

Question #1

1. Qu'est-ce que l'architecture d'une application web?

- a) L'ensemble des technologies utilisés
- b) L'ensemble des composants de haut niveau du logiciel
- c) Le protocole de communication utilisé
- d) L'endroit où sont situés les utilisateurs du système
- e) Aucune de ces réponses

Question #2

Dans l'architecture client-serveur classique, quel composant émet la requête?

- a) Le client
- b) Le serveur
- c) Le client ou le serveur, ça dépend des données échangées
- d) Ni le client, ni le serveur, c'est un composant intermédiaire qui s'en charge
- e) Un microservice
- f) Aucune de ces réponses

Question #3

Dans l'architecture client-serveur classique, quel composant produit la réponse?

- a) Le client
- b) Le serveur
- c) Le client ou le serveur, ça dépend des données échangées
- d) Ni le client, ni le serveur, c'est un composant intermédiaire qui s'en charge
- e) Un microservice
- f) Aucune de ces réponses

Question #4

Quel terme décrit le mieux une application web avec beaucoup de logique d'affaires écrite en Javascript interprété par le navigateur?

- a) Serveur lourd (*fat server*)
- b) Serveur léger (*thin server*)
- c) Client lourd (*fat client*)
- d) Client léger (*thin client*)

Question #5

Dans l'architecture client-serveur classique, le client est connecté à combien de serveurs simultanément?

- a) 0
- b) 1
- c) 2
- d) 3
- e) Le nombre n'est pas connu à l'avance, ça dépend du trafic sur le site web

Question #6

Quelle technologie n'est pas interprétée par un navigateur?

- a) Python
- b) Javascript
- c) HTML
- d) CSS
- e) Aucune de ces réponses

Question #7

Quel port est écouté par défaut par un serveur HTTP?

- a) 25
- b) 5000
- c) 80
- d) 8888
- e) 3000
- f) 21

Question #8

Quel contenant est utilisé pour envoyer un document HTML au navigateur?

- a) Le head d'une requête HTTP
- b) Le body d'une requête HTTP
- c) Le head d'une réponse HTTP
- d) Le body d'une réponse HTTP

Question #9

Quelle affirmation suivante est fausse?

- a) Aucun utilisateur ne peut lire le code du frontend
- b) Le frontend n'accède pas directement à la base de données
- c) Le backend est l'intermédiaire entre le frontend et la base de données
- d) Le frontend est exécuté dans le navigateur
- e) Le backend est exécuté sur le serveur

Question #10

Que doit-on faire si un algorithme breveté ne doit pas être accessible par la concurrence?

- a) Mettre l'algorithme dans le frontend
- b) Mettre l'algorithme dans le backend
- c) Mettre l'algorithme dans le frontend, en maquillant le code pour le rendre illisible
- d) Mettre l'algorithme dans le backend et le frontend
- e) Aucune de ces réponses

Question #11

Quelles sont les méthodes supportées par les formulaires HTML?

- a) GET et PUT
- b) PUT et DELETE
- c) GET et HEAD
- d) POST et PUT
- e) GET et POST
- f) POST et DELETE
- g) Aucune de ces réponses

Question #12

Quelle méthode permet à un formulaire HTML d'envoyer les données encodées dans le body de la requête HTTP?

- a) GET
- b) PUT
- c) DELETE
- d) HEAD
- e) POST
- f) Aucune de ces réponses

Question #13

Quelle méthode permet à un formulaire HTML d'envoyer un mot de passe encrypté automatiquement dans la requête HTTP?

- a) GET
- b) PUT
- c) DELETE
- d) HEAD
- e) POST
- f) Aucune de ces réponses

Question #14

Quelle affirmation suivante est fausse?

- a) Un champ de formulaire est un point d'entrée possible pour l'injection SQL.
- b) Un paramètre d'URL est un point d'entrée possible pour l'injection SQL.
- c) Un cookie est un point d'entrée possible pour l'injection SQL.
- d) Un champ caché dans un formulaire est un point d'entrée possible pour l'injection SQL.
- e) Aucune de ces réponses

Question #15

Quelle affirmation suivante est fausse?

- a) Une URL contient toujours une route.
- b) Une URL contient toujours un nom de domaine ou une adresse IP.
- c) La méthode HTTP GET sert à lire l'état d'une ressource.
- d) Une URL contient toujours un numéro de port.
- e) Une URL peut contenir des paramètres optionnels.
- f) Flask utilise le port 5000 par défaut.

Question #16

Quel code de statut HTTP doit-on retourner lorsqu'une requête de type GET a fonctionné correctement?

- a) 200
- b) 201
- c) 400
- d) 403
- e) 404
- f) 500

Question #17

Quel code de statut HTTP doit-on retourner lorsqu'une requête de type GET fait référence à une ressource inexistante?

- a) 200
- b) 201
- c) 400
- d) 403
- e) 404
- f) 500

Question #18

Quel code de statut HTTP doit-on retourner lorsqu'une requête de type POST ne contient pas toutes les données qu'elle devrait contenir?

- a) 200
- b) 201
- c) 400
- d) 403
- e) 404
- f) 500

Question #19

Quelle expression décrit le mieux la ligne de code suivante :

```
@app.route("/test")
```

- a) Une variable globale définissant une route
- b) Un décorateur définissant une route
- c) Une fonction définissant une route
- d) Une annotation définissant une route
- e) Un paramètre d'objet définissant une route

Question #20

Avec Flask, chaque route est associée à une fonction. Comment la fonction de route envoie-t-elle les données au fureteur?

- a) Avec la fonction `render_template`
- b) Avec la fonction `redirect`
- c) Avec la fonction `send`
- d) Avec la fonction `make_response`
- e) Avec `return`

Question #21

Avec Flask, comment transforme-t-on un gabarit Jinja2 en document HTML?

- a) Avec la fonction `render_template`
- b) Avec la fonction `redirect`
- c) Avec la fonction `send`
- d) Avec la fonction `make_response`
- e) Avec `return`

Question #22

Parmi les éléments suivants, lequel ne devrait pas se retrouver dans le répertoire static de Flask?

- a) Les images
- b) Les documents PDF
- c) Les documents HTML statiques
- d) Les gabarits Jinja2
- e) Les documents CSS
- f) Les bibliothèques Javascript externes

Question #23

Comment Flask permet-il de gérer plusieurs connexions de clients simultanément?

- a) Avec plusieurs processus
- b) Avec des appels de fonctions asynchrones
- c) Avec plusieurs threads
- d) Avec une file d'attente
- e) Un mélange de toutes ces réponses

Question #24

Dans l'éventualité où nous voudrions changer de type de base de données dans le futur, que pouvons-nous faire pour faciliter ce changement futur?

- a) Ne pas utiliser de base de données
- b) Isoler le code qui utilise la base de données
- c) Ne pas changer de base de données
- d) Utiliser une bibliothèque qui fonctionne avec toutes les bases de données
- e) Aucune de ces réponses

Question #25

Considérant l'URL suivante : <https://www.facebook.com/RestaurantChezChose>

Si la partie 'RestaurantChezChose' correspond à l'identifiant d'une page Facebook, quelle serait la ligne définissant la route générique pour les pages Facebook?

- a) `@app.route('https://www.facebook.com/RestaurantChezChose')`
- b) `@app.route('www.facebook.com/RestaurantChezChose')`
- c) `@app.route('/RestaurantChezChose')`
- d) `@app.route('https://www.facebook.com/:page_id')`
- e) `@app.route('www.facebook.com/:page_id')`
- f) `@app.route('/:page_id')`
- g) `@app.route('https://www.facebook.com/<page_id>')`
- h) `@app.route('www.facebook.com/<page_id>')`
- i) `@app.route('/:page_id')`

Question #26

Quelle serait la meilleure route pour une page d'administration d'un site web?

- a) `@app.route('/')`
- b) `@app.route('/admin')`
- c) `@app.route('/?admin=1')`
- d) `@app.route('/:admin')`
- e) `@app.route('/admin/<page_id>')`

Question #27

Quelle technologie est utilisée dans l'extrait de code suivant :

```
<ul>
  {% for artist in artists %}
    <li><a href="/artiste/{{ artist.id }}">{{ artist.nom }}</a></li>
  {% endfor %}
</ul>
```

- a) HTML
- b) CSS
- c) Javascript
- d) Python
- e) Flask
- f) Jinja2
- g) Aucune de ces réponses

Question #28

Quelle ligne de code permet de générer une page 404 avec Flask?

- a) `return 404`
- b) `return make_response(render_template('404.html'))`
- c) `return render_template('404.html'), 404`
- d) `return render_template('404.html')`
- e) `return response.render_template('404.html')`

Question #29

Quel exécutable permet de vérifier que le code source d'un projet respecte les conventions de style de Python?

- a) `flask`
- b) `pep8`
- c) `make`
- d) `python`
- e) `python --style`
- f) Aucune de ces réponses

Question #30

Quels composants peuvent définir un cookie?

- a) Le fureteur seulement
- b) Le serveur seulement
- c) Le fureteur et le serveur
- d) Le fureteur et le client
- e) Aucune de ces réponses