

Guide de gestion des documents en 2^o année BTS SNIR

Sommaire

1. Travaux pratiques	2
1.1. Procédure	2
1.2. Recommandations	3
1.3. Critères	4
1.4. Questions	4
2. Mini-projets et Projets	4
2.1. Gestion des documents	4
2.2. Dossier technique de projet	5
2.3. Dossier de suivi de projet	6
2.4. Organisation	6
2.5. Outils logiciels	7
2.6. Google Docs	8
2.6.1. Modules complémentaires	10
2.7. Google Slides	15
2.7.1. Modules complémentaires	18
2.8. Création de diagrammes	19
2.9. Doxygen	21
2.10. Subversion (svn)	22
2.11. Git	22
2.12. GitHub (git)	23
2.13. Bitbucket (git)	23
2.14. Bouml	23
2.15. Gestion de projet	23
2.15.1. Kanban	23
2.15.2. Trello	24
2.15.3. Trelloquantt	25
2.15.4. Beesbusy	25
2.15.5. Jira	26
3. Bonus : Envoi de courriel	27
3.1. Expéditeur	27
3.2. Objet	27
3.3. Corps	27
3.4. Pièces jointes	28

Ce document précise l'organisation et les exigences pour la production des documents dans le cadre des Travaux pratiques, Mini-projets et Projets de BTS SNIR à LaSalle Avignon.

1. Travaux pratiques

Les TP de 2^e année BTS SNIR sont gérés via la plate-forme [Google Classroom](#).

[Google Classroom](#) est une plate-forme d'apprentissage gratuite dédiée aux écoles. Son but est de simplifier la création et la diffusion de cours et d'exercices de façon numérique.

Le logo Google Classroom



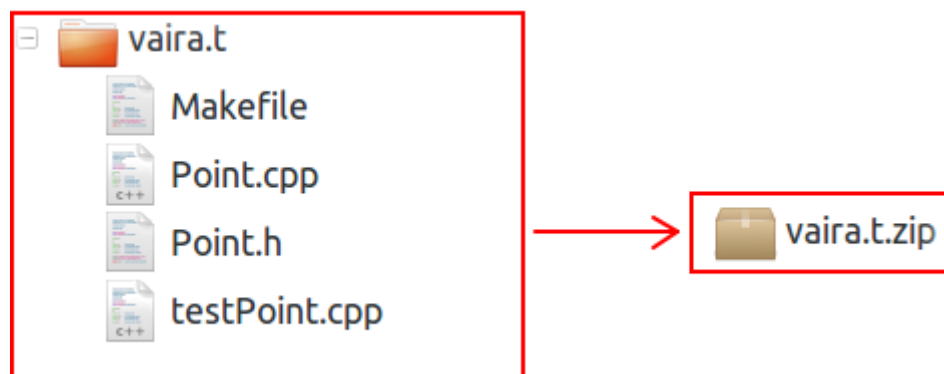
Lien : https://fr.wikipedia.org/wiki/Google_Classroom

1.1. Procédure

Dans [Google Classroom](#) :

- Vous devez récupérer la planche de TP (sujet, codes sources fournis, ressources, ...).
- Vous devez éditer le document [Google Docs](#) fourni qui représente le compte-rendu du TP.
- Vous prendrez soin de placer vos fichiers sources de votre TP dans un répertoire portant en minuscules votre nom suivi d'un point . et de l'initiale de votre prénom, et de me fournir une archive compressée **nom.p.zip** (autre format accepté **.tar.gz** ou **.tgz**) de ce répertoire.

*Exemple de contenu pour une archive **vaira.t.zip***



Une convention de nommage permet, entre autres, d'identifier, classer naturellement les documents. La nomenclature doit être adaptée au contexte.



Tous documents rendus sans respect de la convention de nommage ne seront pas pris en compte pour la notation et seront considérés comme travail non rendu et donneront la note de 0.

Pour chaque TP, vous devez toujours :

- fournir un compte-rendu
- rendre une archive
- respecter la date limite (généralement le lundi à 23h59)



Tous les travaux pratiques rendus en retard donneront la note de 0.

Rendre un devoir dans Google Classroom

The screenshot shows a Google Classroom assignment page for 'TP C/C++ : Prérequis 1' by Thierry Valra. The page includes a date limit, a description of the task, and a list of attachments. Annotations with red arrows and numbers 1 through 4 point to specific elements:

- ① Télécharger les pièces du TP: Points to the attachments 'tp-prerequis-1.pdf' and 'recherche-dichotomique.cpp'.
- ② Éditer (si besoin) son compte-rendu dans Google Docs: Points to the 'Ajouter ou créer' button in the 'Votre devoir' section.
- ③ Ajouter l'archive nom.p.zip contenant les codes sources du TP à rendre: Points to the 'Ajouter ou créer' button in the 'Votre devoir' section.
- ④ Rendre le devoir: Points to the 'Rendre le devoir' button in the 'Votre devoir' section.

Le TP sera notée et corrigée directement dans Google Classroom.

1.2. Recommandations

Vous prendrez garde à :

- ne pas mettre les fichiers objets (.o ou .obj suivant les systèmes) ni les fichiers exécutables ni les fichiers backup (~ sous Linux par exemple)
- fournir si possible un moyen de fabriquer facilement vos programmes (.pro ou Makefile)
- de respecter l'encodage **UTF-8** pour les fichiers texte (codes sources, scripts, ...)

Le minimum attendu : on doit pouvoir fabriquer un exécutable et le lancer !

1.3. Critères

Les points suivants (entre autres) seront considérés pour la notation :

- respect de la procédure de rendu (nom, retard, encodage, ...)
- respect des consignes indiquées sur les planches de TP (nom des fichiers, des classes, des méthodes, ...)
- respect des règles de codage et bonnes pratiques

1.4. Questions

Si vous avez des questions sur le TP, vous privilégiez l'usage des **commentaires privés** dans [Google Classroom](#).

2. Mini-projets et Projets

Ce document précise l'organisation et les exigences pour la production des documents dans le cadre d'un projet de BTS SN.

2.1. Gestion des documents

Les documents produits pendant la phase d'un projet seront stockés sur [Google Drive](#) (Google One). Un espace par projet sera créé et partagé.

La suite bureautique [Google Drive](#) sera utilisée pour produire :

- le dossier technique ([Google Docs](#)),
- le dossier de suivi de projet ([Google Docs](#)) et les documents annexes comme les notes, organigrammes, synoptiques, notes de calcul, résultats d'essais, ... ([Google Docs](#), [Google Sheets](#), [draw.io](#), fichiers au format Markdown/Asciidoc, ...),
- le diaporama ([Google Slides](#)),

Cette suite permet le travail collaboratif afin de créer et de modifier des documents en ligne et de travailler en équipe, en temps réel.

Le logo Google Drive





Google Drive est un service de stockage et de partage de fichiers dans le *cloud* lancé par la société **Google**.

Google Drive, qui regroupe **Google Docs**, **Sheets** et **Slides**, Drawings, est une suite bureautique permettant de modifier des documents, des feuilles de calcul, des présentations, des dessins, des formulaires, etc.

Aide en ligne : <https://support.google.com/>

Pour créer un document, il suffit de cliquer sur “Nouveau” ou utiliser le bouton droit de la souris.

Lien : https://fr.wikipedia.org/wiki/Google_Drive

2.2. Dossier technique de projet

À l'issue du projet, l'équipe d'étudiants remet au centre d'examen un **dossier technique** unique représentatif de l'ensemble du projet. Ce dossier comprend une partie commune à tous les membres de l'équipe et la partie personnelle traitée par chacun d'entre eux.

Dans les 30 pages au maximum qui sont allouées à chaque étudiant, et dans le cadre de son autonomie de réflexion et d'action au sein du projet, il est souhaitable qu'une partie de ce qu'il rédige puisse montrer sa participation à une réflexion commune. L'autre partie contiendra les éléments qui permettront d'évaluer son action individuelle.

Partie commune : (de 20 à 30 pages)

- introduction, situation du projet dans son contexte industriel ;
- dossier de spécifications ;
- dossier d'étude préliminaire et plan de tests des performances au regard du cahier des charges. Suivant la nature du projet et ses points d'entrée, certains éléments de ce dossier peuvent être présents dans les parties personnelles.
- éléments nécessaires à la recette de la maquette ou du prototype final ;
- résultats des essais de la maquette ou du prototype final ;
- conclusion par rapport au cahier des charges fourni par le donneur d'ordre : test intégration, procédure et résultats de la recette.

Partie personnelle : (de 20 à 30 pages)

- situation de la partie personnelle dans l'ensemble du projet ;
- dossier d'étude et de réalisation détaillée, essais unitaires.

En fonction des spécificités du projet et des contraintes de documentation imposées par le cahier des charges, des **documents annexes** peuvent être joints sous forme électronique (annexes techniques, programmes complets, manuel d'utilisation, notice de maintenance, sources complets, ...).

Une soutenance orale aura lieu en fin de projet. Elle est organisée en plusieurs phases. Dans la

phase de présentation de 20 minutes, le candidat expose oralement à la fois la globalité du projet et son action personnelle dans le projet en utilisant un **diaporama**.

2.3. Dossier de suivi de projet

Au fur et à mesure du déroulement du projet, il est indispensable que les étudiants consignent les éléments des Tâches professionnelles qu'ils réalisent au sein d'un **dossier de suivi de projet** (*roadbook*). Ce dossier personnel a plusieurs utilités :

- formaliser l'avancement du travail de l'étudiant (notes, organigrammes, notes de calcul, résultats d'essais, mesure, simulation, modes opératoires, éléments de procédure) ;
- compiler les ressources utilisées (notices techniques, document constructeur, ...) ;
- préparer les revues de projets qui seront au nombre de trois sur la durée du projet ;
- consigner les éléments qui serviront à préparer le dossier technique de projet.

Chaque membre de l'équipe de projet consigne, dans le dossier de suivi, les Tâches professionnelles qu'il réalise pour une période donnée. Les documents relatifs à la vie du projet (devis, bons de commandes, etc.) sont joints au dossier.

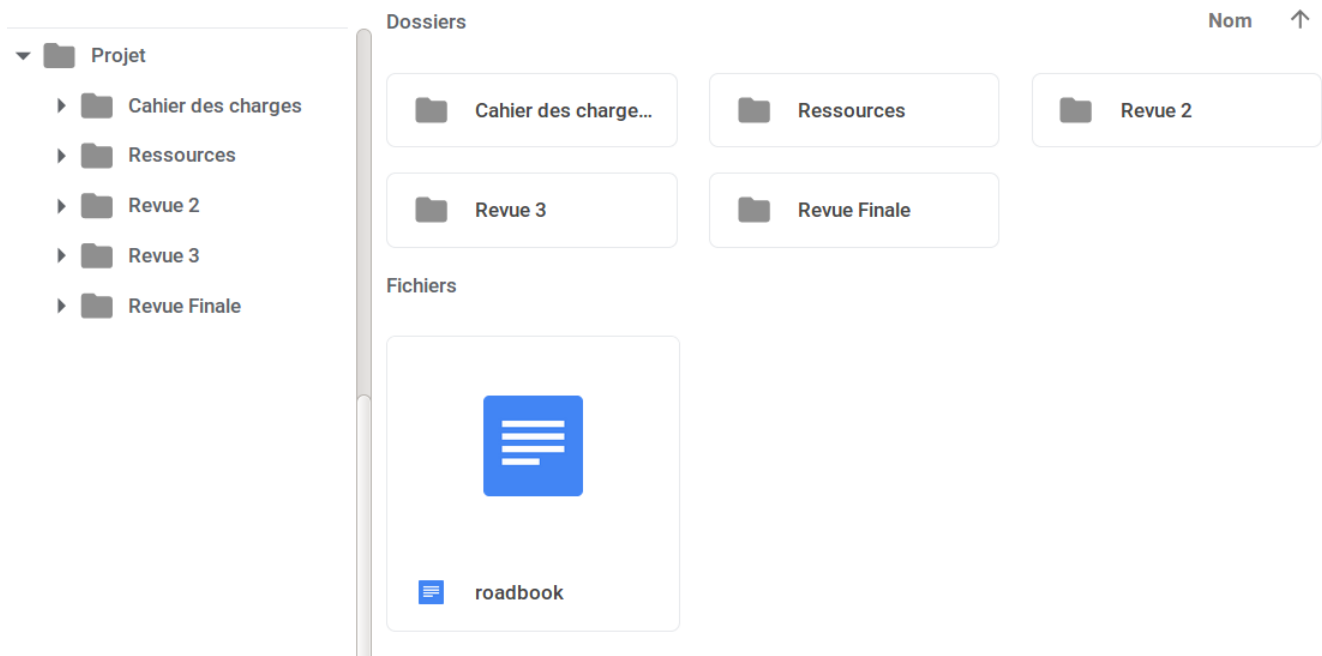
2.4. Organisation

Le projet s'articule autour de 3 + 1 revues :

Jalons	Nom	Version	Livraison
Revue n°1	Revue 1		non
Revue n°2	Revue 2	0.1	oui
Revue n°3	Revue 3	0.2	oui
Rendu dossier	Rendu dossier	1.0	oui
Revue finale	Revue finale	1.1	oui

Un répertoire (dossier) par revues sera créé et partagé dans [Google Drive](#).

Exemple d'organisation



2.5. Outils logiciels

Un certain nombre d'outils logiciels sont nécessaires lors du projet :

Logiciel	Production
Google Docs (Traitement de texte)	Dossier technique, du dossier de suivi de projet, manuel d'utilisateur et d'installation
Google Sheets (Tableur)	Feuille de calcul, résultats d'essais, tableau et courbe de mesures
Google Slides (PAO)	Diaporama
draw.io (Création de diagrammes)	Diagrammes, schémas, ...
Doxygen (Générateur de documentation)	Documentation logicielle
Subversion (svn) (Gestion de versions)	Code source
GitHub (git) (Gestion de versions)	Code source
Bitbucket (git) (Gestion de versions)	Code source
Bouml (Modélisation UML)	Diagrammes UML
Jira (gestion de développement et de projets)	Tableau Kanban, ...
Beesbusy (Gestion de planification)	Tableau Kanban et Diagramme de Gantt

PAO

Présentation Assistée par Ordinateur

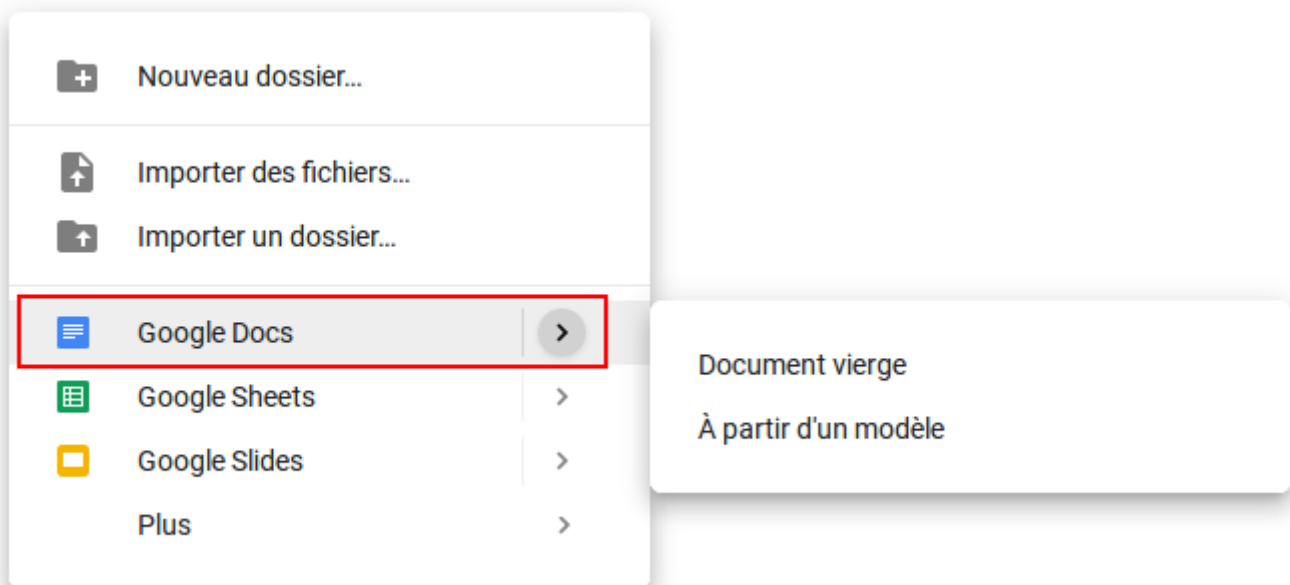
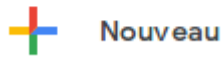


Trello et TrelloGantt pourront être remplacés par [Beesbusy](#), ainsi que Subversion (svn) par Git.

2.6. Google Docs

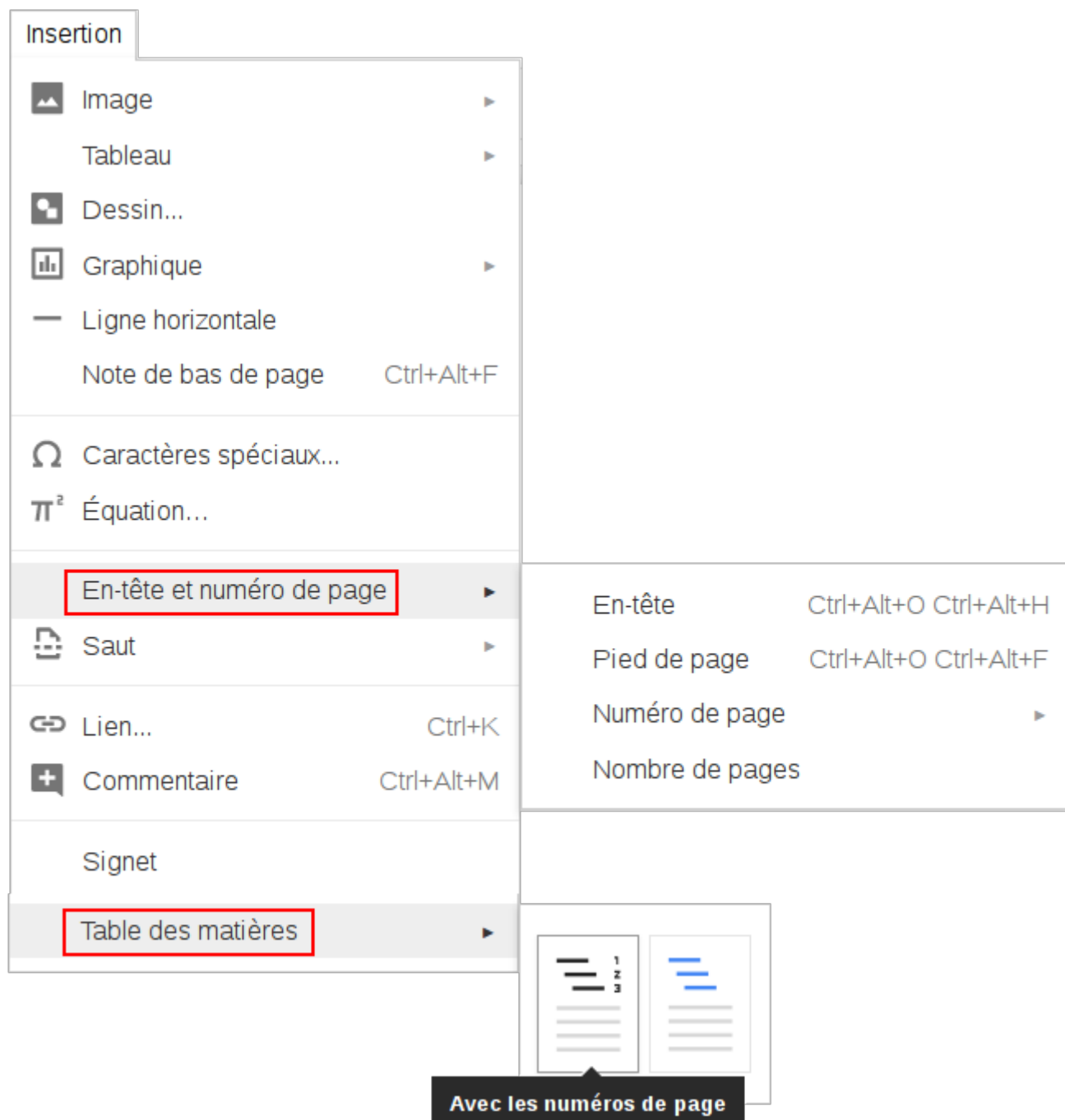
Google Docs est un logiciel de traitement de texte basé sur le Web.

Ce traitement de texte est suffisant pour rédiger les documents du projet. La taille des fichiers est limitée à 50 MO.



Le dossier technique sera structuré et devra comporter :

- une table des matières (sommaire)
- un en-tête
- un pied de page



La table des matières n'est pas mise à jour automatiquement. Il faut penser à la rafraîchir avant d'imprimer ou publier le document.

Chaque page du dossier technique devra être clairement identifiée par :

- un entête comportant : le nom de l'établissement, le nom du projet et l'année de la session du BTS.
- un pied de page comportant : l'identifiant du document (nom, version et date), le numéro de page et le nombre total de pages et le nom du ou des auteurs (les initiales sont acceptées).

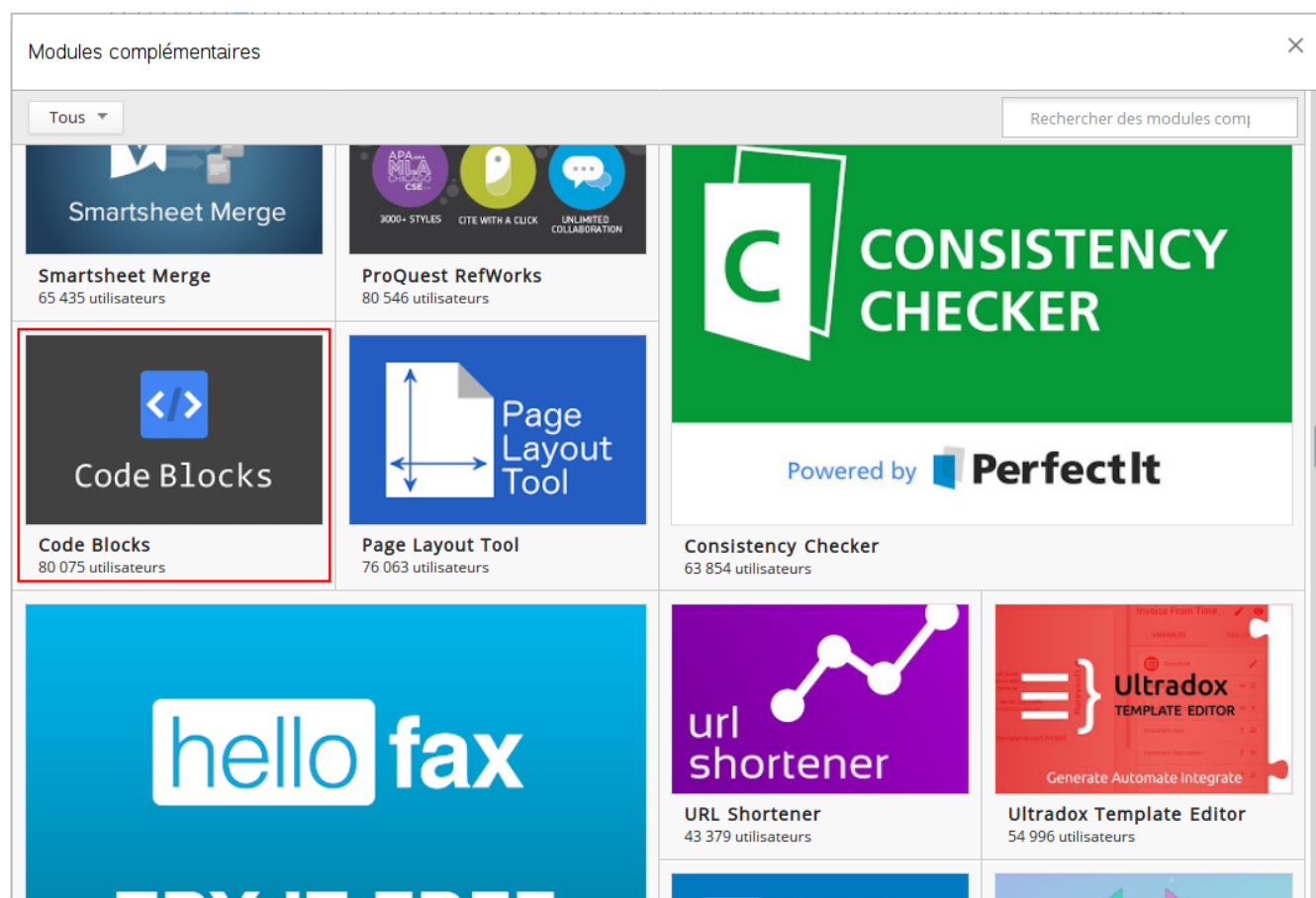
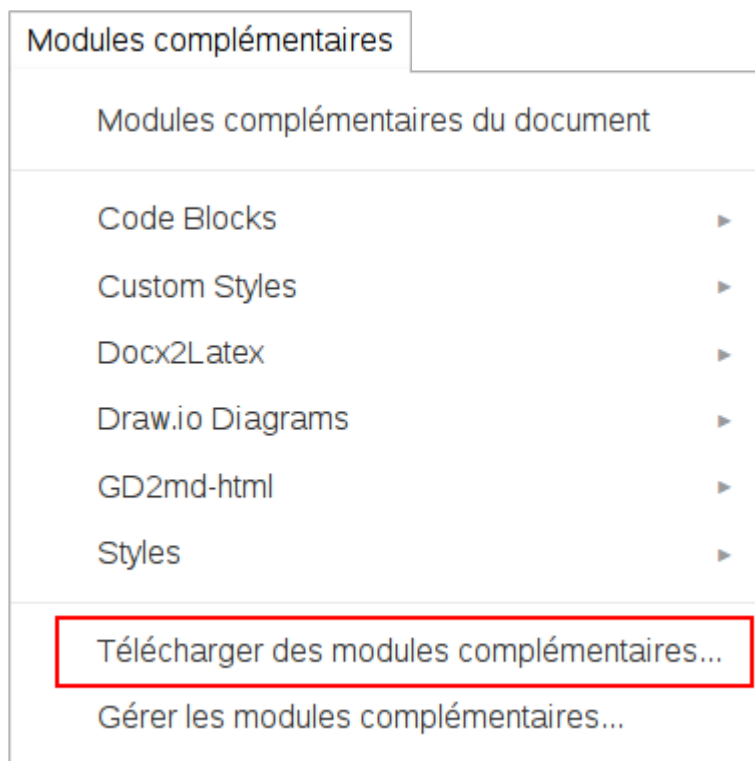
L'en-tête et le pied de page peut être différent pour la première page (page de garde) :

☐ Autre en-tête/pied de page pour la première page

2.6.1. Modules complémentaires

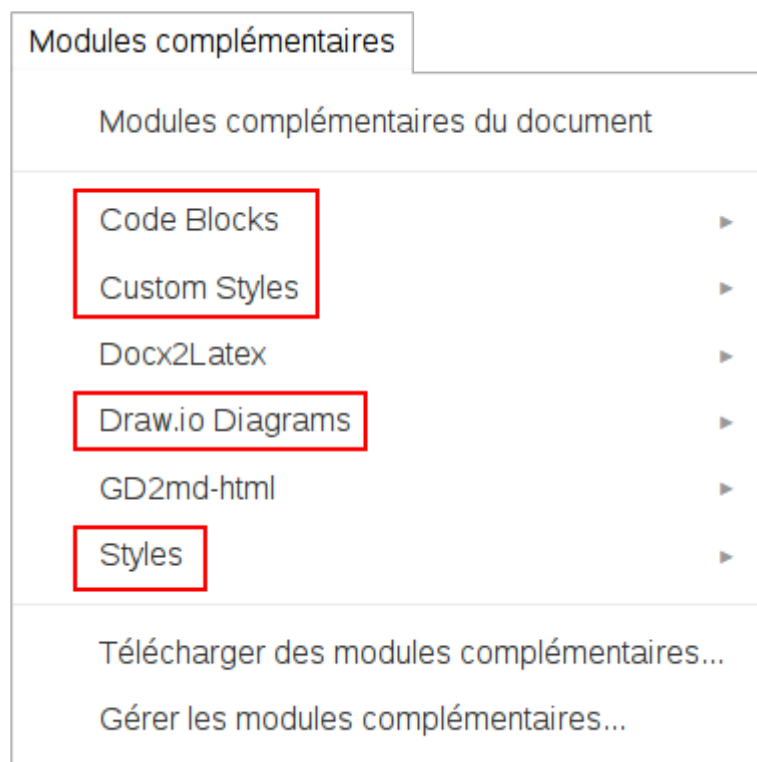
Il est possible d'ajouter des modules complémentaires afin d'enrichir les fonctionnalités de [Google Docs](#).

Pour ajouter un module, il suffit de le télécharger à partir du menu “Modules complémentaires” :



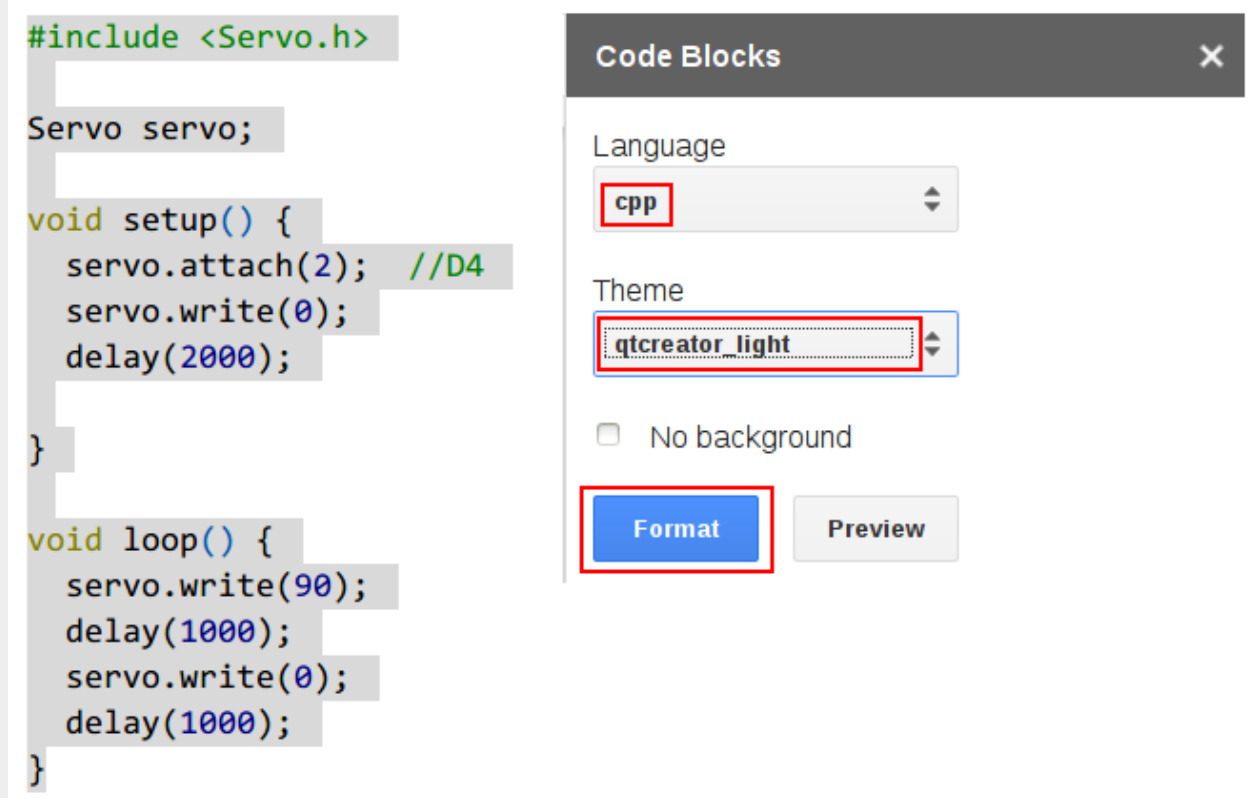
Nous aurons au moins besoin :

Besoin	Module
intégrer et formater du code source et des commandes du <i>shell</i>	Code Blocks
intégrer des diagrammes et des schémas	draw.io
appliquer des styles	Styles
personnaliser des styles	Custom Styles



Intégrer et formater du code source

On utilisera le module **Code Blocks** qui permet de choisir un langage (pour la colorisation syntaxique) et un thème. Il faut sélectionner le texte à formater :



The screenshot shows the Code Blocks application interface. On the left, a code editor displays C++ code for a servo motor. The code is as follows:

```
#include <Servo.h>

Servo servo;

void setup() {
  servo.attach(2); //D4
  servo.write(0);
  delay(2000);
}

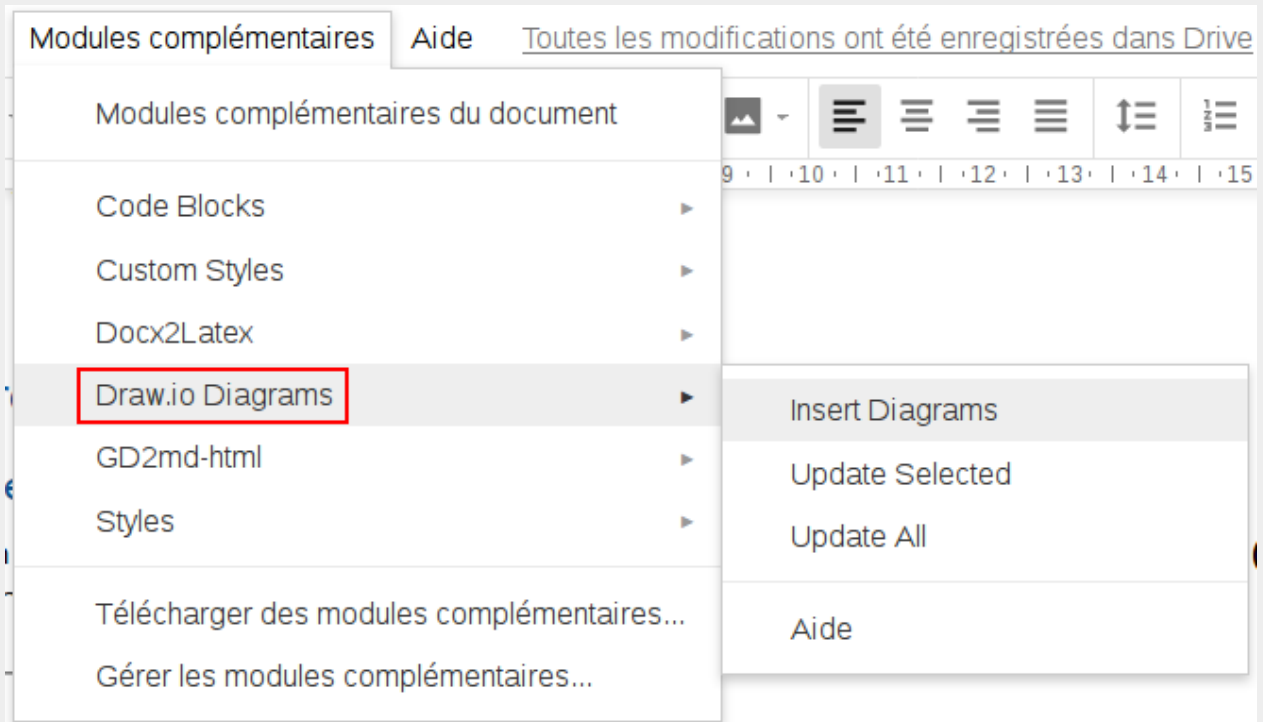
void loop() {
  servo.write(90);
  delay(1000);
  servo.write(0);
  delay(1000);
}
```

On the right, the 'Code Blocks' settings dialog is open. It features a 'Language' dropdown menu set to 'cpp' and a 'Theme' dropdown menu set to 'qtcreator_light'. There is an unchecked checkbox for 'No background'. At the bottom, there are two buttons: 'Format' and 'Preview'. Red rectangular boxes highlight the 'cpp' selection, the 'qtcreator_light' selection, and the 'Format' button.

On l'utilisera aussi pour les lignes de commande du *shell*.

Intégrer des diagrammes et des schémas

On utilisera le module draw.io :



draw.io permet de créer des diagrammes **SysML**. Il permet aussi de réaliser des diagrammes UML mais on ne l'utilisera pas pour cela !

Appliquer des styles

Pour la mise en forme des titres, on pourra utiliser le module **Styles** qui offre une liste de personnalisation “prête à l’emploi”.

The screenshot shows a text editor with a code example for a servo motor. The code is as follows:

```
#include <Servo.h>

Servo servo;

void setup() {
  servo.attach(2); //D4
  servo.write(0);
  delay(2000);
}

void loop() {
  servo.write(90);
  delay(1000);
  servo.write(0);
  delay(1000);
}
```

The 'Styles' panel on the right shows a list of styles: Normal text, Title, Subtitle, Heading 1, Heading 2, Heading 3, Heading 4, Heading 5, and Heading 6. The 'Modern' style is selected. The 'Apply style' button is visible at the bottom of the panel.

Personnaliser des styles

Il est souvent nécessaire de personnaliser la forme du texte.

Dans un dossier technique, c’est indispensable :

- pour les références à des éléments du code source (classe, attribut, méthode, ...) ou à des noms de commande. On impose d’utiliser une police de type **Courier**, éventuellement en **gras** et de préférence en taille 10.
- pour les abréviations, remarques, termes anglo-saxons. On préconise d’utiliser *l’italique*.

Le module **Custom Styles** facilite la gestion de styles personnalisés :

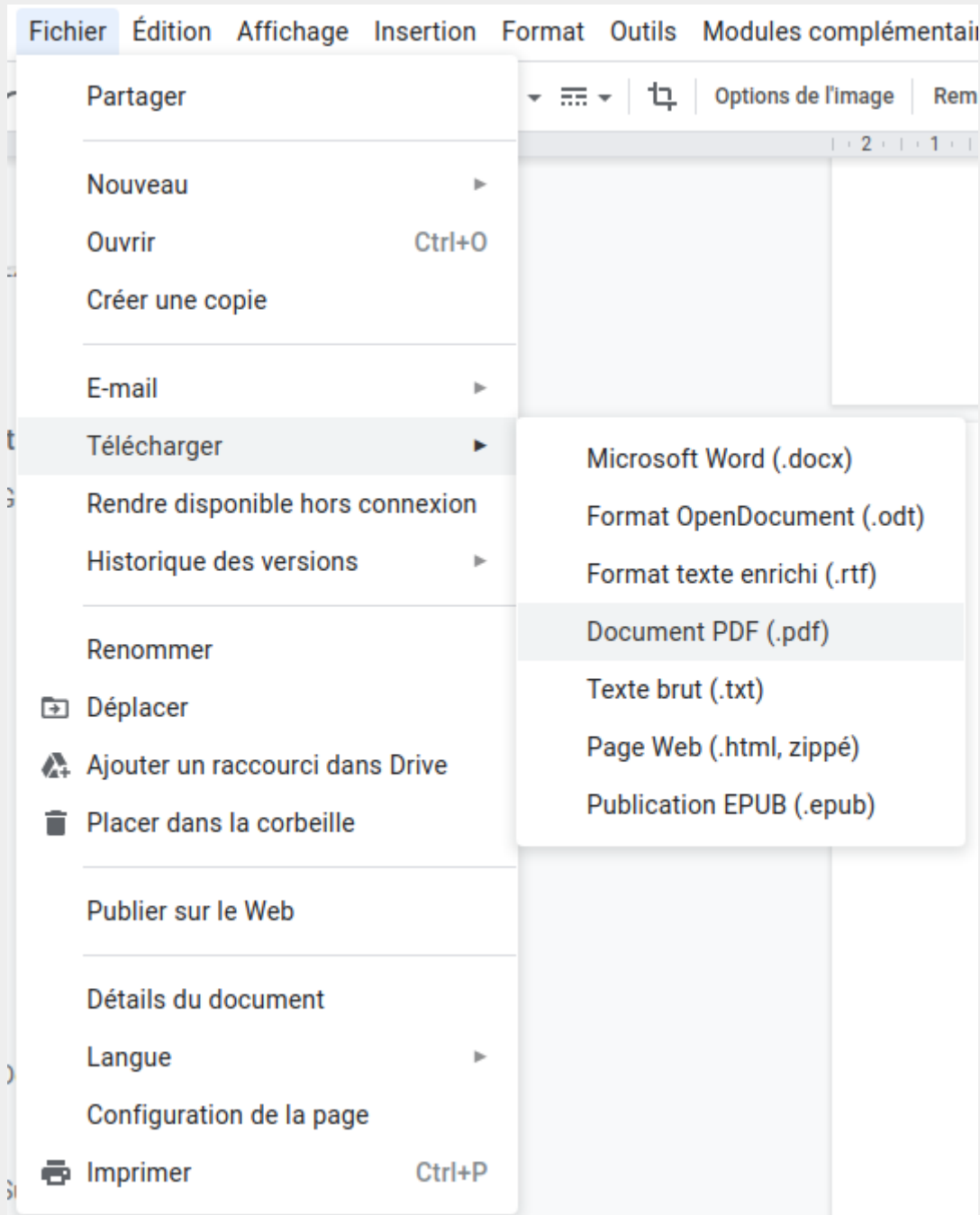
The screenshot shows a text editor with a code example for a servo motor. The code is as follows:

```
On instancie un objet de la classe Servo. On le relie avec la méthode attach() à l'entrée 4. Pour commander le servomoteur, on utilise la méthode write() :
```

The 'Custom styles' panel on the right shows a list of styles: Like this addon? Buy me a cup of coffee, a beer or one stone for my home :)! Or, well, not now. The 'code' style is selected. The 'Delete' button is visible at the bottom of the panel.

Exporter

Google Docs propose de télécharger le fichier dans plusieurs formats :



2.7. Google Slides

[Google Slides](#) est un logiciel de PAO (Présentation Assistée par Ordinateur) qui permet de réaliser des présentations (diaporamas).

Les présentations créées dans [Google Slides](#) sont limitées à 50 Mo, soit environ 200 diapositives.

Le diaporama sera structuré et devra comporter :

- une diapositive de titre

- un sommaire
- un pied de page

Chaque diapositive devra être clairement identifiée par :

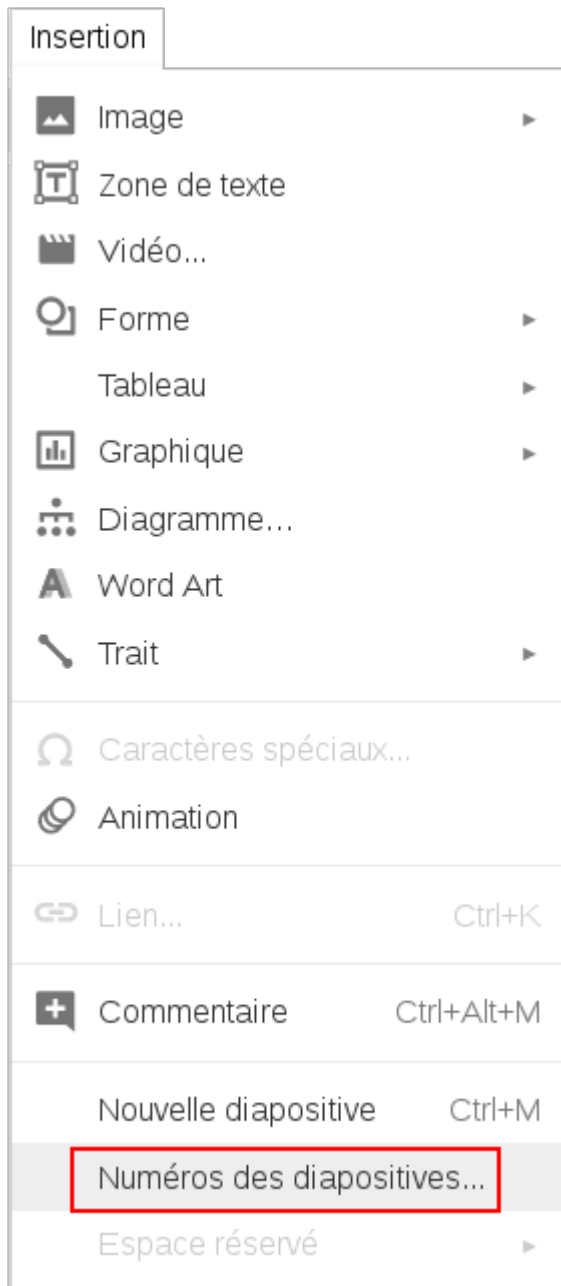
- un titre
- un pied de page comportant : le nom de la section et l'année de la session du BTS, le nom du projet, le nom et prénom de l'étudiant et le numéro de page de diapositive.

Pour cela, il faudra définir un **masque de diapositive** :

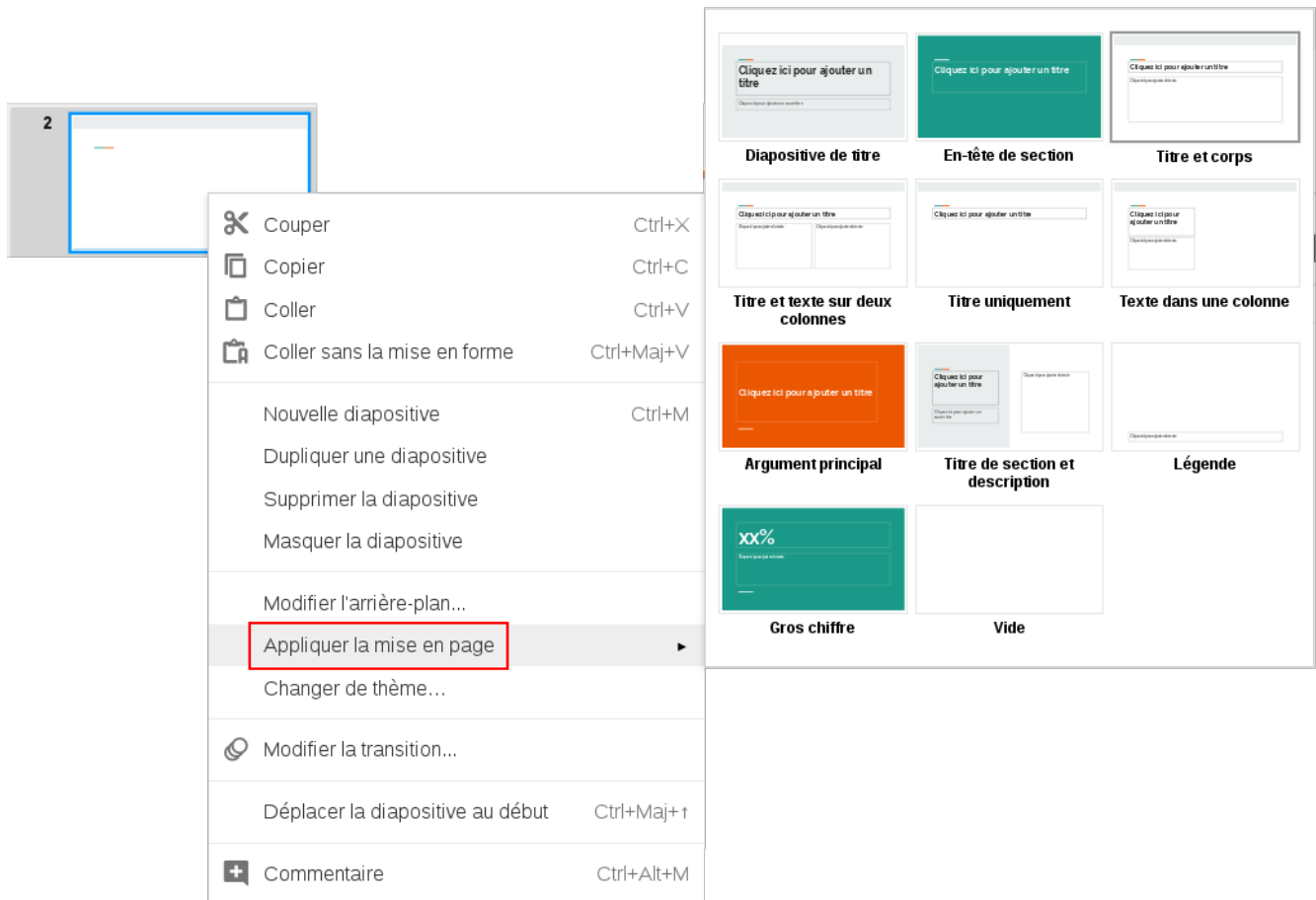


Le # dans le pied de page représente le numéro de diapositive.

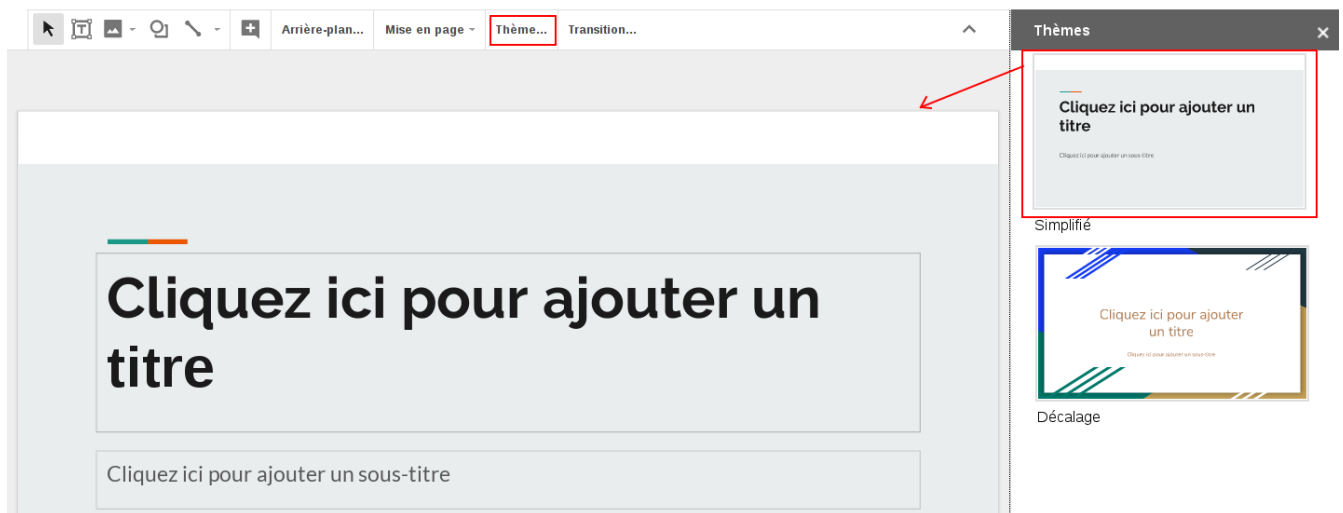
Pour numéroté les diapositives, il faudra faire :



Il est aussi possible de personnaliser la mise en page des diapositives. Ensuite, il suffit de l'appliquer en cliquant avec le bouton droit sur la diapositive concernée :



Des thèmes pré-existants sont aussi disponibles :

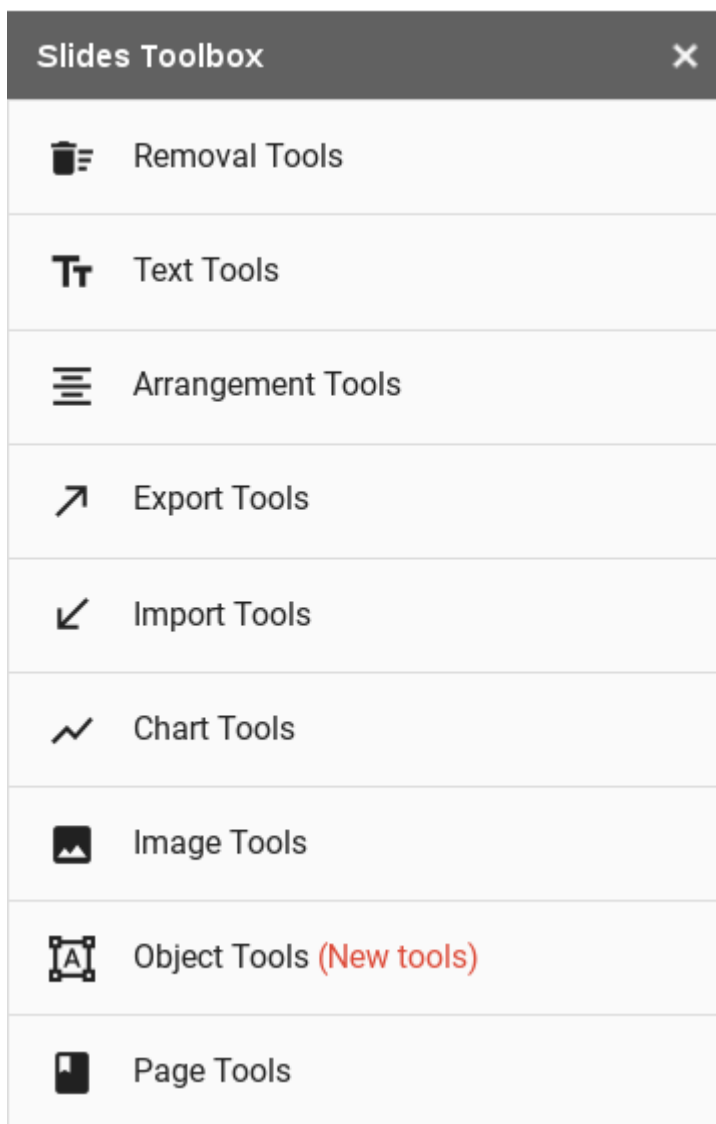
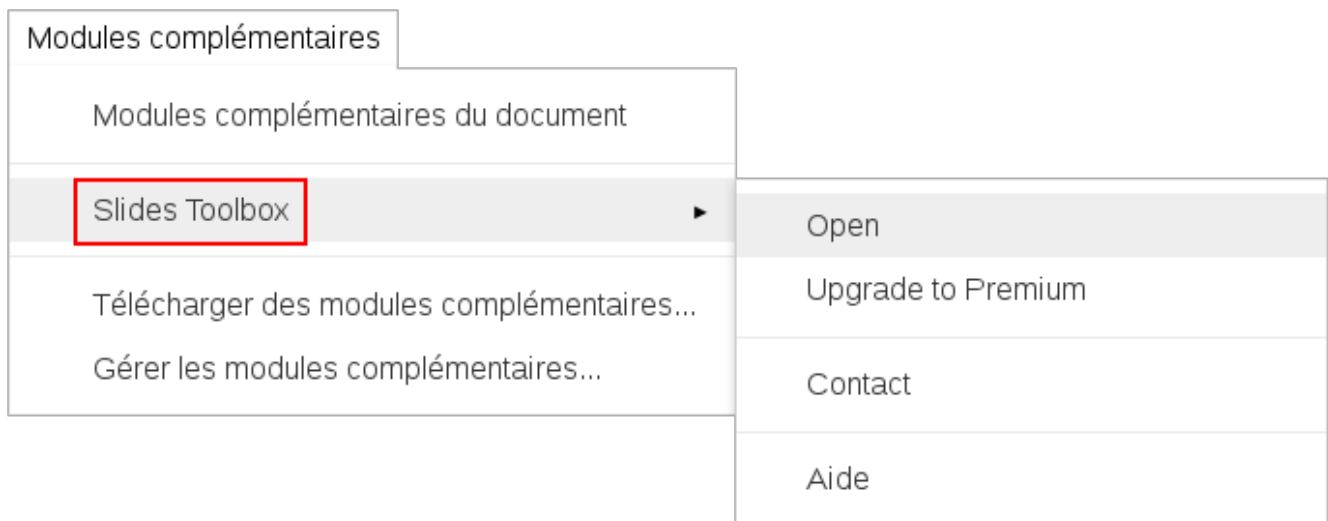


Il sera judicieux d'utiliser des effets de transition (notamment le zoom) pour améliorer la lisibilité des diagrammes.

2.7.1. Modules complémentaires

Il est possible d'ajouter des modules complémentaires afin d'enrichir les fonctionnalités de [Google Slides](#).

Il sera utile d'installer le module **Slides Toolbox**. Ce module offre des fonctionnalités qui facilitent la mise en forme du diaporama :



2.8. Création de diagrammes

diagrams.net (draw.io) est un logiciel de diagramme en ligne gratuit pour créer des organigrammes, des diagrammes de processus, des organigrammes, UML, ER et des diagrammes de réseau.



On n'utilisera pas draw.io pour les diagrammes UML mais Bouml.













Applications supplémentaires : Google Drive permet d'associer des applications externes à votre compte. Vous pouvez ajouter l'application [draw.io](#) qui vous permettra de réaliser des diagrammes et schémas directement dans Google Drive.

Connecter des applications à Drive

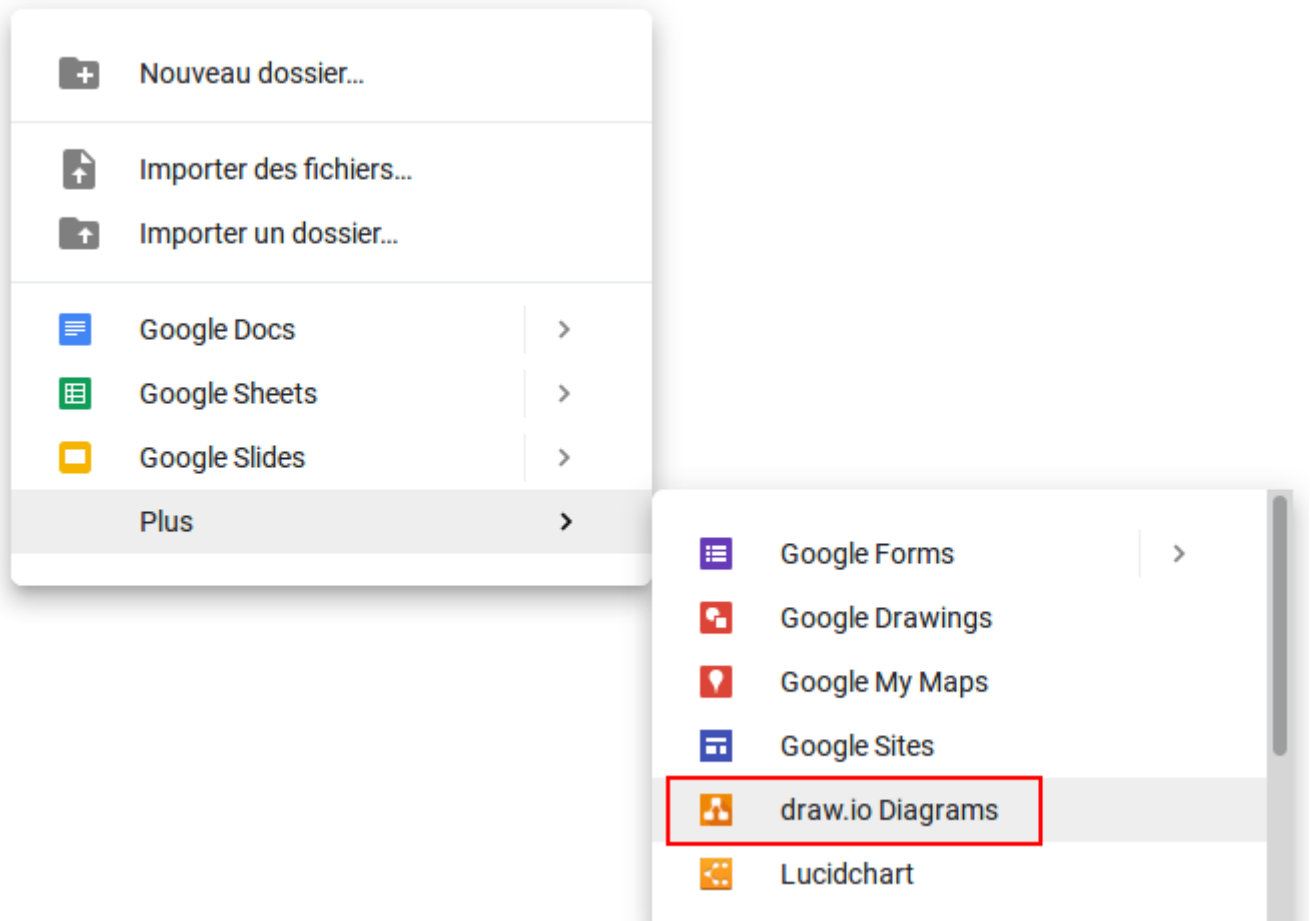
×

Tous

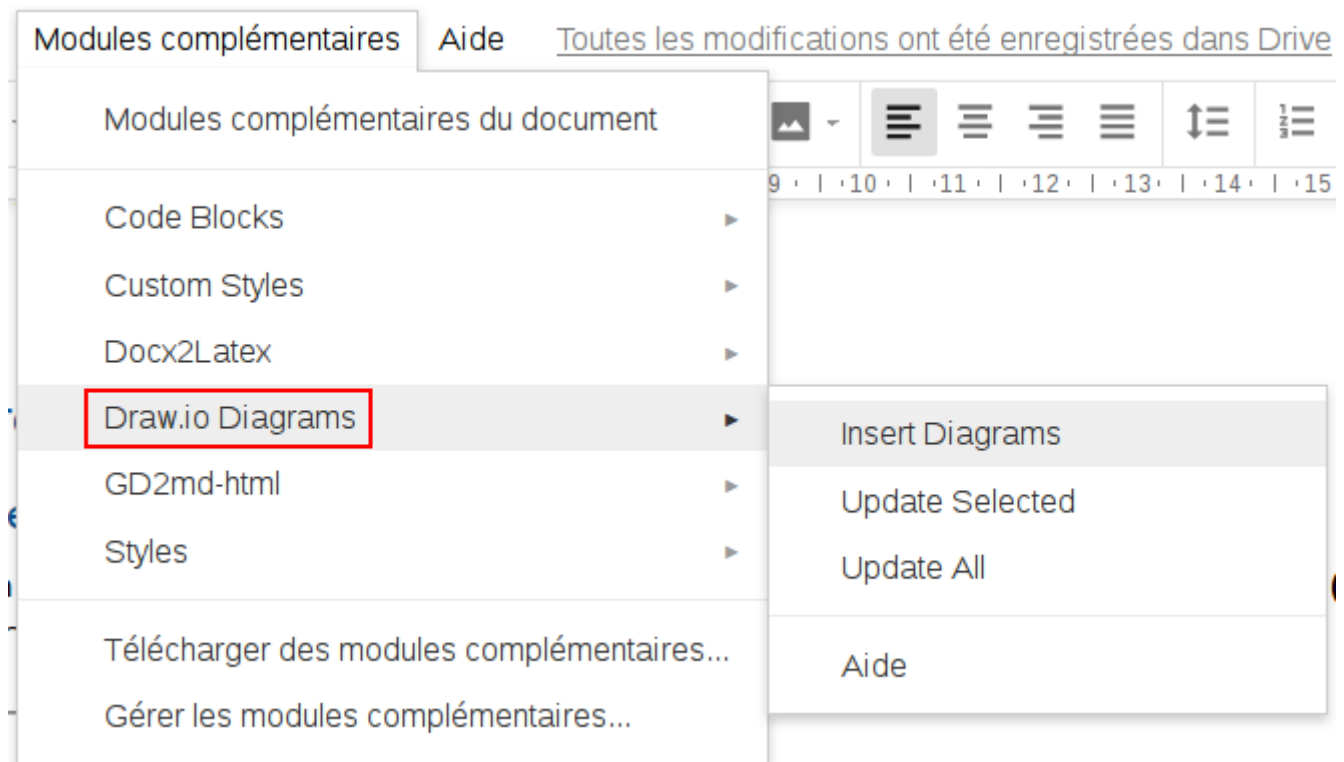
Rechercher dans les applicati

 <p>GeoGebra Classique 5 636 582 utilisateurs</p>	 <p>WeVideo - Éditeur et créateu... 7 600 162 utilisateurs</p>	 <p>Convertisseur Audio 681 600 utilisateurs</p>
 <p>Zoho Sheet 56 178 utilisateurs</p>	 <p>Google Drawings 6 200 169 utilisateurs</p>	 <p>Zoho Writer 80 041 utilisateurs</p>
 <p>vGet Cast (DLNA Controller) 193 846 utilisateurs</p>	 <p>Formulaires Google 4 759 425 utilisateurs</p>	 <p>PDF Mergy - Merge PDF files 742 318 utilisateurs</p>
 <p>Lucidchart Diagrams done right</p>		

L'application [draw.io](#) permet de créer directement des diagrammes et schémas dans Google Drive :



Et de les insérer dans un document :



2.9. Doxygen

[Doxygen](#) est un système de documentation pour C, C++, Java, Python, Php et autres langages.

Doxygen est un logiciel libre, publié sous licence GPL V2.0.

Il permet de générer la documentation de développement d'un projet :

- à partir des commentaires insérés dans le code source ;
- à partir de la structure du code lui même.

Les options de génération de **Doxygen** seront enregistrées dans un fichier **Doxyfile** faisant partie du dépôt Subversion (**svn**).

La documentation sera produite avec l'outil **doxywizard** :

- au format HTML : elle sera intégrée à l'archive ZIP avec le code source
- au format PDF : elle sera imprimée et mise à disposition du jury



Il existe un guide dédié à l'utilisation de Doxygen en projet.

2.10. Subversion (svn)

Subversion est un outil qui permet de maintenir l'ensemble des versions d'un logiciel. Conçu à la base pour faciliter le travail de développement seul ou en équipe, il est surtout utilisé pour le code source, mais peut s'appliquer à tout type de document "texte".

Le dépôt (*repository*) est l'emplacement où sont stockées toutes les informations concernant le projet géré par subversion. Il contient l'historique des versions des fichiers stockés, les journaux de chaque modification, ainsi que toutes les informations comme la date, l'auteur d'une modification etc...

Le dépôt sera hébergé sur Internet par RiouxSVN.

Sous GNU/Linux, on utilisera le client subversion **svn** en ligne de commande. Sous Windows®, on peut utiliser le client **TortoiseSVN**. Les environnements de développement **VS Code**, **Qt Creator** et **Android Studio** intègrent des extensions pour **Subversion**.

2.11. Git

Git est un logiciel de **gestion de versions décentralisé** (DVCS). C'est un logiciel libre créé par **Linus Torvalds** en 2005. Il s'agit maintenant du logiciel de gestion de versions le plus populaire devant **Subversion** (**svn**) qu'il a remplacé avantageusement.

Site officiel : <https://git-scm.com/>

Le dépôt sera hébergé sur Internet par GitHub ou Bitbucket.

Sous GNU/Linux, on utilisera **git** en ligne de commande. Les environnements de développement **VS Code**, **Qt Creator** et **Android Studio** intègrent la prise en charge de Git.

2.12. GitHub (git)

GitHub est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions **Git**.

Site officiel : <https://github.com/>

Ressources GitHub :

- [Git Handbook sur Github](#)
- [Collaborating with pull requests](#)

2.13. Bitbucket (git)

Bitbucket est un service web d'hébergement et de gestion de développement logiciel utilisant le logiciel de gestion de versions **Git**. Bitbucket est gratuit pour les particuliers et les petites équipes comptant jusqu'à 5 utilisateurs, avec des référentiels publics et privés illimités.

Lien : [Tutoriels](#)

2.14. Bouml

BOUML est un logiciel, programmé par Bruno Pagès en Qt, permettant la création de diagrammes UML.

Il est multiplateforme, supporte la génération de code et la rétro-ingénierie pour les langages C++, Java, PHP et MYSQL. Parmi les gratuits UML, il est jugé extrêmement efficace pour la « rétro-modélisation » (créer un modèle UML à partir de codes sources) et pour le développement *roundtrip* (faire des aller-retour entre modèle UML et code source).

Auparavant distribué gratuitement sous licence GPL (version 4.23), puis commercialisé pour les versions 5 et 6, il est de nouveau distribué gratuitement depuis la version 7 (mais pas intégralement sous licence GPL).

On utilisera une [version 7.x en français](#).

2.15. Gestion de projet

2.15.1. Kanban

Issue d'un mot japonais signifiant tableau, la méthode **Kanban** a vu le jour dans les usines Toyota au milieu du XXème siècle. Cette image de tableau vient de l'utilisation des porte-étiquettes permettant d'organiser le travail dans les usines.

Sa reprise pour la gestion de projet date de l'apparition des [méthodes dites agile](#).

La méthode **Kanban** tire sa force de sa simplicité. En effet, toujours dans un esprit agile, cette méthode simplifie au maximum le concept d'organisation des tâches.

Elle repose sur quatre étapes principales aboutissant à un tableau de quatre colonnes :

- *backlog (todo)* : la liste des tâches à effectuer pour l'itération ;
- *ready* (prêt) : la liste des tâches que vous souhaitez effectuer dans l'itération ;
- en cours : les tâches en cours de réalisation (en général une à la fois, voire deux) ;
- terminé : Les tâches terminées.

On peut encore simplifier la méthode en ne retenant que 3 colonnes : **à faire, en cours et terminé**.

2.15.2. Trello

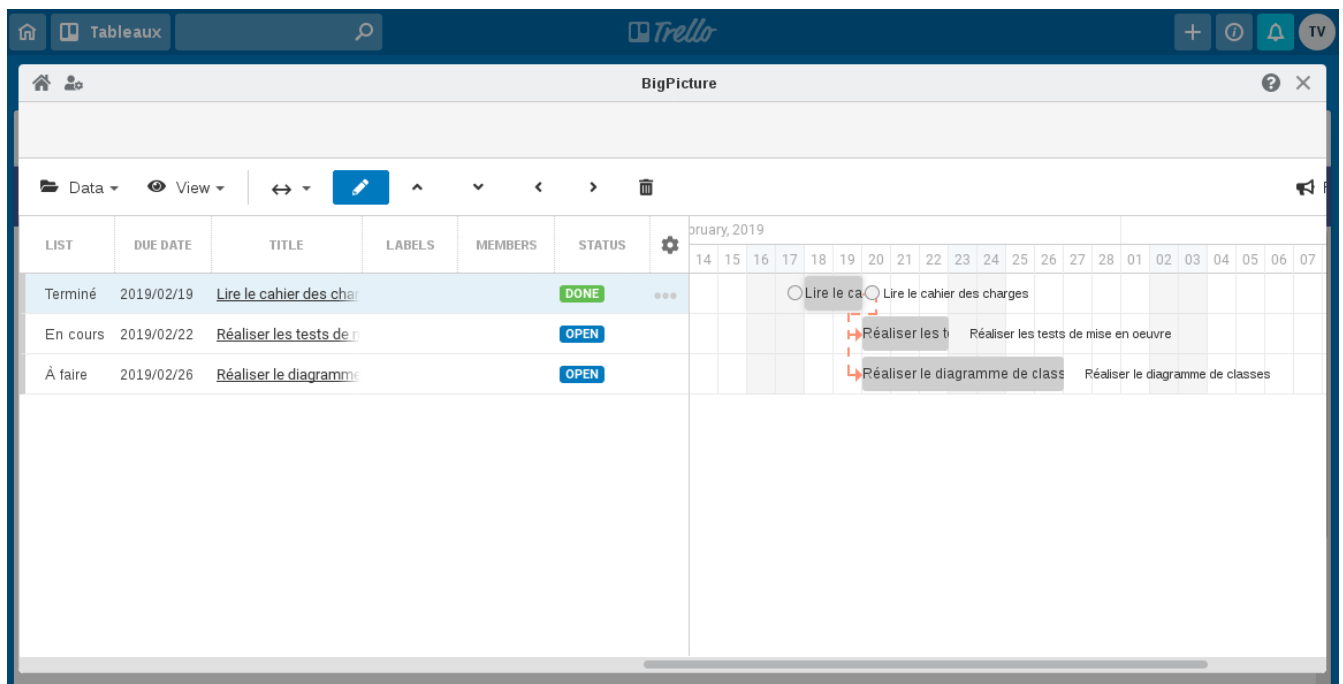
[Trello](#) est un outil de gestion de projet en ligne, lancé en septembre 2011, et inspiré par la méthode [Kanban](#) de Toyota. Il est basé sur une organisation des projets en planches listant des cartes, chacune représentant des tâches.



Lien : [Guide d'utilisation](#)

Il existe des **Power-ups**, comme :

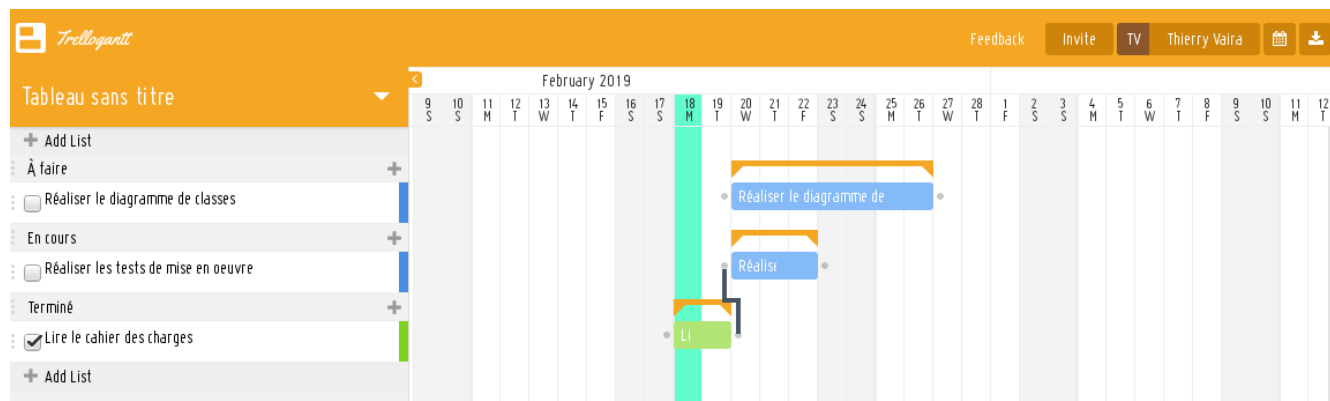
- [BigPicture](#)
- [Planyway Calendar](#)



2.15.3. Trelloantt

Trelloantt est un outil qui permet de réaliser des **diagrammes de Gantt** à partir de Trello.

Lien : [Tutoriels](#)



Voir aussi :

- [Planner](http://live.gnome.org/Planner) est un gestionnaire de planning et de projets (Linux). Lien : <http://live.gnome.org/Planner>
- [GanttProject](https://www.ganttproject.biz/) est un logiciel libre de gestion de projet écrit en Java, ce qui permet de l'utiliser sur divers système d'exploitation (Windows, Linux, MacOS). Lien : <https://www.ganttproject.biz/>
- [ProjectLibre](https://sourceforge.net/projects/projectlibre/) est un outil de planification de projet (Windows, Linux, MacOS). Il est le résultat d'un fork du logiciel OpenProj et s'inspire de Microsoft Project. Lien : <https://sourceforge.net/projects/projectlibre/>

2.15.4. Beesbusy

Beesbusy est un outil de gestion de projet français et complet pour le travail en équipe permettant un suivi de projets et une planification de tâches.

La plateforme propose 3 vues différentes pour permettre de visualiser les projets en cours et d'optimiser la planification :

- une vue « quoi » : vue sur les tâches sous la forme de listes (tableau [Kanban](#))
- une vue « quand » : vue sur les projets avec **diagramme de Gantt**
- une vue « qui » : vue sur les plannings d'équipes et la répartition des tâches

Beesbusy est proposé **gratuitement pour les enseignants et les étudiants**. Excellent !

FREE	BUSINESS	ENTREPRISE
Gratuit	319 € / mois <small>Toutes Taxes Comprises (TTC)</small> 3,19 € / mois / utilisateur	<div>~100%</div> <div>Mon abonnement</div>
<ul style="list-style-type: none"> ✓ Projets illimités ✓ Listes de tâches illimitées ✓ Tâches illimitées ✓ Membres illimités ✓ Vue "Quoi" Organisation en listes de tâches ✓ Fonction de rappel ✓ Classement de projets ✓ Tableau de bord personnel ✓ Envoi illimité de notifications personnalisées (Buzz) 	<ul style="list-style-type: none"> ✓ Toutes les fonctionnalités FREE, avec en plus : ✓ Vue "Quand" Diagramme de Gantt ✓ Vue "Qui" Planning des tâches des membres ✓ Les 3 vues multi-projets Vision d'ensemble de vos projets ✓ Nombre de pièces jointes illimité ✓ Tableau de bord personnalisé et statistiques ✓ Archivage et suppression des projets ✓ Synchronisation Google et Outlook Retrouvez vos projets et vos tâches dans votre calendrier ✓ Import Trello Importez vos tableaux Trello en tant que projets avec leurs listes, cartes et checklists 	<ul style="list-style-type: none"> ✓ Toutes les fonctionnalités BUSINESS, avec en plus : ✓ Gestion des temps Permet aux utilisateurs de saisir leurs temps et donne aux managers une vision globale : <ul style="list-style-type: none"> - Temps initial - Temps planifié - Temps réalisé - Temps restant - Temps total - Pourcentage d'avancement - Export Excel du suivi des temps ✓ Tableau de bord global de suivi des projets Pilotage de tous vos projets avec une vue synthétique ✓ Personnalisation de l'interface Habillez l'application avec votre logo et vos couleurs ! ✓ Création de smart-tags sur les membres Identifiez vos membres avec des smart-tags pour les gérer plus facilement dans le reste de l'application ✓ Champs personnalisés Ajoutez des champs personnalisés sur les membres et les projets

Présentation :

- [Fonctionnalités](#)
- [Démonstration complète de Beesbusy \(25 minutes\)](#)
- [Comment utiliser Beesbusy en 5 minutes](#)

2.15.5. Jira

À l'origine, [Jira](#) est un système de suivi de *bugs* et de gestion des incidents (*tickets*). Il est maintenant un système de gestion de projets développé par **Atlassian**.

[Atlassian](#) est un éditeur de logiciels, basé en Australie, qui développe des produits pour la gestion de développement et de projets. Ses logiciels les plus connus sont Jira, Confluence, Bitbucket et Trello :

- [Jira Software](#) constitue la plate-forme centrale pour les phases de programmation, de collaboration et de livraison. Jira Software est gratuit jusqu'à 10 utilisateurs, avec 2 Go de stockage.
- [Confluence](#) est un logiciel de wiki, utilisé comme logiciel de travail collaboratif.
- [Bitbucket](#) est un service web d'hébergement et de gestion de développement logiciel utilisant le logiciel de gestion de versions [Git](#). Bitbucket est gratuit pour les particuliers et les petites équipes comptant jusqu'à 5 utilisateurs, avec des référentiels publics et privés illimités.
- [Trello](#) est un outil de gestion de projet en ligne, inspiré par la méthode Kanban de Toyota.

Ressources Atlassian :

- [Les guides Jira](#)
- [À quoi sert Jira ?](#)
- [Découvrez Agile grâce à Jira](#)

3. Bonus : Envoi de courriel

3.1. Expéditeur

L'adresse d'expéditeur ne doit pas être folklorique et doit contenir le nom et prénom (en intégralité, sans abréviation).

En cas d'homonyme lors de la création de la boîte email, on peut y ajouter son département, ou la notation "pro".

On vérifie également que le compte de messagerie est correctement renseigné avec son nom et prénom.

On peut classer les adresses de courriel en trois catégories :

- Les adresses folkloriques
- Les adresses correctes mais dont le profil (nom, prénom du compte) est mal renseigné ou incomplet.
- Les adresses professionnelles

3.2. Objet

L'objet d'un courriel est obligatoire.

Il permet au destinataire de savoir immédiatement ce que le courriel va contenir. Il peut servir également au classement automatique pour certains logiciels.

Cet objet peut-être :

- Folklorique, non renseigné.
- Ne pas contenir toutes les informations nécessaires.
- Professionnel et il contient :
 - La classe, le NOM, Prénom de l'expéditeur car ça reste un document à faire parvenir à un professeur.
 - L'objet, c'est ce que je fais parvenir (un rendu de DM, rendu d'exercice, etc.)
 - Le titre de l'objet Rapport de stage, Rendu DM – Nom du DM, etc.

Exemple : Problème erreur de segmentation dans l'exercice 3 du TP Classe PileChar

3.3. Corps

Si l'objet peut être le titre, le corps peut être comparé à un résumé. Il doit également être professionnel.

Exemple :

Bonjour Monsieur,

Vous trouverez ci-joint l'archive contenant mon rattrapage pour le devoir C++ n°1.

Cordialement,

Thierry Vaira
Enseignant Informatique & Réseaux
BTS SNIR - LaSalle Avignon

Pour les questions techniques, n'oubliez pas de préciser :

- le nom et version de votre système d'exploitation
- le nom et version des logiciels concernés
- les commandes et résultats obtenus (format texte conseillé)

3.4. Pièces jointes

La pièce jointe est la partie la plus importante de votre envoi.

Il vous faut cependant :

- Vérifier en premier lieu si on ne vous a pas demandé un fichier particulier (archive, présentation, etc.).
- S'assurer avant de créer l'archive que vous avez bien respecté l'arborescence et la convention de nommage.
- S'assurer de fournir le format qui a été demandé.
- N'envoyer pas votre code source dans le corps du message (sauf si c'est demandé).
- N'envoyer pas de multiple fichier (sauf si c'est demandé).
- En cas d'envoi de documents multiples, il est nécessaire de regrouper les fichiers dans une archive. L'archive doit :
 - Être de type universelle (**zip** pour une archive).
 - Le nom de l'archive doit être éloquent, clair et précis.

Thierry Vaira - <tvaira@free.fr> - version v0.2 - 23/08/2021 - btssn-lasalle84.github.io