

**Université Côte d'Azur - Institut Universitaire de Technologie**  
**Département Informatique**  
**R301 - Développement WEB - PHP**  
**Jeudi 18 janvier 2024**

|                      |   |                      |   |                      |   |                      |   |                      |   |                      |   |                      |   |                      |   |                      |   |                      |   |
|----------------------|---|----------------------|---|----------------------|---|----------------------|---|----------------------|---|----------------------|---|----------------------|---|----------------------|---|----------------------|---|----------------------|---|
| <input type="text"/> | 0 | <input type="text"/> | 1 | <input type="text"/> | 2 | <input type="text"/> | 3 | <input type="text"/> | 4 | <input type="text"/> | 5 | <input type="text"/> | 6 | <input type="text"/> | 7 | <input type="text"/> | 8 | <input type="text"/> | 9 |
| <input type="text"/> | 0 | <input type="text"/> | 1 | <input type="text"/> | 2 | <input type="text"/> | 3 | <input type="text"/> | 4 | <input type="text"/> | 5 | <input type="text"/> | 6 | <input type="text"/> | 7 | <input type="text"/> | 8 | <input type="text"/> | 9 |
| <input type="text"/> | 0 | <input type="text"/> | 1 | <input type="text"/> | 2 | <input type="text"/> | 3 | <input type="text"/> | 4 | <input type="text"/> | 5 | <input type="text"/> | 6 | <input type="text"/> | 7 | <input type="text"/> | 8 | <input type="text"/> | 9 |
| <input type="text"/> | 0 | <input type="text"/> | 1 | <input type="text"/> | 2 | <input type="text"/> | 3 | <input type="text"/> | 4 | <input type="text"/> | 5 | <input type="text"/> | 6 | <input type="text"/> | 7 | <input type="text"/> | 8 | <input type="text"/> | 9 |
| <input type="text"/> | 0 | <input type="text"/> | 1 | <input type="text"/> | 2 | <input type="text"/> | 3 | <input type="text"/> | 4 | <input type="text"/> | 5 | <input type="text"/> | 6 | <input type="text"/> | 7 | <input type="text"/> | 8 | <input type="text"/> | 9 |
| <input type="text"/> | 0 | <input type="text"/> | 1 | <input type="text"/> | 2 | <input type="text"/> | 3 | <input type="text"/> | 4 | <input type="text"/> | 5 | <input type="text"/> | 6 | <input type="text"/> | 7 | <input type="text"/> | 8 | <input type="text"/> | 9 |
| <input type="text"/> | 0 | <input type="text"/> | 1 | <input type="text"/> | 2 | <input type="text"/> | 3 | <input type="text"/> | 4 | <input type="text"/> | 5 | <input type="text"/> | 6 | <input type="text"/> | 7 | <input type="text"/> | 8 | <input type="text"/> | 9 |
| <input type="text"/> | 0 | <input type="text"/> | 1 | <input type="text"/> | 2 | <input type="text"/> | 3 | <input type="text"/> | 4 | <input type="text"/> | 5 | <input type="text"/> | 6 | <input type="text"/> | 7 | <input type="text"/> | 8 | <input type="text"/> | 9 |

← Veuillez coder votre numéro d'étudiant ci-contre et écrire votre nom dans la case ci-dessous.

Nom et prénom :

.....

.....

Questionnaire à Choix Unique ou Multiple (♣) - Durée ~ 45mn

Document autorisé : une feuille A4 recto-verso.

Calculatrice, Téléphone Portable et Ordinateur non-autorisés.

**Question 1** Soit le code suivant :

```
$filters = array (
    "name" => array( "filter" => FILTER_CALLBACK,
        "flags" => FILTER_FORCE_ARRAY, "options" => "ucwords"),
    "age" => array ( "filter" => FILTER_VALIDATE_INT,
        "options" => array( "min_range" => 18, "max_range" => 120)),
    "email" => FILTER_VALIDATE_EMAIL
);
print_r( filter_input_array( INPUT_POST, $filters));
```

Sachant que la constante « FILTER\_FORCE\_ARRAY » force à retourner un tableau et « ucwords » exige une majuscule sur le premier mot d'une chaîne. Sélectionnez le cas où la réponse sera vraie :

- ☐ array( "name" => "sabine", "age" => 30, "email" => 'sabine@exemple.com')
- ☒ array( "name" => "Peter", "age" => 41, "email" => 'peter@exemple.com')
- ☐ array( "name" => "John", "age" => 15, "email" => 'john@exemple.com')
- ☐ array( "name" => "Daniel", "age" => 66, "email" => 'daniel@xz\*y.com')

**Question 2** Soit le code PHP suivant :

```
function xyz() {
    static $col;
    $c1 = "lightgray";
    $c2 = "white";

    if ($col == $c1) {
        $col = $c2;
    } else {
        $col = $c1;
    }
    return $col;
}
echo "<table>";
while( $row = $sth->fetch(PDO::FETCH_ASSOC) {
    echo "<tr style=\"background-color: " . xyz() . "\">";
    foreach( $row as $val)
        echo "<td>$val</td>";
    echo "</tr>";
}
echo "</table>";
```

Comment le tableau est-il affiché ?

- ☐ Les cellules du tableau ont une couleur de fond en quinconce.
- ☐ L'entête du tableau est d'une couleur différente de celle des lignes.
- ☒ Les lignes du tableau ont une couleur de fond en alternance.
- ☐ Les colonnes du tableau ont une couleur de fond en alternance.

**Question 3** ♣ Soit \$v = "1"; \$t = 1. Sélectionnez les cas où les opérateurs logiques suivants renvoient TRUE :

- |                                                  |                                                        |
|--------------------------------------------------|--------------------------------------------------------|
| <input type="checkbox"/> (\$t === \$v)           | <input checked="" type="checkbox"/> (\$v !== \$t)      |
| <input checked="" type="checkbox"/> (\$v == \$t) | <input checked="" type="checkbox"/> (\$v.\$t === "11") |

**Question 4** Soit le code PHP suivant :

```
$date = time();
function yyy( $date){
    // l'appel de la fonction date avec "z" calcule le nombre de jour depuis le début de l'année
    $jour = date( "z", $date);
    $num = ( $jour / 7) + 1;
    $num_res = intval( $num);
    return $num_res;
}
```

Que fait la fonction « yyy() » ?

- ☐ Renvoie le nombre de jours depuis le début de l'année.
- ☐ Renvoie le nombre de mois depuis le début de l'année.
- ☒ Renvoie le nombre de semaine depuis le début de l'année.
- ☐ Renvoie le nombre de dimanche depuis le début de l'année.

**Question 5 ♣** Soit un objet défini à partir de la classe myObject :

```
class myObject {
    function __destruct() {
        echo "My Object is destroyed :(";
    }
}

$a = new myObject();
$b = $a;
$c = $b;
$d = clone $a;
unset($a);      // (1)
unset($b);      // (2)
unset($c);      // (3)
unset($d);      // (4)
```

Quand s'affiche la phrase "My Object is destroyed :(" ?

☒ après (4)

☐ après (1)

☐ après (2)

☒ après (3)

**Question 6** On suppose qu'une session a été lancée et que les variables de session « favcolor » et « favanimal » ont été affectées respectivement à « green » et « cat ». Soit le code suivant :

```
<?php
// Start the session
session_start();
?>
<!DOCTYPE html>
<html>
<body>
    <?php
        echo "Variables session : " . $_SESSION["favcolor"] . " " . $_SESSION["favanimal"];
    ?>
```

Que fait le code ci-dessus ?

☐ Les variables \$\_SESSION["favcolor"] et \$\_SESSION["favanimal"] sont vides

☐ La session étant relancée les variables \$\_SESSION["favcolor"] et \$\_SESSION["favanimal"] n'existent plus

☐ La réutilisation de l'instruction « session\_start(); » déclenche une erreur

☒ L'instruction echo affiche : « Variables session : green cat »

**Question 7** Soit le code PHP suivant :

```
$v = 3;
function xyz($a, $b, $x) {
    $x = $a*$a + $b*$b ;
}
function kwz($a, $b, $z) {
    print("1) La valeur de v est $v <br/>") ;           // premier print
    $z = $a + $b - $z;
    $v = 4*$z ;
    print("2) La valeur de v est $v <br/>") ;           // deuxième print
}
xyz(1,2,&$v);
kwz(2,$v,5);
print("3) La valeur de v est $v <br/>") ;           // troisième print
```

Quelle est la valeur de la variable \$v au troisième print ?

☐ 8

☒ 5

☐ 2

☐ 3

**Question 8** Que fait la constante \_\_CLASS\_\_ ?

☐ Donne le chemin d'accès à la classe en cours

☒ Donne le nom de la classe en cours

☐ Donne le nom de l'objet utilisé

☐ Donne le nom du fichier utilisé par la classe en cours

**Question 9** ♣ Soit la variable globale « y » déclarée entre les balises <head> dans index.php dont un extrait est :

```
<head>
    <?php
        $GLOBALS["y"] = 1;
    ?>
</head>
<body>
    <?php
        function xyz() {
            echo "y = " . $GLOBALS["y"];           // (1)
            // ...
            return();
        }
        echo "y = " . $GLOBALS["y"];           // (2)
    ?>
    <!-- ... -->
</body>
```

Sélectionnez la/les réponse(s) exacte(s) ?

☒ La valeur « y » utilisée dans entre les balises <body> de index.php (cf 2) est conservée

☒ La valeur « y » utilisée dans la fonction xyz déclarée dans index.php (cf 1) est conservée

☐ La valeur de « y » a une durée de vie limitée

☐ La valeur de « y » dans le fichier est visible dans un autre fichier que index.php

**Question 10** Soit le fichier « exemple.txt » possédant les droits « -rwxr-xr-x », appartenant à un utilisateur quelconque et se trouvant sous « /files/mytexts/ ». Nous voulons ouvrir le fichier en écriture avec l'instruction suivante :

```
$fp = fopen("/files/mytexts/exemple.txt", "w+");
```

Quelle sera la valeur de la ressource \$fp ?

- ☐ un pointeur sur la dernière ligne du fichier  
☐ TRUE

- ☒ FALSE  
☐ un pointeur sur la première ligne du fichier

**Question 11** Soit le code PHP suivant :

```
$tableau = ["0","1","2","3","4","5","6","7","8","9",
"a","b","c","d","e","f","g","h","i","j","k","l","m","n",
"o","p","q","r","s","t","u","v","w","x","y","z",
"A","B","C","D","E","F","G","H","I","J","K","L","M","N",
"O","P","Q","R","S","T","U","V","W","X","Y","Z"];
$str = array_rand( $tableau, 8);
$new_str = "";
foreach( $str as $i) {
    $new_str .= $tableau[$i];
}
echo "La chaine est : " . $new_str;
```

Que contient la chaîne \$new\_str ?

- ☒ 8 caractères du tableau pris au hasard.  
☐ 8 lettres prises au hasard.  
☐ 8 chiffres pris au hasard.  
☐ Une suite de 8 caractères du tableau (ex : "6","7","8","9", "a","b","c","d") pris au hasard.

**Question 12** Soit le code HTML suivant :

```
<form name="f4" action="./f4.php" method="post" enctype="multipart/form-data">
    <input type="file" name="f4[p] []">
    <!-- ... -->
    <input type="submit" name="f4[submit]" value="Submit">
</form>
```

Quelle est la variable globale de PHP qui permet de récupérer les fichiers téléchargés ?

- ☐ \$\_REQUEST  
☒ \$\_FILES  
☐ \$\_GET  
☐ \$\_POST

**Question 13** Soit le code HTML suivant :

```
<form name="f1" action="./f1.php" method="post">
    <select name="f1[v] []" multiple="multiple" size="4">
        <option value="a">A</option> (1)
        <option value="b">B</option> (2)
        <option value="c">C</option> (3)
        <option value="d">D</option> (4)
    </select>
    <!-- ... -->
    <input type="submit" name="f1[submit]" value="Submit">
</form>
```

Quel est le nombre d'éléments maximal que l'on peut sélectionner simultanément dans le formulaire f1 ?

- ☐ 1  
☐ 2  
☐ 3  
☒ 4

**Question 14** Soit le code HTML suivant :

```
<form name="f3" action="/f3.php" method="post" enctype="multipart/form-data">
    <input type="file" name="f3[p][]">
    <!-- ... -->
    <input type="submit" name="f3[submit]" value="Submit">
</form>
```

Considérons les droits du répertoire tmp/ se trouvant dans la racine du site, quel doivent être ses droits pour qu'en cliquant sur « Submit » le fichier puisse être téléchargé sur le serveur ?

☐ « drwxrwxr-x »

☒ « trwxrwxrwx »

☐ « drwxr-xr-- »

☐ « drwxr-xr-x »

**Question 15** On considère le code PHP suivant :

```
function xxx( string $texte) : string {
    $String = explode( " ", $texte);
    $NewString="";
    for( $i = 0; $i < count( $String); $i++) {
        $NewString .= $String[$i];
    }
    return $NewString;
}
```

Que fait la fonction « xxx() » ?

☐ Renvoie une chaîne de caractère extraite du texte sans le modifier.

☒ Renvoie une chaîne de caractère constituée par les mots du texte sans les espaces.

☐ Renvoie une chaîne de caractère constituée par les mots du texte rangés en sens inverse.

☐ Renvoie une chaîne vide.

**Question 16** Soit le cookie suivant :

```
$cookie_name = "user";
$cookie_value = "Alphonse Karr";
setcookie( $cookie_name, $cookie_value, time() + (86400 * 30), "/");
```

Comment peut on détruire le cookie précédent ?

☐ unset( \$cookie\_name);

☒ setcookie( \$cookie\_name, \$cookie\_value, time(), "/");

☐ unset( \$\_COOKIE["user"]);

☐ \$cookie\_name = FALSE;

**Question 17** Soit le cookie suivant :

```
$cookie_name = "user";
$cookie_value = "Alphonse Karr";
setcookie( $cookie_name, $cookie_value, time() + (86400 * 30), "/");
```

Quelle est la durée de ce cookie ?

☐ 30 jours

☐ 30 heures

☒ Un mois

☐ Un jour

**Question 18 ♣** Soit le code PHP suivant :

```
// Récupère le contenu d'un fichier
$fichier = "xyz.txt";
$cpt = file_get_contents( $fichier);
if( ! isset( $_COOKIE["xyz"])) {
    setcookie( "xyz", "ok", time() + (86400 * 30),"/");
    $cpt++;
    $inf = fopen( $fichier,"w");
    if ( $inf) {
        fputs( $inf, $cpt);
        fclose( $inf);
    }
}
```

Que fait le code précédent ?

- ☒ Génère un cookie "xyz" et lui affecte "ok"
- ☐ Enregistre le cookie "xyz" dans le fichier "xyz.txt"
- ☐ Affiche le contenu du fichier xyz.txt
- ☒ Compte le nombre de visite à une page et l'enregistre dans le fichier "xyz.txt" pour le cookie "xyz"

**Question 19** Soit le code PHP suivant :

```
$user = "root"; $pass = "rt!123"; $host = "localhost";
$dbh = new PDO( "pgsql:host=$host;dbname=staff", $user, $pass);
$sql = 'SELECT * FROM employes;';
$sth = $dbh->query( $sql);
while( $row = $sth->fetch(PDO::FETCH_ASSOC) {
    foreach( $row as $val) {
        echo "$val <br />";
    }
}
```

Que se passe t'il lors de l'exécution de ce code ?

- ☐ On a une erreur de requête SQL.
- ☒ Tout se passe correctement.
- ☐ On a une erreur de connexion au SGBD.
- ☐ On a une erreur de connexion à la base de données « staff ».

**Question 20** ♣ Pour déclarer une session dans le fichier session.php, on écrit le code :

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      // Start the session
      session_start();
      $_SESSION["favcolor"] = "green";
      $_SESSION["favanimal"] = "cat";
      echo "Variables session : " . $_SESSION["favcolor"] . " " . $_SESSION["favanimal"];
    ?>
    <!-- ... -->
```

Sélectionnez les réponses exactes :

- ☐ Les variables \$\_SESSION["favcolor"] et \$\_SESSION["favanimal"] sont vides
- ☒ La session ne démarre pas
- ☐ L'instruction echo affiche : « Variables session : green cat »
- ☒ La session doit être lancée avant les balises <!DOCTYPE html><html>

**Question 21** Soit le code suivant :

```
<form>
  <select>
    <?php
      $sql = "SELECT * FROM mytable;";
      $result = $conn->query( $sql);
      if ( $result === TRUE) {
        if ($result->num_rows > 0) {
          // output data of each row
          while( $row = $result->fetch_assoc()) {
            echo "<option value=\"\" . $row['myfield'] . \">";
            echo $row['myfield'] . "</option>\n";
          }
        }
      }
    ?>
  </select>
</form>
```

Que fait le code précédent si on suppose que la connection \$conn à la base de donnée est établie et que la table « mytable » contient plusieurs colonnes dont « myfield » ?

- ☐ Affiche un sélecteur constituée par les valeurs du champ "myfield"
- ☒ Génère une erreur d'exécution
- ☐ Affiche le contenu de la table "mytable"
- ☐ Affiche un sélecteur constituée par les champs de "mytable"

**Question 22** Soit la fonction `add()` déclarée dans les fichiers `model.php`, `view.php` et `controller.php`. On inclut les 3 fichiers par des `"require_once"` dans un fichier `main.php`.

```
=> model.php :
namespace MVC\Model;
function add(int $a, int $b) { // (1)
}
```

```
=> view.php :
namespace MVC\View;
function add(int $a, int $b) { // (2)
}
```

```
=> controller.php :
function add(int $a, int $b) { // (3)
}
```

```
=> main.php :
require_once './model.php';
require_once './view.php';
require_once './controller.php';
function add(int $a, int $b) { // (4)
}
echo "Happy New Year !";
```

Que se passe-t-il lors de l'exécution de `main.php` ?

- ☐ on affiche : Happy New year !
- ☒ on a une erreur en (4) : Redéclaration de la fonction `add()`
- ☐ on a une erreur en (1) : Erreur MVC est inconnu
- ☐ on a une erreur en (2) : Redéclaration de la fonction `add()`
- ☐ on a une erreur en (3) : Redéclaration de la fonction `add()`

**Question 23** Soit le code suivant :

```
$filters = array (
    "name" => array( "filter" => FILTER_CALLBACK,
                    "flags" => FILTER_FORCE_ARRAY, "options" => "ucwords"),
    "age" => array ( "filter" => FILTER_VALIDATE_INT,
                    "options" => array( "min_range" => 18, "max_range" => 120)),
    "email"      => FILTER_VALIDATE_EMAIL
);
print_r( filter_input_array( INPUT_POST, $filters));
```

Sur quoi s'applique ce filtre ?

- ☐ Les champs « name », « age » et « email » d'une table associative
- ☐ Les propriétés « name », « age » et « email » d'un objet
- ☐ Les variables `$name`, `$age`, `$email`
- ☒ Les champs d'un formulaire soumis avec la méthode « post »

**Question 24** Soit le code HTML suivant :

```
<form name="f2" action="/f2.php" method="post">
  <select name="f1[v][]" multiple="multiple" size="4">
    <option value="a">A</option> (1)
    <option value="b">B</option> (2)
    <option value="c">C</option> (3)
    <option value="d">D</option> (4)
  </select>
  <!-- ... -->
  <input type="submit" name="f2[submit]" value="Submit">
</form>
```

En supposant que l'on a sélectionné 2 et 4 quel est la valeur de `$_REQUEST["f2[v]"]` ?

- ☒ `array("b", "d")`
☐ `"B"`  
☐ `"AD"`
☐ `array("A", "D")`

**Question 25 ♣** Dans le code HTML :

```
<form name="f1" action="exemple.php?v1=10&v2=20" method="get">
```

Comment récupère-t-on les variables `v1` et `v2` passées dans l'« action » du formulaire lors de sa soumission ?

- ☐ `$v1` et `$v2`
☒ `$_REQUEST["v1"]` et `$_REQUEST["v2"]`  
☒ `$_GET["v1"]` et `$_GET["v2"]`
☐ `$_POST["v1"]` et `$_POST["v2"]`

**Question 26 ♣** On considère le code PHP suivant :

```
function xxx( string $texte) : string {
    $String = explode( " ", $texte);
    $NewString = "";
    for( $i = 0; $i < count( $String); $i++) {
        $NewString .= $String[$i];
    }
    return $NewString;
}
```

Que fait l'instruction « `explode()` » ?

- ☒ Crée un tableau formé par les mots de `$texte`.  
☐ Crée une liste formée par les mots de `$texte` séparés par des espaces.  
☐ Supprime les espaces dans `$texte`.  
☒ Crée un tableau formé par les mots de `$texte` séparés par des espaces.

**Question 27** ♣ Soit la définition de classe :

```
class Fruit {
    // Properties
    public $name;
    public $color;
    private $region;
    // Methods
    function set_name( $name) {
        $this->name = $name;
    }
    function set_region( $name) {
        $this->region = $name;
    }
    function get_name() {
        return $this->name;
    }
}
$apple = new Fruit();
```

Sélectionnez la/les syntaxe(s) exacte(s) ?

☐ \$apple->region = "Normandie";

☒ \$apple->name = "Pomme";

☒ \$apple->set\_name("Normandie");

☐ \$apple->get\_name("Apple");

**Question 28** ♣ Soit le fichier « exemple.txt » possédant les droits « -rwxr-x--x » et se trouvant sous le répertoire « /files/mytexts/ ». Dans le fichier « open.php » se trouvant sous le dossier « /work/read/ » lesquelles de ces instructions retourne « TRUE » ?

☐ is\_file( "files/mytexts/exemple.txt")

☒ is\_file( ".././files/mytexts/exemple.txt")

☐ is\_file( "exemple.txt")

☒ is\_file( "/files/mytexts/exemple.txt")

**Question 29** Soit un objet défini par :

```
class Fruit {
    private $name;
    private $color;
    public function __construct( $name, $color) {
        $this->name = $name;
        $this->color = $color;
    }
    protected function intro() {
        echo "Ce fruit est un(e) {$this->name} et sa couleur est {$this->color}.";
    }
}

class Pomme Extends Fruit {
    private $region;
    public function message() {
        echo "Je suis un fruit ou une pomme?";
    }
    public function __construct( $name, $color, $region) {
        $this->region = $region;
        parent::__construct($name, $color);
    }
}

$F = new Fruit( "pomme", "rouge");
$P = new Pomme( "pomme", "rouge", "var");
$P->message();
$P->intro();
```

Indiquez l'instruction qui génère une erreur :

- ☐ \$P->message();
- ☒ \$P->intro();
- ☐ \$P = new Pomme( "pomme", "rouge", "var");
- ☐ \$F = new Fruit( "pomme", "rouge");

**Question 30** Soit le code HTML suivant :

```
<form name="f5" action="/f5.php" method="post" enctype="multipart/form-data">
    <input type="file" name="f5[p] []">
    <!-- ... -->
    <input type="submit" name="f5[submit]" value= "Submit">
</form>
```

Supposons que nous avons sélectionné deux fichiers images img1.jpg et img2.jpg. Comment sont récupérés les noms des fichiers img1.jpg et img2.jpg ?

- ☐ \$\_GET["f5[p]"]["tmp\_name"][0] et \$\_GET["f5[p]"]["tmp\_name"][1]
- ☒ \$\_FILES["f5[p]"]["name"][0] et \$\_FILES["f5[p]"]["name"][1]
- ☐ \$\_FILES["f5[p]"]["tmp\_name"][0] et \$\_FILES["f5[p]"]["tmp\_name"][1]
- ☐ \$\_REQUEST["f5[p]"]["tmp\_name"][0] et \$\_REQUEST["f5[p]"]["tmp\_name"][1]