**Digistart en ligne**

Manuel à l’usage du professeur

Date : 24/08/2020

Copyright : © 2020 Konexio

|  |
| --- |
| **INTRODUCTION** |

Le cours se déroule à l’aide de 4 outils :

* Google Meet
* VS Code & Live Share
* Navigateur
* Lecteur de slides

On ajoute à ça deux outils qui servent en dehors des cours mais qui peuvent être sollicités pendant les cours, notamment par les apprenants n’arrivant pas à rejoindre le cours :

* WhatsApp
* Slack
* Git

|  |
| --- |
| **DONNER UN COURS** |

**Avant le cours :**

* Aller dans le clone local du repository git du cours, dans la branche correspondant à la promo, puis git pull.
  + https://github.com/konexio
* Aller dans le dossier correspondant à la session du jour (ou rester à la racine si on passe d’une session à l’autre pendant le cours), démarrer VS Code sur ce dossier
* Créer dans ce dossier de session tous les dossiers pour les apprenants et le dossier konexio si ce n’est pas déjà fait, et y copier le contenu du dossier de session.
* Démarrer la session live share
* Démarrer un serveur en local qui sert le dossier du jour sur le port 8000 python -m http.server 8000 ou autre.
* Partager le port 8000 via Live Share
* Rejoindre le meeting Google Meet
* Coller les liens Live Share et <http://localhost:8000>
* Télécharger les PDFs des cours du jour et les ouvrir à la bonne page
* Ouvrir dans Google Drive le suivi pour la présence et compte rendu

**Après le cours :**

* Dans le repository, git add .; git commit -m “[20DSx0x] Cours du jj MMM”; git push;
* Remplir le compte rendu

|  |
| --- |
| **GOOGLE MEET** |

**C’est la solution de visioconférence choisie par Konexio pour donner les cours.**

Quand on crée le cours on invite tous les assistants et apprenants via Google Calendar en y associant un évènement Google Meet. Les participants n’ont qu’à cliquer sur le lien pour rejoindre le cours. Dans Google Meet les rôles sont :

**Pour le professeur :**

* Parler pour donner le cours :).
* Donner les liens vers la session Live Share et localhost (voir la partie VS Code).
* Partager son écran pour montrer les slides.
* Rappeler aux apprenants leurs rôles (ci-dessous).
* Suivre les questions posées par les apprenants par texte s’il n’y a pas d’apprenant pour ça. C’est un peu compliqué avec un seul écran car normalement cet écran ne regarde pas trop Google Meet.
* Demander aux autres participants de renvoyer les liens aux retardataires (voir subtilités).
* Demander régulièrement si les apprenants ont des questions et laisser un peu de temps pour qu’ils puissent prendre la parole.
* Couper son micro pour que les apprenants puissent se concentrer quand ils font les exercices.

**Pour les assistants :**

* Suivre les questions posées par les apprenants par texte ou à l’oral.
* Parler aussi en collaboration avec le professeur
* Redonner les liens aux retardataires (voir subtilités).
* Penser à couper leur micro quand ils ne posent pas de question.

**Pour les apprenants :**

* Poser leur question, par écrit lorsque quelqu’un parle déjà ou à l’oral lorsque c’est possible.
* Penser à couper leur micro quand ils ne posent pas de question.

**Subtilités à connaître**

* Parfois certains participants au meeting se retrouvent à devoir être “admis” dans le meeting par quelqu’un. La logique derrière ce mécanisme est encore floue mais pour diminuer les chances que cela arrive on:
  + Fait créer le meeting dans Google Agenda par le professeur du cours qui devient ainsi propriétaire du Google Meet et voit ainsi la notification pour laisser entrer les gens.
  + Demande aux participants de rejoindre le Google Meet sous Chrome, connectés à leur compte Google qui a reçu l’invitation Google Agenda.
* Un participant arrivant en retard dans le meeting ne vois pas les messages envoyés auparavant, il faut donc que quelqu’un remette un message avec les liens quand un retardataire arrive. Ça peut être fait par des apprenants donc autant leur déléguer (surtout quand il n’y a pas d’assistant).
* Chaque participant peut activer/désactiver son micro, partager l’écran et couper le micro des autres. Le professeur et les assistants doivent couper le micro de ceux qui font du bruit inutilement, reprendre le partage d’écran quand un apprenant le coupe, et aussi réactiver le micro du professeur ou d’un assistant si un quelqu’un d’autre le désactive.

|  |
| --- |
| **VS CODE & LIVE SHARE** |

**VS Code est un IDE qui offre de nombreux avantages pour les formations à distance, dont le principal est Live Share.**

C’est une extension gratuite qui permet à tous les participants d’une session de collaboration via Live Share de lire et écrire dans les fichiers du dossier ouvert actuellement chez celui qui a créé la session.

Tous les participants doivent installer VS Code et Live Share sur leur poste avant les cours. La difficulté principale de Live Share est qu’il demande de se connecter à un compte GitHub ou Microsoft pour rejoindre une session de collaboration. Konexio recommande d’utiliser un compte GitHub car il est nécessaire par ailleurs dans les cours. 90 % des problèmes des apprenants (d’après des stats très solides sur au moins 10 personnes ^^) vont venir de l’impossibilité de se connecter à leur compte GitHub à cause d’un login/mot de passe incorrect.

La grande différence entre un cours en présentiel et un cours avec Live Share est que les fichiers d’exercices dans lesquels les apprenants vont écrire n’est plus sur leur ordinateur, il est sur l’ordinateur de celui qui a créé la session de collaboration : le professeur.

**Sauvegarder le contenu des apprenants**

Le contenu créé par les apprenants n’étant plus sur leur ordinateur, il doit donc être transféré d’une session à l’autre, d’un professeur à l’autre de manière à garder ce qu’ils ont fait. Git est parfait pour ça étant donné que les fichiers sur lesquels les apprenants travaillent sont déjà dans un repository git, soit digistart-html-css, soit digistart-js. Chaque promotion a donc sa branche git et chaque cours est un commit qu’on push sur cette branche.

À la racine de ces repository on trouver les dossiers sessionX qui contiennent, sur la branche master, directement les fichiers de la session. Pour que tous les apprenants puissent écrire en même temps, à l’intérieur de chaque dossier sessionX on crée un dossier par apprenant en utilisant leur prénom et un dossier konexio dans lequel le professeur écrit les solutions des exercices. Le professeur doit donc penser à créer tous ces dossiers et y copier le contenu de la session avant chaque cours.

À la fin de chaque cours le professeur doit donc penser à sauvegarder le contenu écrit par les apprenants et à l’envoyer sur le repository GitHub, cela se fait avec git via :

git add .

git commit -m “[20DSx0x] Cours du jj MMM”

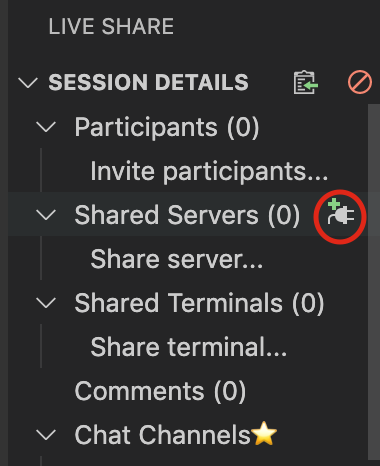
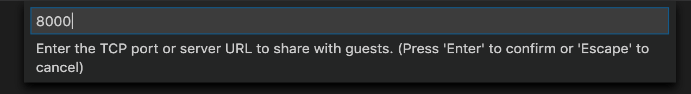
git push

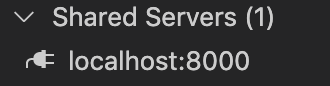
**Regarder le contenu des apprenants**

Le contenu créé par les apprenants n’étant plus sur leur ordinateur, ils ne peuvent plus juste double cliquer sur le fichier index.html pour voir le rendu dans un navigateur.

La solution avec Live Share est de démarrer un serveur sur l’ordinateur du professeur. Cela peut être fait avec python3: python -m http.server 8000

Ce serveur sert juste à servir les fichiers du dossier de la session sur <http://localhost:8000> . Ensuite grâce à Live Share on va partager ce port dans la session Live Share :

1. Ajouter le serveur comme “serveur partagé” dans Live Share
2. Entrer le numéro du port à partager: 8000
3. Live Share indique ensuite que le serveur est partagé



Une fois le partage activé, n’importe quel participant à la session Live Share peut accéder aux fichiers de la session en allant sur <http://localhost:8000> .

**Récupération sur leur ordinateur**

Si les apprenants veulent faire les exercices entre deux cours, ou juste sauvegarder ce qu’ils ont fait ou la correction du professeur sur leur ordinateur, ils peuvent faire clic-droit sur un dossier dans leur VS Code et dans le menu qui s’ouvre choisir “Télécharger”.

Le sens inverse : copier les fichiers de leur ordinateur vers la session live share est beaucoup plus intuitif, il leur suffit d’ouvrir un VS Code sur le dossier où ils ont fait leur modification en local, et un autre VS Code qui rejoint la session Live Share, et de copier coller les fichiers vers Live Share.

**Subtilités à connaître**

* Le copier coller des images depuis l’ordinateur d’un participant vers une session Live Share ne fonctionne pas (l’image à l’arrivée est illisible), c’est un bug connu de Live Share. C’est gênant pour la session du CV où les apprenants doivent mettre leur photo dans le CV. Les solutions sont soit d’envoyer leur image au professeur via WhatsApp et c’est le professeur qui copie-colle l’image au bon endroit sur son ordinateur, soit d’uploader leur image sur un site comme imgur et mettre le lien vers imgur dans le CV.
* Parfois rafraîchir la page dans le navigateur ne va pas montrer la dernière version modifiée dans Live Share (pour des raisons encore non éclaircies sûrement liées au mini-serveur qui devrait être mieux configuré).

|  |
| --- |
| **POUR ALLER PLUS LOIN** |

* Instructions que vous pouvez partager aux apprenants
  + <https://github.com/konexio/digistart-help/blob/222ea63f145648a2893e471dc4cf9d298507ae20/online/help-students.md>
* Instructions pour les bénévoles
  + <https://github.com/konexio/digistart-help/blob/222ea63f145648a2893e471dc4cf9d298507ae20/online/help-teachers.md>
* Modules d’aide (pour les bénévoles) contenant, entre autres, un script pour créer les dossiers des apprenants sur les sessions Live Share.
  + <https://github.com/konexio/digistart-html-css>
  + <https://github.com/konexio/digistart-js>