

05. URL, protocole HTTP et modèle client-serveur



Description de l'activité

Dans cette activité, on présente un principe de base du web (le modèle client-serveur), la notion d'URL et le protocole HTTP de communication entre le client et le serveur.

Objectifs pédagogiques ou compétences

Objectifs généraux	Objectifs intermédiaires	Compétences
Notions de cours	<ul style="list-style-type: none">- Connaître les grandes lignes du modèle client-serveur- Comprendre ce qu'est une url et une url enrichie- Connaître la notion de protocole HTTP (et HTTPS pour sa version sécurisée)	<ul style="list-style-type: none">- Analyser des schémas et des lignes de code
Restitution des savoirs	<ul style="list-style-type: none">- Résumer ses recherches, sélectionner les points-clés pertinents	<ul style="list-style-type: none">- Rechercher des informations en ligne

Matériel et outils

- Ordinateurs et connexion Internet
- Fiche activité élève à imprimer

Tags

#web #modèle client-serveur #url #protocole HTTP #protocole HTTPS

Déroulé de l'activité

Introduction : (~15 minutes)

- **Présenter les objectifs de la séance (contenu théorique et productions attendues) (2-3 minutes)**
- **Introduire les notions : (~10 minutes)**

Pour lancer la thématique, on peut lancer une première discussion sur le Web. L'objectif est de vérifier que les élèves ont assimilé les concepts suivants :

- Signification et utilité d'une URL
- Composition d'une URL
- Signification du sigle HTTP et son utilité
- Signification du sigle HTTPS et son utilité
- Principaux domaines et leur signification

Dans la fiche activité, nous avons intégré une fiche « quiz », que les élèves peuvent remplir en binôme ou petits groupes, ou bien qui peut guider une discussion au sein de la classe.

Proposition de compléments à la correction :

- **Qu'est-ce qu'une URL et à quoi sert-elle ?**

URL signifie « Uniform Resource Locator ». Elle contient les informations nécessaires au navigateur web pour se connecter au serveur qui héberge le site ou la ressource, et y accéder.

- **Les adresses « <http://www.laposte.fr> » et « <http://www.laposte.net> » sont 2 chemins pour accéder au même site internet.**

Le moindre changement de caractère dans une URL signifie un changement d'emplacement dans le site internet (différente page), voire de site internet. Les extensions « Domaine de Premier Niveau – TLD » représentent l'origine ou la catégorie d'un site web.

Cette technique est notamment utilisée par des arnaqueurs pour tromper les usagers.

- **L'URL « <https://garage.pizza> » existe.**

On peut à ce moment demander aux élèves quels TLD ils connaissent, et voir à quoi ils correspondent.

Dans un premier temps, les TLD étaient peu nombreuses. Depuis une dizaine d'années, le type de TLD s'est grandement diversifié. Cela donne notamment aux propriétaires de sites davantage d'options pour choisir une extension qui correspond à leur activité, leur secteur ou leurs préférences.

- **Le protocole HTTP permet ...**

Les protocoles HTTP et HTTPS sont les plus couramment utilisés pour la navigation et l'échange d'informations sur le Web, bien qu'il en existe d'autres (FTP, ...). Ces protocoles sont les normes de communication utilisées sur Internet pour établir la connexion entre le client (navigateur) et le serveur.

On peut à ce moment demander aux élèves quelle est la différence entre HTTP et HTTPS : le protocole HTTP est un protocole non sécurisé, ce qui signifie que les données transférées entre le client et le serveur ne sont pas chiffrées. C'est là qu'intervient le protocole HTTPS, qui utilise une couche de chiffrement SSL/TLS pour sécuriser les échanges de données.

Activités : (1h30)

1. Le principe de base du Web
2. Les URL
3. Un peu de code ...
4. Les protocoles HTTP et HTTPS

Dans ces parties, les élèves répondent aux questions en autonomie ou par deux. L'enseignant.e corrige étape après étape.

Conclusion (15 minutes)

- **Discussion : (5 minutes)**

Pour conclure, on peut demander aux élèves de revenir sur les points clés retenus.

On peut également discuter de qui décide de créer un protocole réseau.

Pistes de réflexion :

La création de nouveaux protocoles réseau est généralement réalisée par des groupes d'experts et des organisations spécialisées, par exemple :

- L'Internet Engineering Task Force (IETF) : organisation qui développe et promeut les normes Internet.
- Le World Wide Web Consortium (W3C) : organisation internationale qui se consacre au développement de normes pour le Web.
- L'Internet Corporation for Assigned Names and Numbers (ICANN) : organisation responsable de la gestion des ressources Internet, y compris les noms de domaine et les adresses IP. Bien qu'ils ne créent pas directement de nouveaux protocoles, ils sont impliqués dans les processus de gouvernance et de coordination liés aux protocoles réseau.
- Les entreprises et organismes de recherche : De nombreuses entreprises technologiques et institutions de recherche développent également leurs propres protocoles réseau pour répondre à des besoins spécifiques. Par exemple, des entreprises comme Google peuvent créer des protocoles pour leurs services spécifiques.

Il est important de préciser que la création de nouveaux protocoles réseau implique souvent un processus de collaboration, d'expérimentation et de validation au sein de la communauté des experts et des organisations

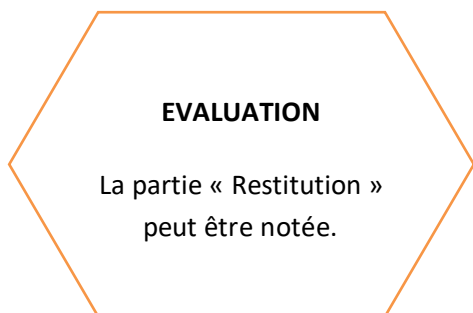
spécialisées. Par la suite, ces protocoles sont documentés et publiés sous forme de spécifications pour permettre leur application.

- **Restitution : (10 minutes)**

Les élèves préparent des supports de révision. L'enseignant choisit le support (qcm, affiches-résumé, schémas ...) et attribue à chaque groupe un thème, par exemple :

- Les différentes étapes décrivant un protocole HTTP
- Les différentes étapes décrivant un protocole HTTPS
- Les principaux protocoles réseau et leurs utilités
- La composition d'une URL
- ...

Il faudra prévoir une liste de critères et contraintes : nombre de questions, informations ou mots-clés, etc.



URL, protocole HTTP et modèle client-serveur

Fiche activité - Correction

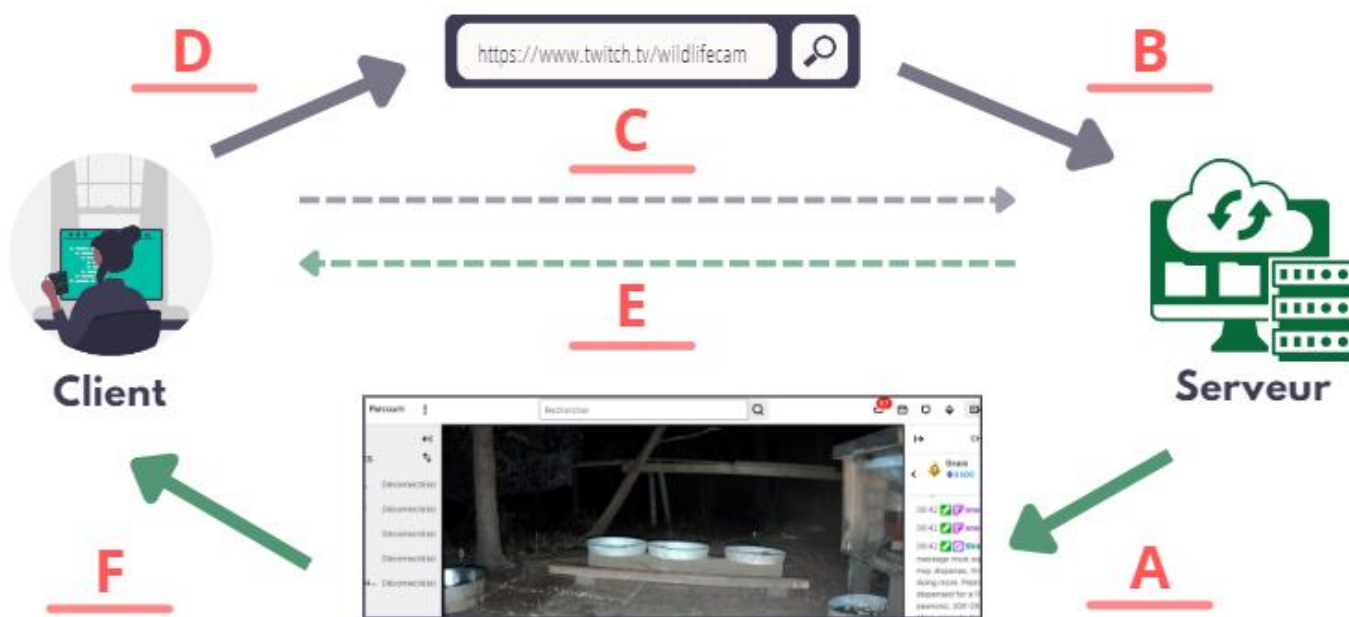
Introduction - Faisons le point ... Répondez aux questions suivantes :

- **1. Qu'est-ce qu'une URL et à quoi sert-elle ? (1 réponse possible)**
 - Une URL est un type de logiciel utilisé pour naviguer sur Internet.
 - Une URL est une adresse web qui identifie et localise une ressource sur Internet.
 - Une URL est un format de fichier spécifique utilisé pour stocker des données sur un ordinateur.
 - Une URL est une méthode de chiffrement utilisée pour sécuriser les transactions en ligne.
- **2. Les adresses « http://www.laposte.fr » et « http://www.laposte.net » sont 2 chemins pour accéder au même site web.**
 - Vrai
 - Faux
- **3. L'URL « https://garage.pizza » existe.**
 - Vrai
 - Faux
- **4. Le protocole HTTP permet ... (1 réponse possible)**
 - Aux navigateurs web de demander et récupérer des ressources sur le Web.
 - D'initialiser une connexion entre notre ordinateur et les satellites, pour ensuite pouvoir utiliser Internet.
 - De demander une connexion haut débit, stable. Il n'est pas indispensable dans une URL.
 - La transmission de données de manière sécurisée.

1. Le principe de base du Web

- Complétez le schéma suivant avec les propositions ci-dessous.

Le client et le serveur communiquent en utilisant un protocole particulier appelé protocole HTTP (= HyperText Transfer Protocol).



- a. Le serveur transmet la page qu'il a stockée ou qu'il crée dynamiquement en fonction de la requête formulée.
- b. Le navigateur web extrait de l'URL des informations qui lui permettent de se connecter à l'ordinateur (serveur) hébergeant le site web.
- c. Requête HTTP.
- d. On saisit une adresse URL dans son navigateur (client).
- e. Réponse HTTP.
- f. Le navigateur affiche la page web.

2. Les URL

Tout comme une adresse postale qui se décompose en plusieurs éléments tels que le numéro de rue, le nom de rue, le code postal, la ville, etc., une URL se décompose également en plusieurs éléments. Plus précisément, une URL est toujours constituée de la façon suivante :

protocole ://**hôte**.**ressource**/**chemin**.**ressource**

- **Le protocole** : C'est la partie qui spécifie le protocole de communication à utiliser, tel que HTTP ou HTTPS.
- **Le nom de domaine** : Il s'agit de l'adresse du site web, par exemple, "www.example.com".
- **Le chemin d'accès** : C'est la partie facultative qui indique l'emplacement spécifique de la ressource sur le serveur.

- **2.1. Identifier les différentes parties des URL suivantes :**

- a. <http://www.fftri.com/competitions/>

Protocole : **http**

Hôte : **www.fftri.com**

Chemin : **/competitions/**

- b. <http://www.fftri.com/competitions/competitions-nationales/triathlon/>

Protocole : **http**

Hôte : **www.fftri.com**

Chemin : **/competitions/competitions-nationales/triathlon/**

- c. <https://www.google.fr/>

Protocole : **https**

Hôte : **www.google.fr**

Chemin : **Aucun (page d'entrée du site)**

Dans les URL mentionnées, tous les chemins utilisés correspondent à des dossiers. Le navigateur se connecte à la page d'entrée de chaque dossier spécifié dans le chemin.

- **2.2. Les URL enrichies :**

Une URL peut indiquer un fichier HTML, une image, etc., **mais également un programme**. Dans ce cas, le schéma simplifié du modèle client-serveur présenté au paragraphe 1.1 devient plus complexe :



La partie "traitement" est effectuée avec les données fournies par le client au programme.

Observez l'adresse suivante : <http://www.google.com/search?q=gateau+pommes&hl=fr> et répondez aux questions :

- Surlignez la partie qui désigne le programme.

search?q=gateau+pommes&hl=fr

- Que fait ce programme ? Quelle partie indique au programme ce qu'il doit faire ?

Il permet d'effectuer une recherche plus poussée grâce à des critères. La partie « search » désigne le programme (*il s'agit d'un CGI – Common Gateway Interface*), et « ? » indique que les données suivantes sont celles à utiliser.

- Qu'indique q=gateau+pommes au programme ?

Ce sont les mots-clés de la recherche.

- Qu'indique hl=fr au programme ?

De sélectionner uniquement les sites en français.

Remarque : Les paramètres dépendent du programme et certains peuvent être très longs. De plus, les paramètres ne sont pas toujours envoyés au serveur via l'URL. Ils peuvent également être transmis via le réseau.

- **2.3. Saisir l'URL suivante : <http://www.adjectif.net/spip/spip.php>**

Une URL peut indiquer un fichier HTML, une image, etc., **mais également un programme**. Dans ce cas, le schéma simplifié du modèle client-serveur présenté au paragraphe 1.1 devient plus complexe :

- Indiquez comment s'appellent les parties suivantes :
 - spip : **chemin de dossier**
 - www.adjectif.net : **hôte**
 - http : **protocole**
 - spip.php : **chemin de fichier**
- Ajouter à cette URL l'élément "?article294", puis le remplacer par l'élément "?article288". Que constate-t-on ?

Une page web s'affiche et est modifiée en fonction de l'élément écrit après le "?" dans l'URL.

- Saisir l'URL <http://www.google.com/search?q=gateau+pommes&hl=fr> et observer le résultat obtenu.

Des liens sont proposés par le moteur de recherche.

- Si on saisit dans le moteur de recherche Google le texte "gâteau pommes », les résultats sont-ils différents ?

Les résultats sont identiques.

- Vérifiez votre hypothèse.

Remarque : Dans la très grande majorité des cas, on ne saisit pas directement dans l'URL les paramètres à envoyer au serveur. Ceux-ci sont saisis en ligne dans un formulaire et l'envoi est effectué automatiquement lorsque l'on clique sur le bouton de validation.

3. Un peu de code ...

- 3.1. Les échanges entre le client et le serveur : côté serveur

Un client remplit un formulaire sur un site qui lui demande son nom et son courriel. Les quatre documents ci-dessous correspondent aux échanges entre le client et le serveur. Observez les différentes étapes et remettez-les dans le bon ordre :

Document 1	
	Nom : <input type="text" value="darkvador"/> Courriel : <input type="text" value="darkvador@etoile.com"/> <input type="button" value="Valider"/>

Document 2	
<pre><!DOCTYPE HTML> <html lang='fr'> <head> <title>Connexion</title> <meta charset='utf-8' /> </head> <body> <p>Bienvenue darkvador
 Votre adresse de courriel est : darkvador@etoile.com</p> </body> </html></pre>	Bienvenue darkvador Votre adresse de courriel est : darkvador@etoile.com

Document 3	
<pre><!DOCTYPE HTML> <html lang='fr'> <head> <title>Connexion</title> <meta charset='utf-8' /> </head> <body> <form action='bienvenue.php' method='GET'> <p> Nom : <input type='text' name='nom' />
 Courriel : <input type='email' name='mail' />
 <input type='submit' value='Valider' /> </p> </form> </body> </html></pre>	Nom : <input type="text"/> Courriel : <input type="text"/> <input type="button" value="Valider"/>

Document 4	
<pre> <!DOCTYPE HTML> <html lang='fr'> <head> <title>Connexion</title> <meta charset='utf-8' /> </head> <body> <p>Bienvenue <?php echo \$_GET['nom']; ?>
 Votre adresse de courriel est : <?php echo \$_GET['mail']; ?></p> </body> </html> </pre>	<p>Bienvenue darkvador Votre adresse de courriel est : darkvador@etoile.com</p>

Ordre des étapes : 3 – 1 – 4 – 2

- **3.2. Les échanges entre le client et le serveur : côté client**

Parfois, le serveur envoie des pages web contenant un code de programme écrit en JavaScript qui est exécutable par le client. On donne les deux documents ci-dessous :

Document 1
<pre> <!DOCTYPE HTML> <html lang='fr'> <head> <title>JavaScript</title> <meta charset='utf-8' /> <script> function maFonction() { document.getElementById("demo").innerHTML = "Le paragraphe a changé."; } </script> </head> <body> <h1>Page web interactive</h1> <p id="demo">un Paragraphe</p> <button type="button" onclick="maFonction()">Appuyer</button>
 <button type="button" onclick="document.getElementById('demo1').innerHTML = Date()">Cliquer pour avoir la date.</button> <p id="demo1"></p> </body> </html> </pre>

Document 2
<h2>Page web interactive</h2> <p>un Paragraphe</p> <p>Appuyer</p> <p>Cliquer pour avoir la date.</p>

À l'aide de ces deux documents, répondez aux questions suivantes :

- À quelles lignes se trouve du code JavaScript ?

Le code JavaScript se trouve entre les balises `<script>` du conteneur d'en-tête `<head>` et les deux lignes correspondantes à la balise `<button>` du conteneur de corps `<body>`.

- Quel code permet de créer le bouton « Appuyer » ?

Le code permettant de créer le bouton se trouve à la troisième ligne du conteneur `<body>`.

- Que se passe-t-il lorsque l'on clique dessus ?

Lorsque l'on clique sur le bouton, la fonction "maFonction" est exécutée et le texte présent au-dessus du bouton est modifié.

- Quel code permet de créer le bouton « Cliquez pour avoir la date » ?

Le code permettant de créer le bouton se trouve à la cinquième ligne du conteneur `<body>`.

- Que se passe-t-il lorsque l'on clique dessus ?

Lorsque l'on clique sur le bouton, la date du serveur (local) s'affiche en dessous du bouton.

4. Les protocoles HTTP et HTTPS

Le protocole HTTP : C'est le protocole utilisé par les navigateurs web (Firefox, Chrome, Edge, etc.) et les serveurs web (Apache, IIS, etc.) pour communiquer entre eux.

Cette communication se déroule en 2x trois temps :

- **Requête HTTP du client :**

1. Le client émet une commande de demande de ressource qu'il transmet au serveur, par exemple :

```
GET/page.html HTTP/1.0
```

2. Le client envoie éventuellement des informations d'en-tête pour informer le serveur de sa configuration et des formats de documents qu'il peut prendre en charge, par exemple :

```
Host: example.com
Accept: image/png, image/jpeg, */*
Referer: http://example.com/
User-Agent: Mozilla/5.0
```

3. Le client termine les informations d'en-tête par une ligne blanche. Ensuite, il peut envoyer des données supplémentaires principalement utilisées par les programmes utilisant la méthode *POST*.

- **Réponse HTTP du serveur :**

1. Le serveur émet une ligne d'état contenant trois champs :

- La version HTTP utilisée par le serveur pour sa réponse.
- Un code d'état à trois chiffres qui indique le résultat de l'interprétation de la requête client.
- Une description textuelle associée au code d'état.

Par exemple, en cas de réussite, on a :

```
HTTP/1.0 200 OK
```

2. Le serveur envoie des informations sur sa configuration et sur la ressource sollicitée, par exemple :

```
Date: Fri 31 Dec 2018 23:59:59 GMT
Server: Apache/0.8.4
Content-Type: text/html
Content-Length: 59
Expires: Sat 01 Jan 2000 00:59:59 GMT
Last-modified: Fri 09 Aug 1996 14:21:40 GMT
```

Une ligne blanche termine l'en-tête.

3. Les données sont envoyées au client, par exemple :

```
<title>Exemple</title>
<p>Ceci est une page d'exemple.</p>
```

• 4.1. À quoi correspondent les codes suivants ?

HTTP 102	HTTP 400	HTTP 405
Traitement en cours	Syntaxe requête erronée	Ressource non trouvée
HTTP 200	HTTP 401	HTTP 500
Méthode traitée avec succès	Authentification nécessaire	Erreur interne du serveur
HTTP 302	HTTP 403	HTTP 504
Document déplacé temporairement	Accès interdit	Temps d'attente de la réponse écoulé

Plus précisément, on a la classification suivante :

- 1xx : Informations
- 2xx : Succès
- 3xx : Redirection
- 4xx : Erreur du client web
- 5xx : Erreur du serveur web

- **4.2. Le protocole HTTPS :**

Ce protocole signifie *HyperText Transfer Protocol Secure* et résulte de la combinaison du protocole HTTP avec une couche de chiffrement. Il permet notamment d'authentifier le site web via un certificat numérique.

Répondre, à l'aide d'Internet, aux questions suivantes :

- Quel est le principe de l'attaque appelée "attaque de l'homme du milieu" ? Quelles en sont les parades ?

L'attaque de l'homme du milieu est une attaque qui vise à intercepter les communications entre deux parties sans que ni l'une ni l'autre ne se rendent compte que le canal de communication a été compromis. Le canal le plus couramment utilisé est une connexion à Internet. Ainsi, l'attaquant a la possibilité de lire et de modifier les messages échangés.

Pour se prémunir contre cette attaque, il existe plusieurs parades :

- Obtenir la clé publique de son interlocuteur par le biais d'un tiers de confiance.
- Échanger les clés de manière sécurisée en utilisant un moyen qui ne permet pas cette attaque.
- Vérifier le niveau de confiance accordé à la clé que l'on possède.
- S'authentifier avec un mot de passe ou un système avancé tel que la reconnaissance vocale ou biologique.

- Qu'est-ce que le phishing ? Quelles en sont les parades ?

Le phishing, ou hameçonnage, est une technique utilisée par des fraudeurs pour obtenir des renseignements personnels dans le but de commettre une usurpation d'identité. Cette technique consiste à faire croire à la victime qu'elle s'adresse à un tiers de confiance (banque, administration, etc.) afin de lui soutirer des informations personnelles telles que des mots de passe, des numéros de carte de crédit, etc.

Pour se prémunir contre le phishing, il existe plusieurs parades :

- Vérifier l'orthographe du nom de domaine.
- Vérifier l'absence d'arobase dans l'URL.
- Vérifier l'absence de caractère Unicode suspect.
- Vérifier les certificats électroniques.
- Écrire manuellement les URL plutôt que de cliquer sur des liens douteux.
- Savoir qu'aucune banque ou administration ne demande de données personnelles via un e-mail.

URL, protocole HTTP et modèle client-serveur

Fiche activité élève

Introduction - Faisons le point ... Répondez aux questions suivantes :

- **1. Qu'est-ce qu'une URL et à quoi sert-elle ? (1 réponse possible)**
 - Une URL est un type de logiciel utilisé pour naviguer sur Internet.
 - Une URL est une adresse web qui identifie et localise une ressource sur Internet.
 - Une URL est un format de fichier spécifique utilisé pour stocker des données sur un ordinateur.
 - Une URL est une méthode de chiffrement utilisée pour sécuriser les transactions en ligne.
- **2. Les adresses « <http://www.laposte.fr> » et « <http://www.laposte.net> » sont 2 chemins pour accéder au même site internet.**
 - Vrai
 - Faux
- **3. L'URL « <https://garage.pizza> » existe.**
 - Vrai
 - Faux
- **4. Le protocole HTTP permet ... (1 réponse possible)**
 - Permet aux navigateurs web de demander et récupérer des ressources sur le Web.
 - Permet d'initialiser une connexion entre notre ordinateur et les satellites, pour ensuite pouvoir utiliser Internet.
 - De demander une connexion haut débit, stable. Il n'est pas indispensable dans une URL.
 - La transmission de données de manière sécurisée.

1. Le principe de base du Web

- Complétez le schéma suivant avec les propositions ci-dessous.

Le client et le serveur communiquent en utilisant un protocole particulier appelé protocole HTTP (= HyperText Transfer Protocol).



- a. Le serveur transmet la page qu'il a stockée ou qu'il crée dynamiquement en fonction de la requête formulée.
- b. Le navigateur web extrait de l'URL des informations qui lui permettent de se connecter à l'ordinateur (serveur) hébergeant le site web.
- c. Requête HTTP.
- d. On saisit une adresse URL dans son navigateur (client).
- e. Réponse HTTP.
- f. Le navigateur affiche la page web.

2. Les URL

Tout comme une adresse postale qui se décompose en plusieurs éléments tels que le numéro de rue, le nom de rue, le code postal, la ville, etc., une URL se décompose également en plusieurs éléments. Plus précisément, une URL est toujours constituée de la façon suivante :

protocole **://**hôte_ressource/chemin_ressource

- **Le protocole** : C'est la partie qui spécifie le protocole de communication à utiliser, tel que HTTP ou HTTPS.
- **Le nom de domaine** : Il s'agit de l'adresse du site web, par exemple, "www.example.com".
- **Le chemin d'accès** : C'est la partie facultative qui indique l'emplacement spécifique de la ressource sur le serveur.

● **2.1. Identifiez les différentes parties des URL suivantes :**

d. <http://www.fftri.com/competitions/>

Protocole :

Hôte :

Chemin :

e. <http://www.fftri.com/competitions/competitions-nationales/triathlon/>

Protocole :

Hôte :

Chemin :

f. <https://www.google.fr/>

Protocole :

Hôte :

Chemin :

● **2.2. Les URL enrichies :**

Une URL peut indiquer un fichier HTML, une image, etc., **mais également un programme**. Dans ce cas, le schéma simplifié du modèle client/serveur présenté au paragraphe 1.1 devient plus complexe :



La partie "traitement" est effectuée avec les données fournies par le client au programme.

Observez l'adresse suivante : <http://www.google.com/search?q=gateau+pommes&hl=fr> et répondez aux questions :

- Surlignez la partie qui désigne le programme.
- Que fait ce programme ? Quelle partie indique au programme ce qu'il doit faire ?

.....

.....

- Qu'indique q=gateau+pommes au programme ?

.....

.....

- Qu'indique hl=fr au programme ?

.....

.....

Remarque : Les paramètres dépendent du programme et certains peuvent être très longs. De plus, les paramètres ne sont pas toujours envoyés au serveur via l'URL. Ils peuvent également être transmis via le réseau.

- Saisissez l'URL <http://www.google.com/search?q=gateau+pommes&hl=fr> et observez le résultat obtenu.

.....

.....

- Si on saisit dans le moteur de recherche Google le texte « gâteau pommes », les résultats sont-ils différents ?

.....

.....

- Vérifiez votre hypothèse.

Remarque : Dans la très grande majorité des cas, on ne saisit pas directement dans l'URL les paramètres à envoyer au serveur. Ceux-ci sont saisis en ligne dans un formulaire et l'envoi est effectué automatiquement lorsque l'on clique sur le bouton de validation.

● **2.3. Saisissez l'URL suivante : <http://www.adjectif.net/spip/spip.php>**

- Indiquez comment s'appellent les parties suivantes :

spip

www.adjectif.net

Http

spip.php

- Ajoutez à cette URL l'élément "?article294", puis le remplacer par l'élément "?article288". Que constatez-vous ?

3. Un peu de code ...

- 3.1. Les échanges entre le client et le serveur : côté serveur

Un client remplit un formulaire sur un site qui lui demande son nom et son courriel. Les quatre documents ci-dessous correspondent aux échanges entre le client et le serveur. Observez les différentes étapes et remettez-les dans le bon ordre :

Document 1	
	Nom : <input type="text" value="darkvador"/> Courriel : <input type="text" value="darkvador@etoile.com"/> <input type="button" value="Valider"/>

Document 2	
<pre><!DOCTYPE HTML> <html lang='fr'> <head> <title>Connexion</title> <meta charset='utf-8' /> </head> <body> <p>Bienvenue darkvador
 Votre adresse de courriel est : darkvador@etoile.com</p> </body> </html></pre>	Bienvenue darkvador Votre adresse de courriel est : darkvador@etoile.com

Document 3

```
<!DOCTYPE HTML>
<html lang='fr'>
  <head>
    <title>Connexion</title>
    <meta charset='utf-8' />
  </head>
  <body>
    <form action='bienvenue.php' method='GET'>
      <p>
        Nom : <input type='text' name='nom' /><br/>
        Courriel : <input type='email' name='mail' /><br/>
        <input type='submit' value='Valider' />
      </p>
    </form>
  </body>
</html>
```

Nom :
Courriel :

Document 4

```
<!DOCTYPE HTML>
<html lang='fr'>
  <head>
    <title>Connexion</title>
    <meta charset='utf-8' />
  </head>
  <body>
    <p>Bienvenue <?php echo $_GET['nom']; ?><br/>
    Votre adresse de courriel est : <?php echo $_GET['mail']; ?></p>
  </body>
</html>
```

Bienvenue darkvador
Votre adresse de courriel est : darkvador@etoile.com

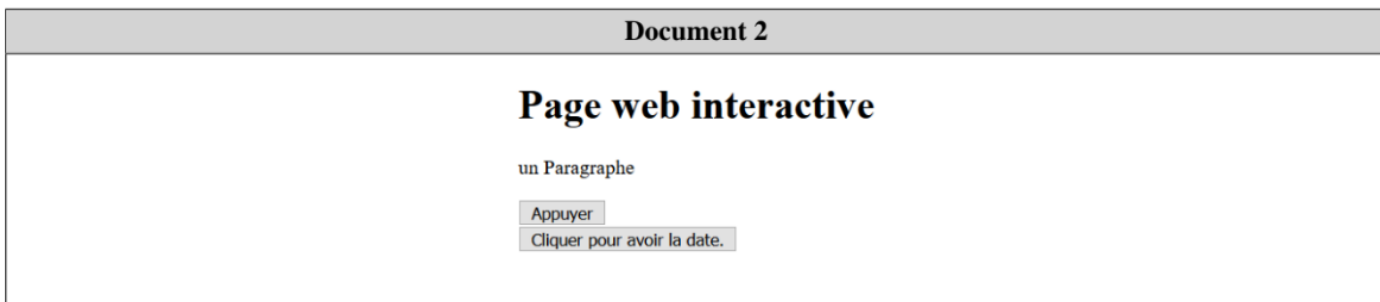
Ordre des étapes :

- **3.2. Les échanges entre le client et le serveur : côté client**

Parfois, le serveur envoie des pages web contenant un code de programme écrit en JavaScript qui est exécutable par le client. On donne les deux documents ci-dessous :

Document 1

```
<!DOCTYPE HTML>
<html lang='fr'>
<head>
  <title>JavaScript</title>
  <meta charset='utf-8' />
  <script>
    function maFonction()
    {
      document.getElementById("demo").innerHTML = "Le paragraphe a changé.";
    }
  </script>
</head>
<body>
  <h1>Page web interactive</h1>
  <p id="demo">un Paragraphe</p>
  <button type="button" onclick="maFonction()">Appuyer</button>
  <br/>
  <button type="button" onclick="document.getElementById('demo1').innerHTML = Date()">Cliquer pour avoir la date.</button>
  <p id="demo1"></p>
</body>
</html>
```



À l'aide de ces deux documents, répondez aux questions suivantes :

- À quelles lignes se trouve du code JavaScript ?

.....

.....

- Quel code permet de créer le bouton « Appuyer » ?

.....

.....

- Que se passe-t-il lorsque l'on clique dessus ?

.....

.....

- Quel code permet de créer le bouton « Cliquez pour avoir la date » ?

.....

.....

- Que se passe-t-il lorsque l'on clique dessus ?

.....

.....

4. Les protocoles HTTP et HTTPS

Le protocole HTTP : C'est le protocole utilisé par les navigateurs web (Firefox, Chrome, Edge, etc.) et les serveurs web (Apache, IIS, etc.) pour communiquer entre eux.

Cette communication se déroule en 2x trois temps :

- **Requête HTTP du client :**

1. Le client émet une commande de demande de ressource qu'il transmet au serveur, par exemple :

```
GET/page.html HTTP/1.0
```

2. Le client envoie éventuellement des informations d'en-tête pour informer le serveur de sa configuration et des formats de documents qu'il peut prendre en charge, par exemple :

```
Host: example.com
Accept: image/png, image/jpeg, */*
Referer: http://example.com/
User-Agent: Mozilla/5.0
```

3. Le client termine les informations d'en-tête par une ligne blanche. Ensuite, il peut envoyer des données supplémentaires principalement utilisées par les programmes utilisant la méthode *POST*.

- **Réponse HTTP du serveur :**

1. Le serveur émet une ligne d'état contenant trois champs :
 - La version HTTP utilisée par le serveur pour sa réponse.
 - Un code d'état à trois chiffres qui indique le résultat de l'interprétation de la requête client.

- Une description textuelle associée au code d'état.

Par exemple, en cas de réussite, on a :

```
HTTP/1.0 200 OK
```

2. Le serveur envoie des informations sur sa configuration et sur la ressource sollicitée, par exemple :

```
Date: Fri 31 Dec 2018 23:59:59 GMT
Server: Apache/0.8.4
Content-Type: text/html
Content-Length: 59
Expires: Sat 01 Jan 2000 00:59:59 GMT
Last-modified: Fri 09 Aug 1996 14:21:40 GMT
```

Une ligne blanche termine l'en-tête.

3. Les données sont envoyées au client, par exemple :

```
<title>Exemple</title>
<p>Ceci est une page d'exemple.</p>
```

• **4.1. À quoi correspondent les codes suivants ?**

HTTP 102	HTTP 400	HTTP 405
.....
.....
HTTP 200	HTTP 401	HTTP 500
.....
.....
HTTP 302	HTTP 403	HTTP 504
.....
.....

