UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES
Centro de e-Learning
Secretaría de Cultura y Extensión Universitaria

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Curso:

PROGRAMADOR WEB INICIAL

UTN.BA | Centro de e-Learning
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES
Secretaría de Cultura y Extensión Universitaria

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Módulo 3:

PHP Y JAVASCRIPT

Unidad 2:

Estructuras de control: selectivas y repetitivas en PHP

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Presentación

En esta unidad, abordaremos las estructuras de control selectivas y repetitivas, y cómo influyen en el flujo de nuestro programa.

Asimismo, veremos el pasaje de valores de una variable de un programa a otro utilizando unas variables especiales de las que dispone PHP para tal fin.

Finalmente, conoceremos la función include para optimizar nuestro código.

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Objetivos

Que los participantes logren...

- Conocer las sentencias condicionales: IF, ELSE, ELSEIF y SWITCH.
- Conocer las sentencias repetitivas: FOR, WHILE, FOREACH.
- Comprender su comportamiento y determinar cuándo utilizar cada una.
- Implementar la función include con el propósito de optimizar código.

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Bloques temáticos

1. Estructuras de control
2. Estructuras de control selectivas
    a. Sentencia IF
    b. Sentencia ELSE
    c. Sentencia ELSEIF
    d. Sentencia SWITCH-CASE
3. Estructuras de repetición
    a. Los bucles for, while y foreach
        i. Los bucles for
        ii. Los bucles while
        iii. Los bucles foreach
4. La función INCLUDE

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Estructuras de control

PHP está construido por una serie de sentencias que componen nuestro script.
Hay diferentes tipos de sentencias, y hoy veremos las estructuras de control, presentes en la mayoría de los lenguajes de programación. Estas sentencias nos permitirán realizar acciones más complejas en nuestro sitio, pudiendo "decidir" qué hacer en cada caso particular.

Debemos recordar que muchas veces realizaremos la misma acción múltiples veces, pero en este caso sólo nos centraremos en las estructuras de control selectivas o condicionales. Más adelante, veremos las repetitivas o iterativas.

Estructuras de control selectivas

Estas estructuras evaluarán una condición fijada y según su valor de verdad (TRUE o FALSE) se ejecutará un bloque de código u otro.
Para realizar estos flujos de decisiones se utilizarán las sentencias **if, else y elseif**.

Sentencia IF

El constructor **if** permite la ejecución condicional de fragmentos de código. Es una estructura de control que se utiliza para tomar decisiones según se cumpla una condición (o varias) o no. La estructura en PHP es la siguiente:

```
if (condición){
  bloque de código a ejecutar si la condición devuelve TRUE
}
```

La condición o expresión es evaluada a su valor booleano (TRUE o FALSE).

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

- ➔ Si la expresión devuelve **true**, PHP ejecutará la sentencia,
- ➔ Si devuelve **false,** la ignorará.

Las expresiones que utilizaremos serán las de comparación que vimos en la unidad anterior. Estas expresiones evalúan si algo es false (falso) o true (verdadero). Las recordamos a continuación:

>	mayor que
>=	mayor o igual que
==	igual
!=	distinto
<	menor que
<=	menor o igual que

También podremos utilizar los operadores de equivalencia estricta:

- === (igual y del mismo tipo)
- !== (diferente o de distinto tipo).

El siguiente ejemplo mostraría "a es mayor que b" **si** $a es mayor que $b:

```
<?php
      if ($a > $b) {
        echo "a es mayor que b";
      }
?>
```

Muchas veces vamos a necesitar realizar una o varias acciones cuando hacemos nuestra comparación. No debemos repetir el condicional, ya que podremos colocar la cantidad de sentencias que queramos dentro de nuestro IF.
En nuestro ejemplo realizaremos algo más luego de mostrar "a es mayor que b" si $a es mayor que $b y asignaremos el valor de $a a $b:

```
<?php
      if ($a > $b) {
```

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

```
        echo "a es mayor que b";
        $b = $a;
      }
?>
```

Las sentencias if pueden anidarse: colocarse una dentro de otra infinitamente lo que nos brinda completa flexibilidad para la ejecución condicional de diferentes partes del programa.

Sentencia ELSE

Muchas veces vamos a necesitar ejecutar una acción si nuestra condición del **IF** es FALSA, y por lo tanto en este caso utilizaremos la sentencia **ELSE.**
Es una expresión que siempre va junto al IF.

```
if (condición){
   bloque de código a ejecutar si la condición devuelve TRUE
} else {
   otro bloque de código si la condición devuelve FALSE
}
```

Por ejemplo, el siguiente código podrá imprimir en pantalla dos cadenas distintas según el resultado de la condición dentro del IF.

- "a es mayor que b" **si** $a es mayor que $b (IF - TRUE).
- "a NO es mayor que b" en el **caso contrario** (IF- FALSE).

```
<?php
      if ($a > $b) {
        echo "a es mayor que b";
      } else {
        echo "a NO es mayor que b";
      }
?>
```

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

La sentencia **else** solo será ejecutada si la expresión **if** es evaluada como FALSE. Recordemos que, como vimos anteriormente, podremos anidar IF dentro de nuestras estructuras principales para evaluar situaciones más complejas.

```
if (condición1) {
      Instrucción 1;
      Instrucción 2;
      ...
} else {
      if (condición2){
            Instrucción A;
            Instrucción B;
            ...
       } else {
            Instrucción X ... }
}
```

En el siguiente ejemplo, utilizaremos dos condiciones en nuestro IF principal, utilizando el operador lógico &&. Dicha condición será VERDADERA, si AMBAS condiciones se cumplen.

```
if (($edad>=18) && ($dinero>5)){
      echo "<p>Puedes comprar cerveza porque tienes 18 y tu dinero es mayor que 200
      pesos</p>";
} else{
      echo "<p>O no tienes más de 200 pesos o no tienes 18 años, aléjate de la
      cerveza! </p>";
}
```

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Sentencia ELSEIF

Como pudimos ver en los ejemplos anteriores, muchas veces necesitaremos combinar else con if, y allí es dónde utilizaremos la sentencia combinada **elseif**.
Funciona de la misma manera que else, ejecuta una sentencia diferente en caso que la expresión if principal se evalúe como FALSE. Sin embargo, a diferencia de else, esta expresión solo se ejecutará si la expresión condicional del elseif se evalúa como TRUE. Utilizar elseif nos simplificará la sintaxis cuando utilizamos IF ANIDADOS.

```
if (condicion1){
      Instrucción 1;
      Instrucción 2;
      ...
} elseif (condicion2){
      Instrucción A;
      Instrucción B;
      ...
} else{
      Instrucción X
      ...
}
```

Por ejemplo, el siguiente código deberá mostrar:

- “a es mayor que b”.
- “a es igual que b”.
- “a es menor que b”.

```
<?php
      if ($a > $b) {
          echo "a es mayor que b";
      } elseif ($a == $b) {
          echo "a es igual que b";
      } else {
```

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

```
            echo "a es menor que b";
        }
?>
```

Podremos tener varias sentencias elseif dentro del mismo IF, y se ejecutará la que primero se evalúe como TRUE.

Hay dos maneras de escribir nuestra sentencia en PHP:

- ➔ 'else if' separados o
- ➔ 'elseif' juntos.

El comportamiento de ambas maneras es idéntico, van a realizar exactamente lo mismo.

Ejemplo:

```
<?php
        $test = 33;
        if ($test > 40) {
                echo " $test es mayor que 40.";
                elseif ($test > 35) {
                        echo " $test es mayor que 35.";
                }
                elseif ($test > 30) {
                echo " $test es mayor que 30.";
                } else {
                echo " $test es menor que 40, 35 y 30.";
        }
?>
```

En este caso, la respuesta sería: 33 es mayor que 30.

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Sentencia SWITCH-CASE

También conoceremos la sentencia switch, que es similar a una serie de sentencias IF en una misma expresión.
En muchas oportunidades, vamos a querer comparar una misma variable (o expresión) con diferentes opciones o valores, y ejecutar un código distinto que dependa del valor que contiene. Vamos a pasar de caso en caso, utilizando la sentencia **break;** al finalizar el caso actual, y pasar al siguiente.

En el siguiente ejemplo, realizaremos el mismo caso utilizando una serie de sentencias if y elseif, y por otro lado utilizaremos la sentencia switch, que acabamos de aprender:

Ejemplo #1: Estructura switch

```
<?php
	if ($i == 0) {
	    echo "i es igual a 0";
	} elseif ($i == 1) {
	    echo "i es igual a 1";
	} elseif ($i == 2) {
	    echo "i es igual a 2";
	}

	switch ($i) {
		case 0:
			echo "i es igual a 0";
		break;
		case 1:
			echo "i es igual a 1";
		break;
		case 2:
			echo "i es igual a 2";
		break;
	}
?>
```

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Ejemplo #2 La estructura switch nos permite utilizar strings

```
<?php
	switch ($i) {
		case "manzana":
			echo "i es una manzana";
		break;
		case "barra":
			echo "i es una barra";
		break;
		case "pastel":
			echo "i es un pastel";
		break;
	}
?>
```

Debemos conocer varios puntos para entender por qué utilizamos la sentencia switch:

- Es ejecutada con el fin de evitar errores.
- La sentencia **switch** ejecuta línea por línea.
- PHP ejecutará la expresión dentro del **case correspondiente,** cuando coincida con la expresión que tenemos en el switch.
- PHP seguirá ejecutando todas las sentencias **case** hasta el final del bloque switch, o hasta que encuentre una sentencia **break**. Si no se escribe una sentencia break al final de la lista de sentencias de un caso, PHP seguirá ejecutando las sentencias del caso siguiente.

Por ejemplo:

```
<?php
	switch ($i) {
		case 0:
			echo "i es igual a 0";
		case 1:
			echo "i es igual a 1";
		case 2:
			echo "i es igual a 2";
	}
?>
```

- Si $i **es igual a 0**, PHP ejecutaría todas las sentencias echo.
- Si $i **es igual a 1**, PHP ejecutaría las últimas dos sentencias echo.

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

- Se obtendría el comportamiento esperado (se mostraría 'i es igual a 2') sólo si $i es **igual a 2**.

Por lo tanto, no nos debemos olvidar de colocar las sentencias **break;** dentro de nuestros **case,** para evitar resultados inesperados.

➔ Si utilizamos la sentencia **switch**, nuestra condición será evaluada solo una vez y el resultado se compara con cada una de las sentencias **case.**

➔ En cambio, si utilizamos una sentencia **elseif** la condición será evaluada otra vez. Esto puede provocar, teniendo una condición compleja o en bucle, que el elseif tarde más tiempo en ejecutarse que un switch.

No deberemos poseer un bloque de código para cada caso, por lo cual podremos dejarlo vacío. Se salteará las instrucciones hasta llegar a la que coincida.

```
<?php
	switch ($i) {
		case 0:
		case 1:
		case 2:
			echo "i es menor que 3 pero no negativo";
		break;
		case 3:
			echo "i es 3";
	}
?>
```

Podremos utilizar un caso “predeterminado” o default cuando queremos realizar una caso más genérico que **no coincida** con nuestros demás casos. Por ejemplo:

```
<?php
	switch ($i) {
		case 0:
			echo "i es igual a 0";
		break;
		case 1:
			echo "i es igual a 1";
```

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

```
                break;
                case 2:
                        echo "i es igual a 2";
                break;
                default:
                        echo "i no es igual a 0, 1 ni 2";
        }
?>
```

Estructuras de repetición

Los bucles for, while y foreach

Los **bucles** son estructuras de control de flujo que ejecutan código, pero no de forma "condicional" o "selectiva" como **if/else** y **switch**, sino que definen la ejecución "repetitiva" de un código.

Debido a su naturaleza iterativa, son perfectos para iterar sobre vectores o conjuntos de filas o registros recuperados de una base de datos.

Hay tres funciones básicas en la estructura del bucle for en PHP:

- **for**
- **while**
- **foreach**.

Los bucles for

Se utilizan cuando queremos repetir una acción un determinado número de veces. Permite ejecutar el comando un número exacto de veces, es decir, el número de repeticiones debe saberse de antemano.

El bucle **for** funciona modificando el valor de una variable que se utiliza como indicador del "índice" o número de vuelta en la que se encuentra el bucle, el "contador de vueltas", también conocido como contador.

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

```
for (valorInicial; valorFinal; incremento){
    bloque a ejecutar;
}
```

- La variable "valorInicial" debe ser el primer argumento dentro del for, fijando el valor original que tendrá esa variable al inicio del bucle
- Luego, como segundo parámetro colocaremos "valorFinal" que establece el valor más alto (o último) de la variable, es decir, la cantidad de veces que se repite el comando.
- En tercer lugar, se especifica qué incremento o decremento sufrirá esa variable luego de terminar de dar cada vuelta del bucle.

Ejemplo:

```
for ($indice=1; $indice<10; $indice=$indice + 1) {
      Aquí van las instrucciones que ejecutará 10 veces
      (mientras $indice valga menos que 10);
}
```

Los bucles while

Muchas veces nos sucederá que debemos realizar una acción hasta que pase algo, pero no sabemos a ciencia cierta cuándo sucederá.

Los bucles **While** ejecutan un bloque de código mientras se cumpla la condición especificada.

```
While (Condicion){
      bloque a ejecutar;
```

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

```
}
```

Ejemplo:

```
while ($condicion !="Salir") {
      Aquí van las instrucciones se que ejecutarán mientras el
      valor de la variable $condicion sea distinto a "Salir"
}
```

El siguiente ejemplo muestra los números del 1 al 5:

```
$numero = 1;

while($numero <= 5) {

  echo "El número es: $numero <br>";

  $numero++;

}
```

`$numero = 1;` - Inicializa el contador de bucle (`$numero`) y establece el valor de inicio en 1

`$numero <= 5` - Continúa el ciclo mientras `$numero` sea menor o igual a 5

`$numero++;` - Aumenta el valor del contador de bucle en 1 para cada iteración.

Los bucles foreach

Para simplificar el recorrido de vectores, hay una construcción llamada foreach que es específicamente para recorrer vectores (si pasa algo que no sea un vector como argumento, generará un error).

Su estructura "mínima" es la siguiente:

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

```
foreach ($vector as $valor) {
      bloque a ejecutar;
}
```

Ejemplo, aquí recorremos el array y multiplicamos su valor * 2.

```
$array = array(1, 2, 3, 4);
foreach ($array as &$valor) {
    $valor = $valor * 2;

  // $array ahora es array(2, 4, 6, 8)
    echo "$valor <br>";
}
```

La función INCLUDE

Al usar la función include(), se puede colocar y reutilizar parte del código (script o html simple) en otra página tantas veces como quieran. Por ejemplo se suele utilizar para los encabezado y pie de página, entonces si necesitamos cambiar uno de ellos, deberemos cambiar el archivo específico y se cambiará en todos los archivos que lo contengan.
La sentencia include() inserta y evalúa el archivo especificado. Aquí no solo se pueden incluir archivos de nuestro servidor, sino también sitios remotos (indicar URL).

```
<?php
      include(“pagina.php”);
?>
```

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

En este caso, Include(); llama al archivo pagina.php y lo inserta en el propio punto del script donde hacemos la llamada.

Si usamos include() para insertar un archivo, debemos considerar que PHP entra en modo html, por lo que si el archivo a insertar contiene código php para ser evaluado (ejecutado), debe ser encerrado dentro de etiquetas de comienzo y fin de PHP.

También podemos usar múltiples include anidados (es decir, usar una función include para llamar a otro archivo en el archivo incluido), con la única precaución de tener en cuenta que los archivos que se van insertando se ejecutan en el entorno del primer archivo que contiene la llamada.

En resumen

En esta unidad hemos conocido las estructuras de control selectivas llamadas IF, ELSE, ELSEIF y SWITCH. Vimos cómo se utilizan ejemplificándolas, y aprendimos a combinarlas utilizándose una dentro de otra, o mejor dicho anidadas.

También vimos estructuras llamadas repetitivas, las cuales nos permiten ejecutar un bloque de código un número determinado de veces.

Bibliografía utilizada y sugerida

- PHP - News Archive - 2021. Recuperado de: https://www.php.net/archive/2021.php
- PHP: Estructuras de Control - Manual. Recuperado de: https://www.php.net/manual/es/language.control-structures.php
- W3Schools. PHP Include Files. Recuperado de: https://www.w3schools.com/php/php_includes.asp
- W3Schools. PHP Loops. Recuperado de: https://www.w3schools.com/php/php_looping.asp