**Documentation**

**Internet et le protocole « http » :**

L'Hypertext Transfer Protocol (HTTP, littéralement « protocole de transfert [hypertexte](https://fr.wikipedia.org/wiki/Hypertexte) ») est un [protocole de communication](https://fr.wikipedia.org/wiki/Protocole_de_communication) [client-serveur](https://fr.wikipedia.org/wiki/Client-serveur) développé pour le [World Wide Web](https://fr.wikipedia.org/wiki/World_Wide_Web). [HTTPS](https://fr.wikipedia.org/wiki/HyperText_Transfer_Protocol_Secure) (avec S pour secured, soit « sécurisé ») est la variante du HTTP sécurisée par l'usage des [protocoles](https://fr.wikipedia.org/wiki/Protocole_de_communication) [SSL](https://fr.wikipedia.org/wiki/Transport_Layer_Security) ou [TLS](https://fr.wikipedia.org/wiki/Transport_Layer_Security).

**Troubleshooting :**

Le Tshoot consiste à éliminer un ou plusieurs problèmes.

Le processus de recherche se doit d’être réfléchie, intelligent et logique.

Il convient tout d’abord de bien identifier le problème, avant de rechercher sa cause, pour ensuite le résoudre.

Enfin, il est préférable de garder une trace de tout cela.

Le Tshoot peut être nécessaire à toutes les étapes.

A la conception d’un réseau, il est possible de rencontrer un comportement inattendu.

Dans ce cas-là, l’urgence est rarement forte.

En revanche, sur un réseau en production, un incident va nécessiter un Troubleshooting rapide et efficace.

<https://www.networklab.fr/introduction-au-troubleshooting/>

**debugging :**

Le débogage est le processus de recherche et de résolution de défauts ou de problèmes dans un programme informatique qui empêche le bon fonctionnement d'un logiciel informatique ou d'un système.

Les tactiques de débogage peuvent impliquer le débogage interactif, l'analyse du flux de contrôle, les tests unitaires, les tests d'intégration, l'analyse des fichiers journaux, la surveillance au niveau de l'application ou du système, les vidages de mémoire et le profilage.

**Comprendre HTML « boilerplate » :**

[HTML5 Boilerplate](http://html5boilerplate.com/) est un template (HTML+CSS+JS+...) lancé par Paul Irish (Modernizr) et Divya Manian. Il contient toutes les bonnes pratiques du moment en terme de performances, de compatibilité inter-navigateurs, de JavaScript, avec un soupçon d'accessibilité (si utilisé à bon escient) [HTML5 Boilerplate](http://html5boilerplate.com/) est un template (HTML+CSS+JS+...) lancé par Paul Irish (Modernizr) et Divya Manian. Il contient toutes les bonnes pratiques du moment en terme de performances, de compatibilité inter-navigateurs, de JavaScript, avec un soupçon d'accessibilité (si utilisé à bon escient)

**Le doctype :**

La première ligne d'en-tête a été considérablement simplifiée. On peut maintenant retenir un *doctype* par coeur, sans avoir à le copier-coller.

<!doctype html>

Ce doctype allégé a été mûrement testé et réfléchi : il permet toujours - pour les anciens navigateurs - de rester dans le mode d'interprétation conforme aux standards et d'éviter le mode *quirks*. On peut remarquer qu'il n'y a pas d'allusion à un quelconque numéro de version. Son rôle est toujours de préciser quel est le type de document qui va suivre, afin de permettre au navigateur de savoir quel langage de la vaste famille SGML il devra interpréter.

**Le HTML5 sémantique :**

Le HTML5 sémantique comble cette lacune en définissant des balises spécifiques pour indiquer clairement le rôle du contenu qu'elles englobent. Cette information explicite aide les machines comme Google et Bing à comprendre les rôles et l'importance relative de différentes parties de votre page.

**Exemples**

Balises non sémantiques / génériques :



Ce sont des exemples d'éléments HTML non sémantiques. Ils servent seulement de conteneurs communiquant au navigateur comment le contenu qu'ils englobent doit être affiché. Ils ne donnent aucune information sur le rôle de ce contenu dans la page.

Voici des balises sémantiques :



Ces éléments définissent clairement le rôle du contenu qu’ils englobent.

**Pourquoi dois-je utiliser le HTML5 sémantique ?**

Pour les utilisateurs, il est généralement facile d'identifier les différentes zones d'une page web en un coup d'œil. Les en-têtes, les menus et (espérons-le) le contenu principal sont tous immédiatement apparents et évidents. Mais les bots des moteurs de recherche (les “crawlers”) sont gravement malvoyants. Pour eux, les indices visuels sont quasiment impossibles à comprendre. Ils ont besoin de votre aide.

Si vous pouvez communiquer à Google et Bing quelle partie de la page est l'en-tête, laquelle représente le footer et laquelle contient les éléments de navigation, ils vous seront reconnaissants. Il est encore plus important de pouvoir indiquer quelle partie de la page présente le message principal de votre contenu.

<https://fr.semrush.com/blog/balises-structurelles-html-semantique/>

**Inline, internal and External CSS ?**

Une feuille de style externe est un fichier .css autonome lié à partir d'une page Web. L'avantage des feuilles de style externes est qu'elles peuvent être créées une seule fois et les règles appliquées à plusieurs pages Web. Si vous devez apporter des modifications étendues à la conception de votre site, vous pouvez effectuer une seule modification dans la feuille de style et elle sera appliquée à toutes les pages liées, ce qui vous fera gagner du temps et des efforts.

Une feuille de style interne contient des règles CSS pour la page dans la section d'en-tête du fichier HTML. Les règles s'appliquent uniquement à cette page, mais vous pouvez configurer des classes CSS et des ID qui peuvent être utilisés pour styliser plusieurs éléments dans le code de la page. Encore une fois, une seule modification de la règle CSS s'appliquera à tous les éléments balisés de la page.

Les styles en ligne se rapportent à une balise HTML spécifique, en utilisant un attribut de style avec une règle CSS pour styliser un élément de page spécifique. Ils sont utiles pour des modifications rapides et permanentes, mais sont moins flexibles que les feuilles de style externes et internes, car chaque style en ligne que vous créez doit être modifié séparément si vous décidez de modifier la conception.

**box-sizing :**

Le CSS box-sizing propriété nous permet d'inclure le rembourrage et la bordure en largeur et la hauteur totale d'un élément.

**currentColor :**

Les variables CSS se fraient lentement leur chemin vers une mise en oeuvre prochaine dans les navigateurs. Une variable existe pourtant depuis des années : currentColor ! Cette fonctionnalité CSS a une bonne compatibilité navigateurs et quelques applications pratiques intéressantes. Voyons comment nous pouvons l’utiliser.

**L'affichage: inline-block Valeur**

Par rapport à display: inline, la différence majeure est que display: inline-blockpermet de définir une largeur et la hauteur de l'élément.

En outre, avec display: inline-block, les marges supérieure et / rembourrages bas sont respectés, mais display: inlineils ne sont pas.

**display: flex**

La propriété flex est une propriété raccourcie qui définit la capacité d'un élément flexible à modifier ses dimensions afin de remplir l'espace disponible de son conteneur.

**justify-content: space-between**

La propriété CSS justify-content indique la façon dont l'espace doit être réparti entre et autour des éléments selon [l'axe principal](https://developer.mozilla.org/fr/docs/Web/CSS/CSS_Box_Alignment#Concepts_majeurs_et_terminologie) d'un conteneur flexible ou selon l'axe en ligne lorsque le conteneur est une grille

**grid-template-columns**

La propriété grid-template-columns définit les noms des lignes et les fonctions de taille (track sizing) pour les colonnes de la grille.

**Media Queries**

La [spécification CSS3 **Media Queries**](http://www.w3.org/TR/css3-mediaqueries/) définit les techniques pour l'application de feuilles de styles en fonction des périphériques de consultation utilisés pour du HTML. On nomme également cette pratique **Responsive Web Design**, pour dénoter qu'il s'agit d'adapter dynamiquement le design à l'aide de CSS.

**svg**

L'élément svg peut être utilisé pour intégrer des fragments de code SVG à l'intérieur d'un document (par exemple, un document HTML). Ce fragment de code SVG dispose de ses propres viewport et système de coordonnée.