TP6 – Internet & Web – Animations (Corrigé)

Licence Informatique – 1ère année

Question 1 – Animation fadeOut() avec jQuery

On crée un répertoire TP6 et on y ajoute la page principale. Le paragraphe 'cliquez-moi' agit comme un lien et déclenche une animation de disparition sur l'image.
Commandes Linux :

```
cd ~
mkdir -p public_html/TP6
cd public_html/TP6
cp ~/chemin/vers/dom.html .
touch question1.html
gedit question1.html &
```

Code HTML :

```
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <title>TP6 Question 1</title>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    <style>
        .clickable {
            color: blue;
            text-decoration: underline;
            cursor: pointer;
        }
    </style>
</head>
<body>
    <p class="clickable" id="clickMe">cliquez-moi</p>
    <img src="images/js.jpeg" alt="Photo" id="photo" style="width: 200px; height: auto;">

    <script>
        $("#clickMe").click(function() {
            $("#photo").animate(
                {
                    width: "0px",
                    height: "0px",
                    opacity: 0
                },
                1000,
                function() {
                    $(this).hide();
                }
            );
        });
    </script>
</body>
</html>
```

Lorsqu’on clique, l’image reste dans le DOM mais jQuery modifie ses styles inline. L’opacité diminue jusqu’à 0, puis l’image est masquée. Le DOM n’est pas modifié structurellement.

Question 2 – Deux images et texte masqué/affiché

On affiche deux photos côte à côte. Au survol, un cadre bleu apparaît. Un clic sur l'image de droite masque le texte, et un clic sur celle de gauche le réaffiche.

```
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <title>TP6 Question 2</title>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    <style>
        .image-container img {
            margin-right: 10px;
            border: 2px solid transparent;
            cursor: pointer;
        }
        .image-container img:hover {
            border-color: blue;
        }
    </style>
</head>
<body>
    <div class="image-container">
        <img src="images/js1.jpeg" alt="js1" id="js1">
        <img src="images/js2.jpeg" alt="js2" id="js2">
    </div>
    <p id="texte">Texte à afficher ou masquer</p>

    <script>
        $("#js1").click(function() {
            $("#texte").fadeIn();
        });

        $("#js2").click(function() {
            $("#texte").fadeOut();
        });
    </script>
</body>
</html>
```

L'effet est fluide grâce à fadeIn() et fadeOut(). On comprend comment jQuery interagit avec le DOM sans le modifier directement.

Question 3 – Boutons interactifs avec jQuery

On crée trois boutons différents et un bouton 'Clear'. Chaque bouton affiche son contenu (paragraphe + image). Seul le bouton 2 affiche uniquement son contenu.

```
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <title>TP6 Question 3</title>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    <style>
        .button {
            padding: 10px;
            margin: 5px;
            background-color: lightgrey;
            border: none;
            cursor: pointer;
            transition: background-color 0.3s;
        }
        .button:hover {
            background-color: darkgrey;
        }
        #content1, #content2, #content3 {
            display: none;
            margin-top: 10px;
        }
    </style>
</head>
<body>
    <button class="button" id="button1">Bouton 1</button>
    <button class="button" id="button2">Bouton 2</button>
    <button class="button" id="button3">Bouton 3</button>
    <button class="button" id="clear">Clear</button>

    <div id="content1">
        <p>Contenu du Bouton 1</p>
        <img src="images/imageB1.jpg" alt="Image Bouton 1">
    </div>
    <div id="content2">
        <p>Contenu du Bouton 2</p>
        <img src="images/imageB2.jpg" alt="Image Bouton 2">
    </div>
    <div id="content3">
        <p>Contenu du Bouton 3</p>
        <img src="images/imageB3.jpg" alt="Image Bouton 3">
    </div>

    <script>
        $("#button1").click(function() {
            $("#content1").toggle();
            $("#content2, #content3").hide();
        });

        $("#button2").click(function() {
            $("#content2").toggle();
            $("#content1, #content3").hide();
        });

        $("#button3").click(function() {
            $("#content3").toggle();
            $("#content1, #content2").hide();
```

```
        });

        $("#clear").click(function() {
            $("#content1, #content2, #content3").hide();
        });
    </script>
</body>
</html>
```

On obtient une interface dynamique claire. Chaque bouton agit indépendamment et jQuery simplifie la logique d’affichage.

Question 4 – Jeu de cartes et survol d’images

On affiche 4 cartes superposées. Le roi se transforme en as au survol grâce au CSS, et le 10 change via jQuery.

```
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <title>TP6 Question 4</title>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    <style>
        .cards {
            position: relative;
            width: 200px;
            height: 300px;
            margin: auto;
        }
        .cards img {
            position: absolute;
            top: 0;
            left: 0;
            width: 100%;
            height: auto;
        }
        #roi { z-index: 4; }
        #dame { z-index: 3; }
        #valet { z-index: 2; }
        #dix { z-index: 1; }
        #roi:hover {
            content: url("images/as.jpg");
        }
    </style>
</head>
<body>
    <div class="cards">
        <img src="images/roi.jpeg" alt="Roi" id="roi">
        <img src="images/dame.jpeg" alt="Dame" id="dame">
        <img src="images/valet.jpeg" alt="Valet" id="valet">
        <img src="images/10.jpeg" alt="10" id="dix">
    </div>

    <script>
        $("#dix").hover(function() {
            $(this).attr("src", "images/as.jpg");
        }, function() {
            $(this).attr("src", "images/10.jpeg");
        });
    </script>
</body>
</html>
```

Le positionnement relatif et l’utilisation du z-index permettent une superposition correcte. Les événements hover montrent l’interaction entre CSS et jQuery.

Question 5 – Menu déroulant CSS

On crée une liste à deux niveaux. Grâce aux CSS, elle devient un menu horizontal avec sous-menus visibles au survol.

```
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <title>TP6 Question 5</title>
    <style>
        .menu li {
            display: inline-block;
            position: relative;
        }
        .submenu {
            display: none;
            position: absolute;
            background-color: #f9f9f9;
            box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
        }
        .menu li:hover .submenu {
            display: block;
        }
    </style>
</head>
<body>
    <ul class="menu">
        <li>Entrées
            <ul class="submenu">
                <li><a href="soupe.html">Soupe</a></li>
                <li><a href="salade.html">Salade</a></li>
            </ul>
        </li>
        <li>Plats
            <ul class="submenu">
                <li><a href="poisson.html">Poisson</a></li>
                <li><a href="poulet.html">Poulet</a></li>
            </ul>
        </li>
    </ul>
</body>
</html>
```

Le menu est simple, responsive et élégant. On comprend comment les CSS seules peuvent créer un effet d’interactivité sans JavaScript.