



Code 1 – Hello World

```
<html>
<head>
<title> Exemple PHP </title>
</head>
<body>
<?php echo "Bonjour tout le monde" ; ?>
</body>
</html>
```

Exercice 1

Hello World

Essayez et déployez l'exemple Hello World ! Placez-le dans le répertoire `home.iut/www`. Pour tester, placez-vous dans votre dossier `www` et lancez le serveur web embarqué de PHP : `php -S localhost :8000` puis pointez votre navigateur sur : `http://localhost:8000`. Vous pouvez également lancer le serveur de PHP avec la commande : `commande php -S 0.0.0.0 :8000` si vous voulez que votre serveur soit visible sur le réseau local (depuis un autre poste). On peut aussi utiliser le serveur Web embarqué de symfony en créant au préalable un projet symfony, puis en se plaçant dans le répertoire de ce projet : `php bin/console server :start --docroot= /www` (qu'on peut ensuite arrêter par `php bin/console server :stop`) Votre fichier peut contenir exclusivement du PHP mais il y a généralement un mélange de code HTML/PHP dans les fichiers PHP ... Améliorez l'affichage de votre Hello World.

Code 2 – phpinfo

```
<?php phpinfo (); ?>
```

Exercice 2

infos PHP

Essayez un exemple `info.php` qui contient seulement la ligne ci-dessus. Etudier les résultats obtenus. Observez en particulier la version actuelle de PHP et les variables listées en fin de page.

Exercice 3

infos sur le client

A l'aide de certaines valeurs fournies par le tableau associatif `$_SERVER`, affichez en PHP l'adresse IP et le nom du navigateur du client. Consultez la documentation de PHP : <http://php.net/manual/fr/reserved.variables.server.php> à ce sujet et profitez en pour observez un peu cette documentation. Comment afficher le nom courant du Navigateur? Vous pouvez ajouter les 2 lignes ci-dessus pour vous assurer que les erreurs PHP seront bien reportées (pas les parse errors ...)

Exercice 4

Formulaire

Concevez à présent un petit formulaire HTML posant une dizaine de questions de culture générale style "quizz" Votre formulaire contiendra au moins des checkbox, une liste déroulante, un radio et un champ de texte. Eventuellement des nouveaux champs html5 (email, tel, number, etc.).
Ecrire un script "reponse.php" qui va simplement afficher les variables PHP créées automatiquement lors de la soumission du formulaire.

Exercice 5

Reponse au Formulaire en PHP

Complétez le script "reponse.php" en disant à l'internaute si ses réponses son exactes. Déployez le et testez-le sur votre serveur. Vérifiez bien si les champs du formulaire :

1. existent
2. sont non-vides
3. ont la valeur attendue

Vérifiez tous les cas et rendez robuste votre script PHP. Que doit-on modifier si on met POST dans la méthode de soumission du formulaire au lieu de GET? Y a-t-il une manière de traiter simultanément les 2 types de requêtes? Voir ci-dessous l'utilisation de l'opérateur coalescent de PHP7 par exemple.

Code 3 – Opérateur Coalescent PHP 7

```
// Operateur coalescent PHP >= 7
$username = $_GET['user'] ?? 'nobody';
// Ceci equivaut a:
$username = isset($_GET['user']) ? $_GET['user'] : 'nobody';

// On peut chainer les appels:
$username = $_GET['user'] ?? $_POST['user'] ?? 'nobody';
// Ceci retourne la premiere valeur definie parmi
// $_GET['user'], $_POST['user'] et 'nobody'.
```

Exercice 6

Votre propre Quizz

Proposez maintenant votre propre quizz comportant au moins une dizaine de questions. Améliorez aussi votre formulaire html en effectuant les vérifications sur les champs de saisie à l'aide notamment de l'attribut *pattern* des champs de formulaires en HTML5 et rendez certains champs obligatoires à l'aide l'attribut *required* Ecrire la réponse en testant tous les cas possibles, calculez éventuellement un score global.

Pour améliorer la gestion de différents types de questions et de réponses, on peut utiliser une approche un peu plus méthodique avec des fonctions. On peut par exemple écrire des fonctions spécifiques aux types de widgets qu'on souhaite afficher et puis des handlers pour afficher les formulaires HTML correspondants et y répondre. Voir par exemple le code ci-dessous et ajoutez-y par exemple un handler pour la gestion des checkbox.

Code 4 – Quizz avec fonctions

```

<?php
$questions = [
    array(
        "name" => "ultime",
        "type" => "text",
        "text" => "Quelle est la reponse a la question ultime de la vie ?",
        "answer" => "42",
        "score" => 1
    ),
    array(
        "name" => "cheval",
        "type" => "radio",
        "text" => "Quelle est la couleur du cheval blanc d'Henri IV?",
        "choices" => [
            array(
                "text" => "Bleu",
                "value" => "bleu"),
            array(
                "text" => "Blanc",
                "value" => "blanc"),
            array(
                "text" => "Rouge",
                "value" => "rouge"),
        ],
        "answer" => "blanc",
        "score" => 2
    )
];
$question_total = 0; $question_correct = 0;
$score_total = 0; $score_correct = 0;

function question_text($q) {
    echo ($q["text"] . "<br><input type='text' name='{$q[name]}'><br>");
}

function answer_text($q, $v) {
    global $question_correct, $score_total, $score_correct;
    $score_total += $q["score"];
    if (is_null($v)) return;
    if ($q["answer"] == $v) {
        $question_correct += 1;
        $score_correct += $q["score"];
    }
}

```

```

    }
}

function question_radio($q) {
    $html = $q["text"] . "<br>";
    $i = 0;
    foreach ($q["choices"] as $c) {
        $i += 1;
        $html .= "<input type='radio' name='$q[name]'
                value='$c[value]' id='$q[name]-$i'>";
        $html .= "<label for='$q[name]-$i'>$c[text]</label>";
    }
    echo $html;
}

function answer_radio($q, $v) {
    global $question_correct, $score_total, $score_correct;
    $score_total += $q["score"];
    if (is_null($v)) return;
    if ($q["answer"] == $v) {
        $question_correct += 1;
        $score_correct += $q["score"];
    }
}

$question_handlers = array(
    "text" => "question_text",
    "radio" => "question_radio",
);
$answer_handlers = array(
    "text" => "answer_text",
    "radio" => "answer_radio",
);
if ($_SERVER["REQUEST_METHOD"] == "GET") {
    echo "<form method='POST' action='quiz.php'><ol>";
    foreach ($questions as $q) {
        echo "<li>";
        $question_handlers[$q["type"]]( $q);
    }
    echo "</ol><input type='submit' value='Envoyer'></form>";
} else {
    $question_total = 0;
    $question_correct = 0;
    $score_total = 0;
    $score_correct = 0;
    foreach ($questions as $q) {
        $question_total += 1;
        $answer_handlers[$q["type"]]( $q, $_POST[$q["name"]] ?? NULL);
    }
    echo "Reponses correctes: " . $question_correct . "/" . $question_total."<br>";
    echo "Votre score: " . $score_correct . "/" . $score_total . "<br>";
}
}

```