 <?php

//BASICS :

$nom = "John"; var\_dump($nom); // string(4) "John"

$age = 30; echo gettype($age); // integer

echo strlen("Bonjour"); // 7

is\_int($age);//checking if the variable is integer

intval($age);//transforming from any type to int

$position = strpos($texte, "tout"); // $position vaut 8

$partie = substr($texte, 3); // $partie vaut "jour"

$majuscules = strtoupper($texte); // $majuscules vaut "BONJOUR"

$nouveau\_texte = str\_replace("Bonjour", "Salut", $texte);// $nouveau\_texte vaut "Salut tout le monde"

$partie = trim($texte); // $partie vaut « Bonjour"

$fruits = array("pomme", "banane", "orange");

$chaine\_fruits = implode(", ", $fruits);

echo $chaine\_fruits; //pomme, banane, orange

$result\_sqrt = sqrt(16); // $result\_sqrt = 4

$result\_abs = abs(-25); // $result\_abs = 25

$result\_pow = pow(3, 4); // $result\_pow = 81

$result\_round = round(5.65); // $result\_round = 6

$result\_floor = floor(5.65); // $result\_floor = 5

$result\_ceil = ceil(5.65); // $result\_ceil = 6

$result\_pi = pi(); // $result\_pi = 3.1415926535898...

$result\_min = min([4, 9, -2, 17, 0]); // $result\_min = -2

$result\_max = max([4, 9, -2, 17, 0]); // $result\_max = 17

$result\_rand\_range = rand(1, 100); // Par exemple : 42

define("PI", 3.14);//constant variable

echo PI;//3.14

echo defined("PI"); // true

//-------------------------------------------------------------------------------------------------------------

//ASSOC ARRAY :

$personne = ["nom" => "Jean", "age" => 30,"ville" => "Paris"];

echo $personne["age"];//assoc array return 30

$etudiants = array(

    array("nom" => "Alice", "age" => 20),

    array("nom" => "Bob", "age" => 22),

    array("nom" => "Claire", "age" => 21)

    );//Multidimensional Array

echo $etudiants[1]["nom"]; // Bob

$fruits = array("Apple", "Banana", "Orange");

print\_r($fruits);//Array([0] => Apple[1] => Banana[2] => Orange)

//-------------------------------------------------------------------------------------------------------------

//ARRAY :

$array = ["apple", "banana", "orange", "apple"];

$count = count($array); // 4

array\_push($array, "grape", "melon"); // $array = ["apple", "banana", "orange", "apple", "grape", "melon"]

$last = array\_pop($array); // pop last element

$first = array\_shift($array); // pop first element

array\_unshift($array, "kiwi"); // $array = ["kiwi", "banana", "orange", "apple", "grape"]

$sliced = array\_slice($array, 1, 3); // $sliced = ["banana", "orange", "apple"]

$more\_fruits = ["mango", "pineapple"];

$merged = array\_merge($array, $more\_fruits); // Combine both arrays

$keys = array\_keys($array); // [0, 1, 2, 3, 4]

$values = array\_values($array); // ["kiwi", "banana", "orange", "apple", "grape"]

$search\_index = array\_search("apple", $array); // e.g., 3

$exists = in\_array("orange", $array); // true

$unique = array\_unique($array); //remove duplicates

sort($sort\_array); // ["apple", "banana", "grape", "kiwi", "orange"]

rsort($rsort\_array); // ["orange", "kiwi", "grape", "banana", "apple"]

asort($asort\_array); // Same values sorted, keys preserved

$assoc\_array = ["b" => "banana","a" => "apple","c" => "cherry"];

ksort($assoc\_array); // Sorted by keys: a, b, c

//--------------------------------------------------------------------------------------------------------------

$x = 10;$y = 20;

function addition1() {

$GLOBALS['z'] = $GLOBALS['x'] + $GLOBALS['y'];

}

addition1();

echo $z; // Affiche la valeur de la variable z (30)

function addition2() {

global $x, $y;

return $x + $y;

}

echo $\_SERVER['SERVER\_NAME']; // display server name

echo $\_SERVER['REQUEST\_METHOD']; // display requested method

echo $\_SERVER['REMOTE\_ADDR']; // display IP address of the client

echo $\_SERVER['HTTP\_USER\_AGENT']; //display user navigator

//---------------------------------------------------------------------------------------------------------------

//DATE :

$date\_actuelle = date("d/m/Y H:i:s");

echo "La date actuelle est: $date\_actuelle"; // La date actuelle est : 15/02/2024 11:02:55

//strtotime() usage

$timestamp = strtotime("+3 days");

echo date("Y-m-d", $timestamp); // 2024-06-21

// modify() usage

$date = new DateTime("2024-06-18");

$date->modify("+3 days");

echo $date->format("Y-m-d"); // 2024-06-21

//diff() usage with dateTimeZone

$date1 = new DateTime('2024-02-14 00:00:00', new DateTimeZone('Europe/Paris'));

$date2 = new DateTime('2024-02-14 00:00:00', new DateTimeZone('Asia/Beirut'));

$difference = $date1->diff($date2);

echo $difference->format('%R%a jours %H heures %i minutes');//+0 jours 01 heures 0 minutes

date\_default\_timezone\_set('Asia/Beirut');//set time zone

$date = new DateTime('now');

echo $date->format('Y-m-d H:i:s');

$timestamp = time(); // Timestamp actuel in seconds from 1970-01-01 00:00:00 UTC

echo $timestamp; //1707995197

echo date('Y-m-d H:i:s', $timestamp); // 2024-02-15 12:06:37

//---------------------------------------------------------------------------------------------------------------

//CSV , TXT ,JSON :

// Mode Description

// "r"  Lire uniquement. Fichier doit exister.

// "w"  Écrire uniquement. Écrase ou crée le fichier.

// "a"  Ajouter uniquement. Crée s'il n'existe pas.

// "x"  Crée uniquement. Erreur si déjà existant.

// "r+" Lire + écrire sans effacer.

// "w+" Lire + écrire. Écrase le contenu.

// "a+" Lire + ajouter.

// "x+" Lire + écrire. Crée uniquement si n'existe pas.

$filename = "fichier.txt";

if(file\_exists($filename))echo "Le fichier existe.";//checking if the file exists

else echo "Le fichier n'existe pas.";

$file = fopen("nouveau\_fichier.txt", "r") or die("Impossible de créer le fichier!");

echo fread($file, filesize("fichier.txt"));//return indenxed array;

fclose($file);

// Read file line by line until end of file

while(!feof($file)) {

    $line = fgets($file); // Get one line

    echo $line . "<br>";  // Print the line with a line break for browser output

}

$file = fopen("data.csv", "r") or die("Impossible d’ouvrir le   fichier!");

$data = [];

while(!feof($file)){//checking if end of file

    $line = fgets($file);

    if(empty($line)) continue;//passing the empty line

        $data[] = explode(",", trim($line));

}

//using fgetcsv

$file = fopen("data.csv", "r") or die("Impossible d’ouvrir lefichier!");

$data = [];

while(($csvRow = fgetcsv($file) !== false)){

$data[] = $csvRow;

}

print\_r($data);

$filename = "data.json";

$file = fopen($filename, "r") or die("Impossible d’ouvrir le fichier!");

$fileContent = fread($file, filesize($filename));

$data = json\_decode($fileContent,true);

print\_r($data);

echo $data['name']; // we access data like this

fclose($file);

function lire\_fichier($nom\_fichier) {

    $arrayAssoc = [];

    $file = fopen($nom\_fichier, "r") or die("Impossible d'ouvrir le fichier");

    $keys = fgetcsv($file);//getting first line

    while(($row = fgetcsv($file)) !== false){

        $arrayAssoc[] = array\_combine($keys, $row);//combining each key with his value

    }

    fclose($file);

    return $arrayAssoc;

}

$file = fopen("fichier.txt", "w") or die("Impossible...");

fwrite($file, "Contenu à écrire dans le fichier\n");//same method for 'w' and 'a'

fclose($file);

$data = array(

    array('Nom', 'Âge', 'Ville'),

    array('Dupont', 30, 'Paris'),

    array('Durand', 25, 'Lyon')

);

$file = fopen('donnees.csv', 'w') or die("Impossible...");

for($i = 0; $i < count($data); $i++) {

    fputcsv($file, $data[$i]);//adding each mini array consicutively

}

fclose($file);

//transforming from assoc array to json String and add it to the file

$data = array('nom' => 'Dupont','age' => 30,'ville' => 'Paris');

$json\_data = json\_encode($data);//transforming from assoc array to json String

$file = fopen('donnees.json', 'w') or die("Impossible...");

fwrite($file, $json\_data);

fclose($file);

//-------------------------------------------------------------------------------------------

//REGULAR EXPRESSION (REGEX) :

$passwordPattern = "/^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\d).{8,}$/";

$phonePattern = "/^\(?\d{3}\)?[-.\s]?\d{3}[-.\s]?\d{4}$/";

$urlPattern = "/\bhttps?:\/\/[^\s]+/i";

$datePattern = "/^\d{4}-\d{2}-\d{2}$/";

$passwordPattern = "/^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\d).{8,}$/";

$onlyNumber = "/\d+/";

//exemple

$phone = "(123) 456-7890";

echo preg\_match($phonePattern, $phone) ? "Valid phone\n" : "Invalid phone\n";

$text = "fff123fff";

echo preg\_replace("/\d/","#",$text);//fff###fff

//-------------------------------------------------------------------------------------------

// PDO :

$servername = "127.0.0.1";

$username = "root";

$password ="";

$dbname = "nfa008";

$conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);

//connection to the database

$conn->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$nom = "Dupont";

$prenom = "Jean";

$sql = "SELECT \* FROM auteur WHERE nom = :nom and prenom = :prenom";

$stmt = $conn->prepare($sql);

$stmt->bindParam(':nom', $nom, PDO::PARAM\_STR);

$stmt->bindParam(':prenom', $prenom, PDO::PARAM\_STR);

// • PDO::PARAM\_STR

// • PDO::PARAM\_INT

// • PDO::PARAM\_BOOL

// • PDO::PARAM\_NULL

// • PDO::PARAM\_LOB

$stmt->execute();

$data = $stmt->fetchAll(PDO::FETCH\_ASSOC);

if(count($data) > 0){

echo json\_encode($data);

}

$sql = "INSERT INTO livre (isbn, titre, prix, auteur\_id) VALUES (:isbn, :titre, :prix, :auteurID)";

$stmt = $conn->prepare($sql);

$stmt->bindParam(':isbn', $isbn, PDO::PARAM\_STR);

$stmt->bindParam(':titre', $titre, PDO::PARAM\_STR);

$stmt->bindParam(':prix', $prix, PDO::PARAM\_INT);

$stmt->bindParam(':auteurID', $auteurID, PDO::PARAM\_INT);

$stmt->execute();

$rowsAffected = $stmt->rowCount();

if ($rowsAffected !== false) {

echo "Nouvel enregistrement créé avec succès - Nb de ligne(s) ajoutée(s): $rowsAffected";

}

$stmt = null;

$conn = null;

//MySqli

$conn = new mysqli("localhost","root","","dbName");

$sql = "INSERT INTO auteur (id, nom, prenom) VALUES (NULL, ?, ?)";

$stmt = $conn->prepare($sql);

if (!$stmt) {

die("Error preparing statement: " . $conn->error);

}

$stmt->bind\_param("ss", $nom, $prenom);

// • s: string

// • i: integer

// • d: double

// • b: binary

if (!$stmt->execute()) {

die("Error executing query: " . $stmt->error);

}

$last\_id = $conn->insert\_id;

$affectedRows = $stmt->affected\_rows;

echo "Nouvel enregistrement créé avec succès - ID: $last\_id - Affected Rows: $affectedRows";

//-------------------------------------------------------------------------------------------

//download file :

$data = "Ceci est une ligne de texte à écrire dans le fichier.\n";

$filename = "example.txt";

$file = fopen($filename, "w");

fwrite($file, $data);

fclose($file);

header('Content-Type: application/octet-stream');

header('Content-Disposition: attachment; filename="' . basename($filename) . '"');

header('Content-Length: ' . filesize($filename));

readfile($filename);

unlink($filename);

exit;

//---------------------------------------------------------------------------------------------

setcookie("nom\_utilisateur", "John", time() + 3600, "/");//set cookies

setcookie("nom\_utilisateur", "", time() - 3600, "/");//remove cookies

$name = htmlspecialchars($name, ENT\_QUOTES, 'UTF-8');

$email = filter\_var(trim($\_POST['email']), FILTER\_SANITIZE\_EMAIL);

$age = filter\_var(trim($\_POST['age']), FILTER\_SANITIZE\_NUMBER\_INT);

$price = filter\_var(trim($\_POST['price']), FILTER\_SANITIZE\_NUMBER\_FLOAT, FILTER\_FLAG\_ALLOW\_FRACTION);

$website = filter\_var(trim($\_POST['website']), FILTER\_SANITIZE\_URL);

$fileName    = $file['name'];      // "document.pdf"

$fileTmpName = $file['tmp\_name'];  // "/tmp/phpA1B2C3" (for example)

$fileSize    = $file['size'];      // 512000

$fileType    = $file['type'];      // "application/pdf"

$fileError   = $file['error'];     // 0

//5 \* 1024 \* 1024 :for comparaison avec $fileSize

$uploadDir = 'uploads';

if (!is\_dir($uploadDir)) {

    mkdir($uploadDir, 0755, true);//creating the folder if not exists

}

$fileExt = pathinfo($fileName, PATHINFO\_EXTENSION);

$newFileName = uniqid('CV\_', true) . '.' . $fileExt;

move\_uploaded\_file($fileTmpName, "uploads/$newFileName");

//-----------------------------------------------------------------------------------------------------------

class Person{

    // Properties

    public $name; // Accès partout

    private $species; // Accès uniquement dans la classe Animal

    protected $color; // Accès dans Animal et ses classes dérivées

    public $firstName;

    public $lastName;

    // Constructor

    public function \_\_construct($firstName, $lastName){

        $this->firstName = $firstName;

        $this->lastName = $lastName;

    }

    // Functions

    public function getFulleName(){

    return $this->firstName . " " . $this->lastName;

    }

    public function setFirstName($firstName){

        $this->firstName = $firstName;

        return "success";

    }

}

$pers1 = new Person("John", "Doe");

echo $pers1->getFulleName();

//still not done