

Corrigé d'examen Technologies Web**Exercice 1**

1. Quelle est la différence entre `<div>` et `<span>` en HTML?
2. C'est quoi la différence entre une « balise HTML » et un « élément HTML » ?
3. Quels sont les attributs importants pour la balise `<img>`?
4. Comment créer une liste déroulante en utilisant HTML?
5. Quelle est la différence entre `margin` et `padding`?
6. C'est quoi un sélecteur CSS? donnez un exemple?
7. Quel est l'intérêt de l'utilisation de JS dans les pages web?
8. Comment changer le contenu d'un élément HTML avec javascript?
9. Quelle est la différence entre `==` et `===` en JavaScript?
10. Comment soumettre le contenu d'un formulaire en HTML en utilisant JavaScript?

**Corrigé:**

1. La différence entre `<div>` et `<span>` en HTML :

`<div>` : est un élément de bloc qui commence sur une nouvelle ligne et occupe toute la largeur disponible.

`<span>` : est un élément en ligne qui ne modifie pas le flux et n'occupe que l'espace nécessaire.

2. La différence entre une « balise HTML » et un « élément HTML »

balise HTML: Représente la syntaxe comme `<p>` ou `</p>`

élément HTML: C'est l'ensemble composé de la balise ouvrante, du contenu et de la balise fermante

Exemple : `<p> Texte </p>`

3. Les attributs obligatoires ou importants pour la balise `<img>` sont :

`src` : chemin de l'image.

`alt` : texte alternatif (obligatoire pour l'accessibilité).

`width / height` : pour définir les dimensions (optionnel mais recommandé).

4. La création une liste déroulante en utilisant HTML :

```
<select>
  <option value="option1">Option 1</option>
  <option value="option2">Option 2</option>
</select>
```

5. La différence entre `margin` et `padding` :

`Margin` : Espace à l'extérieur de l'élément, entre l'élément et ses voisins.

`Padding` : Espace à l'intérieur de l'élément, entre le contenu et la bordure.

- 6.

Un sélecteur CSS permet de cibler des éléments HTML pour leur appliquer des styles. Exemple :

```
<style>
p { color: blue; } /* Sélectionne tous les <p> */
</style>
```

- 7.

JavaScript permet de rendre les pages interactives, dynamiques et réactives (modification de contenu, gestion d'événements, animations, appels API).

- 8.

On peut utiliser `textContent` ou `innerHTML` Exemple :

```
<script>
  document.getElementById("monId").textContent = "Nouveau texte"; /*plus
sécurisé*/
  document.getElementById("monId").innerHTML = "Nouveau texte";
  document.getElementById("monId").innerText = "Nouveau texte";
</script>
```

9. La différence entre `==` et `===` en JavaScript :

`==` : Comparaison avec conversion de type (ex : `5=="5" → true`).

`===` : Comparaison stricte sans conversion de type (ex : `5==="5" → false`).

10.

On peut utiliser `submit()` ou intercepter l'événement `submit` :

```
<script>
  document.getElementById("monForm").submit();
</script>
```

## Exercice 2

Créez un formulaire en HTML/CSS avec les champs suivants :

Nom, Prénom, Email (Doit être contient « @ et. »), Téléphone (le numéro de téléphone doit être 8 chiffre )

Ajoutez un bouton "Ajouter". Lorsque l'utilisateur clique sur le bouton :

- Les données saisies doivent être affichées dans un tableau sous le formulaire (en utilisant du JS).
- Le tableau doit contenir toutes les données ajoutées. Si l'utilisateur ne remplit pas les champs, une alerte doit s'afficher pour lui demander de les remplir.

**ajouter un utilisateur**

Prénom :

Nom :

Email :

Téléphone :

**les données d'utilisateur**

| Prénom    |  |  |
|-----------|--|--|
| Nom       |  |  |
| Email     |  |  |
| Téléphone |  |  |

NB.

La première colonne (Prénom, Nom,...) doit être statique et la deuxième, change selon les données saisies par l'utilisateur.

**Corrigé:**

```
<!doctype html>
<html lang="fr">
  <head>
    <meta charset="UTF-8" />
    <title>Exam25 EXO 2</title>
    <style>
      body {
        margin-left: 400px;
        margin-top: 300px;
      }
      fieldset {
        max-width: 400px;
      }
      .container {
        background: #f4f1e6;
        padding: 25px;
        box-shadow: 0 4px 10px #0000001a;
        padding: 20px;
        width: 400px;
        margin: auto;
        margin-top: 10px;
      }
      .donnee {
        display: flex;
        margin-bottom: 15px;
        align-items: center;
      }
      .donnee label {
        width: 100px;
        font-weight: bold;
      }
      .donnee input {
        flex: 1;
        padding: 5px;
      }
      button {
        background-color: #555;
        color: white;
        padding: 10px;
        width: 100%;
```



```

        <b>Les données d'utilisateur</b>
    </caption>
    <tr>
        <th>Prénom</th>
        <td id="prenomUti"></td>
        <td style="background-color: gray;" rowspan="4" ></td>
    </tr>
    <tr>
        <th>Nom</th>
        <td id="nomUti"></td>
    </tr>
    <tr>
        <th>Email</th>
        <td id="emailUti"></td>
    </tr>
    <tr>
        <th>Téléphone</th>
        <td id="TeleUti"></td>
    </tr>
</table>
</div>
</fieldset>
<script>
function ajouter() {
    event.preventDefault();
    const prenom = document.getElementById("prenom").value;
    const nom = document.getElementById("nom").value;
    const email = document.getElementById("email").value;
    const telephone = document.getElementById("telephone").value;
    if (!prenom || !nom || !email || !telephone) {
        alert("Veuillez remplir tous les champs.");
        return;
    }
    if (!email.includes("@") || !email.includes(".")) {
        alert("Email invalide (doit contenir @ et .)");
        return;
    }
    if (telephone.length !== 8) {
        alert("Le numéro de téléphone doit être 8 chiffres.");
        return;
    }
    document.getElementById("prenomUti").textContent = prenom;
    document.getElementById("nomUti").textContent = nom;
    document.getElementById("emailUti").textContent = email;

```

```
document.getElementById("TeleUti").textContent = telephone;  
}  
</script>  
</body>  
</html>
```

