

# 1 - Introduction

1MODE - Modélisation d'applications



# *Sommaire*

1. Contextualisation
2. Outils utilisés

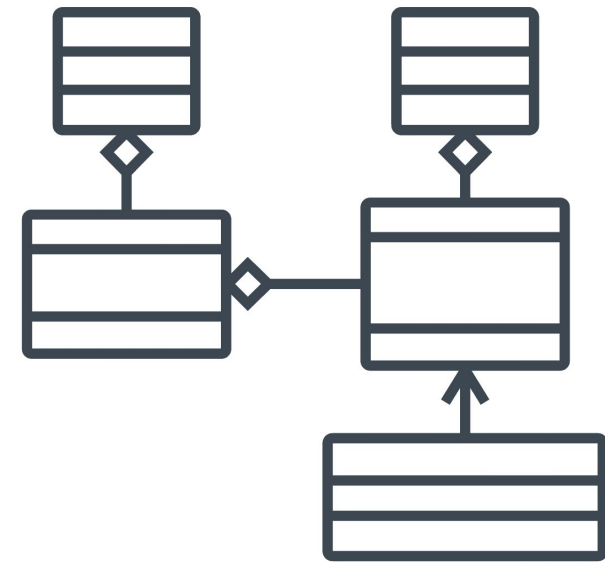


# 1. Contextualisation

# 1. Contextualisation

## Qu'est-ce que la modélisation ?

- Il s'agit d'un processus de création de **représentations abstraites** d'un système logiciel
- Il a pour but de mieux comprendre, concevoir et communiquer la **structure** et le **comportement** de ce système

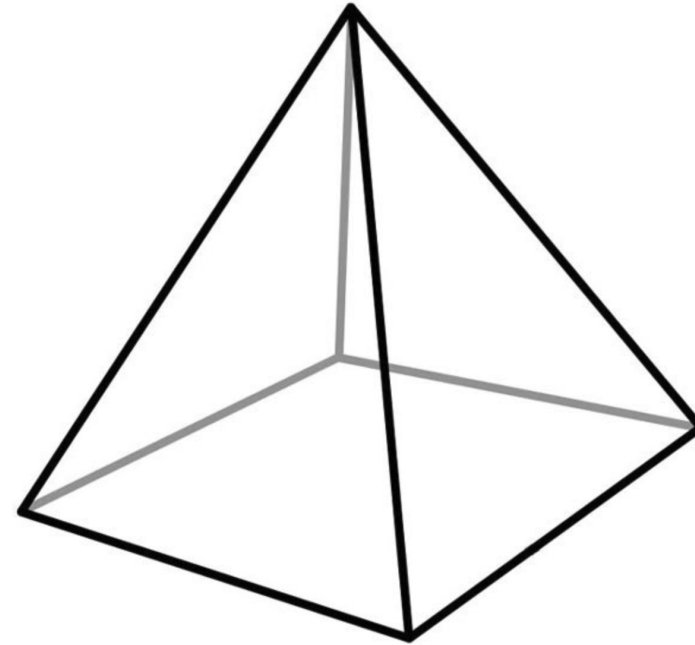


# 1. Première partie

## Exemple de modélisation



Pyramide de Mykérinos



Pyramide à base carrée

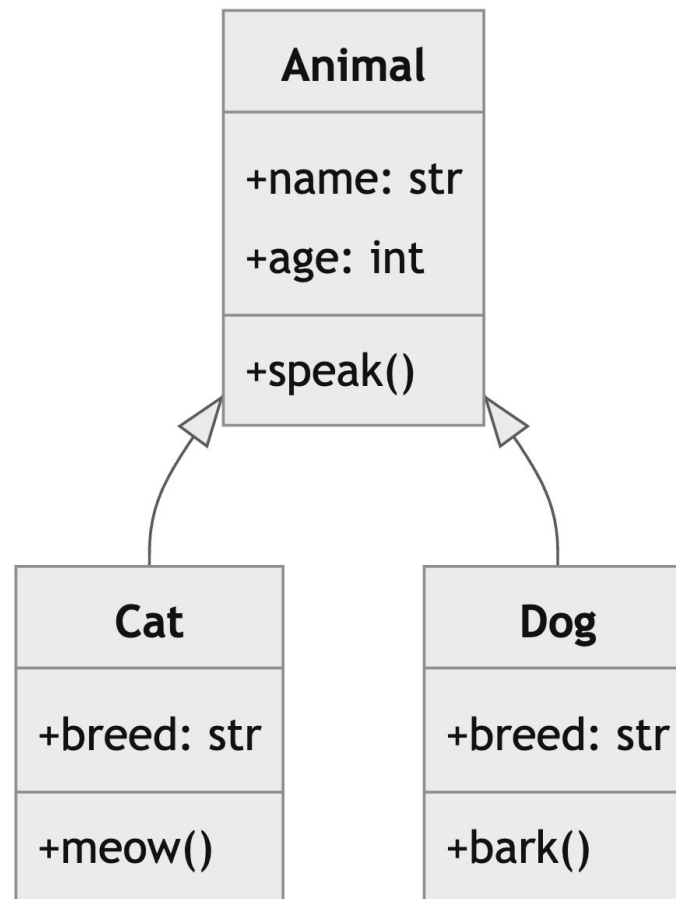
# 1. Contextualisation

**Pourquoi passer par un processus de modélisation lorsqu'on développe des logiciels ?**

- **Visualiser** l'architecture d'une application pour faciliter la communication entre les parties prenantes (développeurs, clients, etc.)
- Spécifier la **structure** et le **comportement** du système de manière précise et non ambiguë
- Guider l'**implémentation** en fournissant un modèle à partir duquel le code peut être écrit
- **Documenter** les décisions de conception prises lors du développement

# 1. Contextualisation

## Premier exemple de modélisation logicielle



# 1. Contextualisation

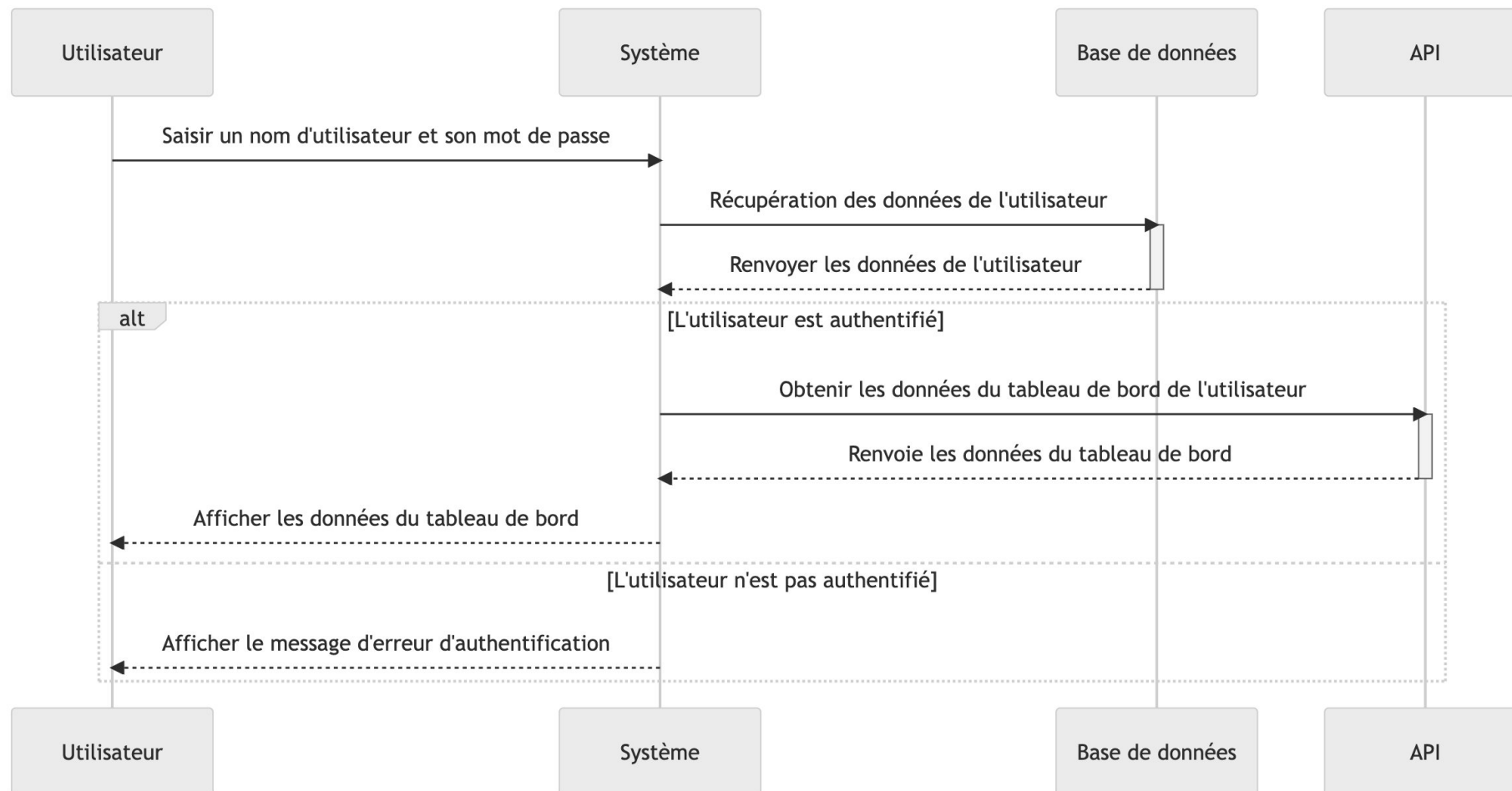
## **Premier exemple de modélisation logicielle**

- Dans cet exemple, nous avons trois classes : Animal, Cat, et Dog
- La classe Animal possède trois attributs (name, age) et une méthode (speak)
- Les classes Cat et Dog héritent de Animal
- Chacune des classes a également ses propres attributs et méthodes spécifiques (breed, meow, bark)



# 1. Contextualisation

## Second exemple de modélisation logicielle



# 1. Contextualisation

## Second exemple de modélisation logicielle

- Dans cet exemple, il y a deux participants : **Utilisateur** et **Système**
- L'utilisateur entre son nom d'utilisateur et son mot de passe, et le système interroge la base de données pour obtenir les informations de l'utilisateur
- Si l'utilisateur est authentifié, le système récupère les données du tableau de bord de l'utilisateur à partir d'une **API**, les affiche à l'utilisateur, et la séquence se termine
- Si l'utilisateur n'est pas authentifié, le système affiche un message d'erreur à l'utilisateur

# 1. Contextualisation

## Remarques

- Ces deux diagrammes ont pour seul but de donner une idée des concepts explorés dans ce cours
- Ces diagrammes suivent une notation formelle et **standardisée** qu'il convient de respecter
- Nous apprendrons en détail ces notations dans la suite

# 1. Contextualisation

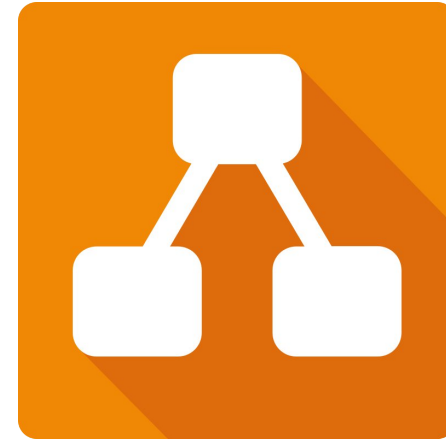


## 2. Outils utilisés

## 2. Outils utilisés

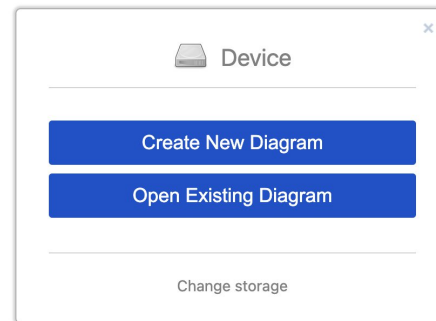
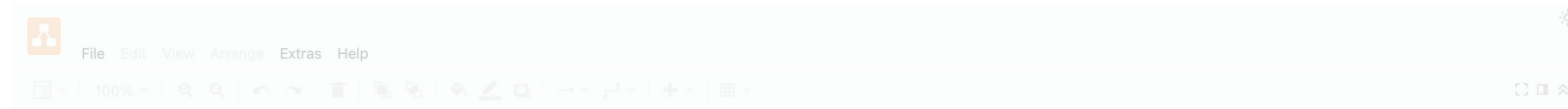
### Diagrams.net

- <https://app.diagrams.net/>
- Conception de multiples types de diagrammes, y compris ceux de modélisation logicielle
- Application web
- Gratuit
- Export au format PDF, images, ou vectoriel



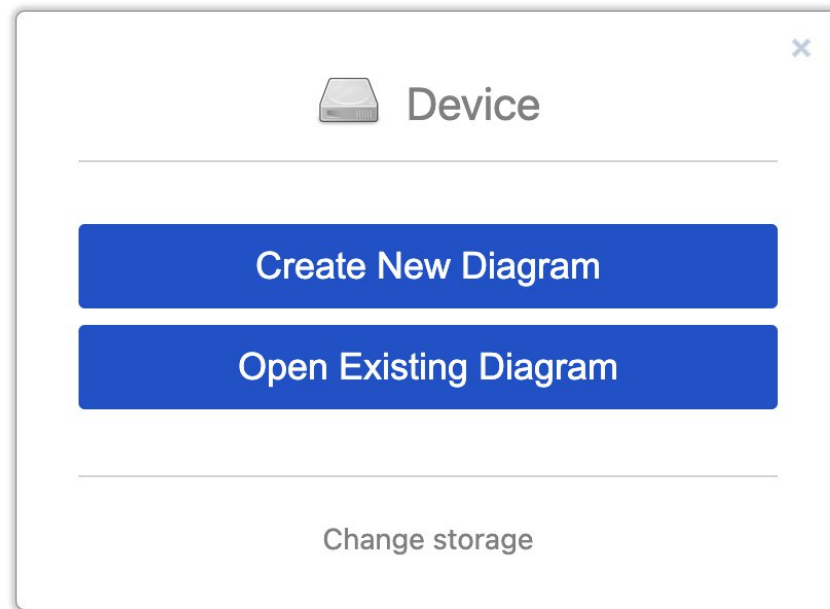
## 2. Outils utilisés

### Diagrams.net



## 2. Outils utilisés

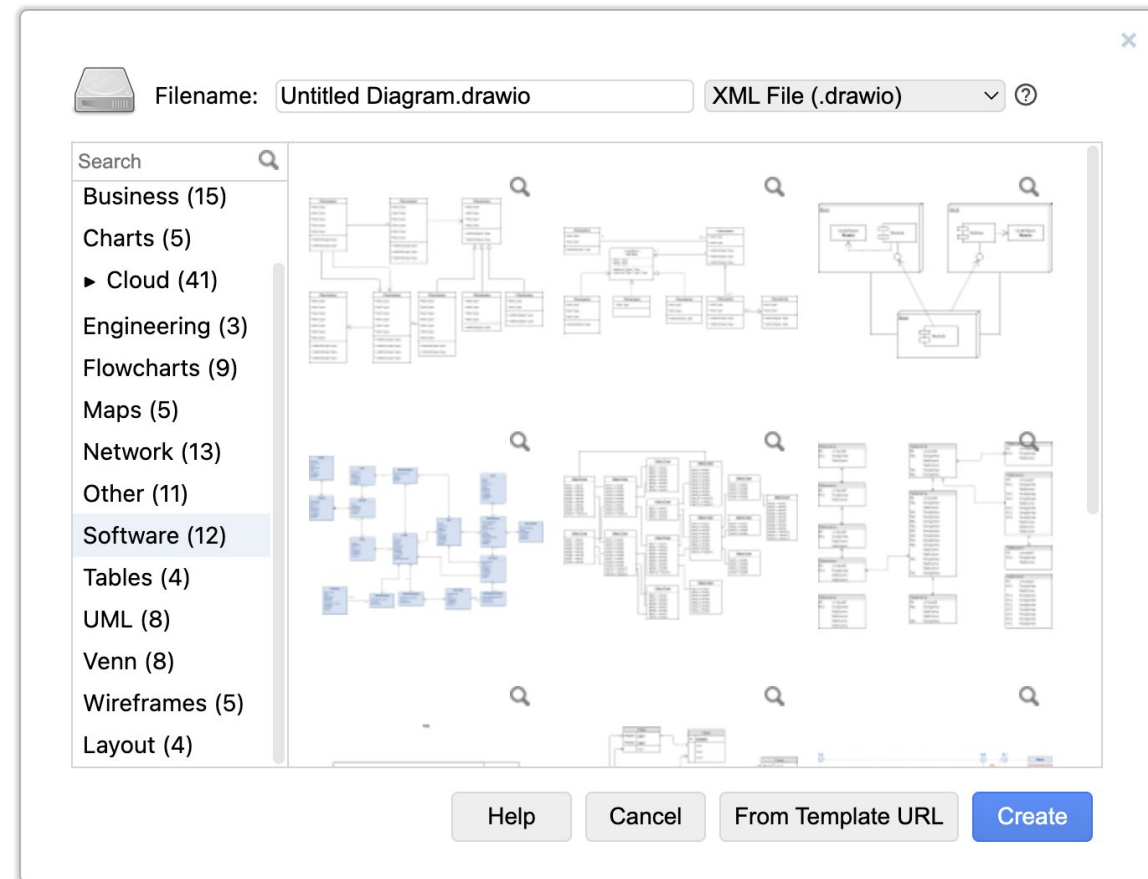
### Diagrams.net





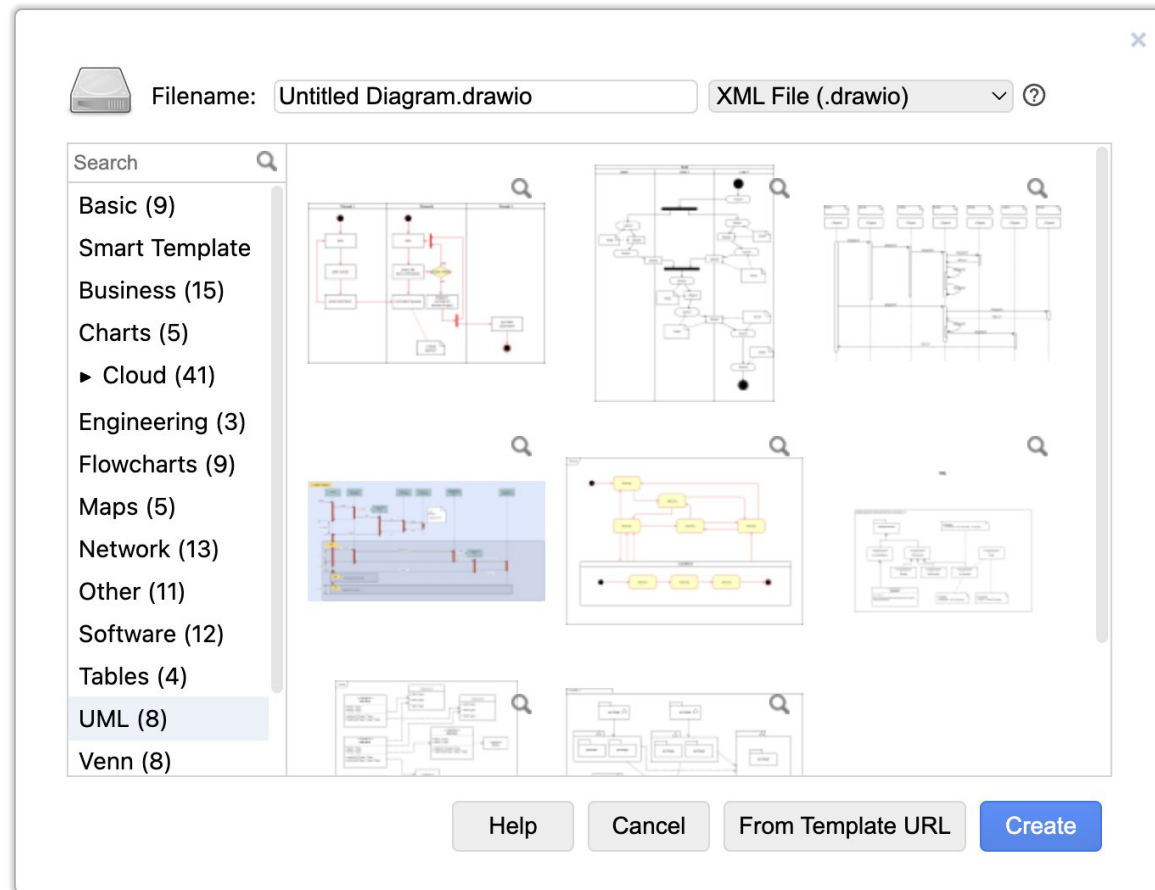
## 2. Outils utilisés

### Diagrams.net



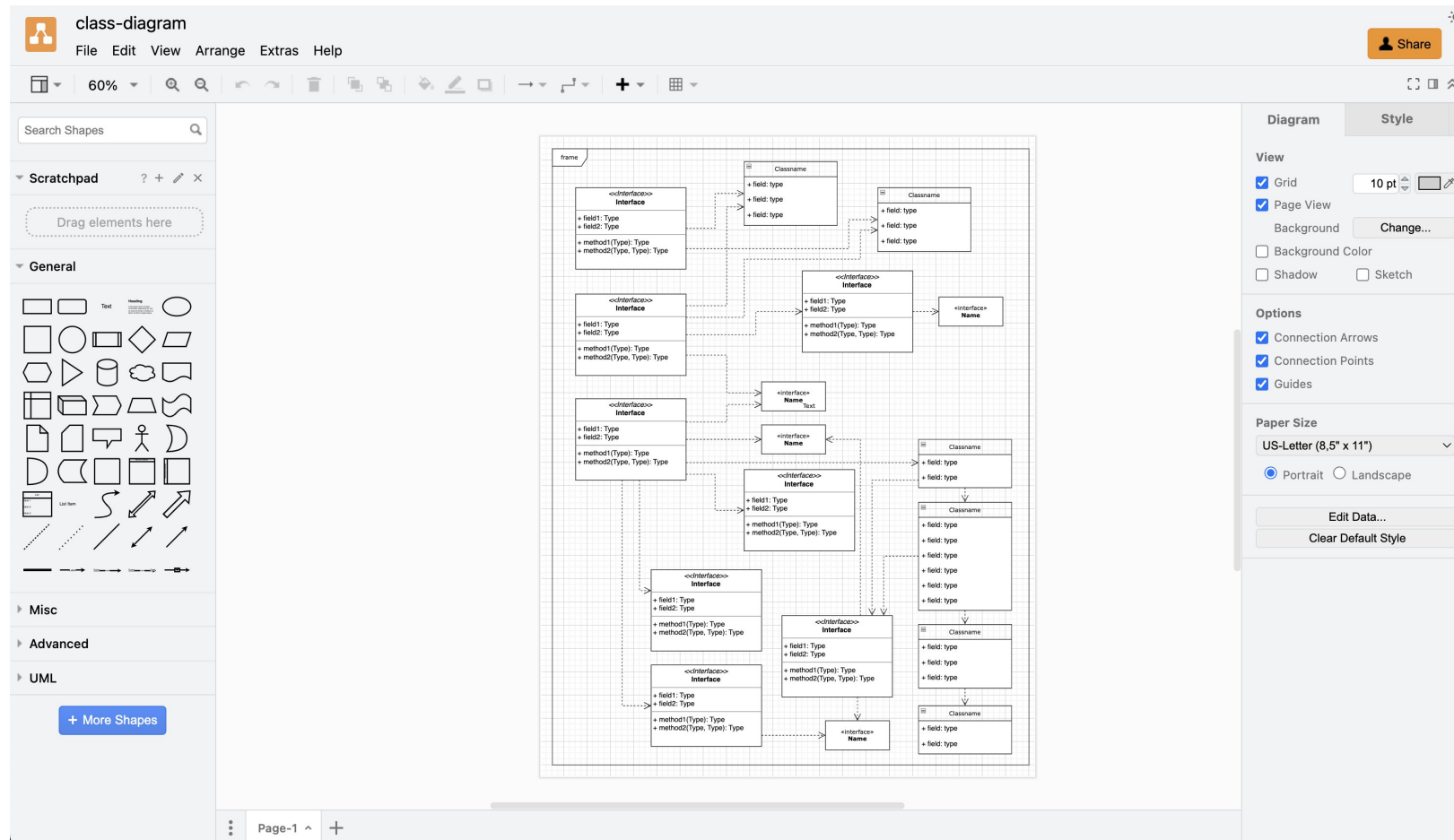
## 2. Outils utilisés

### Diagrams.net



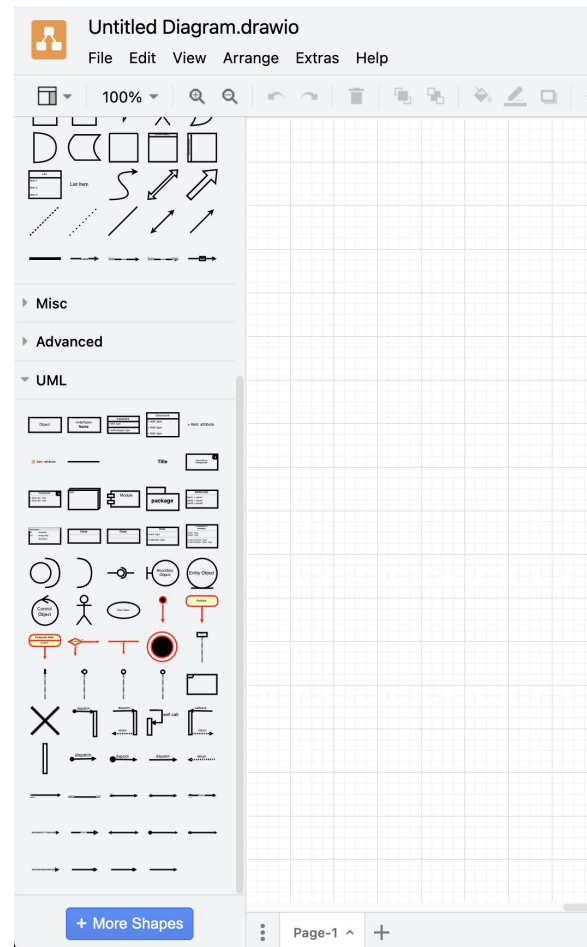
## 2. Outils utilisés

### Diagrams.net



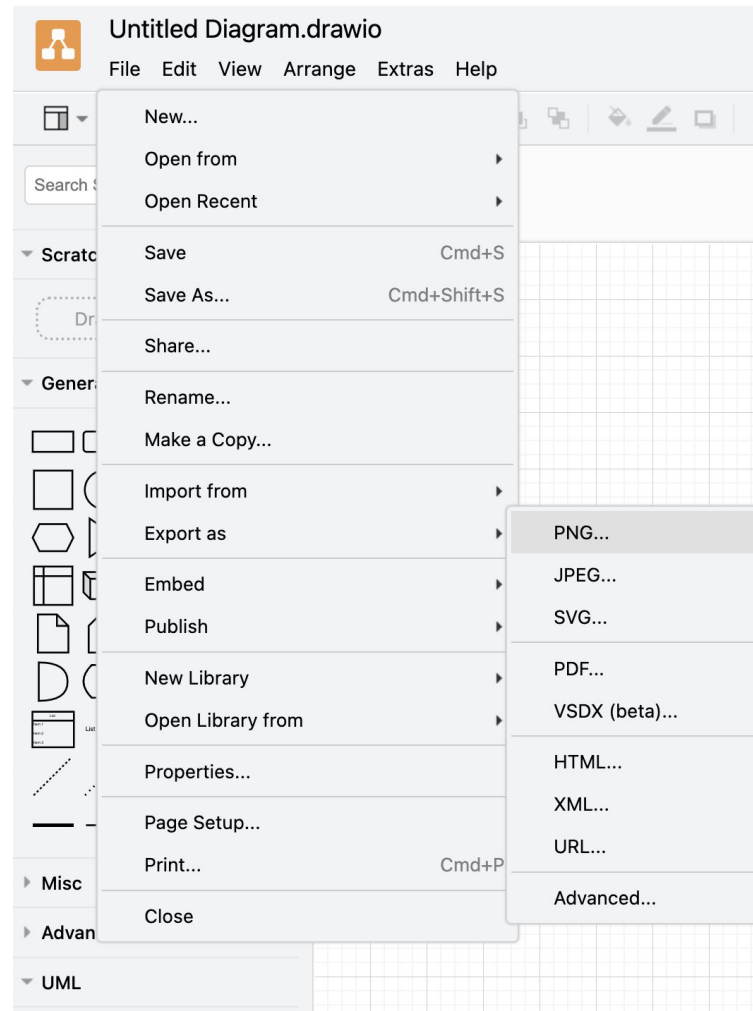
## 2. Outils utilisés

### Diagrams.net



## 2. Outils utilisés

### Diagrams.net



## 2. Outils utilisés

### TypeScript

- Dans ce cours, nous utiliserons le langage **TypeScript** (TS)
- Il s'agit d'un **sur-ensemble** de JavaScript, signifiant que tout code JS valide est également valide en TS
- À la différence de JavaScript, c'est un langage **fortement typé**
- Il ajoute à JavaScript un certain nombre de fonctionnalités, dont la plupart ont un lien avec l'orienté objet



## 2. Outils utilisés



