

Bonnes pratiques et méthodologies de développement

Brice AIELLO – brice.aiello@tomorrow-security.com



Tomorrow learning





Table des matières

- Gestion de versions
- Environnement
- Code design et Style guides
- Documentation
- Mini-projet

Gestion de versions

- Historiser le code
- Repérer les bugs
- Propager la connaissance
- Amélioration de la qualité



Changes requested
[Hide all reviewers](#)

1 review requesting changes

✓	github — Build #5630178 Details
✗	mdo requested changes See review
●	yourfrienderin was requested for review
●	stephbwillis was requested for review

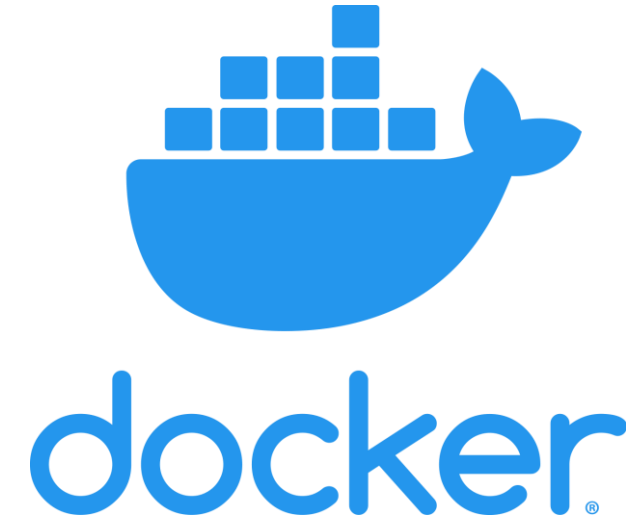
Environnement standardisé

Développement

Test

Production

- Faciliter l'onboarding
- Fonctionnement ISO
- Déploiement testé et facilité



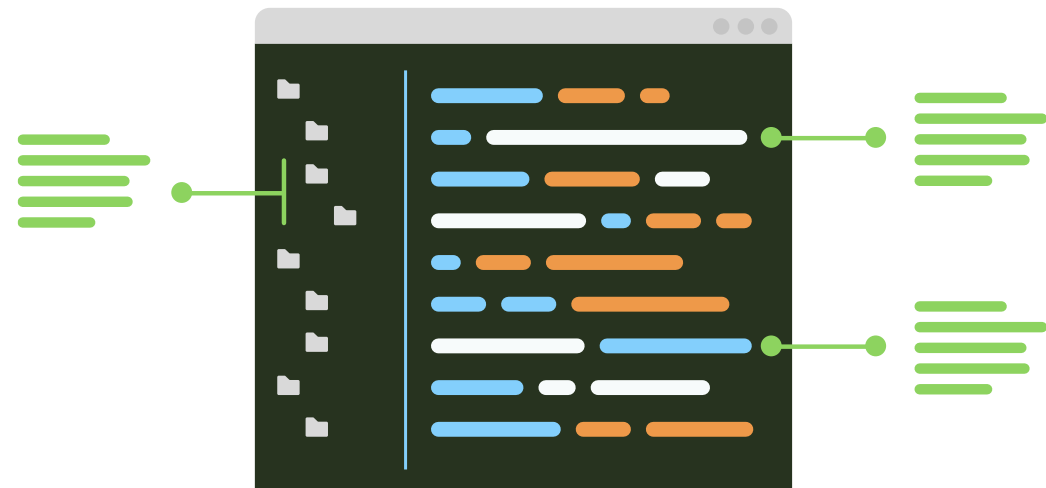
Code design

- Nommage des variables
- Division des choses longues / grandes
- Séparation des responsabilités
- Refactoring agile
- Commentaires



Documentation

- Facilite l'onboarding
 - Installation et démarrage du projet
 - Partage des règles
- Propage la connaissance
- Rappel les besoins
- Communication





Style guides

Ensemble de règles et outils permettant d'améliorer la qualité de votre code, pour faciliter la reprise et la maintenabilité de celui-ci.

Langage	Standard	Linters (outils)
PHP	<ul style="list-style-type: none">• PSR-12	php-cs-fixer / phpstan
Python	<ul style="list-style-type: none">• PEP8	Flake8 / Black
Go	<ul style="list-style-type: none">• Google Go Style Guide	revive
Javascript	<ul style="list-style-type: none">• AirBnB JavaScript Style Guide• Javascript Standard Style• Google JavaScript Style Guide	ESLint / Prettier
C++	<ul style="list-style-type: none">• Google C++ Style Guide	clang-tidy





Mini-projet : Queezie

Introduction

- L'objectif de ce mini-projet est de mettre en application les concepts précédents dans un contexte technique.
- Ce projet pourra être réalisé dans le langage de votre choix, sur la plateforme de votre choix (web, mobile, desktop, ...)

1ère partie : Structure

- Mettre en place l'architecture de base permettant de démarrer facilement le projet (versionning, environnement, gestion de dépendances, ...)
- Expliquer, à l'aide d'une documentation, comment participer à ce projet (readme, guidelines, base documentaire, ...)

2ème partie : Code Style

- Définir les règles de style de ce projet
- Mettre en place un linter automatisé

3ème partie : Dev

- Développer le projet pour répondre aux besoins exprimés sur page suivante

Mini-projet : Queezie

Queezie est une application de ... quizz ! 

Queezie devra répondre aux besoins suivants :

- Une **question** pourra apparaître dans plusieurs **questionnaires**
- Un **domaine** sera attribué à chaque **question**
- Une **question** pourra être de type simple, multiple ou booléenne
- Une **question** pourra contenir jusqu'à 4 **réponses**
- Une **réponse** pourra être correcte ou incorrecte
- Un utilisateur pourra participer à une **session de quizz**
- Une **session de quizz** sera valide pendant une période définie
- Un **questionnaire** aura une durée définie

A toi de jouer !

