

Ejercicio 4:

- **¿Cómo declaramos una variable?**
 - R: Hay dos formas, una de ellas es usando la palabra reservada **var** seguido de su identificador, por ejemplo: **var \$nombre**;
La otra forma puede omitir la palabra reservada **var** y definir el nombre de la variable, seguido del valor que queramos que tenga, por ejemplo: **\$Nombre = "Kriz"**;
- **¿Importan las mayúsculas y minúsculas en la declaración de las variables?**
 - R: Si, para PHP los nombres de las variables son sensibles a mayúsculas o minúsculas, por lo que su dos nombres de variables se diferencian por esto, se tomarán como dos variables distintas, por ejemplo: **\$Numero** es diferente de **\$numero**.

Ejercicio 5:

- **¿Qué es la concatenación?**
 - R: La concatenación consiste en unir una o más caracteres o cadenas de caracteres con el fin de formar una cadena mayor, usando ya sea elementos como variables que pueden cambiar respecto de algún procedimiento o método previo que le brinde un valor en específico.
- **¿Cómo hacemos la concatenación en PHP?**
 - R: La concatenación en PHP se realiza uniendo una cadena con una variable o con otra cadena mediante un punto. Ejemplo: **\$Name = "Kriz"**;
"Hola ".\$Name." ¿Cómo estás?";
El ejemplo anterior daría como resultado:
Hola Kriz ¿Cómo estás?.

Ejercicio 6:

- **¿Qué es lo que hace el script del ejercicio 6?**
 - R: El script realiza, mediante la etiqueta **
** un salto de línea en el texto que se va a mostrar en pantalla. Es una forma de darle formato al texto que queremos que se muestre.

Ejercicio 7:

- **¿Cuál es la diferencia con el ejercicio 6?**
 - R: Aparentemente no hay diferencias visuales entre el salto de línea con **
** y con **\n**.

Ejercicio 8:

- **¿Qué es un array?**
 - R: Un **array** es un concepto que representa un conjunto de elementos del mismo tipo, en la programación, el **array** o **arreglo** es un objeto que ocupa un espacio en la memoria dedicado a almacenar datos que son del mismo tipo, es decir, un grupo de números enteros, un grupo de caracteres, un grupo de cadenas, un grupo de objetos incluso. Cabe mencionar que las cadenas se consideran un **array** ya que almacenan un grupo de caracteres.
- **¿Cómo declaramos un array?**
 - R: Existen algunas formas de declarar un **array**, éstas dependerán del lenguaje de programación que se está utilizando, en PHP una de las formas es la "Forma larga", en la que se declara un elemento del **array** por cada línea de código.

Ejercicio 9:

- **¿Qué diferencia hay con el script 7?**
 - R: La diferencia principal está en la forma de declarar el **array**, ya que el método largo requiere de más líneas de código para declarar todos los elementos que estarán dentro, el método corto lo hace en un sola línea.
- **¿Cómo se ponen los comentarios?**
 - R: PHP permite poner comentarios dentro del código de 3 formas distintas:
// - Permite comentarios en una sola línea de código.

- Permite comentarios en una sola línea de código.

/**/ - Permite comentarios en un bloque indefinido de código.

Ejercicio 10:

- **¿Cómo se declara una variable numérica en PHP?**
 - R: Para declarar una variable numérica en PHP sólo basta con declarar el identificador de la variable y asignarle un valor numérico, ya sea entero o decimal.
- **¿Cómo se realiza una operación con variables numéricas?**
 - R: Existen los operadores básicos como en la gran mayoría de los lenguajes de programación: +, -, *, /.
- **¿Es posible mostrar en pantalla en una misma línea cadenas y números?**
 - R: Si, es posible mediante la concatenación, ya que en muchas ocasiones y lenguajes la concatenación es muy sencilla con valores de distintos tipos de datos, en lenguajes como Java, es necesario que valores que no son caracteres sean convertidos a caracteres para poder usarse como elementos de una cadena y de esta forma mostrarlos en pantalla más fácilmente.
- **¿Cómo incrementamos una variable?**
 - R: Se utiliza un operador básico para incrementar una variable numérica por una unidad mediante ++.

Ejercicio 11:

- **Anota el significado de las operaciones de comparación**
 - == : “Igual que”
 - != : “Diferente de”
 - < : “Menor que”
 - > : “Mayor que”
 - <= : “Menor o Igual que”
 - >= : “Mayor o Igual que”

Ejercicio 12:

- **Anota el significado de las operaciones de comparación**
 - && : “And”
 - || : “Or”

Ejercicio 13:

- **No hay preguntas a contestar en este ejercicio**

Ejercicio 14:

- **¿Qué funciones de PHP se usan en el script del ejercicio 14?**
 - R: **round()** y **sprintf()**.

Ejercicio 15:

- **¿Qué hace la sentencia if...else?**
 - R: La sentencia if-else es una sentencia condicional básica en programación, su utilidad consiste en evaluar una expresión y realizar una acción basada en el resultado de dicha evaluación, si la expresión da como resultado un valor verdadero, ejecutará un código determinado, **si no**, dependiendo de la declaración de la sentencia ejecutará alguna otra acción en específico o no ejecutará nada.

Ejercicio 17:

- **¿Cuál es la diferencia entre la instrucción SWITCH y la instrucción IF?**

- La sentencia **if** evalúa una expresión y realiza una o dos acciones basándose en el resultado, la sentencia **switch** también evalúa una expresión, pero la diferencia está en que se pueden definir más de dos acciones para un grupo de resultados esperados para la evaluación de la expresión, además de ello, se puede definir el realizar una acción si el resultado no se encuentra en ninguno de los casos determinados para la sentencia **switch**.

Ejercicio 18:

- **Explica lo que hace el script del ejercicio 18:**
 - R: En el script se define una función que recibirá un valor como parámetro, éste valor se evalúa, si es menor a 0.5 se mostrará de color rojo, de lo contrario, se mostrará de color azul, luego, el valor ya con el color determinado, se colocará en una tabla donde se organizarán todos los datos que se están calculando, fuera del script, se utiliza un ciclo **for**, que hará iteraciones en incrementos de 0.1, que van desde 0 hasta 2, dentro del ciclo, se llama a la función **muestra()** y se le envía como parámetro el valor de **X** que representa los incrementos, y se le envía el valor que da como resultado las funciones **sin()** y **cos()**.

Ejercicio 19:

- **¿Cuál es la diferencia con el script anterior?**
 - R: En este script, se agregan condiciones para determinar si el renglón sobre el que se realizó el cálculo es un renglón par o impar, basado en eso, se definirá un color de fondo para cada renglón.

Ejercicio 21:

- **¿Por qué tenemos dos archivos en este ejercicio, uno php y el otro html?**
 - R: Para este ejercicio se cuenta con dos archivos, uno PHP y otro HTML, esto es con la finalidad de separar la parte visual de la parte lógica de una aplicación Web. El archivo PHP contiene toda la lógica que procesa los datos que se van a mostrar, ya sea desde operaciones aritméticas hasta modificación de información en una base de datos. El archivo HTML contiene toda la parte visual, es decir, todo el código que le dirá al navegador cómo mostrar los elementos en la ventana, además de definir elementos que nos permiten como usuarios ingresar datos. Este modelo de desarrollo de aplicaciones Web nos da la ventaja de separar la parte lógica de la visual en un proyecto, así facilita el mantenimiento e implementación de nuevas funcionalidades en un futuro, además de proveer de organización en el uso de distintos lenguajes de marcas y programación.
- **¿Qué es un formulario?**
 - R: Un formulario es un elemento HTML que permite capturar datos en una aplicación, éstos datos serán procesados posteriormente por alguna función escrita en un lenguaje de programación.
- **¿Explica el ciclo while**
 - R: El ciclo While funciona de una forma muy simple, realiza iteraciones basándose en la evaluación de una expresión, **mientras** esta expresión de un resultado positivo, el ciclo seguirá ejecutándose, y sólo terminará cuando el resultado de la evaluación sea negativo, se diferencia del ciclo **for** en que éste, el ciclo for tiene una parte de la condición que determina cuántas repeticiones se deben hacer o de otra forma, establece un límite muy específico que detendrá la ejecución del ciclo.
- **¿Qué hace la variable \$counter?**
 - R: En este script, la variable **\$counter** se incrementa con cada iteración del ciclo **while**. La expresión del ciclo, evalúa el valor de **\$counter**, comparándola con una variable llamada

\$number, la cuál establece un límite de iteraciones que se van a realizar, otra forma de solucionar este script es mediante un ciclo **for**.

Ejercicio 22:

- **¿Cuál es la diferencia con el script del ejercicio 21?**
 - R: La principal diferencia con el ejercicio anterior está en que esta vez sólo se utilizó un archivo, de tipo PHP. El formulario hace referencia al mismo archivo donde se declaró, ya que en este mismo se encuentra el script que procesará el dato que capturó.
En proyectos muy pequeños es útil esta forma de desarrollo, ya que en un solo archivo tenemos todo lo necesario para la aplicación, sin embargo no es la práctica más recomendable, ya que para proyectos más grandes es necesario establecer un orden para poder dar mantenimiento, implementar nuevas funciones o corregir errores durante su desarrollo.
Así es más recomendable una práctica que nos tenga separadas la parte visual y la parte lógica una de otra.