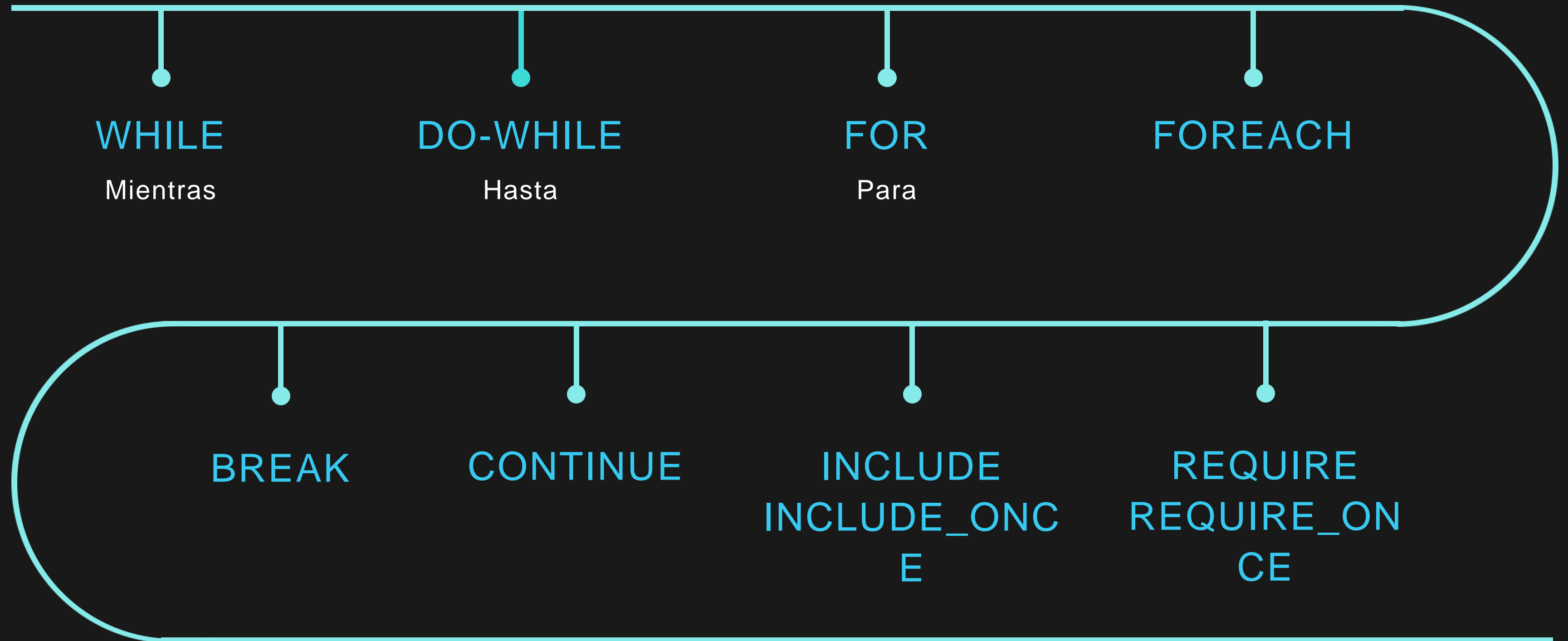


Desarrollo de Sistemas Web

Tec. Superior en Análisis Funcional de Sistemas

Estructuras de control





```
1  $i = 1;
2  while ($i ≤ 10):
3      echo $i;
4      $i++;
5  endwhile;
6
7  $f = 1;
8  while ($f ≤ 10) {
9      echo $f++;
10 }
```

1

While

Se inicia y continúa
ejecutándose siempre
que la condición sea
verdadera



2

Do-While

```
1  $i = 0;  
2  do {  
3      echo $i;  
4  } while ($i > 0);
```

Se inicia al menos 1 vez

Continúa su ejecución
siempre que la condición
sea verdadera



```
1  for (expr1; expr2; expr3) {  
2      # code ...  
3  }
```

3

For

3 expresiones

1- se ejecuta al
comenzar.
2- condición (true).

3- se ejecuta si la
expression 2 se cumple.

3

For



```
1  for ($i = 1; $i ≤ 10; $i++) {  
2      echo "<br>";  
3      echo $i;  
4  }
```

Espera 3 expresiones.

1- se ejecuta al comenzar.

2- condición (true).

3- se ejecuta si la
expresión 2 se cumple.

4

Foreach

Iterar dentro de
arrays u objetos

Espera 2 parámetros, array
a iterar y variable que
almacena el registro
temporal



```
1 foreach ($variable as $key) {  
2     # code ...  
3 }
```

```
1  $alumnos = [  
2      [  
3          "nombre" => "Juan",  
4          "edad" => 20,  
5          "notas" => [  
6              "mate" => 10,  
7              "fisica" => 8,  
8              "ingles" => 7  
9          ]  
10     ],  
11     [  
12         "nombre" => "Pedro",  
13         "edad" => 20,  
14         "notas" => [  
15             "mate" => 10,  
16             "fisica" => 8,  
17             "ingles" => 7  
18         ]  
19     ],  
20 ];
```

```
1  //Foreach Alumnos  
2  foreach ($alumnos as $alumno) {  
3      echo "Nombre: ".$alumno['nombre']." <br>";  
4      echo "Edad: ".$alumno['edad']." <br>";  
5      echo "Notas: <br>";  
6      echo "Mate: ".$alumno['notas']['mate']." <br>";  
7      echo "Fisica: ".$alumno['notas']['fisica']." <br>";  
8      echo "Ingles: ".$alumno['notas']['ingles']." <br>";  
9      echo "<hr>";  
10 }
```



```
1 $alumnos = [  
2     [  
3         "nombre" => "Juan",  
4         "edad" => 20,  
5         "notas" => [  
6             "mate" => 10,  
7             "fisica" => 8,  
8             "ingles" => 7  
9         ]  
10    ],  
11    [  
12        "nombre" => "Pedro",  
13        "edad" => 20,  
14        "notas" => [  
15            "mate" => 10,  
16            "fisica" => 8,  
17            "ingles" => 7  
18        ]  
19    ],  
20 ];
```

```
1 foreach ($alumnos as $key => $alumno) {  
2     echo "Nombre: ".$alumno['nombre']." <br>";  
3     echo "Edad: ".$alumno['edad']." <br>";  
4     echo "Notas: <br>";  
5     foreach ($alumno['notas'] as $key => $nota) {  
6         echo $key." : ".$nota." <br>";  
7     }  
8     echo "<hr>";  
9 }
```

5

Break

```
1  $i = 0;
2  while (++$i) {
3      switch ($i) {
4          case 5:
5              echo "En 5, salgo del switch pero
6                  continúo ejecutando el while.<br />";
7              break 1;
8          case 10:
9              echo "En 10; salgo del switch y del while.<br />";
10             break 2;
11         default:
12             break;
13     }
14 }
```

Se puede utilizar en for, foreach, while, do-while y switch

Sirve para finalizar iteraciones

Puede estar acompañada de un parámetro numérico para indicar estructuras anidadas que debe finalizar

6

Continue



```
1  for ($i = 0; $i ≤ 10; ++$i) {  
2      if ($i % 2 ≠ 0) {  
3          continue;  
4      }  
5      echo "$i\n";  
6  }  
7
```

Se puede utilizar en estructuras repetitivas

Sirve para saltar la ejecución en curso y continuar con la próxima iteración

Puede estar acompañada de un parámetro numérico

7

Return



```
1 function suma($num1, $num2) {  
2     return $num1 + $num2;  
3 }
```

Se utiliza para devolver
el control al módulo que
lo invoca

Es muy usado en funciones



```
1 include "2.incluir.php";  
2 echo $informacion['nombre'];
```



```
1 $informacion = [  
2     "nombre" => "Juan",  
3     "apellidos" => "Pérez",  
4     "edad" => "30",  
5     "direccion" => [  
6         "calle" => "C/ Madrid",  
7         "numero" => "1",  
8         "ciudad" => "Madrid",  
9         "pais" => "España"  
10    ]  
11 ];
```

8

Include
Include_once

Se utiliza para incluir un
archivo en otro y
ejecutarlo

INCLUDE_ONCE solo
incluye el archivo una vez
por más de que lo
volvamos a invocar.



```
1 require "2.incluir.php";  
2 echo $informacion['nombre'];
```



```
1 $informacion = [  
2     "nombre" => "Juan",  
3     "apellidos" => "Pérez",  
4     "edad" => "30",  
5     "direccion" => [  
6         "calle" => "C/ Madrid",  
7         "numero" => "1",  
8         "ciudad" => "Madrid",  
9         "pais" => "España"  
10    ]  
11 ];
```

9

Require Require_once

Se utiliza para incluir un
archivo en otro y
ejecutarlo

REQUIRE_ONCE solo
incluye el archivo una vez
por más de que lo
volvamos a invocar.

La diferencia con include
es que, si hay error de
ejecución en el archivo que
se invoca, se detiene la
ejecución de todo el
programa

```
1 <!DOCTYPE HTML>
2 <html>
3   <head>
4     <title> HTML
5     <meta http-ec
6     <meta name="c
7     <meta name="p
8     <link rel="sl
9     <link rel="s
10    <script type
11  </head>
12  <body>
13    <h1> HTML
14    <div classe
15    <div cla
```

1

Funciones
def.

2

Tipado definido

3

Variables locales

4

Variables estáticas

5

Variables
globales

6

Variables
superglobales



```
1 declare(strict_types=1);
2 function bienvenida(string $usuario = 'Invitado'):string{
3     return "Bienvenido $usuario";
4 }
5 echo bienvenida('Nicolás');
6
7
8 function sumar(int ... $numeros):int{
9     return array_sum($numeros);
10 }
11 echo sumar(1,2,3,4,5);
12
13
14 function sumar2($numero1, $numero2):int{
15     return $numero1+$numero2;
16 }
17 echo sumar2(1,2);
18
19 echo date('d-m-Y');
```

1

FUNCIONES

Conjunto de instrucciones que, a lo largo de la ejecución del programa, se van a poder invocar todas las veces que las necesitamos.



Palabra reservada

Nombre función

Parámetros

1

FUNCIONES

```
1 function bienvenida($usuario = 'Invitado')
2 {
3     echo "Bienvenido $usuario";
4 }
5 bienvenida('Juan');
```

Código de la
función

Invocación

1

FUNCIONES

Las que ejecutan acciones

Las que retornan valores



```
1 function bienvenida($usuario = 'Invitado')
2 {
3     echo "Bienvenido $usuario";
4 }
5 bienvenida('Juan');
```



```
1 function bienvenida($usuario){
2     return "Bienvenido $usuario";
3 }
4 echo bienvenida('Nicolás');
```

1

FUNCIONES



```
1 function bienvenida($usuario = 'Invitado')
2 {
3     echo "Bienvenido $usuario";
4 }
5 bienvenida('Juan');
```



```
1 function sumar( $numero1, $numero2){
2     return $numero1+$numero2;
3 }
4 echo sumar(1,2);
```

RESTRICCIONES DE INGRESO

RESTRICCIÓN
DE SALIDA

2

FUNCIONES
CON TIPADO
DEFINIDO



```
1 function sumar(int $numero1,int $numero2):int
2 {
3     return $numero1+$numero2;
4 }
5 echo sumar(1,2);
```

2

FUNCIONES CON TIPADO DEFINIDO



```
1 declare(strict_types=1);  
2 function sumar(int ... $numeros):int{  
3     return array_sum($numeros);  
4 }  
5 echo sumar(1,2,3,4,5);
```



El ámbito de una variable es el contexto dentro del que la variable está definida.



ÁMBITO DE VARIABLE



VARIABLES LOCALES




VARIABLES GLOBALES



VARIABLES ESTÁTICAS



VARIABLES SUPERGLOBALES



```
1 $varLocal = "Hola<br>";  
2 function miFuncion()  
3 {  
4     $varLocal = "Hola desde función <br>";  
5 }  
6 miFuncion();  
7 echo $varLocal;
```



```
1  function test()
2  {
3      static $a = 0;
4      echo $a++;
5  }
6  test();
7  test();
8  test();
```

4

VARIABLES ESTÁTICAS

Valor: 0

Valor: 1

Valor: 2



```
1 $varGlobal = "Hola <br>";  
2 echo $varGlobal;  
3 function miFuncion2()  
4 {  
5     global $varGlobal;  
6     $varGlobal = "Hola desde función <br>";  
7 }  
8 miFuncion2();  
9 echo $varGlobal;
```



```
1 $varGlobal = "Hola <br>";  
2 echo $varGlobal;  
3 function miFuncion2()  
4 {  
5     $GLOBALS['varGlobal'] = "Hola desde función <br>";  
6 }  
7 miFuncion2();  
8 echo $varGlobal;
```



FORMULARIOS

≡ \$_POST

≡ \$_FILE

≡ \$_GET

≡ \$_REQUEST



SERVER

≡ \$_SERVER



SESIÓN Y COOKIES

≡ \$_SESSION

≡ \$_COOKIE



ENTORNO

≡ \$_ENV

5

VARIABLES
SUPERGLOBA
LES

Desarrollo de Sistemas Web

Tec. Superior en Análisis Funcional de Sistemas