


CORRECTION

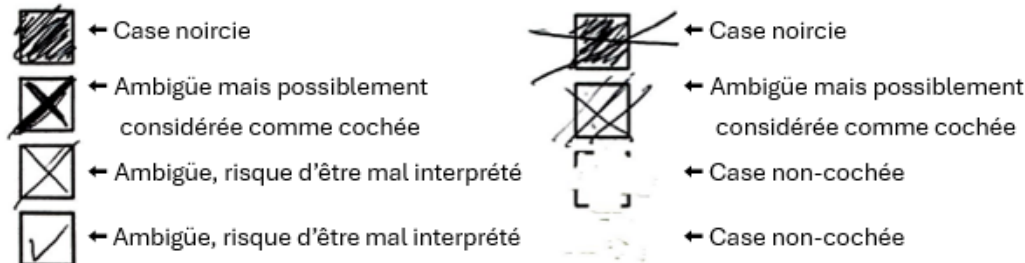
Nom et prénom :

.....

	EXAMEN 2023/2024		
	MATIÈRE : Développement WEB		
	Remis par : Alban MAZEROLLES		
	Promotion : I3-FISA-RIOC		
	Date : 01/03/2024	Durée : 1h	1 feuille de brouillon autorisée.
	Sans calculatrice	Sans document	

Vérifiez que votre copie est bien composée de 14 questions et que toutes vos pages portent le même numéro de copie (en bas du document).

Utiliser un stylo foncé (bleu foncé ou noir) pour foncer les cases. Evitez donc le crayon à papier. Si vous souhaitez "décocher" une case, il faut utiliser du blanc (correcteur, blanco, souris...), quitte à supprimer la case. Rayer une case sera considéré comme une case cochée par le logiciel car le logiciel évalue la quantité de noir dans chaque case. Annoter la case pour dire qu'il ne faut pas la prendre en compte est aussi inefficace :



⚠ Aucune vérification **manuelle** ne sera faite et aucune réclamation en sera acceptée si une case est mal interprétée.

Pour les questions à choix multiples (repérées par le symbole **◆**), il y a zéro, une ou plusieurs cases à foncer. Pour en faciliter la lecture, les questions sont toujours rédigées en utilisant le pluriel.

Une question à choix multiples bien répondue rapporte 1 point. S'il y a une erreur, elle ne rapporte que 0,2. Au delà, elle ne rapporte rien. Une erreur c'est quand une case qui ne doit pas être cochée est cochée ou bien quand une case qui doit être cochée ne l'est pas.

Pour les questions à choix unique, 1 point en cas de bonne réponse, 0 sinon. Pas de point négatif.

Les questions ouvertes valent 1 point si la réponse est correcte, 0.5 point s'il y a une petite erreur ou imprécision, 0 point sinon.

Certaines questions peuvent être coefficientées (indication présente dans la question).

S'il y a du code, il se peut que certaines lignes de code commencent sur une page et finissent sur l'autre. Soyez vigilant.

CORRECTION

Voici quelques fonctions usuelles en python - que vous ne connaissez peut être pas encore - pour les list (a est le nom donné à la liste) :

- ▶ `len(a)` Renvoie le nombre d'éléments dans la liste
- ▶ `del a` Détruit la variable (elle n'est plus définie)
- ▶ `a.append(x)` Ajoute un élément à la fin de la liste.
- ▶ `a.insert(i, x)` Insère un élément à la position indiquée. Le premier argument est la position de l'élément avant lequel l'insertion doit s'effectuer, donc `a.insert(0, x)` insère l'élément en tête de la liste et `a.insert(len(a), x)` est équivalent à `a.append(x)`
- ▶ `a.remove(x)` Supprime de la liste le premier élément dont la valeur est égale à x. Une exception `ValueError` est levée s'il n'existe aucun élément avec cette valeur
- ▶ `a.pop(i)` Enlève de la liste l'élément situé à la position indiquée et le renvoie en valeur de retour. Si aucune position n'est spécifiée, `a.pop()` enlève et renvoie le dernier élément de la liste (les crochets autour du i dans la signature de la méthode indiquent que ce paramètre est facultatif et non que vous devez placer des crochets dans votre code !)
- ▶ `a.clear()` Supprime tous les éléments de la liste. Équivalent à `del a[:]`
- ▶ `a.index(x)` Renvoie la position du premier élément de la liste dont la valeur égale x (en commençant à compter les positions à partir de zéro). Une exception `ValueError` est levée si aucun élément n'est trouvé.
- ▶ `a.count(x)` Renvoie le nombre d'éléments ayant la valeur x dans la liste
- ▶ `a.reverse()` Inverse l'ordre des éléments dans la liste. Met à jour l'élément existant. Valeur de retour `None`

Question 1 On considère le code suivant :

```
liste1 = [1, 2, 3, 4]
liste2 = [10, 20, 30, 40]
nouvelle_liste = [x * y for x, y in zip(liste1, liste2)]
print(nouvelle_liste)
# affiche [10, 40, 90, 160]
```

D'après vos observations, que fait la fonction `zip(...)`

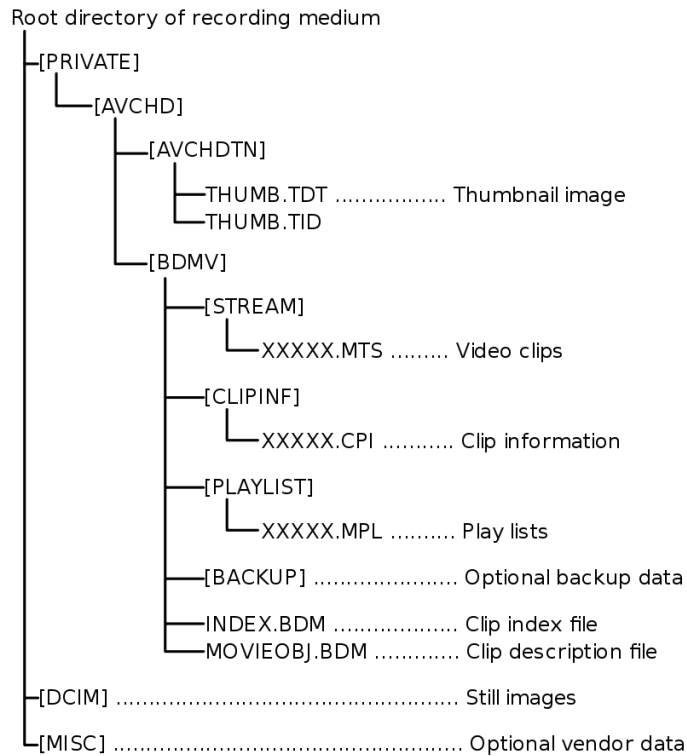
☐ 2+ ☐ 1 ☒ 0 *Reservé - Ne rien cocher*

.....

.....

CORRECTION

Question 2 On considère l'arborescence de fichiers suivantes :



Nous sommes dans le dossier qui contient le fichier THUMB.TID. Nous souhaitons aller dans le dossier DCIM. L'instruction à utiliser est :

☐ ../DCIM
☐ ../../DCIM

☒ ../../../DCIM
☐ ../DCIM

☐ /DCIM
☐ ../../../DCIM

Question 3 Un site web qui s'adapte à la taille des différents dispositifs d'affichage, est dit :

☐ adaptatif ☐ ajustable ☒ responsive ☐ réglable

CORRECTION

Question 4 On considère le code suivant :

```
a = [1,2,[4,5,6],3]
b=a[:]
b.append(3)
b[2].append(3)
print(a)
```

Indiquez ce que la dernière instruction affiche.

☐ 2+ ☐ 1 ☒ 0 *Reservé - Ne rien cocher*

.....
.....

Question 5 ♦ Parmi les produits suivants, lesquels sont des serveurs web (en opposition aux interpréteurs) ?

- | | | |
|--------------------------------|--|---|
| <input type="checkbox"/> flask | <input type="checkbox"/> django | <input type="checkbox"/> Aucune de ces réponses n'est correcte. |
| <input type="checkbox"/> rails | <input checked="" type="checkbox"/> apache | |
| <input type="checkbox"/> php | <input checked="" type="checkbox"/> nginx | |

Question 6 ♦ On a le code suivant : `a = [1, 2, 3, 5, 4, 6, 7]`. On souhaite obtenir `a = [1, 2, 3, 4, 5, 6, 7]`. Lesquelles de ces codes nous permettent d'atteindre ce résultat ?

- ☐ `a.insert(a.index(5),4)`
`a.remove(4)`
- ☒ `b = a[3]`
`a[3] = a[4]`
`a[4] = b`
- ☐ `a.insert(a.index(5),4)`
`a.reverse()`
`a.remove(4)`
- ☒ `a.insert(3,4)`
`a.pop(5)`
- ☐ Aucune de ces réponses n'est correcte.

CORRECTION

Question 7 (coefficient 2) – Laquelle de ces propositions correspond à la sortie du code suivant :

```
a = 3
b = [2, 5, 3, "a", [2, 25.1, "h"]]
c = b[-1]
c.append(3)
print(c)
print(b)
```

- | | |
|---|--|
| <input type="checkbox"/> [2, 5, 3, 'a', [2, 25.1, 'h']]
[2, 5, 3, 'a', [2, 25.1, 'h']] | <input checked="" type="checkbox"/> [2, 25.1, 'h', 3]
[2, 5, 3, 'a', [2, 25.1, 'h', 3]] |
| <input type="checkbox"/> [2, 5, 3, 'a', [2, 25.1, 'h', 3]]
[2, 5, 3, 'a', [2, 25.1, 'h', 3]] | <input type="checkbox"/> [2, 25.1, 'h', 3]
[2, 5, 3, 'a', [2, 25.1, 'h']] |

Question 8 ♦ On considère le code suivant (les ... représentent la ligne manquante) :

```
liste = [1, 2, 3, 4]
...
print(nouvelle_liste)
# affiche [2, 4, 6, 8]
```

Quelles lignes de code conviennent pour remplacer les ... ?

- ☒ nouvelle_liste = [2 * _ for _ in liste]
- ☒ nouvelle_liste = [2 * x for x in liste]
- ☐ nouvelle_liste = [_ for _ in liste if _%2 != 0]
- ☐ nouvelle_liste = [_ for _ in liste if _%2 == 0]
- ☐ Aucune de ces réponses n'est correcte.

Les prochaines questions font partie d'un groupe de questions qui font toute référence au code suivant :

```
1 <div class="container">
2     <aside>
3         <ul>
4             <li><a href="">Elément1</a></li>
5             <li><a href="">Elément2</a></li>
6             <li class="static_elt">Elément static 3</li>
7             <li>
8                 <ul>
9                     <li>Sous-élément 1</li>
10                    <li>Sous-élément 2</li>
11                </ul>
12            </li>
13            <li>Elément5</li>
14        </ul>
15    </aside>
16 </div>
```

CORRECTION

Question 9 On souhaite que les éléments <a> aient la même couleur de fond que leur parent. Quel règle convient ?

☐ `li a { color: auto; }`

☐ `li a { color: inherit; }`

☐ `li > a { color: auto; }`

☒ `a { background-color: inherit; }`

Question 10 On souhaite que la couleur #0f329e soit appliquée à tous les principaux mais pas aux sous-. Quelle sélecteur convient ?

☒ `aside > ul > li`

☐ `ul > li`

☐ `aside ul li`

☐ `aside ul > li`

Question 11 On souhaite que le bloc <aside> prennent toute la largeur de l'écran. Quelle déclaration convient ?

☐ `width: 100vh;`

☐ `width: 100%;`

☐ `width: 1rem;`

☒ `width: 100vw;`

Question 12 On souhaite que l'élément de classe static_elt soit sur un fond bleu clair lorsqu'on passe le curseur dessus. Quel sélecteur convient ?

☒ `li.static_elt:hover`

☐ `li:not(.static_elt):active`

☐ `li.static_elt`

☐ `li.static_elt:active`

— FIN du groupe de question —

Question 13 On considère le code suivant :

```
result = []
for i in range(6):
    i += 1
    result.append(i)
print(result)
```

Compactez ce code en une ligne grâce aux fonctions map et lambda.

☐ 2+ ☐ 1 ☒ 0 *Reservé - Ne rien cocher*

.....

CORRECTION

Question 14 ♦ On considère le code suivant (les ... représentent la ligne manquante) :

```
liste = ["hello", "the", "world"]
...
print(nouvelle_liste)
# affiche ['the']
```

Quelles lignes de code conviennent pour remplacer les ... ?

- ☐ nouvelle_liste = [x for x in liste if len(x) > 4]
- ☐ nouvelle_liste = [len(x) for x in liste if len(x) > 4]
- ☐ nouvelle_liste = [len(x) for x in liste if len(x) < 4]
- ☒ nouvelle_liste = [x for x in liste if len(x) < 4]
- ☐ Aucune de ces réponses n'est correcte.

– FIN DU SUJET –