**AFCI Formation**

**Guide pour la scénarisation pédagogique des modules digitaux**

1. **Mode d’emploi**

**Objet du présent fichier**

Concevoir un module digital, c'est beaucoup écrire ! Les formations sont d'abord écrites sur des documents extérieurs aux outils de digitalisation. Les jeux de relecture sont facilités, vous gardez la maîtrise sur le contenu que vous proposez et la médiatisation peut se faire sur une base sûre.

Cette trame se compose de plusieurs parties (présentation + X sections) formant la base d'un module digitalisé standard.

**Mode d'emploi**

1. Déterminez le nombre de sections dans votre module

2. Si vous avez + de 3 sections, copiez-collez autant de descriptif de section qu’il vous manque.

3. Adaptez la partie 3 de l'onglet présentation de votre formation pour qu'il puisse accueillir le nombre de sections voulues

4. Commencez à remplir l'onglet présentation :

* Auteur (a savoir…vous !)
* Domaine (Par exemple : Sanitaire et social, numérique,...)
* Certification concernées (les diplômes qui peuvent être concernés par ce module)
* Compétences ciblées
* Référence technique : laissez à Thomas ou à l’ambassadeur le soin de le compléter 😉

5.Continuez par l’en-tête :

* Titre (titre du module, quoi d’autre ?)
* Image/GIF/Vidéo (choisissez des images libres de droit, par exemple sur les sites “FREEPIK” ou “PIXABAY”, pour les vidéos youtube, copiez-collez simplement le lien)
* Introduction/résumé : Présentez en quelques lignes le module aux stagiaires, comme vous pourriez le faire en présentiel, une touche d’humour ou de sympathie est fortement appréciée !
* Objectif : ce sont les objectifs pédagogiques, ce que le stagiaire doit apprendre, ils commencent toujours par : “A la fin de ce module de formation, vous serez capable de…” suivit d’un verbe d’action, au besoin, fiez-vous à la taxonomie de Bloom
* Consignes spécifiques : s’il y en a, par exemple : pendant tout le module, ayez avec vous votre carnet XXX…” Sinon laissez cette case vide
* Durée indicative : combien de temps faut-il, à peu près, pour terminer le module

6. Remplissez le tableau “Plan du module”, cela vous permettra de voir plus clair dans les différents chapitres, au besoin, ajoutez des sections

7. Remplissez les sections

Complétez la présentation de la section, si besoin, référez-vous au point 5

8. Par section, déterminez le nombre d'activités.

Attention, une section peut très bien ne contenir qu'une seule activité

9. Commencez à remplir vos activités en rédigeant le contenu.

10. Si vous avez déjà une idée, vous pouvez choisir dès à présent le type d'activité

11. A chaque section son évaluation, elle n’est pas forcément là pour sanctionner le stagiaire, mais plutôt pour lui permettre de s’assurer qu’il à bien compris. Vous devez proposer à minima un exercice par section, choisissez s’il s’agira d’un texte ou d’un devoir, ne remplissez que la partie concernée.

12. Lorsque toutes les sections sont créées, compléter la dernière partie : Section finale

13. Dans la section finale, rédigez le résumé du module, il reprend les axes essentiels à retenir, ajoutez l’image si vous le souhaitez

14. Créer l’évaluation finale du module

15. Une fois l'ensemble des chapitres sections terminés, retournez sur la présentation et vérifiez l'exactitude des informations du plan

**Félicitations ! Votre module de formation est prêt à être digitalisé !**

1. **Lexique**

Activité :

Consigne :

Consigne pédagogique :

Consigne opératoire/opérationnelle :

Devoir :

Evaluation :

Module :

Objectif :

Section :

Test :

1. **Trame d’écriture**
2. **Présentation du module :**

**Auteur : Didier ROLLAND**

**Domaine :** Numérique

**Certifications concernées : TP DWWM**

**Compétences ciblées :**

**Références techniques :**

1. **En-tête**

**Titre : Le versioning**

**Image/ GIF/ Vidéo d’introduction :**

**Introduction/ Résumé du module : Nous allons voir ce qu’est le versioning**

**Objectif :** A l’issue de ce module de formation, vous serez capable de

**Consignes spécifiques :**

**Durée indicative :** 1 h

1. **Plan du module**

|  |  |
| --- | --- |
|  | **Titre** |
| **Section 1** | Qu’est que le versioning, son utilité, pourquoi cela existe et sa place en entreprise |
| **Section 2** | Installer l’environnement GIT |
| **Section 3** | Création d’un repository et envoyer un fichier dessus |
| **Evaluation finale** | Test |
| **Section 5** |  |

1. **Section 1**
   1. **Présentation de la section**

**Titre :** Qu’est que le versioning, son utilité, pourquoi cela existe et sa place en entreprise

**Image/Gif/Vidéo d’introduction :**

**Introduction/Résumé de la section :** Nous allons voir ce qu’est le versioning, son utilité, son existence et sa place en entreprise

**Objectifs :** Comprendre le versioning

**Consignes spécifiques :**

**Durée indicative :** 30 mn

* 1. **Contenu**

|  |  |  |
| --- | --- | --- |
|  | **Contenu** | **Type d’activité** |
| **Activité 1** | Bien connu des développeurs et ingénieurs réseaux, le **versioning** permet de prendre une photo à un moment T de son serveur ou de son fichier pour y revenir facilement si besoin est. Utilisé pour corriger des erreurs ou pour travailler simultanément sur différentes tâches.  Le **versioning** correspond au fait de sauvegarder des versions de son serveur ou de son fichier à certains moments-clés. Ce faisant, on crée des points de contrôle qui vous permettent, si besoin est, d’y revenir facilement et rapidement. On parle alors de restaurer les versions précédentes de votre fichier ou de votre serveur.  Son usage est multiple :   * On peut se servir du versioning pour travailler conjointement sur différents projets * On peut revenir à la version désirée si la situation l’exige * Mais on s’en sert surtout pour contrer les cryptovirus tel que le cryptolocker, ou ransomware, qui bloque les fichiers en demandant de payer une rançon pour les récupérer. Dans ce cas, il est possible de revenir à une version antérieure pour contourner cette menace. * A une plus large échelle, on active le versioning sur les serveurs virtuels pour avoir des sauvegardes supplémentaires en pleine journée et pouvoir revenir à une situation opérationnelle en un minimum de temps.   Ce qui limite le chômage technique en cas de problème lié au réseau. | Leçon |
| **Activité 2** | Avant cette technique de sauvegarde d’un ensemble de fichiers et avec différentes versions, les sauvegardes se faisaient de façon manuelle et distincte. Ce qui avait pour conséquence de perdre du temps et aussi qu’une personne dédiée devait le faire. En ce qui concerne la réinjection des données en cas de problème, cela prenait un certain temps, donc plutôt fastidieux. | Leçon |
| **Activité 3** | Pour la plupart des projets de développement, le code source est comme les joyaux de la couronne : un bien précieux dont la valeur doit être protégée. Pour la plupart des équipes de développement, le code source est un dépôt de connaissances et d'explications inestimable sur le domaine du problème, que les développeurs ont rassemblées et affinées en déployant des efforts minutieux. Le contrôle de version protège le code source à la fois contre les catastrophes et contre les dégradations occasionnelles dues aux erreurs humaines et aux conséquences imprévues. | Leçon |
| **Activité 4** |  | Type d'activité |
| **Activité 5** |  | Type d'activité |

* 1. **Section 2**

**Titre :** Installer l’environnement GIT

**Image/Gif/Vidéo d’introduction :**

**Introduction/Résumé de la section :** Nous allons voir comment installer un environnement GIT

**Objectifs :** Savoir installer et configurer votre outil de versioning

**Consignes spécifiques :**

**Durée indicative :** 15 mn

* 1. **Contenu**

|  |  |  |
| --- | --- | --- |
|  | **Contenu** | **Type d’activité** |
| **Activité 1** | Installation de git sur Windows :   * Téléchargez la dernière version : <https://gitforwindows.org/> * Exécutez le fichier téléchargé via le fichier ‘Git setup’ * Faites ‘Next’ jusqu’à ‘Finish’ * Ouvrez l’invite de commande (tapez ‘cmd’ dans la barre de recherche et si une fenêtre noire apparait, c’est gagné !!!) * Tapez les 2 commandes suivant  1. git config --global user.name "Emma Paris" 2. git config --global user.email [eparis@atlassian.com](mailto:eparis@atlassian.com)   Vérifions si Git est bien installé avec la commande : git -v  S’il vous donne une version comme : git version 2.43.0.windows.1  Vous venez de réussir votre installation.  Pour une installation sur d’autre système d’exploitation, rendez-vous sur : <https://www.atlassian.com/fr/git/tutorials/install-git> | Atelier |
| **Activité 2** |  | Fichier |
| **Activité 3** |  | Type d'activité |
| **Activité 4** |  | Type d'activité |
| **Activité 5** |  | Type d'activité |

* 1. **Section 3**

**Titre :** Création d’un repository et envoyer un fichier dessus

**Image/Gif/Vidéo d’introduction :**

**Introduction/Résumé de la section :** Nous allons découvrir comment créer un repository et déposer un fichier dessus

**Objectifs :** Savoir utiliser GIT

**Consignes spécifiques :**

**Durée indicative :** 1 h

* 1. **Contenu**

|  |  |  |
| --- | --- | --- |
|  | **Contenu** | **Type d’activité** |
| **Activité 1** | Nous allons créer un repository ou dépôt distant, mais pour cela il faut créer un compte GitHub, donc nous allons voir ça tout de suite.  Rendons-nous sur <https://github.com/>  Entrez votre adresse mail et cliquez sur ‘Sign up for GitHub’  On vérifie l’adresse mail, si c’est bon, on continu  On saisit notre mot de passe, on continu  On saisit un nom d’utilisateur, on continu  On clique sur vérifier et on résolve l’énigme  On clique sur ‘create account’  On nous demande de valider via nom adresse mail avec un code reçu  On répond aux différentes questions  On choisit la version gratuite  Voilà votre compte est créé  On clique ensuite sur ‘Create repository’  Dans ‘Repository name’ on mettra ‘premier-repository’  On choisira ‘Public’  Et on cochera ‘Add a README file’ puis ‘Create repository’  Vous venez de créer votre premier repository. | Atelier |
| **Activité 2** | Envoyer un fichier sur notre dépôt distant cela veut dire faire un ‘push’ vers notre ‘repository’.  Dans un premier temps, on va lancer Git Bash et on se place dans ‘Documents’ pour cela il faut lister les dossiers et fichiers de ou vous vous situez avec la commande ‘dir’ si vous voyez le dossier ‘Documents’ faites la commande ‘cd Documents’ suivi de ‘entrée’.  On récupère notre l’adresse https de notre repository ‘premier-repository’ en allant sur GitHub.  Au-dessus de la liste des fichiers, cliquez sur ‘<>Code’, copiez l’URL du dépôt qui commence par ‘https://github.com.github/…’.  On revient dans git Bash et à l’aide de la commande suivante, on va récupérer notre repository pour l’avoir en local (sur notre pc) :  ‘git clone <https://github.com/formation-afci/premier-repository.git>’  Allez dans ‘documents’ puis dans le dossier ‘premier-repository’  On va créer un nouveau fichier ‘fichier.txt’  On va dans notre git Bash et on va dans le dossier de la façon suivante :  ‘cd premier-repository’  Ensuite on va mettre à jour notre repository avec les commandes suivantes :  ‘git add .’ => ajoute tous les nouveaux fichiers  ‘git commit -m ‘’maj080224’’ ‘ => on prépare nos nouveaux fichiers avec un commentaire pour être envoyés  ‘git push origin main’ => on envoi tous les fichiers préparés sur notre repository, une autorisation va vous être demandé, acceptez quand même !!!  Allons vérifier sur GitHub, et là, miracle notre fichier est bien là ! | URL |
| **Activité 3** |  | Type d'activité |
| **Activité 4** |  | Type d'activité |
| **Activité 5** |  | Type d'activité |

* 1. **Evaluation finale**

Au choix :

* + 1. **Test**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Question** | **Réponses possibles** | **Bonne réponse** |
| **Q1** |  |  |  |
| **Q2** |  |  |  |
| **Q3** |  |  |  |
| **Q4** |  |  |  |
| **Q5** |  |  |  |
| **Q6** |  |  |  |
| **Q7** |  |  |  |
| **Q8** |  |  |  |
| **Q9** |  |  |  |
| **Q10** |  |  |  |

* + 1. **Devoir**

**Consigne pédagogique :**

**Consigne opératoire :**

* + 1. **Autre activité**

**Renseignez**

**ésumé et conclusion** :