

UNIDAD II – ESTRUCTURAS DE CONTROL

Sentencias condicionales



Objetivos

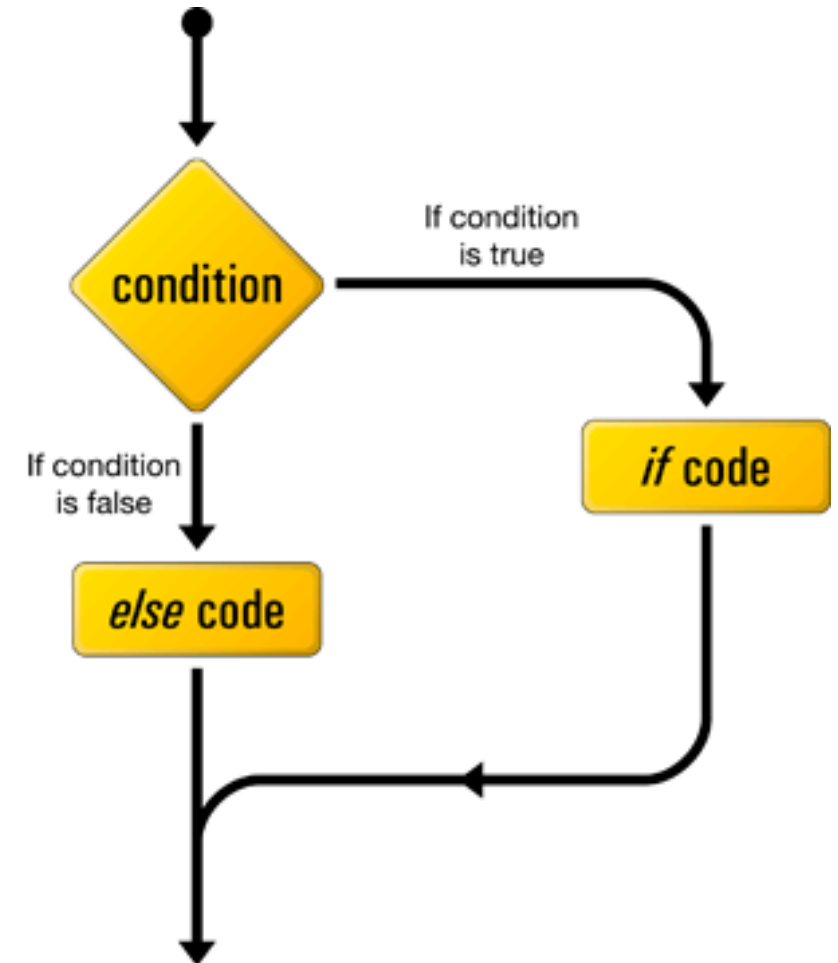
- Adquirir el conocimiento del propósito de las estructuras de control.
- Desarrollar el dominio de la sintaxis de las sentencias condicionales en PHP.
- Lograr un alto nivel de capacidad para construir sentencias condicionales *if* con todas sus variantes.
- Conseguir un alto nivel de dominio en la construcción de sentencias de múltiple decisión con la instrucción *switch*.
- Adquirir dominio en el uso del operador ternario condicional `? :`.

Contenidos

- Sentencias condicionales
 - ▣ Estructuras de control.
 - ▣ Sentencia *if*.
 - ▣ Sentencia *if-else*.
 - ▣ Sentencia *else if*.
 - ▣ Sentencia *switch*.
 - ▣ Sentencias condicionales anidadas.
 - ▣ Operador condicional.

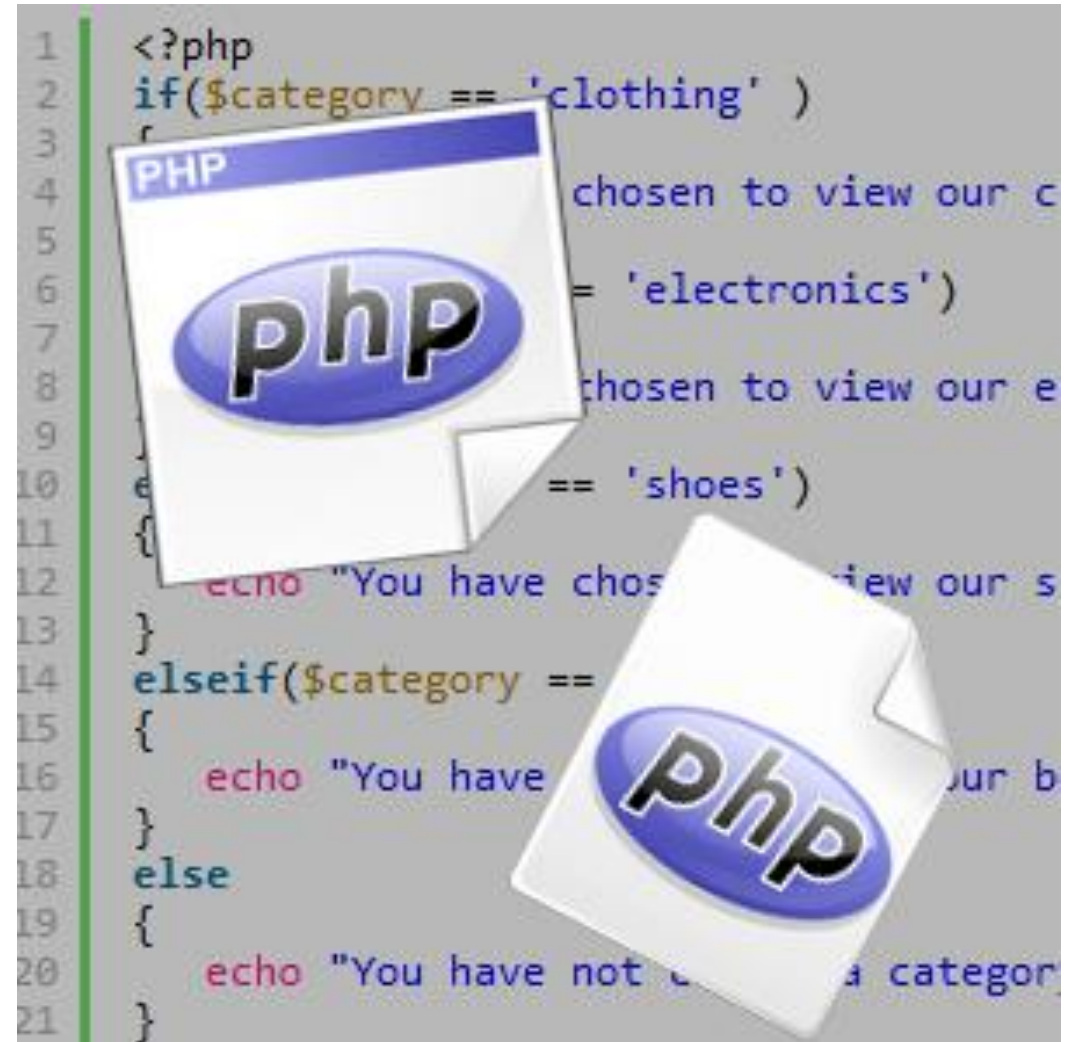
Estructuras de control

- ❑ La mayor parte de problemas que se resuelven con programación no sería posible solucionarlos usando únicamente instrucciones secuenciales.
- ❑ En una gran cantidad de situaciones será necesario tomar decisiones, seleccionar una de varias alternativas o ejecutar varias veces un bloque de código.



Estructuras de control

- ❑ Las estructuras de control son construcciones de los lenguajes de programación que permiten controlar el comportamiento de los programas.
- ❑ Con las estructuras de control es posible determinar bajo qué condiciones del programa se ejecutará un grupo de instrucciones o se realizará más de una vez un conjunto de instrucciones.



```
1 <?php
2 if($category == 'clothing' )
3 {
4     // chosen to view our c
5
6     // = 'electronics')
7
8     // chosen to view our e
9
10    // == 'shoes')
11 }
12 echo "You have chosen to view our s
13 }
14 elseif($category ==
15 {
16     echo "You have
17 }
18 else
19 {
20     echo "You have not chosen a category
21 }
```

The image shows a snippet of PHP code illustrating control structures. The code uses an `if` statement to check if the category is 'clothing', followed by an `elseif` statement. Comments in the code indicate different categories like 'electronics' and 'shoes'. Two PHP document icons, each with a blue 'php' logo, are overlaid on the code. The background is a light gray with a green vertical line on the left side, suggesting a code editor.

SENTENCIAS CONDICIONALES



If, If-Else, If-Else-If-Else y Switch

Sentencias condicionales

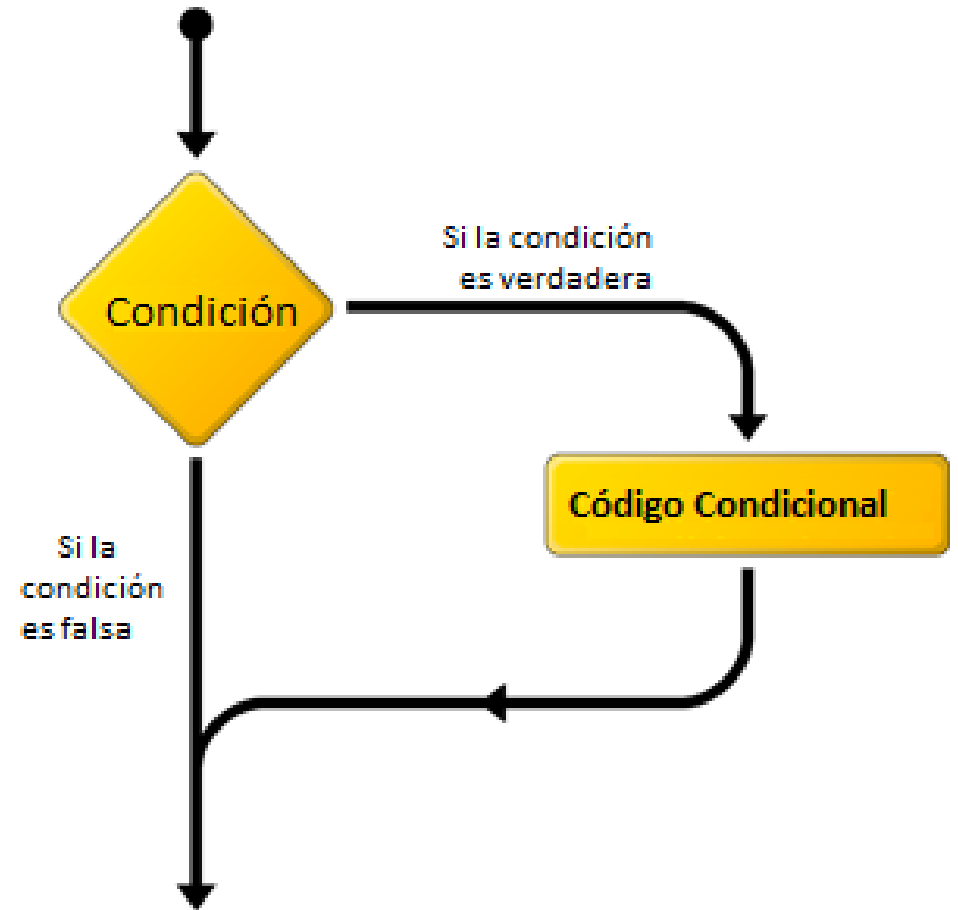
- Se denominan así a las sentencias de un lenguaje que plantean una condición que de evaluarse verdadera, mediante una expresión condicional, provocará la ejecución de un bloque de instrucciones.
- En tanto que, si se evalúa falsa, de forma opcional, se ejecutará un bloque de instrucciones diferente.
- En otras palabras, mediante una sentencia condicional se puede hacer que el programa pueda elegir entre dos posibles rutas de ejecución de instrucciones en función del resultado de la evaluación de una expresión condicional.

Sentencias condicionales

- Bajo ninguna circunstancia se ejecutarán ambos bloques de instrucciones, siempre se ejecutará solamente uno de ellos, el bloque *if* si la condición se evalúa verdadera o el bloque *else*, si la condición se evalúa falsa.
- Con una sentencia condicional puede comprobar si una variable coincide con un valor determinado, o si la variable es mayor, mayor o igual, menor, menor o igual que un valor concreto. Así como si una cadena de texto coincide con un literal de cadena, si posee cierto número de caracteres, e incluso si el resultado de evaluar una expresión lógica resulta verdadero o falso y en base a ese resultado ejecutar una instrucción o un bloque de instrucciones.

Sentencia *if*

- Una sentencia *if* en su forma más básica posee una sola expresión condicional que será evaluada y de resultar verdadera provocará la ejecución de una instrucción o un bloque de instrucciones delimitados por el bloque *if*.
- Si la condición se evalúa falsa, puede ejecutarse un bloque de instrucciones alternativo, si así lo estima conveniente el programador, aunque puede ser que no se ejecute código alguno.

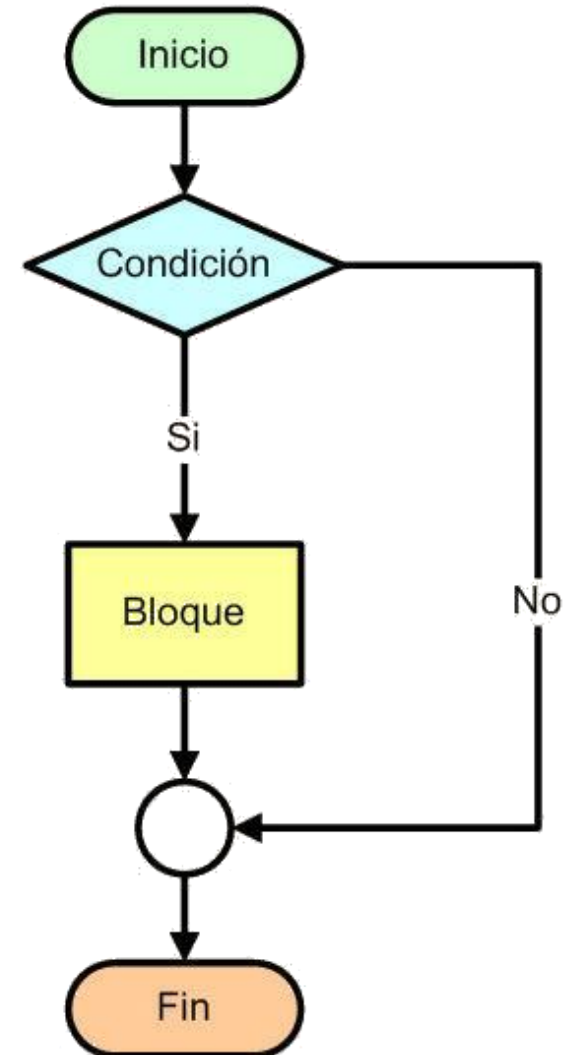


Sintaxis de la sentencia *if*

- Una sentencia *if* básica tiene la siguiente sintaxis:

```
if(condicion) [{  
    //bloque de instrucciones if  
}]
```
- Se evalúa la condición y si esta resulta verdadera se ejecutan las instrucciones del **bloque if**. De lo contrario no.

NOTA: El uso de los corchetes es sólo para indicar que el elemento encerrado entre éstos es opcional en el caso que sólo exista una única instrucción dentro de la sentencia *if*. En este caso nos referimos a las llaves (`{, }`). Sin embargo, es sintácticamente correcto e incluso conveniente usar llaves siempre.



Sentencia *if-else*

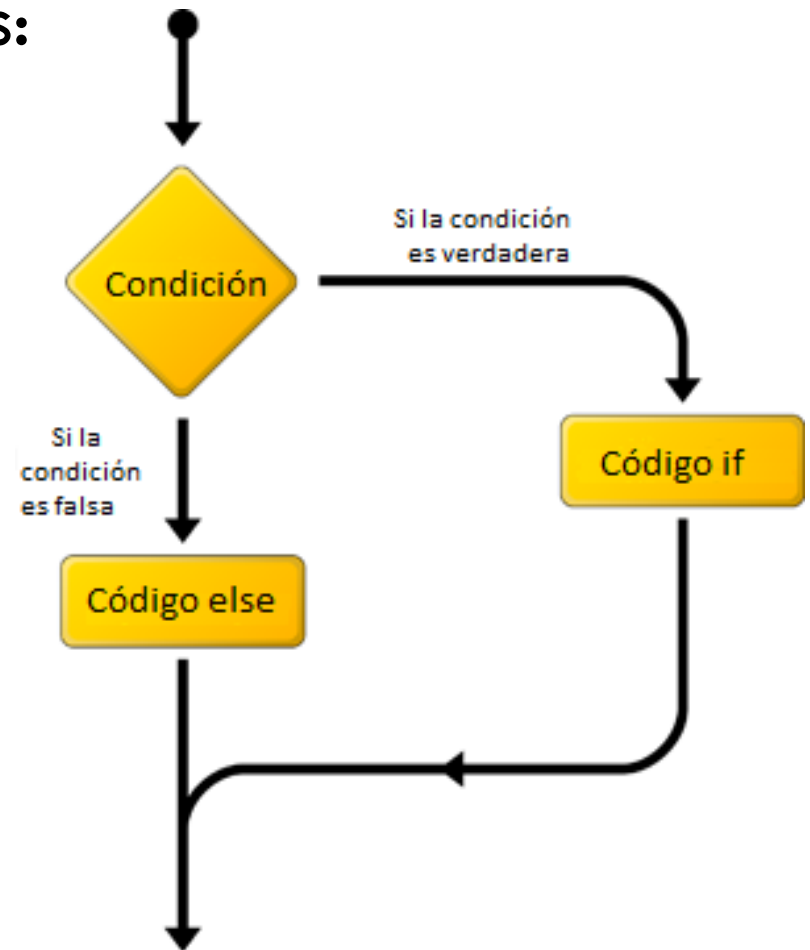
- La sentencia *if* puede ir acompañada de una sentencia *else*, cuando se requiera que se ejecute un bloque de instrucciones alternativo para el caso que la condición a evaluar resulte falsa.
- Debe tener claro que en una instrucción *if-else*, únicamente se ejecutará uno de los bloques de instrucciones. El bloque *if* o el bloque *else*, dependiendo de si la condición se evalúa verdadera o falsa, respectivamente. Nunca se ejecutarán ambos bloques de instrucciones.

Sintaxis de la sentencia *if-else*

- Una sentencia ***if-else*** tiene la siguiente sintaxis:

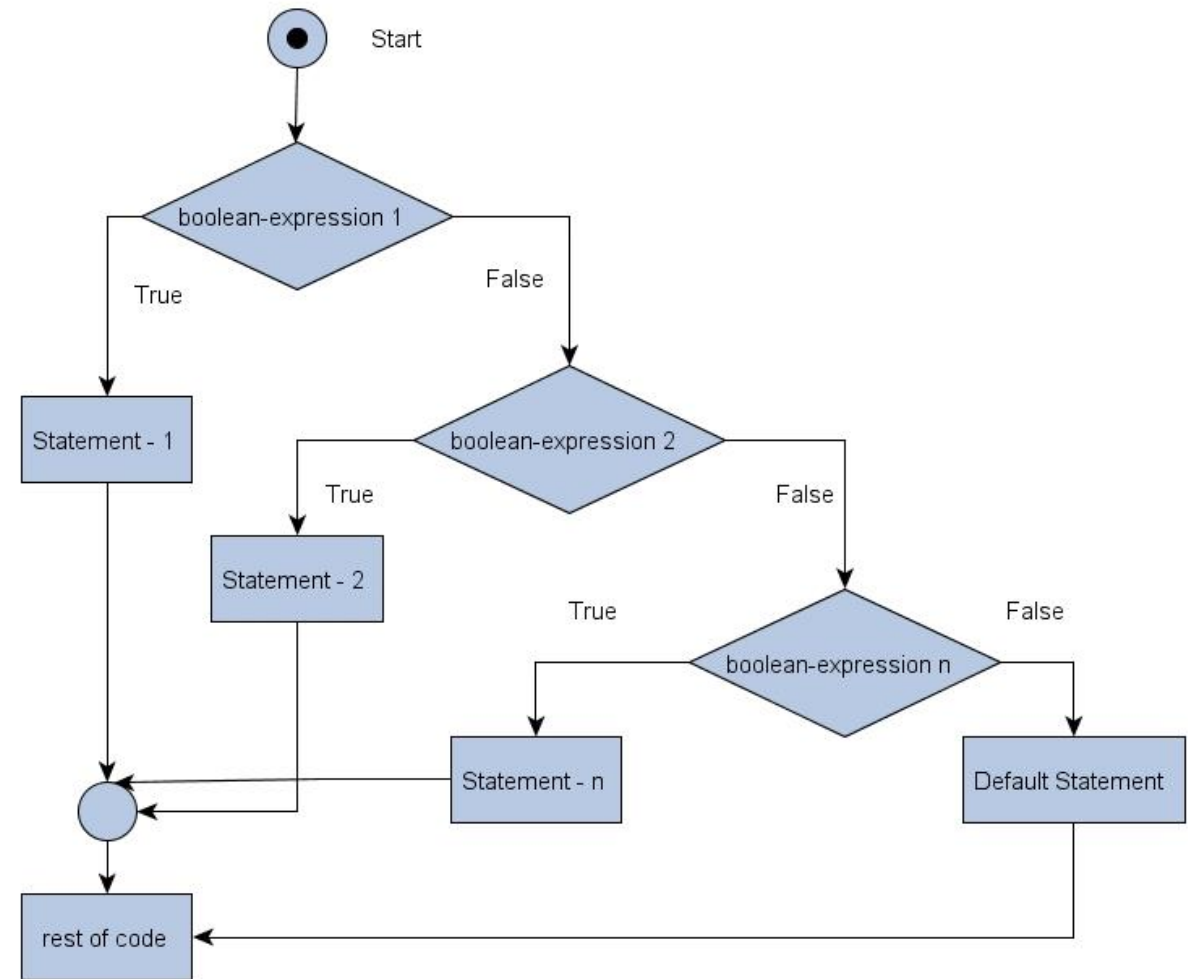
```
if(condición) [{  
    //bloque de instrucciones if;  
}]  
else [{  
    //bloque de instrucciones else;  
}]
```

- Los corchetes indican que los elementos encerrados son opcionales, solo en caso que los bloques *if* y *else* contengan una sola instrucción. No obstante, se recomienda usar las llaves siempre.



Sentencia *if-elseif-else*

- Esta construcción de la sentencia ***if***, resulta muy útil cuando se desea evaluar más de una condición y asociar a cada condición un bloque de instrucciones a ejecutar.
- La instrucción evalúa si la primera condición es verdadera, y luego, ejecuta ese bloque si el resultado es verdadero. De no ser así, evalúa la segunda condición, y de resultar verdadero, ejecuta el segundo bloque de instrucciones. Si resulta falso, pasa a evaluar la siguiente condición, y así sucesivamente. Si se incluye un bloque ***else*** se ejecuta este bloque de instrucciones si todas las evaluaciones anteriores fueron falsas.

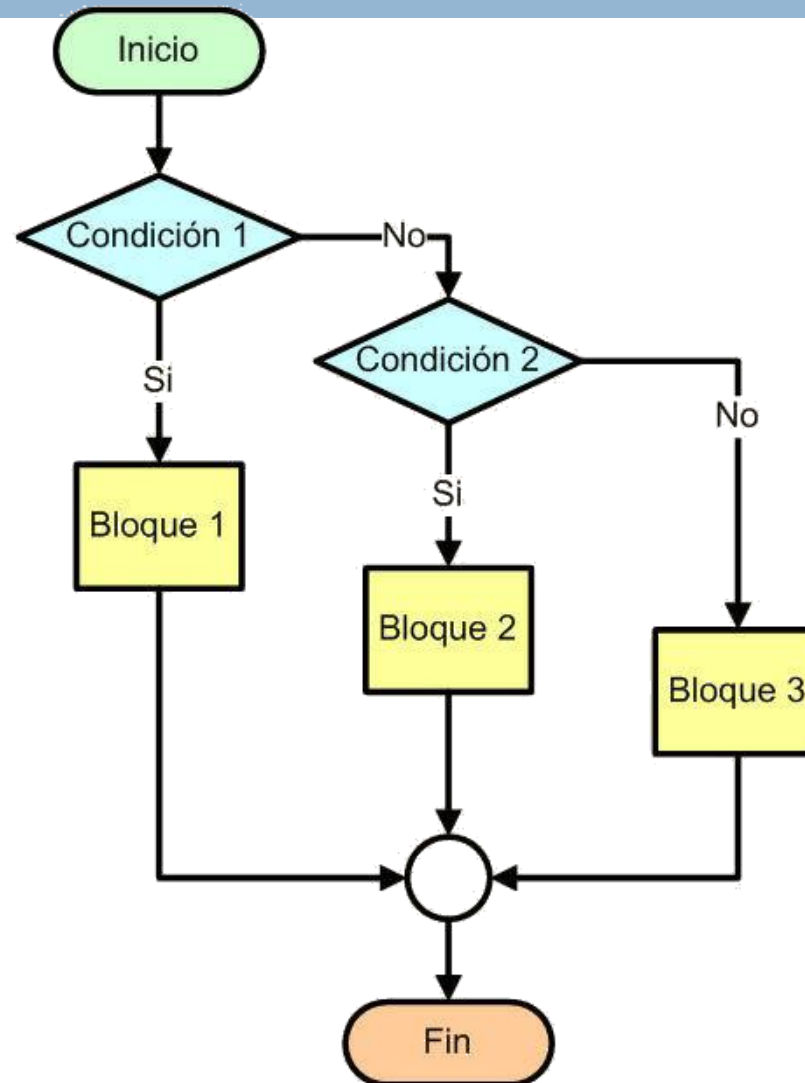


Sintaxis de la sentencia *if-else if-else*

- Una sentencia ***if-elseif-else*** tiene la siguiente sintaxis:

```
if(condicion1) [{  
    //bloque de instrucciones if  
}]  
[elseif(condicion2) [{  
    //primer bloque de instrucciones elseif  
}]  
[elseif(condicion3) [{  
    //segundo bloque de instrucciones elseif  
}]  
...  
[else[{  
    //bloque de instrucciones else  
}]]]
```

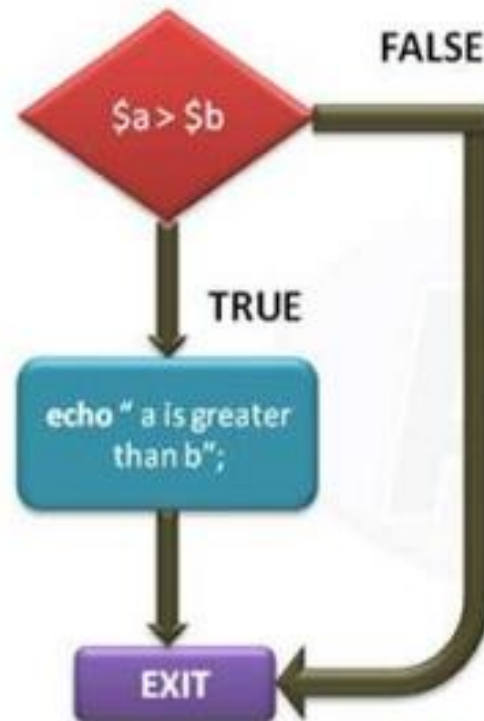
Diagrama de flujo del *if-else if-else*



Ejemplo con sentencia *if-else*

```
if($num % 2 == 0) {  
    echo "$num es par";  
}  
else{  
    echo "$num es impar";  
}
```

```
if($a > $b) {  
    echo "$a es mayor que $b";  
}  
else{  
    echo "$a es menor que $b";  
}
```



Example:

```
<?php  
if ($a > $b)  
{  
    echo "a is bigger than b";  
}  
?>
```


Ejemplo con sentencia *if-else if-else*

```
$dia = date("D");  
if($dia == "Mon"){  
    echo "Hoy es lunes";  
}  
else if($dia == "Tue"){  
    echo "Hoy es Martes";  
}  
else if($dia == "Wed"){  
    echo "Hoy es Miércoles";  
}  
else if($dia == "Thu")  
    echo "Hoy es Jueves";  
}  
...  
else{  
    echo "Hoy es Domingo";  
}
```

if anidados

- ❑ Las sentencias *if* pueden aparecer adentro de otro *if* y también adentro de un *else*.
- ❑ Con un *if* anidado se pueden crear varios niveles de comprobación antes de ejecutar un bloque de instrucciones. Aunque técnicamente dos sentencias *if* anidados se pueden sustituir por un sólo *if* con una condición compuesta por una expresión que contenga un operador lógico And (&&).
- ❑ A veces es mejor o más conveniente plantear las condiciones separadas por simplicidad.

Ejemplo con sentencias *if* anidadas

```
if($numero % 2 == 0) {  
    if($numero > 10 && $numero < 26) {  
        echo "El número es par y está entre 10 y 26";  
    }  
    else{  
        echo "El número es par pero no está en el rango";  
    }  
}  
else{  
    echo "El número no es par";  
}
```

Sintaxis alternativa de las sentencias condicionales

- ❑ PHP ofrece una sintaxis alternativa para todas las sentencias condicionales. Esto es algo que casi sólo se encuentra documentado en el sitio oficial de PHP en <http://php.net>.
- ❑ Se muestran a continuación las sintaxis alternativa para todas las variantes de la sentencia *if* que hemos visto anteriormente y para la sentencia *switch*.
- ❑ Posteriormente, al revisar las sentencias repetitivas, retomaremos la sintaxis alternativas de esas estructuras de control.

Sintaxis alternativa de la SENTENCIA *if*

- La sentencia *if* se puede escribir también de la siguiente forma:

```
if(condicion1):
```

```
    //instrucciones if;
```

```
[else if(condicion2):
```

```
    //instrucciones else if;
```

```
...
```

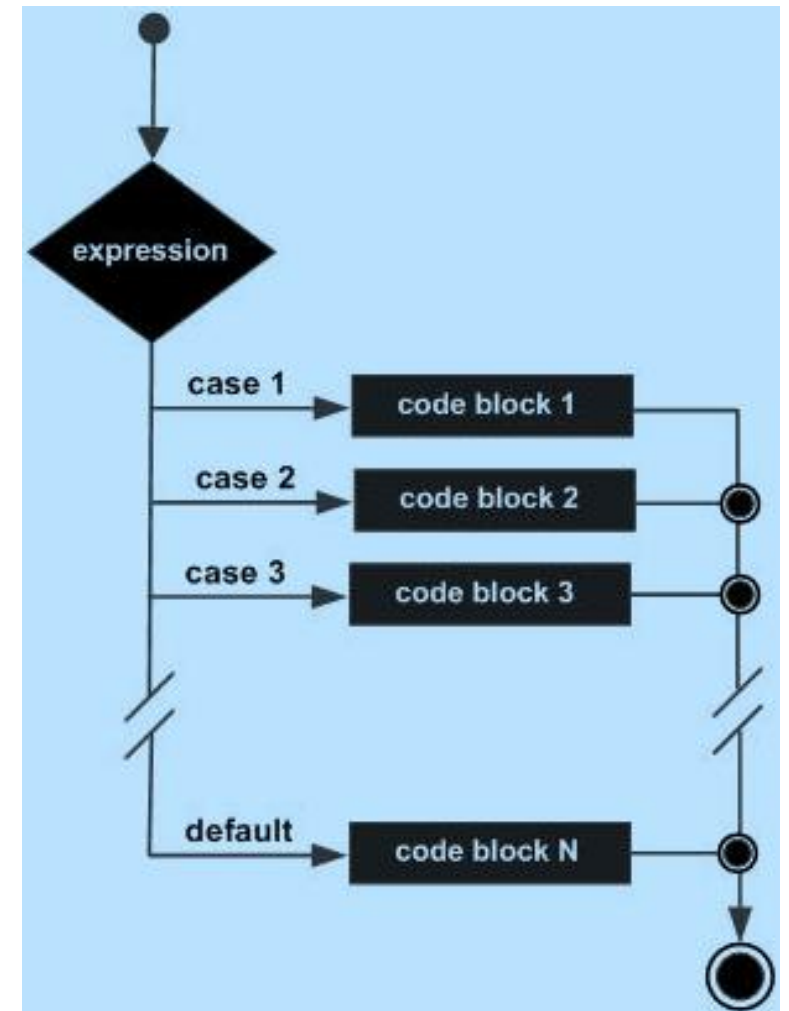
```
else:
```

```
    //instrucciones else;]
```

```
endif;
```

Sentencia *switch*

- Una sentencia ***switch*** es muy similar a una serie de múltiples sentencias del tipo ***if-else if-else***.
- Con la sentencia ***switch*** se pueden comparar variables o expresiones con una serie de múltiples valores de prueba y ejecutar un determinado bloque de código dependiendo de cuál sea el valor final de la variable o expresión.



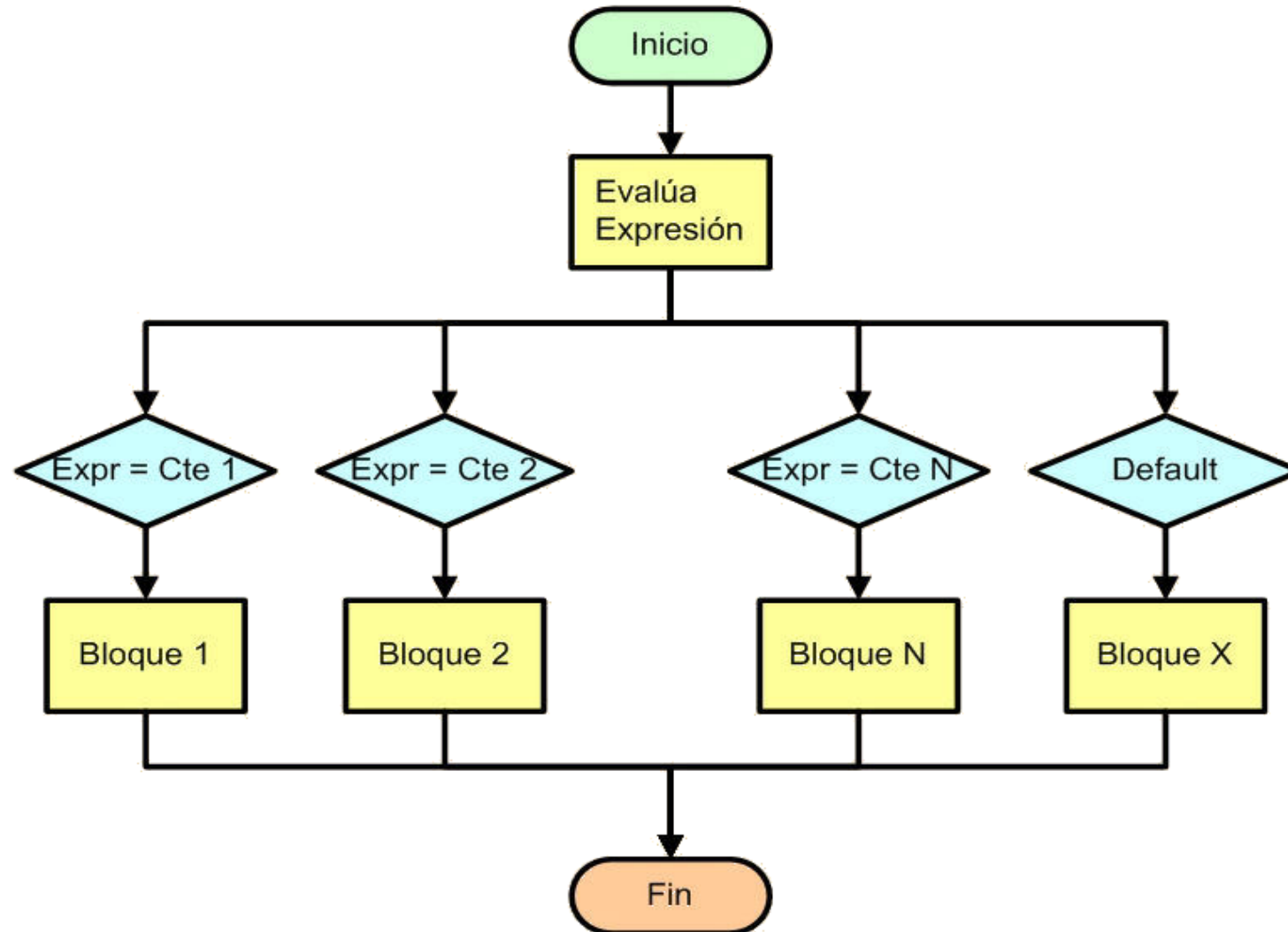
Sintaxis de la sentencia *switch*

```
switch (expresion) {  
    case valor1:  
        [sentencias1;]  
        [break;]  
    case valor2:  
        [sentencias2;]  
        [break;]  
    ...  
    case valorN:  
        [sentenciasN;]  
        [break;]  
    [default:  
        sentencias;  
        break;]  
}
```

Donde, *expresion* arroja un valor que será evaluado y comparado con *valor1*, *valor2*, ..., y *valorN* de cada sentencia *case*. Si no coincide con ninguno de estos valores de prueba, entonces se ejecuta el bloque *default*.

Digamos que la sentencia *switch* evalúa *expresion* y busca en cada uno de los *case* un resultado que coincida con dicha evaluación.

Diagrama de flujo del *switch*



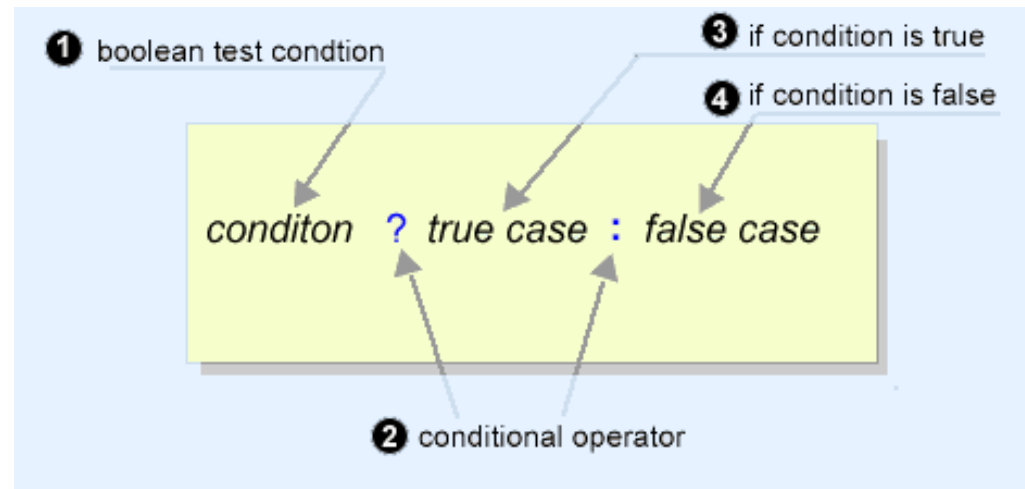
Sintaxis alternativa de la estructura de control condicional *switch*

- Una sentencia ***switch*** puede escribirse alternativamente así:

```
switch (expresion) :  
    case valor1:  
        [sentencias1;]  
        [break;]  
  
    ...  
  
    case valorN:  
        [sentenciasN;]  
        [break;]  
  
    [default:  
        sentencias;]  
endswitch;
```

Operador condicional

- Se llama así a un operador especial, conocido también como operador ternario. Se representa haciendo uso de los siguientes operadores: ? y :.
- Para utilizarlo se requieren tres operandos, donde el primero es una expresión condicional que al ser evaluada como verdadera hará que todo el operador devuelva el segundo operando, en caso de ser falso, devolverá el tercer operando.



Operador condicional

- La sintaxis del operador condicional es la siguiente:

`(expresionCondicional) ? expresionTrue : expresionFalse;`

- La expresión condicional devolverá `expresionTrue` si la `expresionCondicional` es evaluada como `true` y devolverá `expresionFalse`, si la `expresionCondicional` es evaluada como `false`.

- Ejemplos:

`$mayor = ($n1 > $n2) ? $n1 : $n2;`

`$msg = ($acceso == true) ? "Bienvenido" : "Inicie sesión";`

Operador condicional

- Si va a utilizar el operador ternario debe tener presente que únicamente es aplicable cuando se tenga que ejecutar una sola instrucción en la parte del *if* y una sola instrucción en la parte del *else*.
- En caso de no cumplirse ese requerimiento, tendrá que construir una sentencia *if-else* normal. Del mismo modo que si sólo estuviera presente la parte del *if*, sin *else* en la instrucción que se desea realizar.

FIN



If, If-Else, If-Else-If-Else y Switch