
    return largo

texto = "Hola mundo!"
largo = contar_caracteres(texto)
print(largo)
```

FUNCIONES

php

```php
<?php
function fibonacciRecursivo($n) {
    if ($n <= 1) {
        return $n;
    }
    return fibonacciRecursivo($n - 1) + fibonacciRecursivo($n - 2);
}
$n = 10;
for ($i = 0; $i < $n; $i++) {
    echo fibonacciRecursivo($i) . " ";
}
?>
```

python

```python
def fibonacci_recursivo(n):
    if n <= 1:
        return n
    return fibonacci_recursivo(n - 1) + fibonacci_recursivo(n - 2)

n = 10
for i in range(n):
    print(fibonacci_recursivo(i), end=" ")
```

ACTIVIDAD: A POR PHP!

Para practicar tus habilidades de programación de php van a tener que crear las siguiente funciones:

1. Función que lea un archivo: Una función que reciba un nombre de archivo y lea el contenido de este y lo devuelva en un array. ayudantia2-1.csv

2. Función que elimine duplicados: Una función que reciba un array y que retorne un array con elementos no duplicados.
[[1,"Laptop",1200], [2,"Mouse",20], [3,"Teclado",50], [4,"Laptop",1200], [5,"Monitor",300]]

3. Función que escriba un archivo: Una función que reciba un array y nombre de archivo y cree un archivo con ese contenido.
["Hola mundo", "PHP es genial", "Hello world", "Aprender a programar"]

4. Función que muestre una lista de listas: Una función que recibe un array y muestre en consola un array de forma cómoda.
["Hola mundo", "PHP es genial", "Hola mundo", "Aprender a programar"]

¡MUCHAS GRACIAS!