

# Le World Wide Web

On a vu ce qu'était Internet : un réseau de réseaux interconnectés à l'échelle mondiale, basé sur les protocoles TCP et IP. Il est ainsi possible à des ordinateurs de communiquer de manière fiable et ce sur des distances potentiellement très importantes.

Internet n'est cependant qu'une *infrastructure* : si des ordinateurs communiquent effectivement les uns avec les autres que peuvent-ils alors échanger et comment ?

Le **World Wide Web** (souvent abrégé en **Web**) est un *système utilisant le réseau Internet*, mis au point par (entre autres) Tim Berners Lee dès 1989 au CERN, permettant **d'accéder à un ensemble de ressources** (pages, sons, vidéos, documents, fichiers quelconques) **reliées à l'aide de liens hypertextes**.

Il existe évidemment d'autres systèmes utilisant le réseau Internet pour proposer des services différents, par exemple :

- la messagerie électronique ;
- la messagerie instantanée (« chats ») ;
- la VoIP (téléphonie via Internet) ;
- les jeux multijoueurs ;
- ...

Le fonctionnement du Web se base sur plusieurs technologies et principes :

- le **modèle serveur-client**, qui explicite où sont localisées les ressources et comment elles sont partagées ;
- les protocoles **HTTP** (*Hypertext Transfert Protocol*) et **HTTPS** qui régissent les interactions entre les clients et serveurs ;
- le **HTML** (*Hypertext Markup Language*) qui structure le contenu des pages Web ;
- les **navigateurs** qui permettent d'afficher les pages Web et de suivre les liens hypertextes.

Au fil du temps sont également apparus des **moteurs de recherche** sur le Web, qui proposent des listes de pages Web relatives à des mots clés et permettent trouver de nouvelles pages sans connaître directement leur adresse.

## 1) Fonctionnement du Web :

### a) Modèle serveur-client :

Dans un modèle serveur-client, les ressources sont situées sur un ordinateur, appelé « **serveur** », qui *répond aux requêtes* d'autres ordinateurs qui veulent y accéder, appelés « **clients** ».

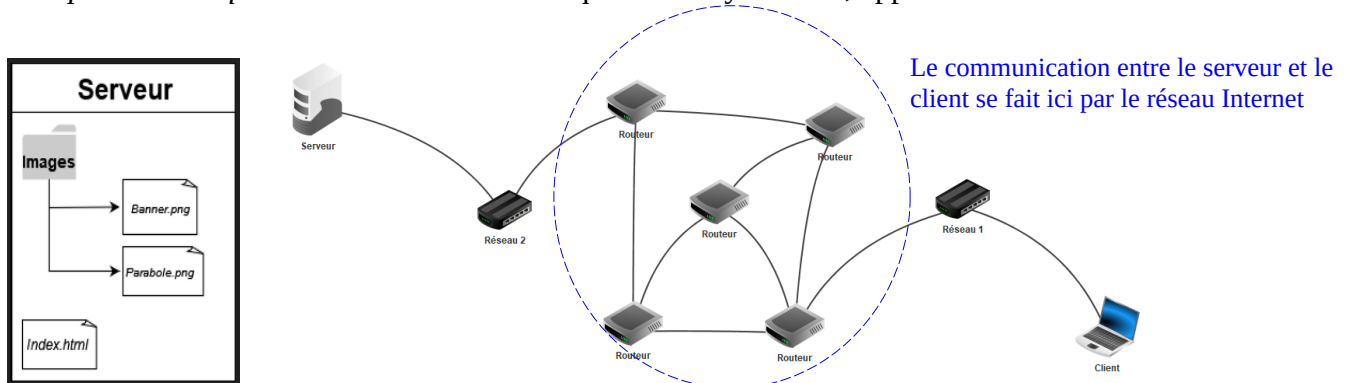


Illustration du modèle serveur-client :

► on a, d'une part, un **serveur** disposant de trois ressources :

- un fichier HTML nommé *index.HTML* qui n'est pas un dossier (on dit qu'il est à la *racine* du serveur) ;
- deux fichiers images au format PNG (Portable Network Graphic) nommés *Banner.png* et *Parabole.png*, tous les deux situés dans un dossier nommé *Images*.

► d'autre part, un **client** qui peut envoyer des *requêtes* au serveur pour accéder à ces ressources.

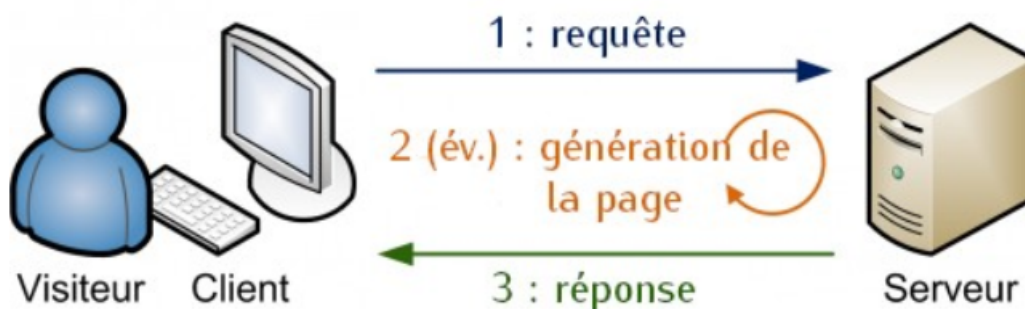
Ces deux machines sont ici reliées par l'intermédiaire du réseau Internet (mais ce n'est pas obligatoire).

## b) Le protocole HTTP :

On l'a vu, pour communiquer des ordinateurs ont besoin de respecter les mêmes règles dans leurs échanges, qui sont spécifiées dans des protocoles. La communication sur Internet est basée sur les protocoles TCP et IP, la communication sur le Web sur le protocole HTTP.

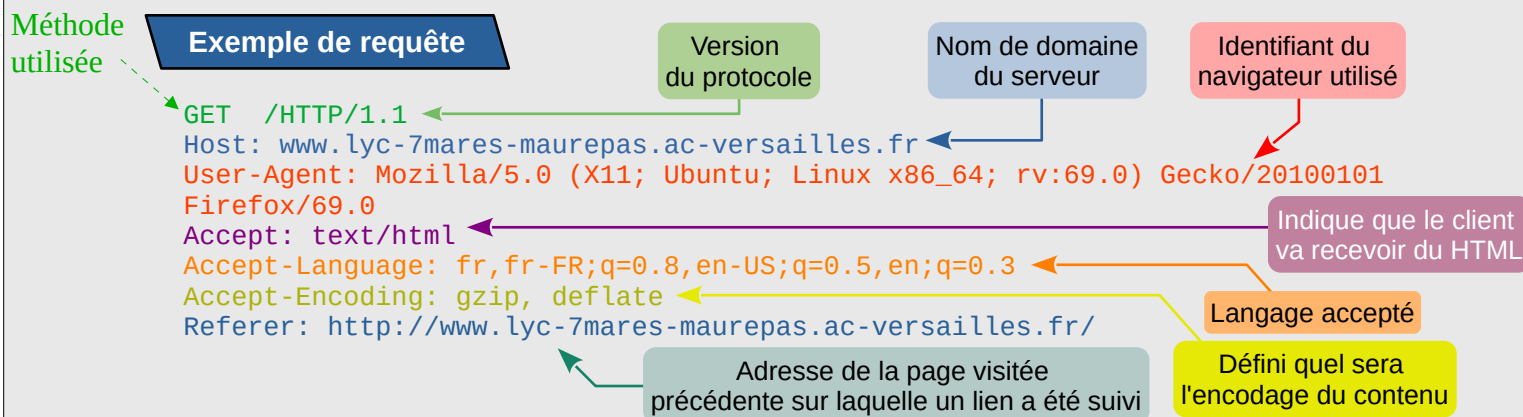
Comment fonctionne (sommairement) le protocole HTTP :

1. Le client émet une **requête** vers le serveur, sous forme textuelle, qui contient un **verbe** (ou **méthode**, ce sont des synonymes ici) pour décrire le type de requête.
2. Le serveur traite la requête, en générant une page si besoin ;



3. Le serveur renvoie une réponse au client qui comporte un code pour lui indiquer comment la requête a été traitée, éventuellement accompagnée de la ressource.

### Exemple de requête



La première ligne commence par le verbe et chaque entête de ligne suivante (tout ce qui est avant « : ») décrit l'information que contient la ligne. Seules les deux premières lignes sont obligatoires.

Quelques exemples de méthodes :

- **GET** : le client demande à obtenir la ressource (c'est-à-dire que le serveur lui envoie) ;
- **PUT** : le client demande à ajouter une ressource sur le serveur (ou la remplacer si elle existe déjà) ;
- **HEAD** : le client demande seulement les informations sur la ressource (sans qu'on lui envoie).

## Exemple de réponse

### Exemple de réponse du serveur

```
HTTP/1.1 200 OK
Date: Fri, 25 Oct 2019 16:44:59 GMT
Server: Apache/2.2.22 (Debian)
X-Powered-By: PHP/5.2.17-0.crdp.0
P3P: CP="NOI ADM DEV PSAI COM NAV OUR OTRo STP IND DEM"
ETag: 38f2df76b2d0d56bfe268b7f84a9f8af
Cache-Control: no-cache
Pragma: no-cache
Set-Cookie: c9alf7677a29c18bee24af0dd03599a1=fr-FR; expires=Sat, 24-Oct-2020 16:44:59 GMT; path=/
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 6127
Connection: close
Content-Type: text/html
```

```
<!DOCTYPE html>
<html dir="ltr" lang="fr-fr">
<head>
  <base href="http://www.lyc-7mares-maurepas.ac-versailles.fr/Joomla/index.php/fr/" />
  <meta http-equiv="content-type" content="text/html; charset=utf-8" />
  <meta name="keywords" content="lycée, lycée, pédagogie, pédagogie, L7M L7m, lycée les 7 mares, lycée sept mares, 7mares maurepas, 7 mares élanecourt, 7 mares élanecourt, BTS , BTSNRC, NRC" />
  <meta name="description" content="Site officiel du lycée les 7 mares - Maurepas bassin de Saint Quentin en Yvelines - Yvelines" />
  <meta name="generator" content="Joomla! - Open Source Content Management" />
</head>
<title>Lycée Les Sept Mares - Accueil</title>
```

En-tête de réponse

Contenu HTML  
qui permet à votre navigateur  
d'afficher la page web



Elle se compose :

- d'une en-tête qui contient le **code de réponse** et des informations sur le traitement de la requête (même format que pour la requête, par exemple ici *Content-Length* : 6127 signifie que la réponse fait 6 127 octets.).
- éventuellement de la ressource demandée (ici le contenu de la page Web).

Quelques exemples de codes réponses :

- 200 (OK) : requête correctement traitée
- 404 (Not Found) : la ressource demandée n'a pas été trouvée
- 403 (Forbidden) : refus d'exécuter la requête demandée (généralement à cause d'un problème de droit)

### c) URL d'une ressource :

Pour qu'un client puisse accéder à une ressource sur un serveur, on doit nécessairement avoir :

1. un moyen de spécifier le protocole utilisé (ici le protocole HTTP)

► pour identifier le protocole utilisé, on a décidé de l'écrire en toutes lettres suivi de « :// »

2. un moyen d'identifier le serveur qu'il doit contacter ;

► il suffit de se rappeler que le serveur est connecté à Internet, donc il est identifiable par son **adresse IP**. En pratique, on préférera toujours utiliser, lorsqu'il existe, le **nom de domaine** associé.

3. un moyen d'identifier la ressource à laquelle il désire accéder sur le serveur en question.

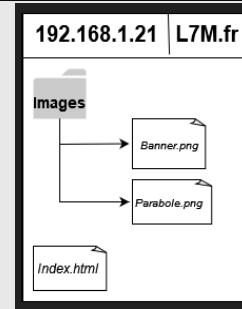
► on a choisi d'utiliser le **nom de la ressource** (avec son extension) sur le serveur, en spécifiant le **chemin d'accès** jusqu'à elle à l'aide de « / » pour indiquer les dossiers.

La combinaison de ces trois identifiants est appelée l'**URL** de la ressource.

## Exemples :

On considère le serveur ci-contre :

► Le fichier *Index.html* a pour URL :



ou encore, en utilisant directement l'adresse IP du serveur :



► Le fichier *Banner.png* a pour URL :



## d) Protocole HTTPS :

Lorsque le Web (et donc le protocole HTTP) a été créé, la sécurité des communications n'était pas une priorité ; les échanges de données via le protocole HTTP se font donc en clair (sans être chiffrés).

Ainsi, les **données envoyées sont lisibles par tous les intermédiaires** (proxy, routeurs, commutateurs, etc) entre le client et le serveur. Une personne qui surveillerait le trafic réseau pourrait ainsi obtenir beaucoup d'informations, parfois confidentielles (mots de passe, identifiants, etc).

Pour éviter cela, on donc mis au point le protocole HTTPS (HTTP **Secure**) qui fonctionne comme le protocole HTTP mais ajoute **une couche de chiffrement des données** (c'est-à-dire qu'elles sont illisibles telles quelles, sauf pour le destinataire). Si le chiffrement est bien effectué, il est alors très difficile pour quelqu'un interceptant les données de prendre connaissance de leur contenu.

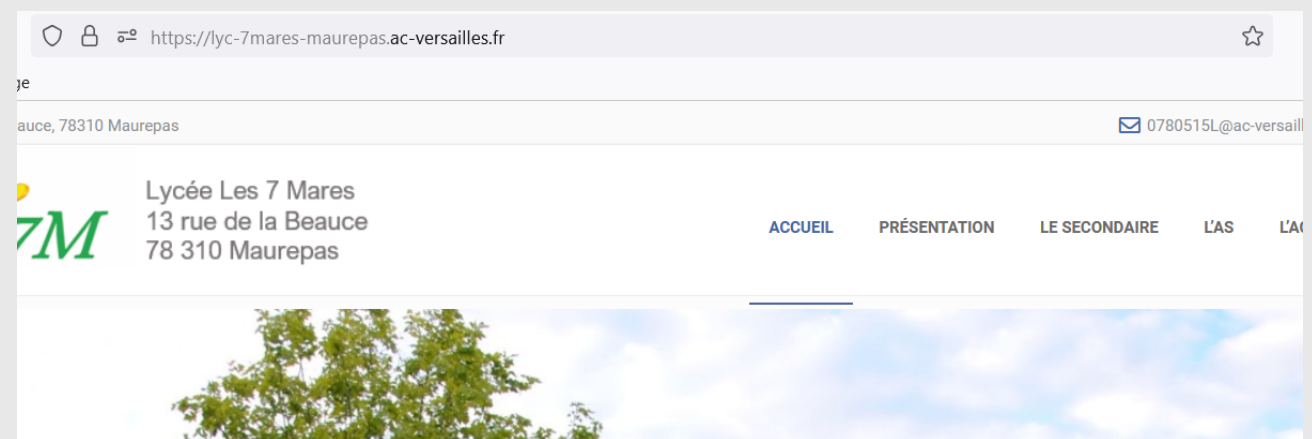
## 2) Navigateurs et moteurs de recherche :

### a) Navigateurs Web :

Un navigateur Web est un **logiciel** (installé sur un client) qui permet, comme son nom l'indique, de naviguer sur le Web en automatisant plusieurs tâches :

- il gère la création et l'envoi des requêtes HTTP ou HTTPS;
- il gère le traitement des réponses des serveurs ;
- il affiche le contenu des pages Web visitées (c'est-à-dire envoyées par les serveurs) et, pour certains, le contenu d'autres ressources reçues (image, son, etc).

Pour utiliser un navigateur, il suffit de spécifier l'URL de la ressource à laquelle on souhaite accéder :



Le navigateur a ici :

- envoyé une requête au serveur dont le nom de domaine est « *lyc-7mares-maurepas.ac-versailles.fr* » en utilisant le protocole HTTPS et en demandant la page principale (puisque l'on n'a spécifié aucune ressource particulière du serveur) ;
- reçu la réponse du serveur avec la page principale ;
- affiché la page Web en la mettant en page.

Quelques exemples de navigateurs :

- *Mozilla Firefox*
- *Microsoft Edge*
- *Google Chrome*
- *Safari*

Mozilla, Microsoft et Google étant les noms des entreprises développant ces navigateurs.

## **b) Moteurs de recherche :**

Un moteur de recherche est un **site Web** permettant d'obtenir une liste de sites Web que l'algorithme du moteur de recherche trouve pertinent par rapport aux mots clés entrés par l'utilisateur.

### **Fonctionnement d'un moteur de recherche**

Un moteur de recherche fonctionne en trois étapes :

- A lieu en permanence et pas seulement lors d'une requête
1. **Collecte de données** : Des robots d'exploration (appelés "crawlers" ou "spiders") explorent le Web en suivant les liens hypertextes qu'ils trouvent et récupérant des données sur les sites visités.
  2. **Indexation** : Les données collectées sont stockées dans des centres de données (*data centers*), triées et organisées selon des critères propres à chaque moteur de recherche.
  3. **Traitement des requêtes** : *en réponse aux requêtes des utilisateurs*, le moteur de recherche propose une liste de sites Web que son **algorithme** aura déterminé **comme pertinent par rapport aux mots clés** de la requête. On appelle cela le **référencement naturel** des sites.

#### Remarques :

- En plus du référencement naturel, certains moteurs de recherche permettent à des entreprises « **d'acheter des liens** » pour apparaître en tête de liste dans ses réponses pour certains mots clés. Ces liens doivent maintenant signaler qu'on a payé pour eux (*sponsorisé, commercial*, etc).
- Certains (beaucoup) de moteurs de recherche utilisent les données personnelles des utilisateurs pour personnaliser leurs résultats (analyse des mots clés tapés pour construire un profil marchand de l'utilisateur par exemple).

#### Quelques exemples de moteurs de recherche :

- *Google* (qui était également le nom de l'entreprise) ;
- *Microsoft Bing* ;
- *Qwant* ;
- *Ecosia*.

### **3) Pages Web, HTML et CSS :**

On a vu comment un navigateur pouvait communiquer avec un serveur pour récupérer des pages Web, mais comment peuvent-elles être interprétées et affichées par le navigateur ?

Les pages Web modernes reposent sur trois langages :

- le HTML (*HyperText Markup Language* ou *Langage de Balise Hypertexte*), un langage de description, qui **décrit le contenu** de la page Web ;
- le CSS (*Cascading Style Sheet* ou *Feuillet de Style en Cascade*) qui va décrire **la présentation ou la mise en page** du contenu
- le JavaScript, un langage de programmation exécuté côté client, qui permet de rendre les pages **interactives** (envoi de formulaires,

Seuls le HTML et le CSS (plus simples) sont au programme de SNT.

#### **a) Le langage HTML :**

► Interprété par le navigateur Web lorsque la page s'affiche, le HTML est basé sur le concept de **balise** : des balises sont insérées dans la page Web et permettent de définir ce que représentent les différentes parties.

Ainsi chaque partie est délimitée par une balise ouvrante `<balise>` et une balise fermante `</balise>` .

#### Exemples :

- décrire un paragraphe : la balise `<p>`

```
<p>Contrairement à une opinion  
répandue, le Lorem Ipsum n'est pas...</p>
```

- décrire un titre de niveau 1 : `<h1>`

```
<h1>Titre principal de la page</h1>
```

► Certaines balises possèdent des **attributs** qui permettent de préciser certaines données (taille d'une image, cible d'un lien, etc). Ils se placent dans la balise ouvrante et sont suivis d'un =.

#### Exemple :

Pour décrire un lien vers un site, on utilise la balise `<a>`. Elle a besoin de connaître la cible du lien et on utilise pour cela l'attribut **href**.

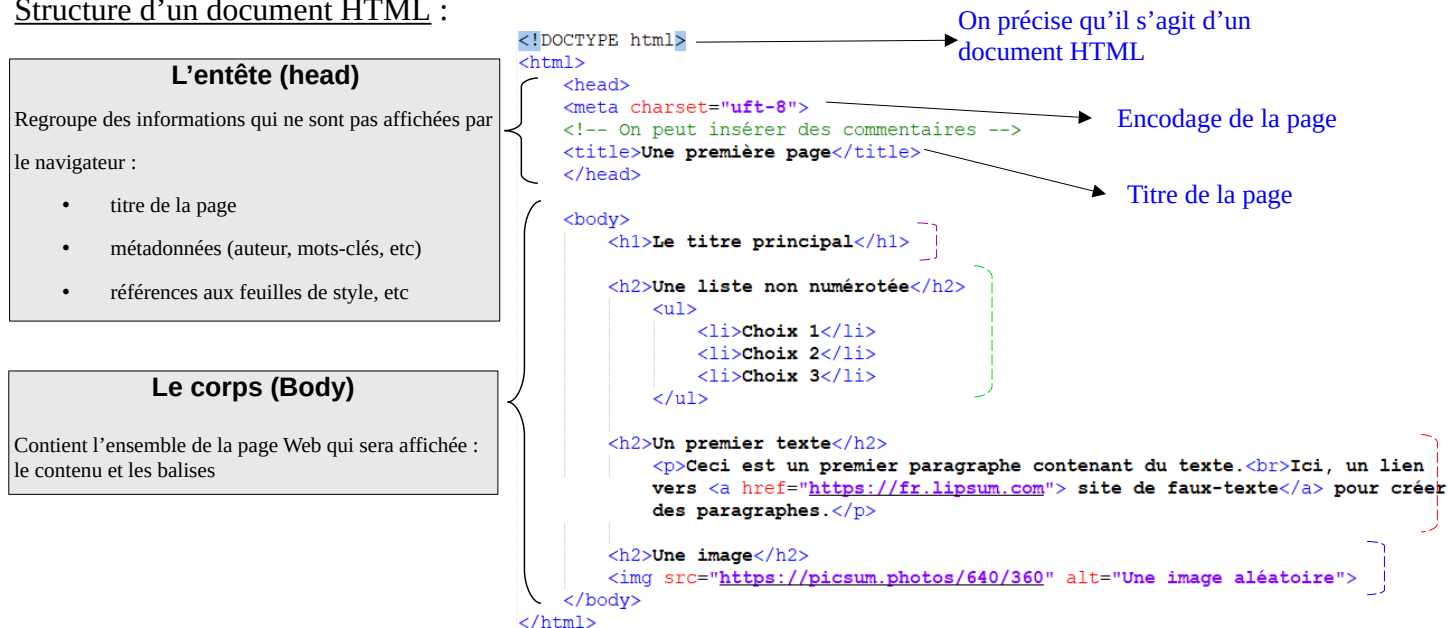
```
<a href="https://fr.lipsum.com">Le site  
Lorem ipsum</a>
```

→ Lien vers le site <https://fr.lipsum.com>

► On peut évidemment imbriquer les balises :

`<p>Un lien vers <a href="https://fr.lipsum.com">le site Lorem ipsum</a> sur lequel vous pouvez cliquer.</p>` → Lien vers le site <https://fr.lipsum.com> au sein d'un paragraphe

## Structure d'un document HTML :



La page précédente interprétée par le navigateur :

### Le titre principal

#### Une liste non numérotée

- Choix 1
- Choix 2
- Choix 3

#### Un premier texte

Ceci est un premier paragraphe contenant du texte.  
Ici, un lien vers [site de faux-texte](https://fr.lipsum.com) pour créer des paragraphes.

#### Une image



## b) Le langage CSS :

Les premières pages Web ne séparaient pas le contenu et la présentation rendant difficile la refonte des sites Web puisqu'il fallait reprendre toute la partie HTML pour en changer le contenu.

Il a donc été décidé de séparer ces deux notions : le HTML décrirait le contenu et le CSS sa mise en page.



## ► Structure d'un fichier CSS :

Un fichier CSS est une suite de déclarations de la forme dans lesquelles :

```
sélecteur {  
propriété1: valeur1;  
propriété2: valeur2;  
}
```

- le **sélecteur** sera (pour nous) la balise HTML (body ; h1 ; p, etc.) à laquelle on doit appliquer la mise en page ;
- la **propriété** est l'attribut qu'on veut appliquer ou modifier (font ; background ; margin ; etc.) ;
- la **valeur** précise les caractéristiques de la propriété.

Exemple :

La déclaration suivante :

- change la couleur du fond de chaque paragraphe en bleu
- affiche le texte dans les paragraphes en rouge et en italique :

```
p {  
background-color: blue;  
color:red;  
font-style:italic;  
}
```

► Pour qu'un feuillet de style soit utilisé par une page Web, il faut l'indiquer dans son entête :

```
<head>  
<link href="style.css" rel="stylesheet">  
...  
</head>
```

Utiliser le fichier `style.css` comme feuillet de style

Exemple de mise en page CSS de la page de la section précédente :

## Feuillet de Style CSS

```
body {  
background-color:grey;  
font-family: liberation sans;  
}  
  
h1 {  
background-color:#636363;  
}  
  
h2 {  
color:orange;  
}  
  
p {  
color:white;  
}
```

### Le titre principal

#### Une liste non numérotée

- Choix 1
- Choix 2
- Choix 3

#### Un premier texte

Ceci est un premier paragraphe contenant du texte.  
Ici, un lien vers [site de faux-texte](#) pour créer des paragraphes.

#### Une image



### Le titre principal

#### Une liste non numérotée

- Choix 1
- Choix 2
- Choix 3

#### Un premier texte

Ceci est un premier paragraphe contenant du texte.  
Ici, un lien vers [site de faux-texte](#) pour créer des paragraphes.

#### Une image

