

CONSIGNES

L'examen est découpé en 2 parties : Questions de cours et Exercice.

Les questions de cours ont toutes été vues en cours, il n'y a pas de piège, servez-vous de votre bon sens.

1 QUESTIONS DE COURS

Question 1 : Donner les 3 notions distinctes qui caractérisent le Web

- ☐ L'adresse URL, le réseau, le client
- ☐ Le transport, l'URL, le rendu
- ☐ La ressource, le réseau, le client Web

Question 2 : De nouvelles balises ont été intégrées à HTML 5, lesquelles ?

- ☐ <section>
- ☐ <a>
- ☐ <link >
- ☐ <details>

Question 3 : En HTML, il existe un certain nombre de balise permettant de créer un tableau, cocher celles-ci.

- ☐ </TABLE>
- ☐ <TL>
- ☐ <TR>
- ☐ <HBODY>

Question 4 : A quoi sert le DOCTYPE ?

- ☐ Le DOCTYPE permet d'organiser correctement notre page
- ☐ Le DOCTYPE permet de valider que les balises que l'on utilise sont valides
- ☐ Le DOCTYPE n'existe pas
- ☐ Le DOCTYPE sert à la mise à page

Question 5 : Quel ensemble de balises permet de gérer des titres ?

- ☐ <t1>, <t2>, <t3> ... <t6>
- ☐ , ,
- ☐ <div>,

Question 6 : Avant HTML5, à quoi ressemblait le DOCTYPE ?

- ☐ <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
- ☐ <DOCTYPE html>
- ☐ <!DOCTYPE html>
- ☐ <!DOCTYPE>
- ☐ Il n'y en avait pas

Question 7 : Question sur les formulaires : parmi les codes suivants, lequel est juste ?

Réponse : ____.

Proposition 1 :

```
<form action="PST">
Enregistrement d'un utilisateur
<br />Login : <input type="text" name="login">Login</input>
<br />Mot de passe : <input type="password">Passwd</input>
<br /><input type="submit">Me connecter</input>
</form>
```

Proposition 2 :

```
<form method="post" action="/mapage">
Enregistrement d'un utilisateur
<br />Login : <input type="text" name="login" name="Login" />
<br />Mot de passe : <input type="password" name="passwd" />
<br /><input type="submit" />
</form>
```

Proposition 3 :

```
<form method="get" action="/mapage">
Enregistrement d'un utilisateur
```

```
<br />Login : <input type="text" name="login">Login</input>  
<br />Mot de passe : <input type="password">Passwd</input>  
<br /><submit>Me connecter</submit>  
</form>
```

Question 8 : In HTML, a section is **a semantic element for creating standalone sections in a web page**. Please could you tell which HTML elements is true ?

- ☐ <head>, <body>, <foot>
- ☐ <header>, <content>, <footer>
- ☐ <header>, <article>, <footer>
- ☐ <head>, <content>, <foot>

Question 9: Quelle est la différence entre Internet et le Web ?

Question 10: Quel définition des Medias Queries est juste ?

- ☐ Les medias queries servent à gérer l'audio et la vidéo depuis du CSS
- ☐ Les medias queries permettent de mieux gérer les différents types d'écrans ainsi que leurs résolutions
- ☐ Les medias queries permettent d'avoir un langage d'interrogation des éléments HTML

Question 11: Flask s'exécute dans le navigateur

- ☐ Vrai
- ☐ Faux

Question 12: Quel(s) définition(s) du moteur de template est juste selon vous (plusieurs réponses possibles) ?

- ☐ Un moteur de Template est une technique de programmation populaire qui permet de séparer distinctement l'interface graphique du reste de votre application
- ☐ Un moteur de modèle est une bibliothèque ou un Framework qui utilise certaines règles/langages pour interpréter les données et rendre les vues.
- ☐ Un moteur de template permet de générer des modèles de code Python

Question 13: Quel(s) élément(s) caractérise(nt) le mieux Flask (plusieurs réponses possibles) ?

- ☐ Il permet d'avoir un serveur de développement
- ☐ Il a une forte empreinte mémoire
- ☐ Il permet d'avoir un moteur de template
- ☐ Il supporte les sessions
- ☐ Aucun de ces éléments

Question 14: Avec Flask, nous avons ce code minimum :

```
from flask import Flask, request
```

```
app = Flask(__name__)
```

```
@app.route('/')
def index():
```

```
    return "My website with a form"
```

```
if __name__ == "__main__":
```

```
    app.run()
```

Nous souhaitons rajouter une route pour récupérer le contenu du formulaire présent sur notre page d'accueil, quel est la bonne route :

Réponse : ____.

Proposition 1 :

```
@app.route('/create', methods=['POST'])
```

```
def create_post():
```

```
    if request.method == 'POST':
```

```
        return 'form'
```

Proposition 2 :

```
@app.route('/create', methods=('POST'))
```

```
def create_post():
```

```
    if request.method == 'POST':
```

```
        args = request.args
```

```
        return 'form'
```

Proposition 3 :

```
@app.route('/create', methods=['PUT'])
```

```
def create_post(args):
```

```
    return 'form'
```

Question 15: Expliquez en quelques lignes à quoi sert la méthode `redirect()` en Flask ?

Question 16: Expliquez la méthode que vous utilisez pour que votre application Flask puisse afficher les images contenu dans votre site web ?

Question 17: Cette route est censée récupérer un paramètre de l'URL, mais elle génère 2 erreurs. Corrigez-les :

```
@app.route('/recherche')
```

```
def search():
```

```
    q = request.args['search']
```

```
    results = perform_search(query)
```

```
    return render_template('results.html', results=results)
```

Question 18: Identifiez l'erreur dans ce code qui tente d'afficher un message de bienvenue personnalisé :

```
@app.route('/hello/<name>')
```

```
def hello(mynome):
```

```
    return f"Bonjour, {mynome} !"
```

Question 19 : Ce code tente d'utiliser un template Jinja2, mais il ne s'affiche pas correctement. Identifiez le problème :

```
@app.route('/dashboard')
```

```
def dashboard():
```

```
    user = get_current_user()
```

```
    return render_template('dashboard.html', user=user)
```

Dans le template dashboard.html

```
<h1>Bienvenue {{ users.name }}</h1>
```

Question 20 : En algèbre relationnel, pourquoi réaliser un produit cartésien est une mauvaise opération ?

- ☐ Ce n'est pas une mauvaise opération
- ☐ C'est une mauvaise opération car elle retourne une erreur
- ☐ C'est une mauvaise opération car elle ne retournera pas le bon résultat
- ☐ C'est une mauvaise opération car c'est extrêmement coûteux en temps CPU

2 EXERCICE

Vous remplacez votre collègue au pied levé car il est malade, il n'a pas fini son code et vous devez livrer son site Web Flask pour .. aujourd'hui ! Malheureusement, le code qu'il a produit est rempli d'erreurs ou d'éléments non terminées, corrigez-le.

2.1 CODE

```
from flask import Flask

app = Flask(__name__)
app.config['SECRET_KEY'] = 'change_me'

@app.route('/')
def index(name):

    return render_template('index.html', welcome_message=name)

@app.route('/create')
def create():
    if request.method == 'POST':
        title = request.form['title']
        content = request.form['content']

        return 'to be finished'
    if not title:
        flash('Title is required!')
    else:
        conn = get_db_connection()
        conn.execute('INSERT INTOU posts (title, content) VALUES (?, ?)',
                      (title, content))
        conn.commit()
        conn.close()
    return redirect(url_for('index'))

    return render_template('create.html')

@app.route('/<int:id>/edit', methods=('GET', 'POST'))
def edit(id):
    get = get_post(ids)

    if request.method == 'POST':
        title = request.form['title']
        content = request.form['content']

        if not title:
            flash('Title is required!')
        else:
            return redirect(url_for('index'))

    return render_template('edit.html', post=post)
```

2.1.1 *INDIQUEZ ICI VOS CORRECTIONS*

2.2 AJOUT D'UNE FONCTIONNALITE

Votre chef se rend compte qu'il manque une route pour aller sur la page de login nommée « /login » avec le verbe GET et allant vers la page login.html, ajoutez cette route ici :

2.3 ERREUR SUR LE TEMPLATE

Votre collègue est finalement, revenu mais il n'arrive pas à comprendre son erreur, il essaye dans sa fonction edit de récupérer la valeur du titre, mais, son code ne retourne rien, voici son template HTML, indiquez l'erreur.

```
{% extends 'base.html' %}

{% block content %}
<h1>{% block title %} Edit "{{ request.get['title'] }}" {% endblock %}</h1>

<form method="post">
  <div class="form-group">
    <label for="title">Title</label>
    <input type="text" name="titre" placeholder="Post title"
      class="form-control"
      value="{{ request.get['title'] }}">
    </input>
  </div>

  <div class="form-group">
    <label for="content">Content</label>
    <textarea name="content" placeholder="Post content"
      class="form-control">{{ request.form['content'] or
post['content'] }}</textarea>
  </div>
  <div class="form-group">
    <button type="submit" class="btn btn-primary">Submit</button>
  </div>
</form>
<hr>
{% endblock %}
```

2.3.1 INDIQUEZ ICI L'ERREUR

2.4 ERREUR D'AFFICHAGE

Votre collègue a créé une route pour afficher une liste d'articles, mais la page reste vide. Identifiez l'erreur dans le code suivant :

```
@app.route('/articles')
def list_articles():
    articles = [
        {'id': 1, 'title': 'Premier article', 'content': 'Contenu du premier article'},
        {'id': 2, 'title': 'Deuxième article', 'content': 'Contenu du deuxième article'},
        {'id': 3, 'title': 'Troisième article', 'content': 'Contenu du troisième article'}
    ]
    return render_template('articles.html')
```

Voici le contenu du template articles.html :

```
{% extends 'base.html' %}

{% block content %}
    <h1>Liste des articles</h1>

    {% for article in list_article %}
        <div class="article">
            <h2>{{ article.title }}</h2>
            <p>{{ article.content }}</p>
        </div>
    {% endfor %}
{% endblock %}
```

2.4.1 INDIQUEZ ICI L'ERREUR

