Introduction: This document is a cheat sheet for FOODIE Website which is built in PHP.

This guide is not intended to shows the functionality of PHP from the ground up, this project uses raw PHP for writing code.

PHP BASICS	Types of Data

Including PHP in a File
  <?php // place PHP code here ?>
Writing Comments
//
  Denotes comments that only span on one line
 	
  #
  Another way of producing single-line comments
 	
  /*...*/
  Everything between /* and */ is not executed,
Outputting Data
  <?php echo " PHP Cheat Sheet "; ?>
Writing PHP Functions
  function NameOfTheFunction() {
       //place PHP code here
  }
	Integers
	
	Floats
	
	Strings
	
	Boolean values
	
	Arrays
	
	Objects
	Objects store both data and information on how to process it.
	
	Resources
	These are references to functions and resources outside of PHP.
	
	NULL
	A variable that is NULL doesn't have any value.

PHP Functions

- **A function is a block of statements that can be used repeatedly in a program.**

- **A function will not execute automatically when a page loads.**

- **A function will be executed by a call to the function.**

Syntax

```php
function functionName() {
    code to be executed;
}
functionName();
```

Function Arguments

```php
<?php
function familyName($fname, $year) {
    echo "$fname Refsnes. Born in $year <br>";
}

familyName("ASEEM", "2000");
?>
```

Default Argument Value

```php
<?php declare(strict_types=1); // strict requirement function setHeight(int $minheight
= 50) {
    echo "The height is : $minheight <br>";
}
setHeight(350);
?>
```

Returning values

```php
<?php declare(strict_types=1); // strict requirement function sum(int $x, int $y) {
    $z =
    $x +
    $y;
    return
    $z;
}
echo "5 + 10 = " . sum(5, 10) . "<br>";
echo "7 + 13 = " . sum(7, 13) . "<br>"; echo "2 + 4 = " .
sum(2, 4);
?>
```

Name conventions – (Variables)

Customer First name - fname, Last Name - lname, Address - address, Postal code - postal, Username - username, Password - password

Functions:

Register page

- **This function hide error. It's created fname, lname, address, postal code, username, password**

function fnameFunction() {

  if(!isEmpty(" ",document.getElementById("firstname").value)) {

    document.getElementById("errorfname").style.display = "none";

  }

}

- **This function is** validation function for form to check the input is valid. It's created for all input fields.

```
function checkForm() {
        if(isEmpty(" ",document.getElementById("firstname").value)) {
                document.getElementById("errorfname").style.display = "block";
                document.getElementById("firstname").focus();
                document.getElementById("errorfname").innerHTML=(" First name cannot be empty. ");
                return false;
        }
```

}

- **This function is created for after validation form submit using ajax**

function submit_host(){

  if(checkForm() == true){ //check validation

-     "all fields e.g. **created fname, lname, address, postal code, username, passworD**

--

  }

}

Login page

This function hide error

function usernameFunction() {

}

function passwordFunction() {

  if(!isEmpty(" ",document.getElementById("password").value)) {

    document.getElementById("errorpassword").style.display = "none";

  }

}

Form validation

Check error

```
function checkForm() {

        if(isEmpty(" ",document.getElementById("username").value)) {

        }

        if(isEmpty(" ",document.getElementById("password").value)) {

}

}
```

Form Submission after validation

```
function submit_host(){

        if(checkForm() == true){ //check validation

                document.getElementById("accForm").submit(); //submit form

        }

}
```

Account page

This function hide error   created for lname, address, city, postal code, username, password

```
function fnameFunction() {

        if(!isEmpty(" ",document.getElementById("firstname").value)) {

                document.getElementById("errorfname").style.display = "none";

        }

}
```

Validation, error lname, address, city, postal code, username, password, picture

```
-      function checkForm() {

        if(isEmpty(" ",document.getElementById("firstname").value)) {

        }

}
```

Validation form submit

```
function submit_host(){

        if(checkForm() == true){ //check validation

                document.getElementById("accForm").submit(); //submit form

        }

}
```

Buy page

This function hide error product ID, Qty

```
function productcodeFunction() {

}
function qtyFunction() {

}
```

Validation Prodyuct id, qty

```
function checkForm() {

}
```

Form submission after validation

```
function submit_host(){

}
```

Order page

```
//function get orders

function get_orders(){

}
```

HTTP Functions

header()
Sends a raw HTTP header to the browser

headers_list()
A list of response headers ready to send (or already sent)

headers_sent()
Checks if and where the HTTP headers have been sent

Classes

class DBConnection {

        var $connection, $statement, $dbQuery, $dbResult;

// /// //

 PHP and MySQL Connection and Error Specific methods

        function __construct()

        {

        }

Purpose: to call mysql_real_escape_string(), stripping slashes before only if necessary

        function sc_mysql_escape($value) {

                // strip out slashes IF they exist AND magic_quotes is on

        }

        function CloseConnection(){

        }

* Execute query for select

        */

        function SelectQuery($fileName="", $methodName=""){

        }

Execute query for Insert

        function InsertQuery($fileName="", $methodName="")

        }

* Execute query for Update/Delete

        function ExecuteQuery($fileName="", $methodName=""){

        }

* Execute queries for webservice

        function SelectQueryResult($fileName="", $methodName="")}

 Use this method to log the database errors.

        function saveIntoErrorLog($fileName="", $methodName="", $sqlQuery="", $exception="")

} //ends the class over here

MySQL

Purpose: to add records into table

//Parameters:

// $dbObj database connection object

// $table table name

// $data array with field names as keys, and values rep. those field values

// $where MySQL where statement, minus the "WHERE" text at the beginning

function add_record($dbObj, $table, $data){

}

//Purpose: To modify a record

function modify_record($dbObj, $table, $data, $where){

}

//Purpose: To delete a record

function delete_record($dbObj, $table, $where){{}

—----------------

PHP Functions

//Parameters:

// $dbObj database connection object

// $table table name

// $data array with field names as keys, and values rep. those field values

// $where MySQL where statement, minus the "WHERE" text at the beginning

//echo "function.database";exit;

// Purpose: to add records into table

function add_record($dbObj, $table, $data){

}

//**Purpose:**

//   **To modify a record**

```php
function modify_record($dbObj, $table, $data, $where){

}
```

//**Purpose:**

//   **To delete a record**

```php
function delete_record($dbObj, $table, $where){}
```

//**Purpose: to call addslashes(), stripping slashes before only if necessary**

```php
function sc_php_escape($value) {

}
```

MYSQL

```php
class DBConnection {

    var $connection, $statement, $dbQuery, $dbResult;

    // /////////////////////////////////////////////// //
    // PHP and MySQL Connection and Error Specific methods
    // /////////////////////////////////////////////// //

    function __construct(){ }

    //Purpose: to call mysql_real_escape_string(), stripping slashes before only if necessary
    function sc_mysql_escape($value) {}

    function CloseConnection(){     }
```

```
/**

* Execute query for select

*/

function SelectQuery($fileName="", $methodName=""){           }


/**

* Execute query for Insert

*/

function InsertQuery($fileName="", $methodName=""){}


/**

* Execute query for Update/Delete

*/

function ExecuteQuery($fileName="", $methodName="")

{}


/**

* Execute queries for webservice

*/

function SelectQueryResult($fileName="", $methodName="")

{}
```