



Université Abdelmalek Essaâdi
Ecole Nationale des Sciences Appliquées
Al Hoceima



Filière :

Transformation digitale Et Intelligence artificielle

Sujet :

web :TP2 et TP3

Réalisé par :

ANAS MATAICH

Encadré par :

• Pr. CHERRADI MOHAMED.

Remerciements

Nos remerciements s'adressent en premier lieu à notre encadreur

Mr.CHERRADI MOHAMED.de l'honneur qu'il a fait en encadrant
notre projet, de la gentillesse qu'il a éprouvée, de ses idées et de
l'attention qu'il porte à ses étudiants, tout en conduisant leurs
travaux dans le respect mutuel et la confiance

Enfin nous tenons également à remercier tous nos enseignants du
département informatique, pour leurs précieux conseils et l'effort
qu'ils n'ont cessé de nous prodiguer au cours de notre formation

SOMMAIRE

WEB

1) TP2

EXERCICE1-----

EXERCICE2-----

EXERCICE3-----

EXERCICE4-----

EXERCICE5-----

2) TP1

EXERCICE1-----

EXERCICE2-----

EXERCICE3-----

EXERCICE4-----


1)TP3

EXERCICE1 :

En utilisant uniquement les langages HTML & CSS, réaliser votre CV sous forme d'un fichier html qui respecte template ci-dessous :

la réponse de exercice est :

il y a le code de html et css dans github.



ANAS MATAICH

OBJECTIVE

Ingénieur passionné et diplômé, j'ai 5 ans d'expérience à mon actif dans de grands groupes. Je recherche actuellement un nouvel emploi pour commencer dès le mois de juin

De nature débrouillard et indépendant dans mon travail, j'aime apprendre de nouvelles technologies, passer du temps à résoudre des problèmes et réaliser du code de qualité. Mes valeurs de travail : clean code, flexibilité, performance et sérieux.

COMPÉTENCES (SKILLS)

- ✓ Angular | Typescript
- Bootstrap

STEVEN TERRY

GRAPHIC DESIGNER

@ anas.mataich6@gmail.com
📞 Téléphone: +212 618 44 32 78

📍 Ville : Maroc Nador
📅 Né le : 4 avril 2001

EXPÉRIENCES PROFESSIONNELLES

2015 → 2021
Fullstack Developer à temps plein chez *Webadev SPRL*

- Intégration de templates Photoshop, Illustrator, Sketch, Figma
- Animations CSS / JS
- Responsive design
- UI / UX Design
- Projets sous npm et Webpack
- Collaboration avec d'autres studios graphique (Studio Debie, SOL,...)
- Optimisation des performances
- Développement de templates et de modules réutilisables
- Projets en équipe, utilisation quotidienne de SVN, Git et Github

2021
Freelance en activité

- Freelance Front-end Developer

- ✓ **Bootstrap**
- ✓ **HTML5 | CSS3 |**
- ✓ **Javascript | python | C**
- ✓ **npm | Webpack**
- ✓ **PHP**
- ✓ **Zend Framework**
- ✓ **MySQL**
- ✓ **Git | Github**

LANGUES

Français, Arabe ,Amazigh
Anglais, compétence professionnelle

HONORS & AWARDS

CERTIFICATIONS

Certificat d'ingénieur en 2026

Certificat de baccalauréat 2021

CENTRES D'INTÉRÊT

Jeux vidéo, jouer et développer
Musique, écoute et composition
Art en général
Sport
Informatique en général

- Freelance Front-end Developer
- Unity Developer / Sound Designer

ÉTUDES

& FORMATIONS

2018 → 2021

Bachelier en Informatique de Gestion.

2018

STE-Formations Informatiques, Formation qualifiante de Web

ACTIVITÉS ET CENTRES D'INTÉRÊT

Coordination entre les équipes techniques et les interfaces externes pour assurer l'amélioration de la performance des plans d'actions.

Veille sur la mise à jour des processus de l'usine

Pilotage d'un projet de validations qualité sur lignes de conditionnement impliquant de nombreux acteurs

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua

EXERCICE2 :

```
style>
p a { ...
}
.enonce div.encadre img {
}
img[width="300"] {
}
article.remarque time[datetime] {
}
div.exemple > p {
}
p#important img.encadre {
}
div > img:first-child {
}
*:nth-child(odd) > img {
}
article.exercice > p:first-child a[href$=".jpg"] {
}
.reponse ~ .remarque {
}
section article > p:nth-of-type(2) {
```

- 1) a : Cette partie du sélecteur cible tous les éléments <a> (liens) qui sont descendants directs des éléments <p> sélectionnés .
- 2) Cette ligne cible les images qui se trouvent à l'intérieur d'un élément div avec la classe encadre et qui sont descendants d'un élément div avec la classe enonce. Cette règle est vide, donc elle ne définit aucun style pour les images sélectionnées.

3) Cette ligne cible toutes les images avec une largeur de 300 pixels. Cette règle est également vide, donc elle ne définit aucun style pour les images sélectionnées.

4) Cette ligne cible les éléments `time` avec un attribut `datetime` qui se trouvent à l'intérieur d'un élément `article` avec la classe `remarque`. Cette règle est vide, donc elle ne définit aucun style pour les éléments sélectionnés.

5)

`div.exemple` : Cette partie du sélecteur cible tous les éléments `div` avec la classe `exemple`.

- `>` : Ce caractère est un opérateur descendant direct. Il signifie que les éléments `p` sélectionnés doivent être des enfants directs des éléments `div` sélectionnés.

- `P` : Cette partie du sélecteur cible tous les éléments `p`

6) Cette ligne cible les images avec la classe `encadre` qui se trouvent à l'intérieur d'un paragraphe avec l'ID `important`. Cette règle est vide, donc elle ne définit aucun style pour les images sélectionnées .

7) Cette ligne cible la première image (élément `img`) qui est un enfant direct d'un élément `div`. Cette règle est vide, donc elle ne définit aucun style pour les images sélectionnées.

8) Cette ligne cible toutes les images qui sont des enfants impairs d'un élément quelconque. Cette règle est vide, donc elle ne définit aucun style pour les images sélectionnées.

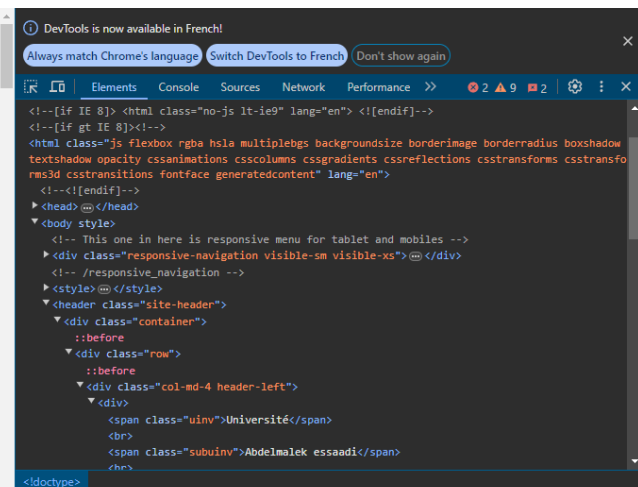
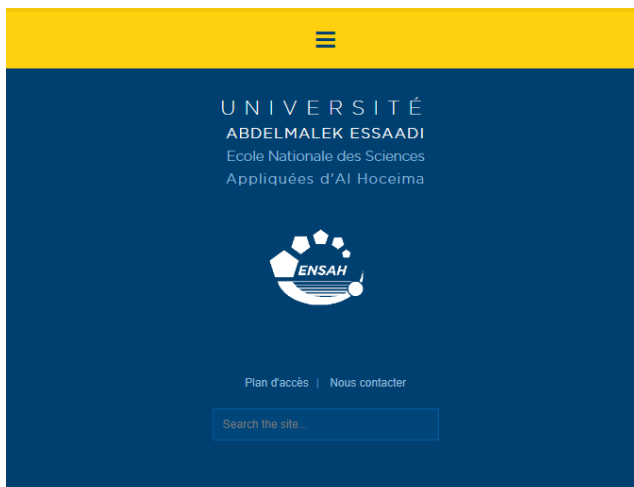
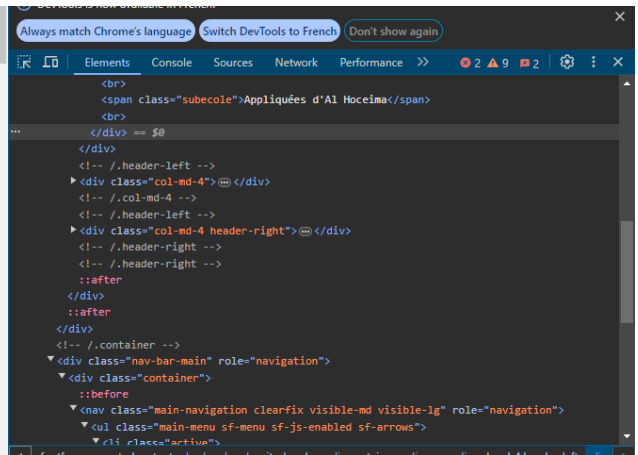
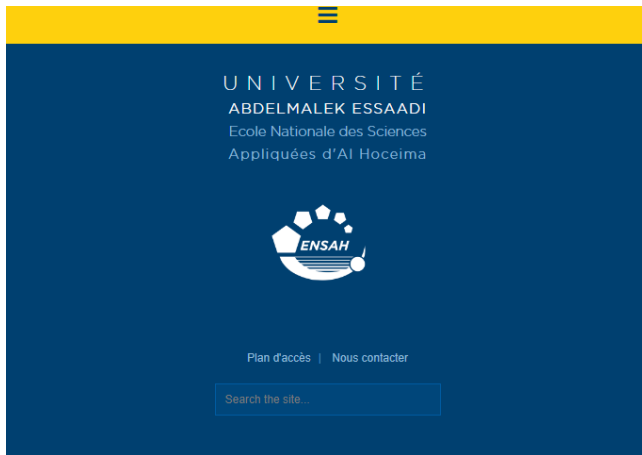
9) Cette ligne cible les liens (éléments `a`) qui se trouvent à l'intérieur du premier paragraphe (élément `p`) d'un élément `article` avec la classe `exercice` et dont l'attribut `href` se termine par `.jpg`. Cette règle est vide, donc elle ne définit aucun style pour les liens sélectionnés.

10) Le sélecteur `.reponse ~ .remarque` cible tous les éléments avec la classe `remarque` qui sont des frères suivants d'un élément avec la classe `reponse`.

11) Cette ligne cible le deuxième paragraphe (élément `p`) qui est un enfant direct d'un élément `article` qui se trouve à l'intérieur d'une section (`section`). Cette règle est vide, donc elle ne définit aucun style pour les paragraphes sélectionnés.

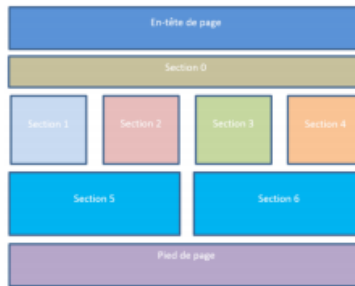
EXERCICE3 :

Donner le code HTML et CSS pour chacun des éléments suivants :



EXERCICE4 :

Utilisez Flexbox pour faire la mise en page suivante :



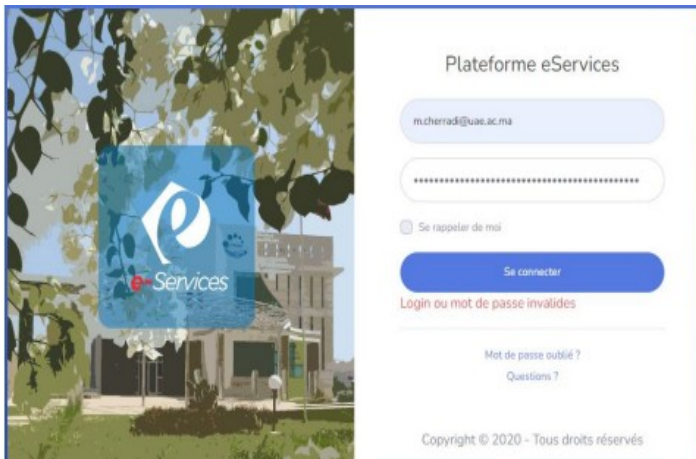
Pour rendre la page responsive design, utilisez les Mediaqueries pour adapter la page aux différents type d'appareils. Ainsi, utiliser Bootstrap.

code html et css dans github :



EXERCICE5 :

En utilisant que l'HTML et CSS (*sans bootstrap*), réaliser une interface responsive comme celle décrite sur la figure (a) (sur desktop) et la figure (b) (mobile et tablette).



(b)

code html et css dans github .



Plateforme eServices

☐ Se rappeler de moi

Se connecter

Login ou mot de passe invalides

• [Mot de passe oublié ?](#)

• [Question ?](#)

Copyright © 2020 - tous droits réservés

TP3

EXERCICE1 :

On appelle nombres d'Armstrong les nombres entiers positifs tels que la somme des cubes de leurs chiffres est

égale au nombre lui-même. Exemple : 153 est un nombre d'Armstrong. En

effet : $1*1*1+5*5*5+3*3*3 = 153$

Écrire un programme PHP qui affiche tous les nombres d'Armstrong inférieurs à 1000 sous forme d'une liste à

puce.

```
<?php

function estArmstrong($nombre) {
    $sommeCubes = 0;
    while ($nombre > 0) {
        $chiffre = $nombre % 10;
        $sommeCubes += $chiffre * $chiffre * $chiffre;
        $nombre = (int)($nombre / 10);
    }
    return $sommeCubes === $nombre;
}

echo "<ul>";
for ($i = 1; $i < 1000; $i++) {
    if (estArmstrong($i)) {
        echo "<li>$i</li>";
    }
}
echo "</ul>";

?>
```

Le sortie:

```
<ul> <li>1</li> <li>153</li> <li>371</li> <li>407</li>

</ul>
```

EXERCICE2 :

Ecrire un programme PHP qui effectue une suite de tirages de nombres aléatoires jusqu'à obtenir une suite

composée de deux nombres pair suivis d'un nombre impair.

```
<?php
function tirageAleatoire($min, $max) {
    return rand($min, $max);
}
function obtenirSuite() {
    $tirage1 = tirageAleatoire(0, 100);
    $tirage2 = tirageAleatoire(0, 100);
    $tirage3 = tirageAleatoire(0, 100);

    while (!($tirage1 % 2 === 0 && $tirage2 % 2 === 0 && $tirage3 % 2 === 1)) {
        $tirage1 = tirageAleatoire(0, 100);
        $tirage2 = tirageAleatoire(0, 100);
        $tirage3 = tirageAleatoire(0, 100);
    }

    return [$tirage1, $tirage2, $tirage3];
}
$suite = obtenirSuite();
echo "Suite obtenue : ";
echo implode(", ", $suite);
?>
```

Le sortie: Suite obtenue : 24, 50, 71

EXERCICE3 :

Ecrire un programme PHP qui permet de générer aléatoirement un nombre de trois chiffres N. Ensuite il

effectue des tirages aléatoires jusqu'à réobtenir le même nombre N puis il affiche le nombre d'essais réalisés.

Réalisez ce script d'abord avec l'instruction *while* puis avec l'instruction *for*.

Pour l'instruction *while* :

```
<?php

function genererNombreAleatoire() {
    return rand(100, 999);
}

function obtenirNombreN($n) {
    $nombre = genererNombreAleatoire();
    $compteur = 0;
    while ($nombre != $n) {
        $nombre = genererNombreAleatoire();
        $compteur++;
    }
    return $compteur;
}

$n = genererNombreAleatoire();

$nombreEssais = obtenirNombreN($n);

echo "Nombre N : $n" . PHP_EOL;
echo "Nombre d'essais : $nombreEssais" . PHP_EOL;
?>
```

pour l'instruction for :

```
<?php

function genererNombreAleatoire() {
    return rand(100, 999);
}

function obtenirNombreN($n) {
    $nombre = genererNombreAleatoire();
    $compteur = 0;
    for (; $nombre != $n; $compteur++) {
        $nombre = genererNombreAleatoire();
    }
    return $compteur;
}

$n = genererNombreAleatoire();

$nombreEssais = obtenirNombreN($n);

echo "Nombre N : $n" . PHP_EOL;
echo "Nombre d'essais : $nombreEssais" . PHP_EOL;

?>
```

EXERCICE4 :

Ecrire une application web qui permet de résoudre les équations du deuxième degré en PHP. L'utilisateur aura

la possibilité de saisir les données nécessaires via un formulaire et l'application

affiche la solution après la

soumission de celui-ci.

Code php :

```

<?php

$a = 0;
$b = 0;
$c = 0;
$solution1 = null;
$solution2 = null;

if (isset($_POST['submit'])) {
    $a = floatval($_POST['a']);
    $b = floatval($_POST['b']);
    $c = floatval($_POST['c']);

    $delta = $b * $b - 4 * $a * $c;

    if ($delta > 0) {
        $solution1 = (-$b + sqrt($delta)) / (2 * $a);
        $solution2 = (-$b - sqrt($delta)) / (2 * $a);
    } elseif ($delta === 0) {
        $solution1 = -$b / (2 * $a);
    } else {

        $solution1 = "Pas de solution réelle";
        $solution2 = "";
    }
}

?>

```

code html :

```

<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Résolution d'équations du second degré</title>
</head>
<body>
    <h1>Résolution d'équations du second degré</h1>
    <form method="post">
        <label for="a">Coefficient a:</label>
        <input type="number" id="a" name="a" value="<?php echo $a; ?>">
        <br>
        <label for="b">Coefficient b:</label>
        <input type="number" id="b" name="b" value="<?php echo $b; ?>">
        <br>
        <label for="c">Coefficient c:</label>
        <input type="number" id="c" name="c" value="<?php echo $c; ?>">
        <br>
        <br>
        <input type="submit" name="submit" value="Calculer">
    </form>

    <h2>Résultats</h2>
    <ul>
        <li>Solution 1: <?php echo $solution1; ?></li>
        <li>Solution 2: <?php echo $solution2; ?></li>
    </ul>
</body>
</html>

```