NOTAS HORAS EMPRESA DISTRIBUIDAS:

dia: 02/Julio/2019.

Tiempo invertido: - Java: 1h 30 min. entre: 3:30 to 5:00 (editado)

- ClaseDual: 30 min. entre: 5:00 to 5:30

dia: 04/Julio/2019.

Tiempo invertido: - Java: 7h. entre: 9:00 to 4:00

- ClaseDual: 1h 30 min. entre: 4:00 to 5:30

dia: 05/Julio/2019.

Tiempo invertido: - Java: 7h. entre: 8:00 to 11:00, y despues entre: 1:30 to 3:00

- ClaseDual: 2h. entre: 11.00 to 1:00pm

dia: 09/Julio/2019.

Tiempo invertido: - Java: 8h. entre: 8:00am to 4:00pm,

- ClaseDual: 30min. entre: 4.00pm to 4:30pm

dia: 11/Julio/2019.

Tiempo invertido: - Java: 8h. entre: 8:00am to 11:00am and 1:30pm to 4:30pm,

- ClaseDual: 1h 30min. entre: 11.00am to 1:00pm

dia: 12/Julio/2019.

Tiempo invertido: - Java: 1h 30 MIN. entre: 1:30pm to 3:00pm,

- ClaseDual: 6h 30min. entre: 9.00am to 1:00pm and 3:00pm to 4:00pm

dia: 16/Julio/2019.

Tiempo invertido: - Java: 5h 30 MIN. entre: 9:00am to 2:30pm,

- ClaseDual: 1h. entre: 2:30pm to 3:30pm

dia: 18/Julio/2019.

Tiempo invertido: - Java: 4h entre: 8:00am to 11:00am and 1:00pm to 2:30pm,

- ClaseDual: 4h. 10 min entre: 11:00pm to 1:00pm and 2:30pm to 4:40pm

dia: 19/Julio/2019.

Tiempo invertido: - Java: 3h 30min entre: 11:00am to 2:30pm,

- ClaseDual: 3h 30min entre: 9:00am to 11:00am and 2:30pm to 4:00pm

dia: 23/Julio/2019.

Tiempo invertido: - Java: 4h 30 MIN. entre: 9:00am to 12:30pm and 4:30 to 5:30,

- ClaseDual: 4h. entre: 12:30pm to 4:30pm

dia: 25/Julio/2019.

Tiempo invertido: - Java: 3h 30 MIN. entre: 8:00am to 11:30pm,

- ClaseDual: 4h 30 min. entre: 11:30pm to 4:30pm

dia: 26/Julio/2019.

Tiempo invertido: - Java: 1h 30 MIN. entre: 1:30am to 3:00pm,

- ClaseDual: 5h 30 min. entre: 8:00pm to 1:30pm

dia: 30/Julio/2019.

Tiempo invertido: - Java: 8h. entre: 9.00 am to 5:00pm,

Trecordatorios php:

Tip: Cuando hagas un bucle recuerda en pensar como ese bucle va ejecutar ese código por vuelta, es decir, lo que tengas en que código ve poco a poco viendo como se va aplicar el bucle en un vuelta y en la siguiete. Y asi te daras cuenta como funciona, esto para que no de error.

TIP: Recuerda que el FOREACH cada vez que hace la vuelta, solo recoge el valor actual, no acumula los valores anteriores de las anteriores vueltas, por lo cual siq ueires acumularlos haz una variable. Mas abajo vas a ver como se hace.

1. Cuando hagas un bucle, el $i ya sea en un for o en un while. Te ejecutara la cantidad de veces el contenido que tengas dentro y que le hayas indicado con el condicional, Ademas puedes usar el $i con los arryas si es el caso para que el $i $array[$i]. Para decirle que busque cualquiera o aleatorio de ese array, dependiendo de la condición que tengamos. IMPORTANTE: Aunque no usemos el $i dentro del contenido del bucle, es importante saber que este ejecutara el bucle de todas formas. EL $i al final es para poder ejecutar el bucle.
2. El condicional PAR e IMPAR: if($array % == 1/0). Lo podemos usar en un array para indicarle que con un ECHO nos muestre solo los pares o impares de ese array, recordemos que si los arrays no tienen llaves, las llaves por defecto son números con sus propias posiciones 0, 1, 2, 3, 4, 5, etc.
3. Podemos poner otro método dentro de otro método con $this->
4. Si vamos a usar el SETTER en una clase, y queremos agrupar todos los atributos dentro de una variable, ponemos en el $this->, el parámetro del método. Ejm:

Public function MostrarAtributos($atributos) {

$this->$atributos;

}

1. Si queremos poner una función dentro de otra función y esta tiene subfunción tiene parámetros. Lo que tenemos que hacer es en la función principal, en los parámetros de esta poner los mismos parámetros de la subfunción, además poner también obviamente los parámetros la función principal si es que tiene.
2. Si queremos poner un método dentro de otro sin tener que usar POO, y además usar esta función que introdujimos para ponerla en los parámetros de otra función que introdujimos. Entonces lo que hacemos es en la función que queremos meter otra función, hacer una variable y en esta meter la subfunción que queremos introducir, esta variable debe estar también en los parámetros de la función principal para que cuando usemos esta variable(que es la subfunción que introdujimos) dentro de los parámetros de otra subfunción, que ya estaba también introducida previamente. Podamos asi tener una función principal con varias subfunciones y parámetros.

EJM:

function totalFactura($subtotal, $IRPF, $IVA )

{ $subtotal = subtotal();

$IRPF = IRPF($subtotal, $IRPF);

$IVA = IVA($subtotal, $IVA);

$totalFilas = $subtotal - $IRPF + $IVA;

return $totalFilas;

};

1. Si queremos que los datos que vienen de otra pagina con $\_POST, se introduzcan dentro de un array para que este compare los $\_POST con los datos que tiene dentro de ese array(que comúnmente este array es con llave => valor) para que el $\_POST coincida con la LLAVE’s del array y asi hacer que el ARRAY nos muestre el VALOR de la llave, entonces lo que hacemos es hacer:

$array[$\_POST['contenido del input']];

1. Dentro de un ARRAY\_PUSH podemos poner funciones, el resultado de estas funciones se meterá dentro del array que le indiquemos, y además podemos aprovechar para que los parámetros de la función(si los tiene) pongamos $\_POST, si e sque la función necesita de ellos.. EJM:

$precios = [];

array\_push($precios, precio(

(int) $\_POST['precioUnidad0'. $i],

(int) $\_POST['unidades0' . $i])

Los vendedores reciben 200\*semana + 9% ventas brutas