

6- Quelle est la règle CSS qui permet d'arrondir de 20 pixels les bords de la bordure d'un élément div :

div {border-collapse : 20px ;}

div{border-radius :20px ;}

div{box-border :20px ;}

7- Quelle est la règle CSS qui permet de modifier les dimensions d'une image au double lorsqu'elle est survolée par le pointeur de la souris ?

img{transform : translate(2) ;}

img :hover{transform : translate(2)}

img :hover{transform : scale(2) ;}

8- Quelle est la règle CSS qui permet de mettre la première lettre de chaque mot en majuscules dans toutes les zones de saisie de type text ?

input.text{text-transform : uppercase ;}

input[type="text"]{text-transform : capitalize;}

#text{transform : capitalize ;}

B/

1- On désire créer un document web qui permet d'afficher la longueur de la chaîne de caractères saisie dans la zone de texte (ch) au fur et à mesure qu'on ajoute des caractères.

Exemple :

ch=
0

ch=
1

ch=
4

Compléter le document

```
<!DOCTYPE html>
<html lang="en">
<head>
  <script>
    fonction afficher(){
      //ajouter le nom de la méthode convenable pour compléter la déclaration
      let ch=document.getElementById("ch").-----;
      //ajouter les nom des méthodes convenables pour compléter l'instruction d'affectation
      document.getElementById("long").----- = ch.-----;
    }
  </script>
</head>
<body>
  <!--compléter avec le nom de l'évènement HTML convenable-->
  ch=<input type="text" id="ch" name="ch" -----="afficher()">
  <p id="long"></p>
</body>
</html>
```

2- En JavaScript, que retourne chacune des instructions ci-dessous ?

On rappelle que `ord("A")=65`.

- a. `document.write(parseInt("123.25abc"));` -----
- b. `document.write("bonbon".indexOf("on"));` -----
- c. `document.write(eval("5<3"));` -----
- d. `document.write("Bonjour".charCodeAt(0));` -----
- e. `document.write(String.fromCharCode(66,65,66,65));` -----
- f. `document.write("Bon123jouR***".toUpperCase());` -----

3- En PHP, que retourne chacune des instructions ci-dessous ?

- g. `echo(strrev("bonjour"));` -----
- h. `echo(strcmp("dur","bonjour"));` -----
- i. `echo(substr("bonjour les amis",3,4));` -----
- j. `echo(str_replace("o","ABC","hello world"));` -----
- k. `echo(strops("bonbon","on"));` -----
- l. `echo(checkdate(02,28,2025))` -----

EXERCICE N°2 (POINTS) :

I. Analyse et interprétation du code HTML

1) En se basant sur les figures **Figure 1 (page 10)** et **Figure 2 (page 11)** complétez le tableau suivant, en associant chaque numéro indiqué dans l'extrait du fichier "*gestion_Menaces.html*" avec les informations correspondantes :

Numéro	Attribut/Element
(1)
(2)
(3)
(4)
(5)
(6)

- 2) Quel est le rôle de l'expression `target='_self'` (ligne 1)?

- 3) Quel est le rôle de l'attribut `disabled` (ligne 9) ?

- 4) Quand se déclenche l'appel de la fonction `activer()` (ligne 4) ?

II. Analyse et interprétation du code CSS

- 1) En se basant sur la **Figure 2 (page 11)** et **Figure 3 (page 12)** et pour chacune des questions ci-dessous, mettre une croix (X) dans la case correspondante à la réponse correcte.

Important : Pour chaque question, toute réponse comportant **plus d'une croix** est considérée **erronée**.

- a- Que se passe-t-il lorsqu'un utilisateur survole un bouton ?

<input type="checkbox"/>	Il change de couleur
<input type="checkbox"/>	Il grossit
<input type="checkbox"/>	Il s'active
<input type="checkbox"/>	Il se transforme en lien hypertexte

- b- Quelle propriété **CSS** est utilisée pour ajouter un effet d'ombre aux formulaires ?

<input type="checkbox"/>	border-radius
<input type="checkbox"/>	box-shadow
<input type="checkbox"/>	text-shadow
<input type="checkbox"/>	opacity

- c- Quelle propriété **CSS** est utilisée pour centrer le formulaire sur la page ?

<input type="checkbox"/>	text-align: center;
<input type="checkbox"/>	margin: auto;
<input type="checkbox"/>	align-items: center;
<input type="checkbox"/>	position : relative ;

- d- Quelle est l'effet de la propriété **transform: scale(1.5)** appliquée à un élément ?

<input type="checkbox"/>	L'élément est déplacé de 1.5 fois sa taille initiale vers la droite
<input type="checkbox"/>	L'élément devient 1.5 fois plus grand
<input type="checkbox"/>	L'élément tourne à 1.5 degré
<input type="checkbox"/>	L'élément devient flou

- 2) Donner une règle **CSS** permettant d'appliquer l'animation "**anim1**" aux titres de niveaux **2** avec une durée de 2s, en boucle infinie et en alternance.

```
.....{
    .....: .....;
}
```

III. Analyse et interprétation du code JavaScript

En se basant sur la **Figure 2 (page 11)** on vous demande de :

- 1) **compléter le corps de la fonction activer() qui permet d'activer le champ description.**

```
function activer()
{
.....
.....
}
```

- 2) **Compléter le corps de la fonction verifNom() qui teste la validité du champ "Nom de la menace" en s'assurant qu'il contient uniquement des lettres et des espaces, et doit contenir entre 3 et 50 caractères :**

```

function verifNom() {

    // Récupération de la valeur du champ et suppression des espaces superflus en début et à la fin de la chaîne
    const nomMenace = .....;
        // Vérification de la longueur (entre 3 et 50 caractères)
    if ((nomMenace.length<3 )|| (nomMenace.length > 50))
    {
        return ..... ;
    }
    let i = 0;
    do {
        let c = nomMenace[i];
        if (c != " "){
            // Vérification de la conversion en majuscules du caractère c est égale à sa conversion en minuscules
            if (.....=.....){
                return .....;
            }
        }
        i=i+1;
    } while (i <nomMenace.length);
    return .....;
}

```

Analyse et interprétation du code PHP

En se basant aux figures : **Figure 1 (page 10)**, **Figure 2 (page 11)** et **Figure 4 (page 13)**, compléter le tableau ci dessous pour relier chaque étiquette par le code convenable.

Etiquette	Code
.....	\$res=mysqli_query(\$con,\$req) or die ("erreur : ".mysqli_error(\$cnx));
.....	\$req="select * from menace;";
.....	<pre> while (\$t=mysqli_fetch_array(\$res)) { echo(" <tr> <td>".\$t[1]."</td> <td>".\$t[2]."</td> <td>".\$t[4]."</td> <td>".\$t[3]."</td> </tr>"); } </pre>
.....	\$req="select * from menace where niveau = '\$niveau' ";";

Exercice 3 : (...../ 5 points)

Dans un contexte de gestion environnementale, une organisation souhaite suivre les incidents liés à la pollution (déversements chimiques, émissions illégales, etc.) affectant différents sites naturels. Une base de données relationnelle simplifiée nommée "**BD_Pollution**" est proposée pour enregistrer ces incidents, mais elle présente des anomalies.

Schéma textuel de la base de données :

Incidents (IdInc, Titre, DateInc, Niveau, DescInc, IdCause#)

Sites (IdSite, NomSite, Etat, IdInc#)

Causes (IdCause, TypeCause, DescCause)

Description des champs :

Champ	Description
IdInc	Identifiant unique d'un incident
Titre	Titre de l'incident (ex. : Déversement chimique, Émission toxique)
DateInc	Date de l'incident (type : date)
Niveau	Niveau de gravité de l'incident (Mineur, Modéré, Grave, Critique) Obligatoire
DescInc	Description détaillée de l'incident
IdSite	Identifiant unique d'un site naturel
NomSite	Nom du site (ex. : Forêt de X, Rivière Y)
Etat	État actuel du site (Sain, Pollué, En restauration)
IdCause	Identifiant unique d'une cause
TypeCause	Type de la cause (Industriel, Naturel, Humain)
DescCause	Description de la cause (texte de maximum 100 caractères)

Règles de gestion :

- **R1** : Un incident doit être associé à une cause et à au moins un site affecté.
- **R2** : Un site peut être affecté par plusieurs incidents différents.
- **R3** : Le niveau de gravité d'un incident doit être spécifié avant son enregistrement.

Extrait des tables :

Incidents					
IdInc	Titre	DateInc	Niveau	DescInc	IdCause
1	Déversement chimique	10/04/2025	Grave	Fuite d'usine chimique	1
2	Émission toxique	12/04/2025		Gaz échappé d'une usine	2
3	Incendie forestier	15/04/2025	Modéré	Feu causé par négligence	3

Sites			
IdSite	NomSite	Etat	IdInc
1	Forêt de Tunis	Pollué	1
2	Rivière Medjerda	En restauration	2
3	Lac Ichkeul	Sain	3
4	Forêt de Tunis	Pollué	2

Causes		
IdCause	TypeCause	DescCause
1	Industriel	Usine chimique polluante
2	Industriel	Rejet de gaz non contrôlé
3	Humain	Feu allumé par un campeur

Partie I : Interprétation de la représentation textuelle

Important : Pour chaque représentation textuelle demandée, indiquez la clé primaire et les clés étrangères de chaque table

1. La colonne Niveau de la table Incidents présente une anomalie.

a) Identifier le type de cette anomalie.

b) Parmi les règles **R1** à **R3**, laquelle est non respectée par cette anomalie ?

c) Proposer une requête SQL pour éviter cette anomalie à l'avenir.

2. Examiner la relation entre les tables **Incidents** et **Sites**.

a) Selon le schéma textuel, quel type de relation existe-t-il entre ces deux tables ? Cocher la bonne réponse :

- | | |
|--------------------------|-----------------------|
| <input type="checkbox"/> | Un-à-un |
| <input type="checkbox"/> | Un-à-plusieurs |
| <input type="checkbox"/> | Plusieurs-à-plusieurs |

b) Parmi les règles **R1**, **R2** et **R3**, laquelle est non respectée par cette relation

c) Proposer une correction pour respecter cette règle de gestion en donnant une nouvelle représentation textuelle des tables **Incidents** et **Sites** ainsi de d'autre nouvelle table.

3. Donner la représentation graphique de la base de données à la suite de cette correction

Partie II : Modification de la base de données

Écrire les commandes SQL pour répondre aux questions suivantes :

1. Ajouter à la table **Incidents** un champ **CoutEstime** pour enregistrer le coût estimé des dégâts :
o Type : Réel (12 chiffres, dont 3 après la virgule).

2. Augmenter la taille du champ **DescCause** à 150 caractères.

3. Renommer la table **Sites** en **SitesNaturals**.

4. Ajouter la valeur "**Climatique**" aux valeurs possibles du champ TypeCause.

Exercice 4 : (...../ 6 points)

Soit la base de données **gestion_terrain** permettant de gérer un système de réservation de terrains de sport, cette base de données simplifier est décrite par la représentation textuelle suivante.

Terrains (id_terrain, nom, type, capacite, localisation)

Clients (id_client, nom, email, telephone, date_inscription)

Reservations (id_reservation, id_client#, id_terrain#, date_reservation, heure_debut, heure_fin)

Paiements (id_paiement, id_reservation#, montant, date_paiement, mode_paiement)

Champ	Type	Description
id_terrain	Entier	Identifiant unique du terrain .
nom	Chaîne de 50 caractères	Nom du terrain.
type	Chaîne de 50 caractères	Type de terrain.
capacite	entier	Capacité maximale du terrain (en personnes).
localisation	Chaîne de 100 caractères	Adresse ou localisation du terrain.
id_client	Entier	Identifiant unique du client .
nom	Chaîne de 50 caractères	Nom complet du client.
email	Chaîne de 50 caractères	Adresse email du client (doit être unique).
telephone	Chaîne de 08 caractères	Numéro de téléphone du client.
date_inscription	Date	Date à laquelle le client s'est inscrit.
id_reservation	Entier	Identifiant unique de la réservation
date_reservation	Date	Date de la réservation.
heure_debut	temps	Heure de début de la réservation.
heure_fin	temps	Heure de fin de la réservation.
id_paiement	Entier (auto incrémenter)	Identifiant unique du paiement .
montant	Réel	Montant total payé par le client (en Dinars).
date_paiement	Date	Date à laquelle le paiement a été effectué.
mode_paiement	Chaîne de 50 caractères	Mode de paiement utilisé par le client ('espèces', 'carte', 'virement')

Les champs des tables sont décrits dans le tableau suivant :

Ecrire les requêtes SQL qui permettent de :

1. Afficher les clients (**nom**, **email**) qui ont effectué une réservation pour un terrain de type "Football", triés par date d'inscription dans l'ordre décroissant.

.....

.....

.....

.....

.....

2. Modifier le mode de paiement de la réservation dont l'ID est 202 en "virement".

.....

.....

3. Afficher les réservations effectuées durant le mois **mars** et l'année **2025**. On affichera les champs `id_reservation`, `nom`, `email`, `nom` et `date_reservation`

.....

.....

.....

.....

4. Afficher pour chaque nom de terrain : l'ID, et le nombre de réservations effectuées.

.....

.....

.....

5. Supprimer les réservations dont la date de réservation est antérieure au 1er janvier 2023.

.....

.....

6. Afficher les terrains (**nom**, **localisation**) qui n'ont jamais été réservés.

.....

.....

.....

7. Afficher le montant total des paiements effectués par chaque client, trié par montant total décroissant.

.....

.....

.....

.....

8. Afficher les noms des terrains qui ont été réservés plus de 5 fois.

.....

.....

.....

.....

9. Afficher les noms et les emails des clients qui ont effectué des paiements supérieurs à 100 Dinars, qui sont inscrits après le 1er janvier 2020, et dont le numéro de téléphone commence par '71'.

.....

.....

.....

.....

.....

2ème Partie : Programmation Web

Gestion des Menaces

Consultez, gérez et ajoutez des menaces actuelles pour protéger vos données en ligne.

Ajouter une Nouvelle Menace

Nom de la Menace :

Ex : Phishing Email

Description :

Type de la Menace :

Niveau de Risque :

Faible Moyen Élevé Critique

Ajouter la Menace

Annuler

Filtrage des Menaces

Filtrer par niveau de risque :

Toutes les menaces

Filtrer

Nom de la Menace	Description	Niveau de Risque	Type de la Menace
Phishing Email	Emails frauduleux visant à voler des informations personnelles.	2	Phishing
Botnet	Un réseau d'ordinateurs infectés contrôlés par un cybercriminel.	3	Autre
Logiciel Espion	Programmes qui collectent des informations sans consentement.	1	Malware
Vol de mots de passe	Utilisation de techniques comme le keylogging pour récupérer les identifiants d'un utilisateur.	0	Spyware

Figure1 : Page web gestion des menaces

```

<form ..(1).. ="verif()" ..(2).. ="ajouter_menace.php" ..(3).. ="POST" target="_self">
<fieldset>
<div class="form-group">
<label for="nom_menace">Nom de la Menace :</label>
<input type="text" id="nom_menace" name="nom_menace" ..(4)..="Ex : Phishing
Email" onfocus="activer()">
</div>
<div class="form-group">
<label for="description">Description :</label>
<..(5).. id="description" name="description" cols="100" rows="7" disabled ></..(5)..>
</div>
<div class="form-group">
<label for="">Type de la Menace :</label>
<input list="type" id="1">
<..(6).. id="types" name="types">
    <option value="Phishing"></option>
    <option value="Malware"></option>
    <option value="Ransomware"></option>
    <option value="Spyware"></option>
    <option value=" Autre"></option>
</..(6)..>
</div>
<div class="form-group">
<label for="niveau_risque">Niveau de Risque :</label> <br>
<input type="radio" name="gr1" class="c" value="1"> Faible
<input type="radio" name="gr1" class="c" value="2"> Moyen
<input type="radio" name="gr1" class="c" value="3">Élevé
<input type="radio" name="gr1" class="c" value="4">Critique
</div>
<input type="submit" value="Ajouter la Menace" class="btn">
<input type="reset" value="Annuler" class="btn">
</fieldset>
</form>

```

Figure 2 : Extrait du code HTML de la page **Gestion_Menaces.html**

```
form {
  background-color: #f8f8f8;
  padding: 30px;
  border-radius: 8px;
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
  margin: auto;}

.btn {
  background-color: #1abc9c;
  color: white;
  padding: 15px 30px;
  font-size: 18px;
  border: none;
  border-radius: 5px;
  cursor: pointer;}

.btn:hover {
  background-color: #16a085;}

@keyframes anim1 {
  from{transform: scale(0.5);}

  to{
    transform: scale(1);
  } }
```

Figure 3 : Extrait du feuille de styles

```

<?php
$cnx=mysqli_connect("localhost","root","","Gestion_ProtectionWeb" ) or die("Problème de connexion");
//récupération du niveau de la gravité de la menace
$niveau=$_POST["filtrer"];
// Identification de la requête à exécuter
if($niveau=="11"){
  ..... 1 .....
}
else{
  ..... 2 .....
}
//Exécution de la requête
..... 3 .....
// affichage du tableau résultat
echo"<table border='1'>
  <tr> <th>Nom de la Menace</th> <th>Description</th> <th>Niveau de Risque</th> <th>Type de la
Menace</th> </tr>";
  ..... 4 .....

echo "</table>";
?>

```

Figure 4 : Extrait du fichier **filtrage.php**