



Introduction to PHP





5. Estructuras de control.

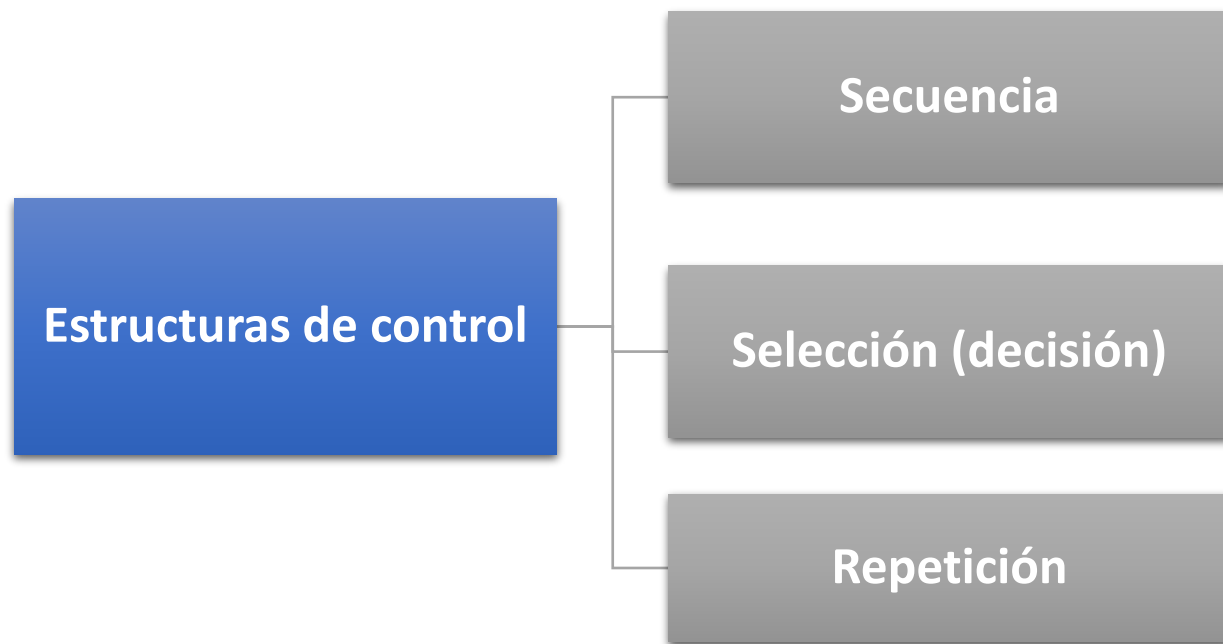


- **Estructuras Condicionales**
- **Sentencias if, else, else if**
- **Sentencias switch**
- **Estructuras de iteración**
- **Ciclo while**
- **Ciclo for**
- **Ciclo for each**



Estructuras de control

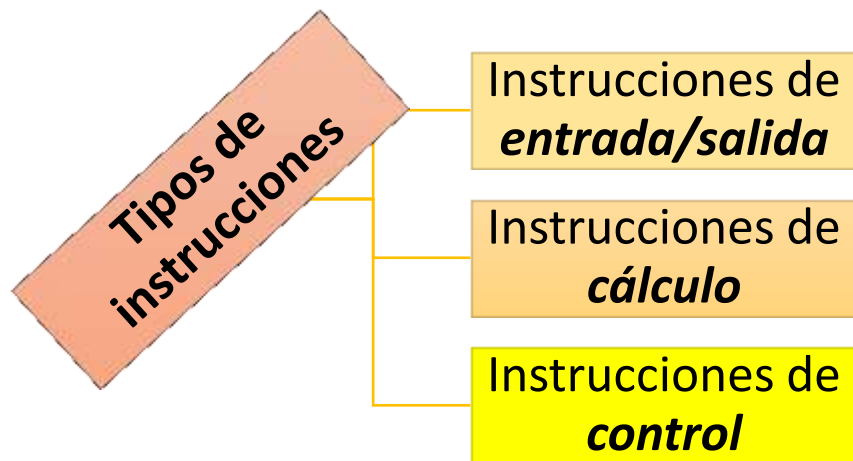
Las estructuras de control controlan el flujo de ejecución de un programa o función.
Tienen un punto de entrada y uno de salida.





Estructuras de control

Dan capacidad de controlar que sentencias se ejecutan y en que momento.



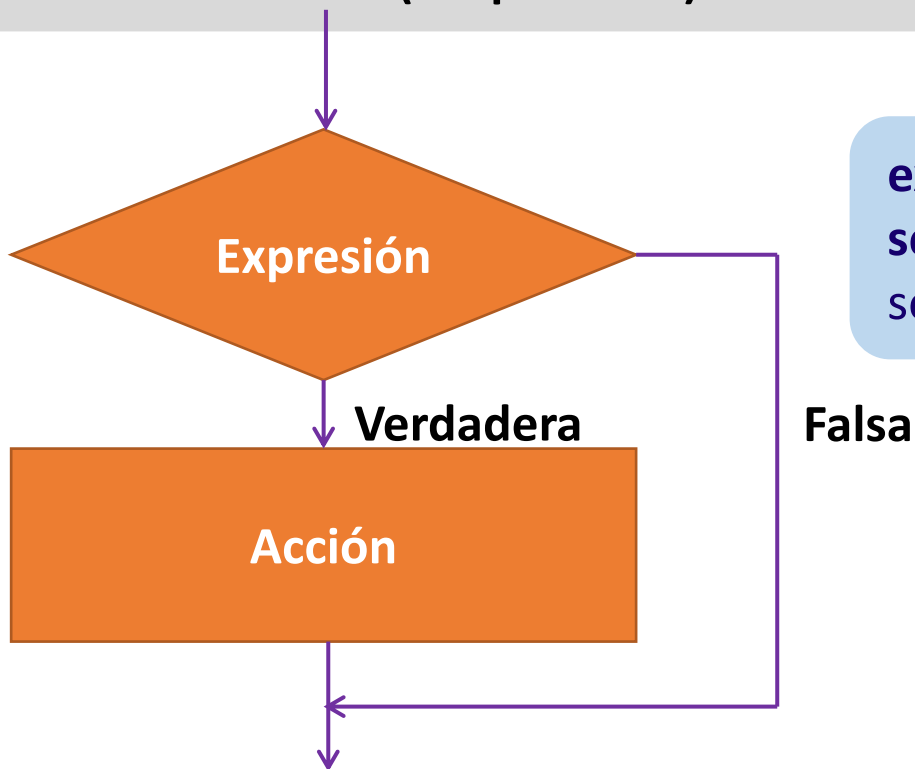
Nota: Hasta el momento solo hemos usado el flujo secuencial.





Sentencia if.

if es la estructura de control principal.
Sintaxis: **if (<expresión>) <sentencia o acción>;**



expresión es una expresión lógica o relacional.
sentencia es cualquier sentencia que se ejecutará si y sólo si la expresión toma valor verdadero.

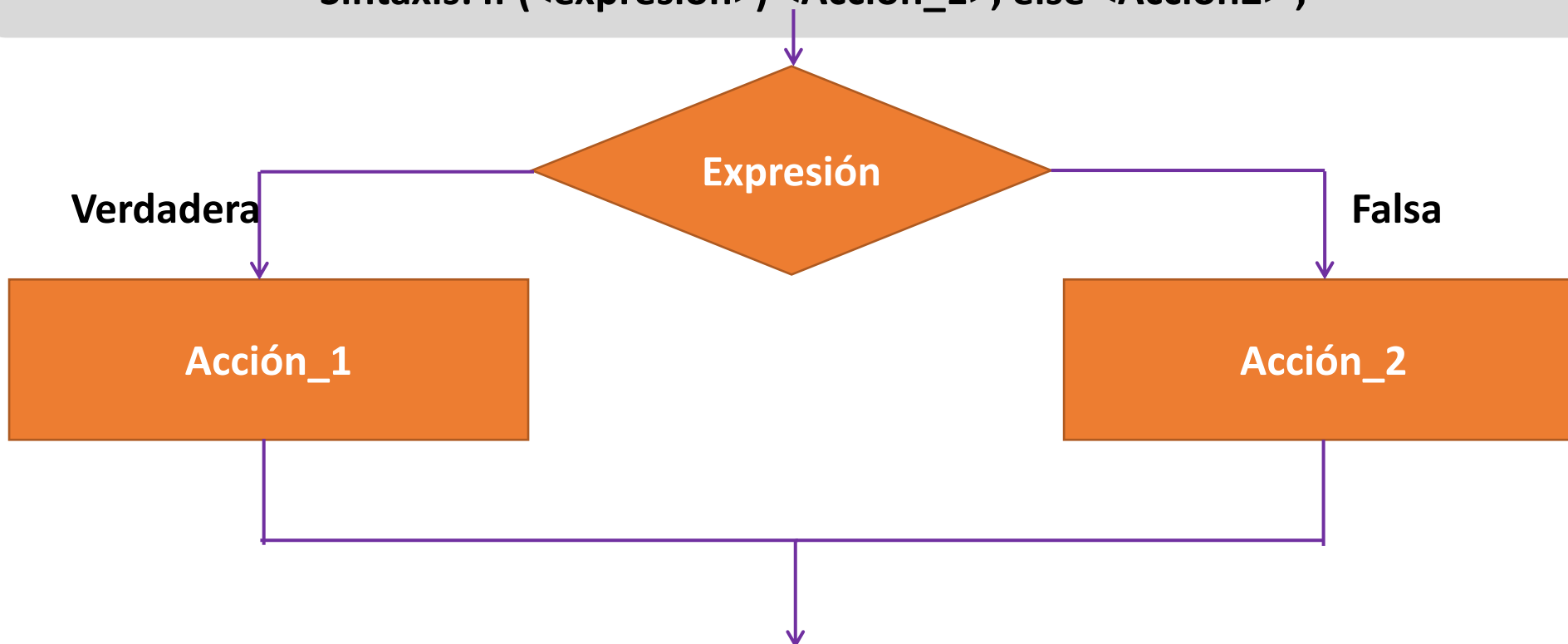


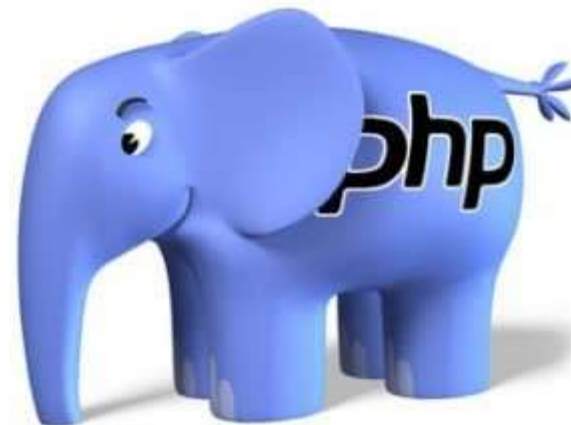


Sentencia if-else.

Esta sentencia tiene dos alternativas, se evalúa la expresión, si es verdadera se ejecuta la primera acción en caso contrario se ejecuta la segunda.

Sintaxis: `if (<expresión>) <Acción_1>; else <Acción2> ;`





Ejercicios:

1. Hacer un programa que pida un número y verifique si están en el rango de (5,10].
2. Hacer una moneda con la función rand().
3. Hacer un programa que pida un número y diga si es negativo, cero o positivo.



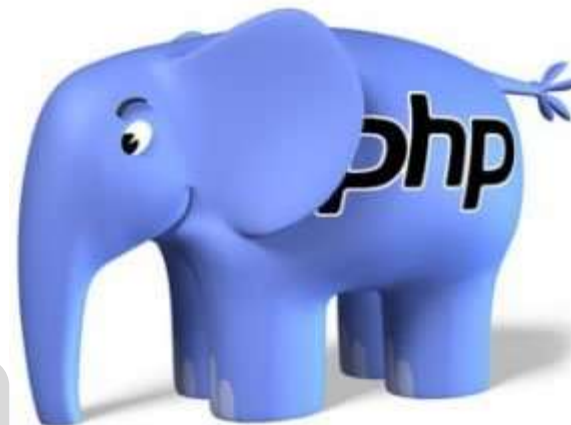


Sentencia elseif.





Sentencia switch.



Switch es útil cuando la selección se basa en el valor de una variable o una expresión simple llamada **selector**.

Sintaxis:

```
switch(selector)
```

```
{
```

```
  case etiqueta1: sentencia1; break;
```

```
  case etiqueta2: sentencia2; break;
```

```
  .
```

```
  .
```

```
  .
```

```
  case etiqueta_n: sentencia_n; break;
```

```
  default: sentencias_d; /*Opcional*/
```

```
}
```

Selector sólo puede ser de tipo **int** o **char**.

etiqueta es un valor único, constante y todas las etiquetas deben ser diferentes.

Break termina la ejecución del switch.

default ejecuta una sentencia en caso de que se introduzca un valor de *selector* no incluido.



Ejercicios:

1. Hacer un menú con opciones del 1 al 4.
2. Hacer un dado con la función rand().



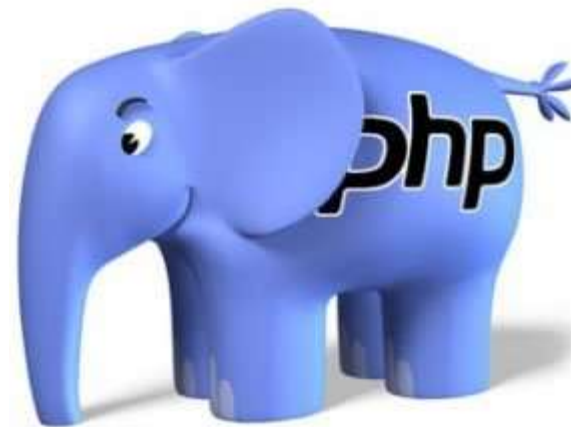


Bucles.

Un **bucle** es cualquier construcción de programa que repite una sentencia o secuencia de sentencias un número de veces.

Las instrucciones que se repiten dentro del bucle se denominan cuerpo del bucle y cada repetición del bucle se llama iteración.



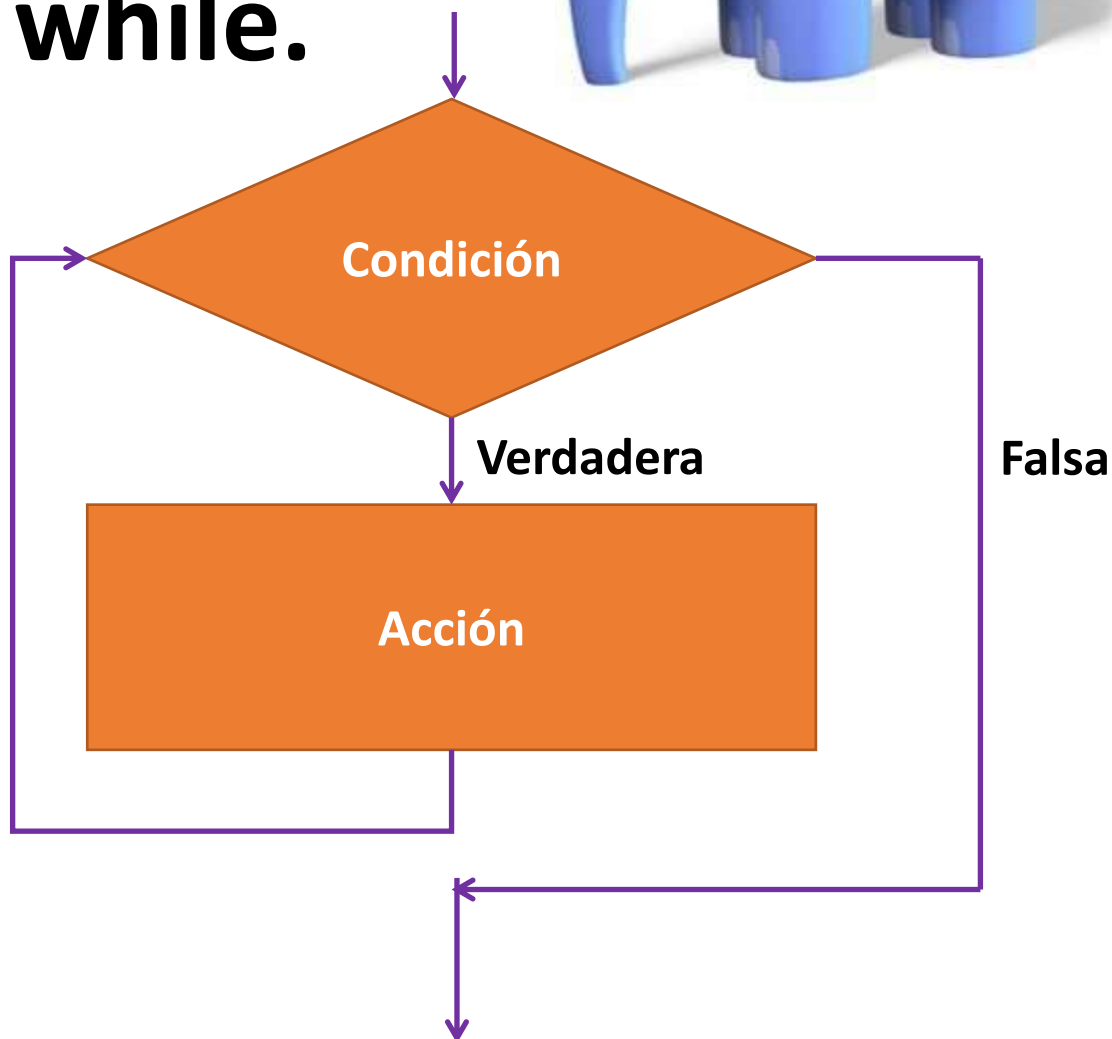


Bucle while.

Sintaxis:

```
while (condición)  
sentencia;
```

```
While (condición)  
{  
sentencia1;  
sentencia2;  
.  
.  
.  
sentenciaN;  
}
```





Ejercicio:

Calcular el valor de la suma $1+2+3+4+5+\dots+100$

Algoritmo

Se utiliza una variable **Contador** como un contador que genere los sucesivos números enteros, y **Suma** almacenar las sumas parciales 1, 1+2, 1+2+3...

1. Establecer Contador a 1
2. Establecer Suma a 0
3. **mientras** Contador \leq 100 **hacer**
 Sumar Contador a Suma
 Incrementar Contador en 1
 fin_mientras
4. Visualizar Suma





Bucle for.

Inicializa la variable de control del bucle.

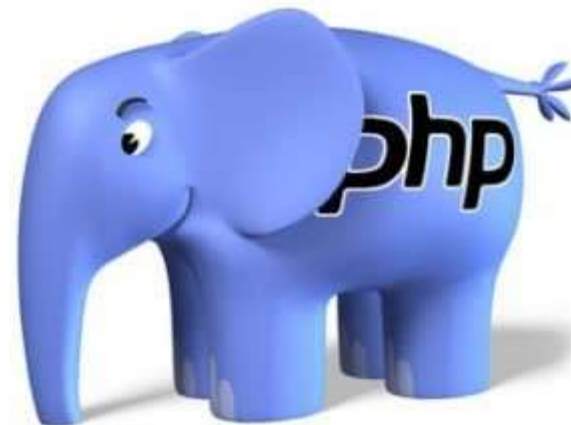
Expresión lógica que determina si han de ejecutar mientras sea verdadera.

Sintaxis:

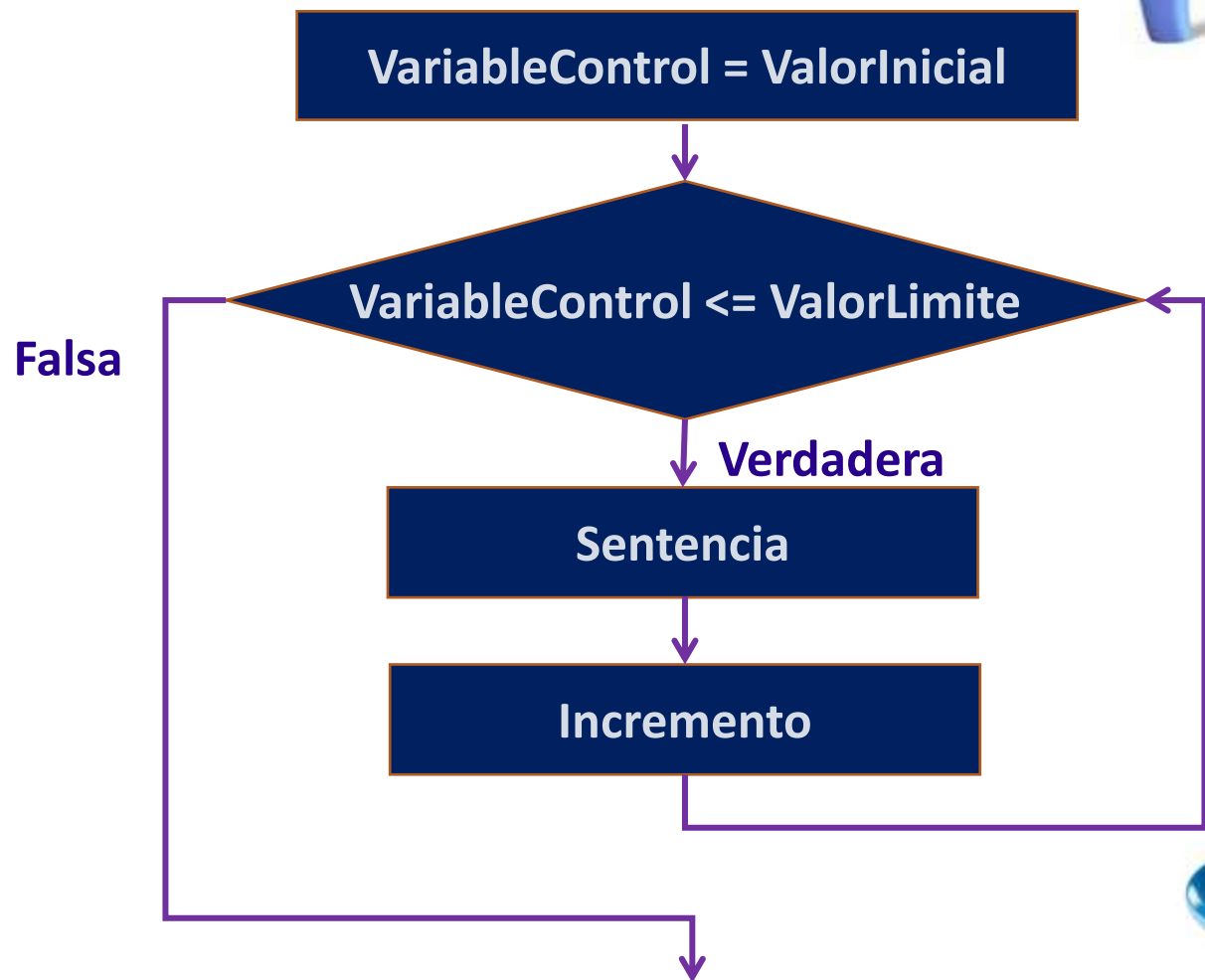
```
for (Iniciación; Condición_Iteracion; Incremento)
{
    sentencias;
}
```

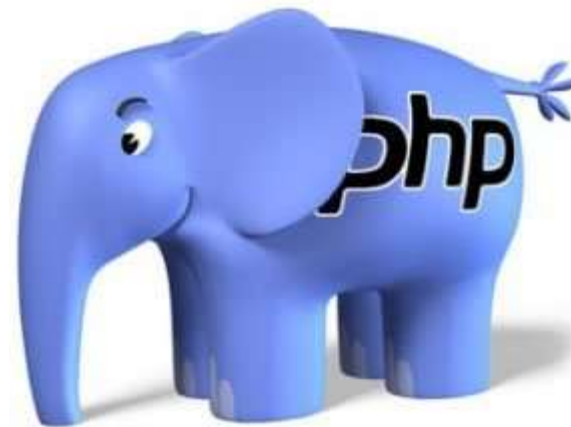
Sentencias a ejecutar en cada iteración del bucle.

Incrementa o decrementa la variable de control del bucle.



Bucle for.





Ejercicio:

1. Hacer un programa que calcule el factorial de un número entero entre 0 y 10.
2. Hacer un programa que imprima las tablas de multiplicar del 1 al 10.
3. Hacer un programa que imprima los números del 1 al 20 pero mencione cuales de ellos son múltiplos de 3.

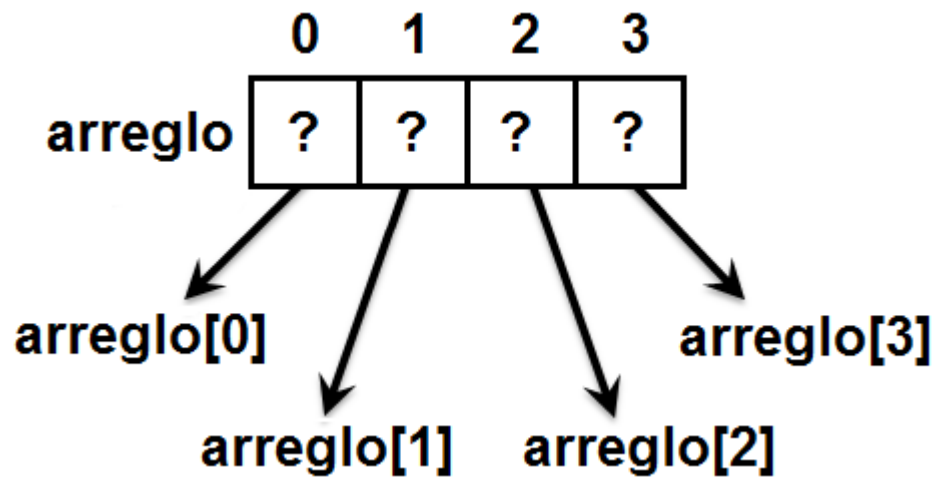




Arreglos (Vectores y matrices).

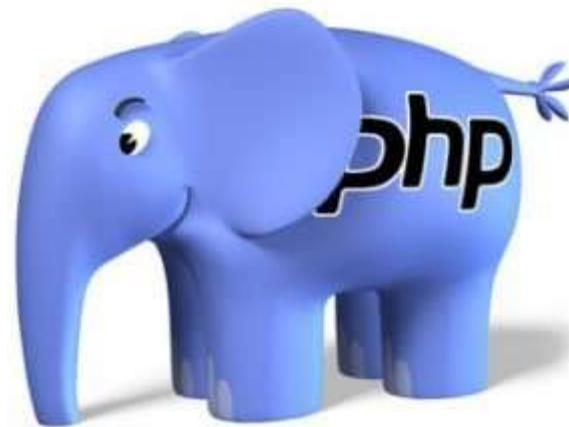


Un arreglo en **PHP** es en realidad un mapa ordenado. Un mapa es un tipo de datos que asocia *valores* con *claves*.

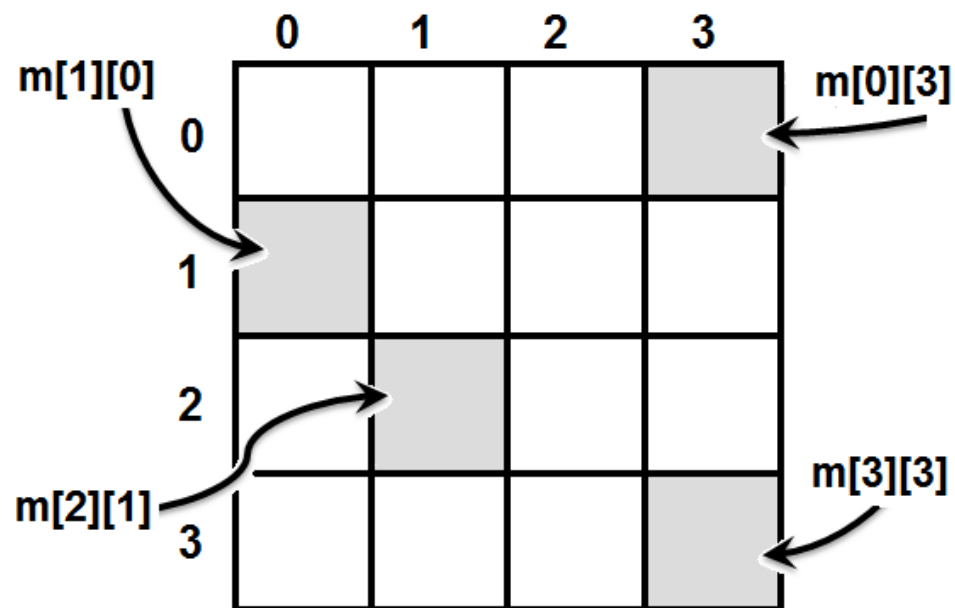




Arreglos (Vectores y matrices).



Los valores de un arreglo pueden ser otros arreglos, entonces los llamamos arreglos multidimensionales.





Arreglos (Vectores y matrices).



Sintaxis:

```
$nums= array(  
    clave => valor,  
    clave2 => valor2,  
    clave3 => valor3,  
    ... );
```

Sintaxis:

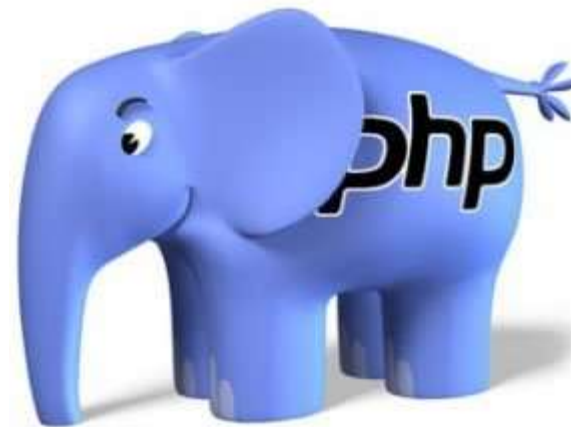
```
$nums= [2,3,4,5,6,7];
```

Sintaxis:

```
$nums= [  
    clave => valor,  
    clave2 => valor2,  
    clave3 => valor3,  
    ... ];
```

Hagamos algunos ejemplos...





Bucle foreach.

Se pasa el nombre del arreglo (\$arreglo) a recorrer.

Sintaxis:

```
for ( $arreglo as $elemento)
{
    sentencias;
}
```

En cada iteración el elemento actual se asigna a \$elemento.



Ejercicio:

1. Hacer un arreglo de 10 números aleatorios e imprimirlo. Luego ordenarlo y mandarlo a imprimir. (for y foreach).
2. Hacer un programa que guarde 5 estados de México con su capital y luego imprimirlos en pantalla. Ejemplo: Pachuca es la capital de Hidalgo. (foreach).
3. Hacer una matriz de 3x3 de números aleatorios y mandarla a imprimir. (for anidado).

