|  |  |  |
| --- | --- | --- |
| C:\Users\Admin\Pictures\JATA KPM.png  **BAHAGIAN PENDIDIKAN TEKNIK DAN VOKASIONAL**  **KEMENTERIAN PENDIDIKAN MALAYSIA**  **ARAS 5 & 6, BLOK E14, KOMPLEKS E,**  **PUSAT PENTADBIRAN KERAJAAN PERSEKUTUAN**  **KERTAS PENERANGAN**  ***(INFORMATION SHEET)*** | | |
| **KOD DAN NAMA NOSS** | IT-010-3: 2016 APPLICATION MODULE DEVELOPMENT | |
| **KOD DAN NAMA CU / WA** | C02 - APPLICATION MODULE DEVELOPMENT | |
| **NAMA PROGRAM** | TEKNOLOGI SISTEM PENGURUSAN PANGKALAN DATA DAN APLIKASI WEB | |
| **TAHAP DAN SEMESTER** | 3 (SEMESTER 2) | |
| **KOD DAN TAJUK KURSUS** | KPD 2023 WEB PROGRAMMING | |
| **NO.D5N TAJUK KOMPETENSI** | K4 WRITE MODULE CODE | |
| **NO. KOD KSKV** | KPD2023 / P(9/13) | Muka Surat : 1 Drp : 14 |
| **NO. KOD NOSS** | IT-010-3:2016-C02/ P(9/13) |

**TAJUK/***TITLE***: WRITE MODULE CODE**

**TUJUAN/***PURPOSE***:**

Kertas penerangan ini adalah bertujuan menerangkan mengenai :

* Array
* Numeric Array
* Associative array
* Control Structure
* If Else
* Elseif
* while Loop
* Foreach Loop
* Switch
* Break
* Continue
* Include and Require

**4.8 ARRAY**

An array is a data structure that stores one or more similar type of values in a single value. For example if you want to store 100 numbers then instead of defining 100 variables its easy to define an array of 100 length.

There are three different kind of arrays and each array value is accessed using an ID c which is called array index.

* **Numeric array** − An array with a numeric index. Values are stored and accessed in linear fashion.
* **Associative array** − An array with strings as index. This stores element values in association with key values rather than in a strict linear index order.
* **Multidimensional array** − An array containing one or more arrays and values are accessed using multiple indices

**Numeric Array**

These arrays can store numbers, strings and any object but their index will be represented by numbers. By default array index starts from zero.

Example

Following is the example showing how to create and access numeric arrays.

Here we have used **array()** function to create array. This function is explained in function reference.

<html>

<body>

<?php

/\* First method to create array. \*/

$numbers = array( 1, 2, 3, 4, 5);

foreach( $numbers as $value ) {

echo "Value is $value <br />";

}

/\* Second method to create array. \*/

$numbers[0] = "one";

$numbers[1] = "two";

$numbers[2] = "three";

$numbers[3] = "four";

$numbers[4] = "five";

foreach( $numbers as $value ) {

echo "Value is $value <br />";

}

?>

</body>

</html>

This will produce the following result −

Value is 1

Value is 2

Value is 3

Value is 4

Value is 5

Value is one

Value is two

Value is three

Value is four

Value is five

**Associative Arrays**

The associative arrays are very similar to numeric arrays in term of functionality but they are different in terms of their index. Associative array will have their index as string so that you can establish a strong association between key and values.

To store the salaries of employees in an array, a numerically indexed array would not be the best choice. Instead, we could use the employees names as the keys in our associative array, and the value would be their respective salary.

**NOTE** − Don't keep associative array inside double quote while printing otherwise it would not return any value.

Example

<html>

<body>

<?php

/\* First method to associate create array. \*/

$salaries = array("mohammad" => 2000, "qadir" => 1000, "zara" => 500);

echo "Salary of mohammad is ". $salaries['mohammad'] . "<br />";

echo "Salary of qadir is ". $salaries['qadir']. "<br />";

echo "Salary of zara is ". $salaries['zara']. "<br />";

/\* Second method to create array. \*/

$salaries['mohammad'] = "high";

$salaries['qadir'] = "medium";

$salaries['zara'] = "low";

echo "Salary of mohammad is ". $salaries['mohammad'] . "<br />";

echo "Salary of qadir is ". $salaries['qadir']. "<br />";

echo "Salary of zara is ". $salaries['zara']. "<br />";

?>

</body>

</html>

This will produce the following result −

Salary of mohammad is 2000

Salary of qadir is 1000

Salary of zara is 500

Salary of mohammad is high

Salary of qadir is medium

Salary of zara is low

**4.7 CARRY OUT SELECTION STATEMENT.**

Mendefinisi pernyataan pilihan berdasarkan konsep standard pengaturcaran web.

* Struktur kawalan berfungsi untuk menentukan aliran operasi yang akan dilakukan dalam atur cara.
* Tiga jenis struktur kawalan :
  + Struktur jujukan
  + Struktur pilihan
  + Struktur gelung

Struktur jujukan

Struktur jujukan sesuai untuk selesaikan masalah yang mudah dan tidak berulang.

Struktur pilihan

* Penggunaan struktur pilihan membenarkan satu atau lebih arahan disediakan.
* Ungkapan syarat memainkan peranan penting kerana pilihan akan dibuat berdasarkan kepada syarat yang diberikan.
* Terdapat tiga jenis struktur pilihan :
  + Satu pilihan
  + Dua pilihan
  + Pelbagai pilihan
* Struktur pilihan menggunakan :-

1. pernyataan if

2. pernyataan else

3.pernyataan else if

4. pernyataan switch

Menulis pernyataan if…else berdasarkan peraturan standard pengaturcaran web.

**4.7.1 Pernyataan if**

Pernyataan if digunakan bagi kes yang mempunyai satu pilihan sahaja iaitu samaada dilaksanakan ataupun tidak.

Format pernyataan if

if (ungkapan syarat)

Kod dilaksanakan jika benar;

Carta alir pernyataan if

Syarat

Blok 1

Ya

Tidak

Pernyataan PHP boleh terdiri daripada satu pernyataan sahaja atau satu blok pernyataan PHP. Blok pernyataan mesti diletakkan di dalam { }.

Contoh

if ($markah >80)

{ echo “<b>tahniah</b><br>”;

echo “Anda dapat gred A <br>”;}

Markah

> 80 ?

Gred A

Ya

Tidak

**4.7.2 Pernyataan else**

* Pernyataan else merupakan pernyataan yang tidak boleh berdiri sendiri.
* Pernyataan ini mesti digunakan bersama dengan pernyataan if.
* Pernyataan else digunakan untuk kes yang menyediakan dua pilihan , iaitu satu pilihan akan dilaksanakan jika syarat benar dan pilihan lain ialah palsu.

Format pernyataan else

if (syarat)

Kod dilaksanakan jika syarat if benar;

else

Kod dilaksanakan jika syarat tidak dipenuhi;

Carta alir pernyataan if else

Syarat

Blok 1

Arahan seterusnya

Ya

Tidak

Blok 2

Menulis pernyataan if…else if…else berdasarkan peraturan standard pengaturcaran web.

**4.7.3 Pernyataan else if**

Pernyataan else if digunakan untuk kes yang menyediakan lebih daripada dua pilihan.

Syarat akan diuji dari atas ke bawah.

Format pernyataan else if

if (syarat)

Kod dilaksanakan jika syarat if benar;

else if (syarat)

Kod dilaksanakan jika syarat else if benar;

……

else

Kod dilaksanakan jika semua syarat tidak dipenuhi ;

Carta alir if else if

Syarat

## 4.7.4 while Loop

The while loop executes a block of code as long as the specified condition is true.

### **Syntax**

while (*condition is true*) {  
*code to be executed*;  
}

The example below first sets a variable $x to 1 ($x = 1). Then, the while loop will continue to run as long as $x is less than, or equal to 5 ($x <= 5). $x will increase by 1 each time the loop runs ($x++):

### Example

<?php   
$x = 1;   
  
while($x <= 5) {  
    echo "The number is: $x <br>";  
    $x++;  
}   
?>

## 4.7.5 PHP Foreach Loop

The foreach loop works only on arrays, and is used to loop through each key/value pair in an array.

### **Syntax**

foreach ($*array*as$*value*) {  
    *code to be executed;*  
}

For every loop iteration, the value of the current array element is assigned to $value and the array pointer is moved by one, until it reaches the last array element.

The following example demonstrates a loop that will output the values of the given array ($colors):

### **Example**

<?php   
$colors = array("red", "green", "blue", "yellow");   
  
foreach ($colors as $value) {  
    echo "$value <br>";  
}  
?>

**4.7.6 PHP include and require Statements**

It is possible to insert the content of one PHP file into another PHP file (before the server executes it), with the include or require statement.

**The include and require statements are identical, except upon failure:**

* require will produce a fatal error (E\_COMPILE\_ERROR) and stop the script
* include will only produce a warning (E\_WARNING) and the script will continue

So, if you want the execution to go on and show users the output, even if the include file is missing, use the include statement. Otherwise, in case of FrameWork, CMS, or a complex PHP application coding, always use the require statement to include a key file to the flow of execution. This will help avoid compromising your application's security and integrity, just in-case one key file is accidentally missing.

Including files saves a lot of work. This means that you can create a standard header, footer, or menu file for all your web pages. Then, when the header needs to be updated, you can only update the header include file.

### **Syntax**

include '*filename*';  
  
or  
  
require '*filename*';

## PHP include Examples

### **Example 1**

Assume we have a standard footer file called "footer.php", that looks like this:

<?php  
echo "<p>Copyright &copy; 1999-" . date("Y") . " W3Schools.com</p>";  
?>

To include the footer file in a page, use the include statement:

### **Example**

<html>  
<body>  
  
<h1>Welcome to my home page!</h1>  
<p>Some text.</p>  
<p>Some more text.</p>  
<?php include 'footer.php';?>  
  
</body>  
</html>

### **Example 2**

Assume we have a standard menu file called "menu.php":

<?php  
echo '<a href="/default.asp">Home</a> -  
<a href="/html/default.asp">HTML Tutorial</a> -  
<a href="/css/default.asp">CSS Tutorial</a> -  
<a href="/js/default.asp">JavaScript Tutorial</a> -  
<a href="default.asp">PHP Tutorial</a>';  
?>

All pages in the Web site should use this menu file. Here is how it can be done (we are using a <div> element so that the menu easily can be styled with CSS later):

### **Example**

<html>  
<body>  
  
<div class="menu">  
<?php include 'menu.php';?>  
</div>  
  
<h1>Welcome to my home page!</h1>  
<p>Some text.</p>  
<p>Some more text.</p>  
  
</body>  
</html>

### **Example 3**

Assume we have a file called "vars.php", with some variables defined:

<?php  
$color='red';  
$car='BMW';  
?>

Then, if we include the "vars.php" file, the variables can be used in the calling file:

### **Example**

<html>  
<body>  
  
<h1>Welcome to my home page!</h1>  
<?php include 'vars.php';  
echo "I have a $color $car.";  
?>  
  
</body>  
</html>

## PHP include vs. require

The require statement is also used to include a file into the PHP code.

However, there is one big difference between include and require; when a file is included with the include statement and PHP cannot find it, the script will continue to execute:

### **Example**

<html>  
<body>  
  
<h1>Welcome to my home page!</h1>  
<?php include 'noFileExists.php';  
echo "I have a $color $car.";  
?>  
  
</body>  
</html>

If we do the same example using the require statement, the echo statement will not be executed because the script execution dies after the require statement returned a fatal error:

### **Example**

<html>  
<body>  
  
<h1>Welcome to my home page!</h1>  
<?php require 'noFileExists.php';  
echo "I have a $color $car.";  
?>  
  
</body>  
</html>

Use require when the file is required by the application.

Use include when the file is not required and application should continue when file is not found.

Menulis pernyataan switch berdasarkan peraturan standard pengaturcaran web.

**Pernyataan Switch**

* Pernyataan switch adalah kaedah alternatif untuk multipilihan.
* untuk mengelakkan kekeliruan dalam multipilihan
* Pernyataan switch ini memiliki kegunaan sama seperti if else.
* Batasan Switch :data yang diperiksa haruslah bertype Integer (int) atau Character (char).

Format pernyataan switch

**switch (ungkapan kawalan)**

**{**

**case pemalar 1 : kod dilaksanakan jika pernyataan = pemalar 1;**

**break;**

**case pemalar 2 : kod dilaksanakan jika pernyataan = pemalar 2;**

**break;**

**case pemalar 3 : kod dilaksanakan jika pernyataan = pemalar 3;**

**break;**

**default : kod dilaksanakan jika pernyataan tidak sama dengan semua pemalar;**

**}**

Anda boleh menggunakan beberapa pernyataan dalam pernyataan switch untuk

memastikan penggunaan kod lebih efisien. Iaitu :-

* Pernyataan break
* Pernyataan default
* Pernyataan continue

**Pernyataan break**

Pernyataan break mengelakkan kod pada pernyataan case yang berikutnya dilaksanakan selepas padanan dilakukan.

Pelaksanaan switch akan tamat apabila pernyataan break dilaksanakan.

Perlaksanaan aturcara seterusnya akan beralih ke pernyataan selepas switch.

**Pernyataan default**

Pernyataan default dilaksanakan jika tiada pernyataan case yang memenuhi syarat. Selalunya diletak sebagai pilihan terakhir dalam pernyataan switch.

**Pernyataan continue**

Pelaksanaan pernyataan ini akan menyebabkan pernyataan-pernyataan selepas pernyataan continue dalam suatu gelung tidak dilaksanakan tetapi kawalan akan terus melaksanakan gelung tersebut

**SOALAN/***QUESTION***:**

1. Jelaskan definisi Numeric Array dan Associative array

2. Senaraikan jenis struktur kawalan.

**RUJUKAN/***REFERENCE* **:**

**Nota KPD 404**

<https://www.w3schools.com/>