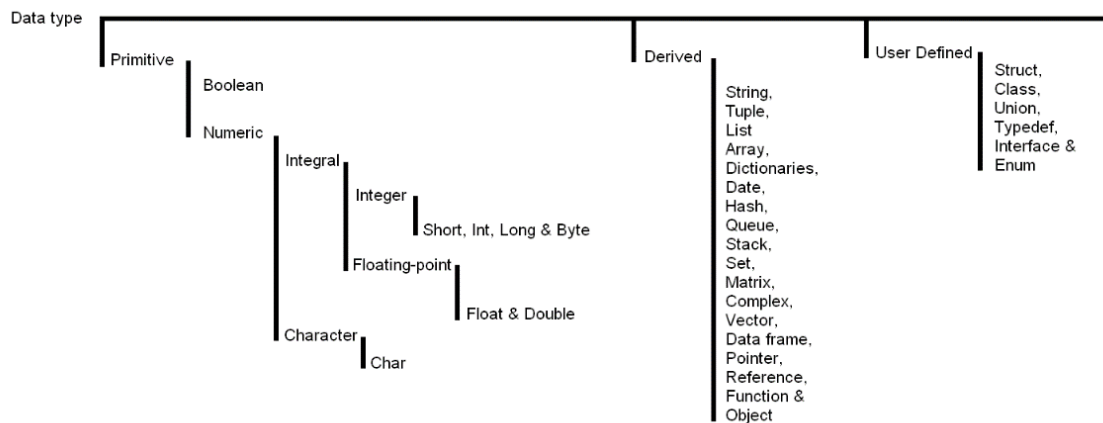


L'univers du développeur 2021

⇒ Environnement de développement

- Environnement virtuel (pipenv, etc.)
- IDE or Text Editor ?(VSC, Vim, Notepad++, outils JetBrains, etc.)
- Cloud platform: Jupyter, Collab, etc.
- Package manager : anaconda (data science), npm (node), pip (python), etc.

⇒ Les types de données



⇒ Structures conditionnelles

- Les boucles : while / do...while (repeat... until) & for.
- Les conditions : if...else & switch.
- Contrôle de boucles : repeat, break, next, continue, pass & txtProgressBar.

⇒ Les principaux paradigmes de programmation



⇒ Fonctions, procédures et arguments

⇒ Built-in functions

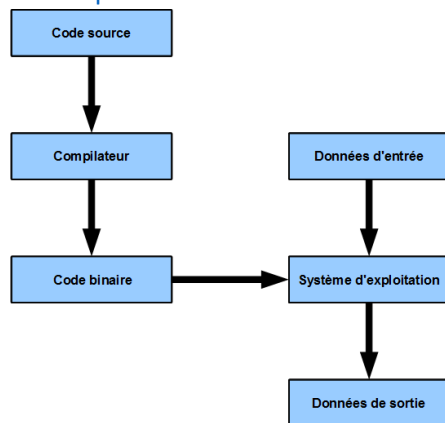
⇒ Les bibliothèques principales

⇒ Spécificité et bonnes pratiques du langage

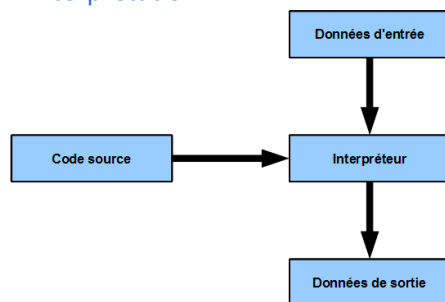
- Point forts et faiblesses
- Les commentaires
 - Single Line Comment
 - Multi Line Comment
 - Documentation Comment
- Convention de nommage (classe, méthode, fonctions, constante, variable, module, package, etc.)

⇒ Les langages interprétés vs compilés + Input/Output

○ La compilation

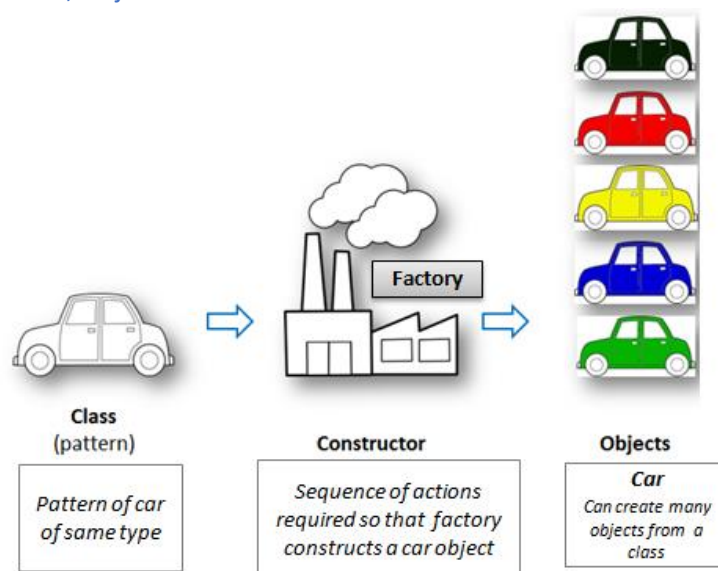


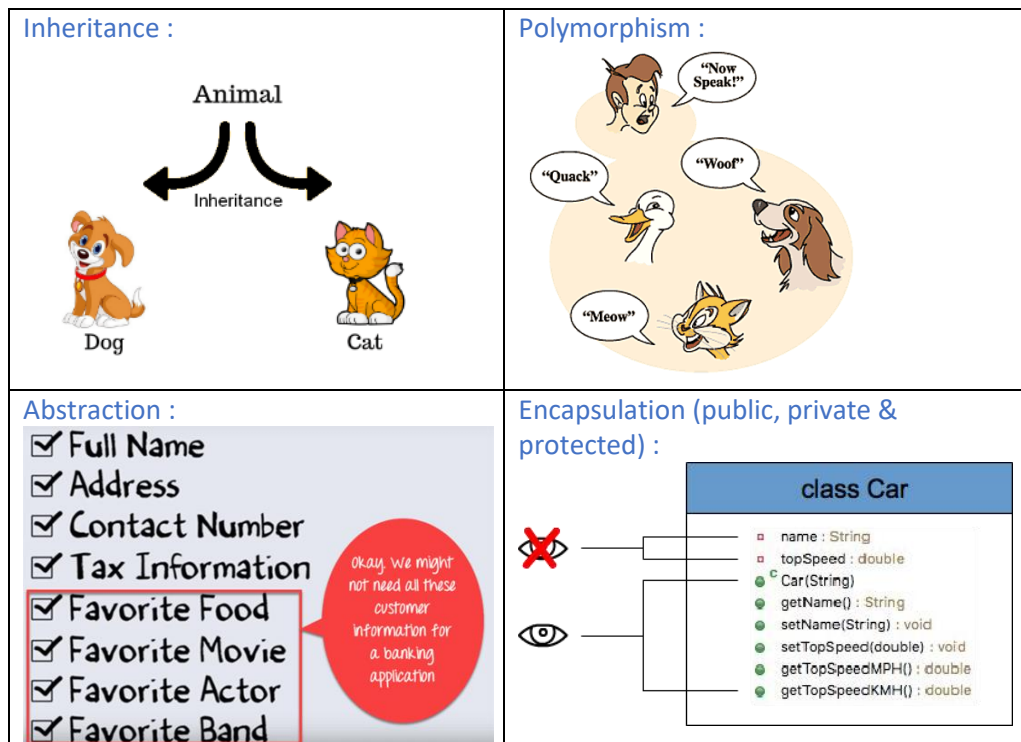
○ L'interprétation



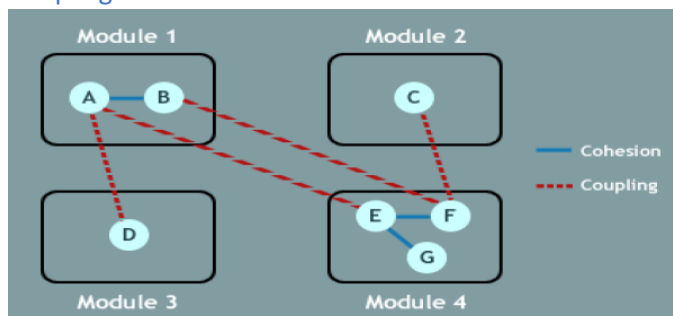
⇒ OOPs (Object-Oriented Programming System)

○ Classes, Object

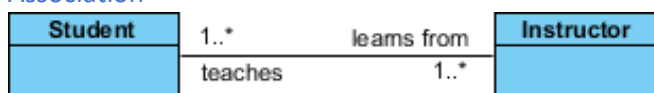




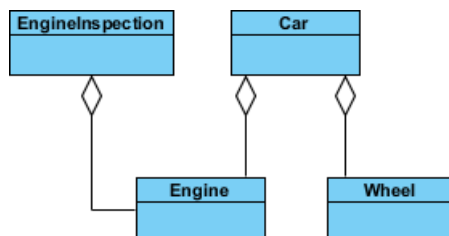
○ Coupling vs Cohesion



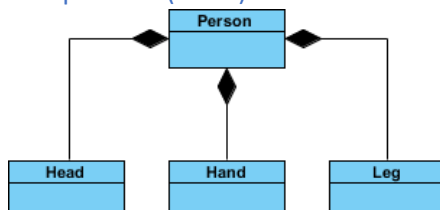
○ Association



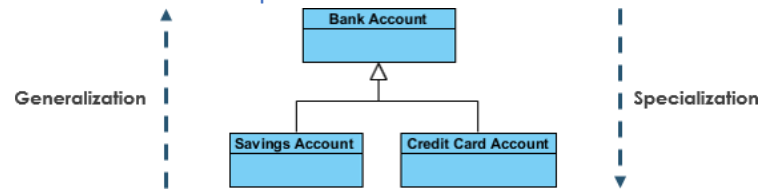
○ Aggregation (uses)



○ Composition (owns)



- Generalization and specialization

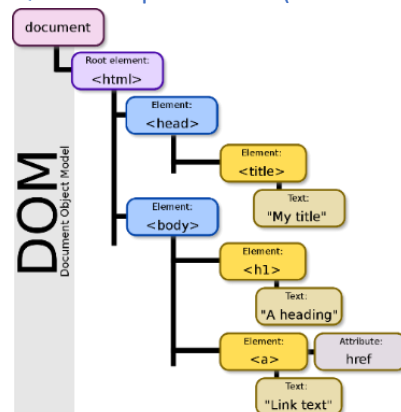


⇒ Développement web

- La standardisation W3C, ECMAScript, RFC, etc.
- CMS (Content Management System) : #1 WordPress, #2 Drupal, #3 Shopify, etc
- Référencement naturel SEO (Search Engine Optimisation)
- Référencement payant SEA (Search Engine Advertising)
- Front-end technologies :

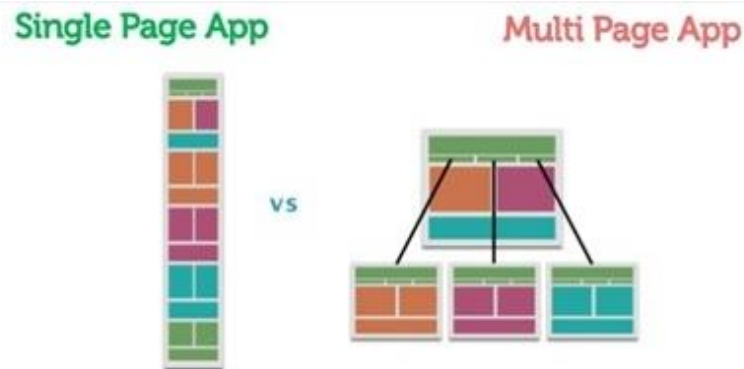


- Qu'est-ce que le DOM (Document Object Model)

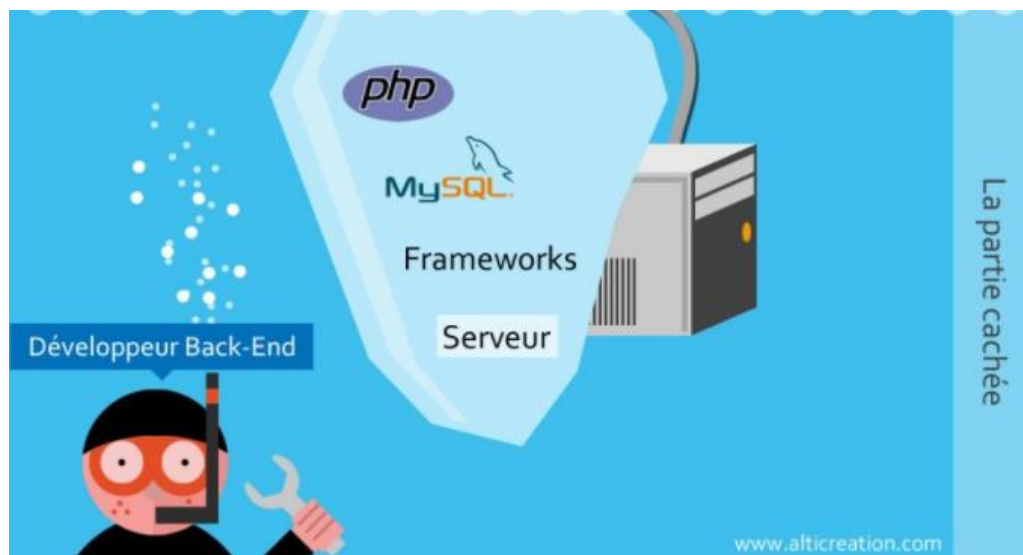


- L'essentiel des tags HTML et des styles CSS
- JavaScript :
 - Dynamisation des pages
 - Manipulation du DOM avec la librairie jQuery
 - Les fonctions à flèches depuis l'ES6.
 - L'asynchronisme avec les promesses (promise, resolve, reject)
- Framework JS :
 - #1 React

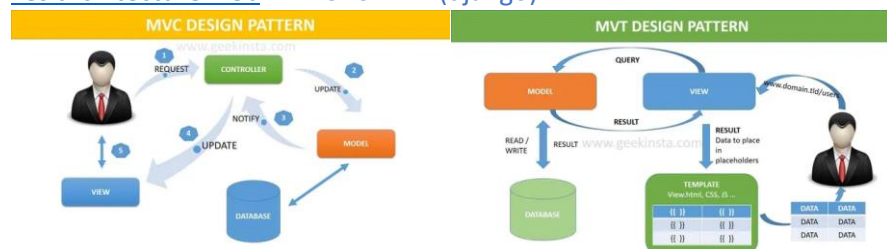
- #2 Angular
- #3 Vue.js
- Framework CSS : #1 Bootstrap, #2 Semantic UI
- Créer une maquette (tous les écrans d'une app) et les coder
 - #1 Adobe XD (Windows et Mac) / #2 Sketch (Mac)
- SPA (Single Page Application) vs MPA (Multi Page Application)



- Back-end technologies :

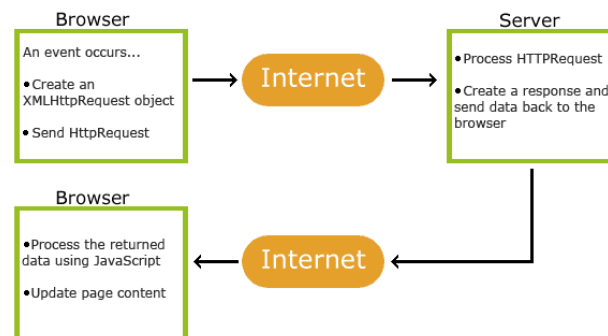


- Server framework :
 - Node : #1 Express.js (Jest, Mocha, Jasmine, Karma, selenium, etc)
 - Python : #1 Django et #2 Flask (unittest, selenium, etc)
 - PHP #1 Laravel et #2 Symfony (testunit, etc)
 - Java : #1 Spring (JUnit, selenium, etc)
 - C# : #1 ASP.NET (MSTest, selenium, etc)
- Les architecture web : MVC vs MVT (django)



- Web services (deux grandes familles) :
 - SOAP/XML (WS)
 - Simple Object Access Protocol
 - Extensible Markup Language
 - REST/JSON (API)
 - REpresentational State Transfer
 - JavaScript Object Notation
- Communication client/serveur
 - La méthode AJAX (Asynchronous JavaScript And XML)

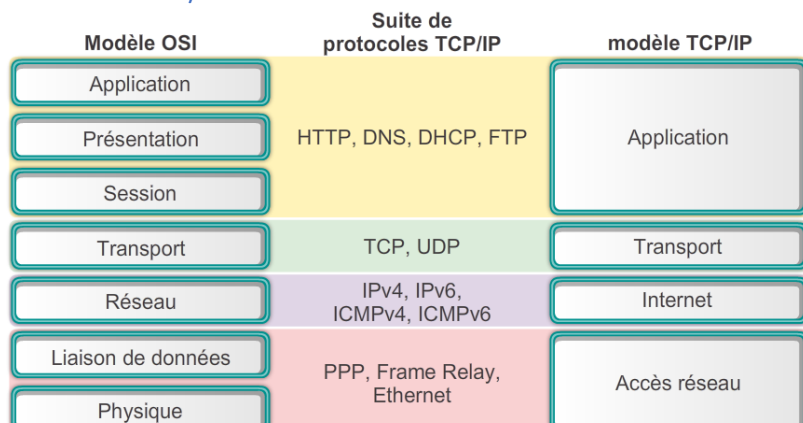
How AJAX Works



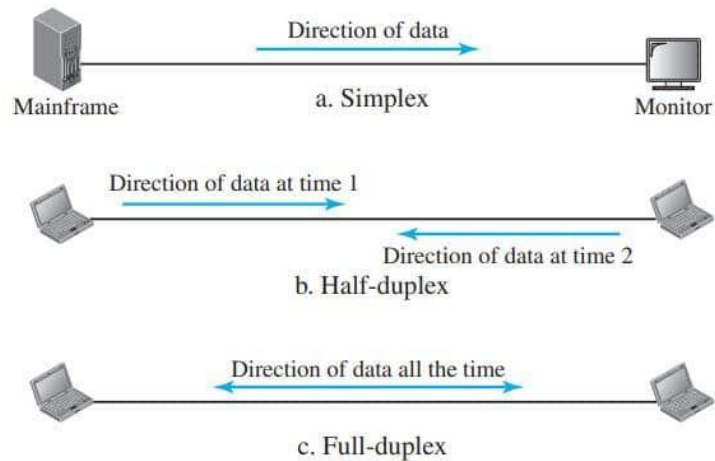
- Les protocoles internet :

Transférer des données (base d'internet)	TCP	Transfert Control Protocol (protocole de contrôle des transmissions) est un protocole qui gère les règles de transport des données sur Internet. Pour cela, il segmente les informations en paquets et les envoie à un ordinateur. L'ordinateur destinataire renvoie un message de confirmation à chaque fois qu'il reçoit un paquet. L'ordinateur émetteur réessaye d'envoyer les paquets pour lesquels il n'a pas reçu de message de confirmation. Lorsque l'ordinateur destinataire a reçu tous les paquets, il reconstitue l'information.
	IP	Internet Protocol (Protocole Internet) est un protocole qui identifie chaque ordinateur en lui adressant une adresse IP unique.
Naviguer sur le Web	HTTP	Hyper Text Transfert Protocol est le protocole qui permet à un client web (un navigateur) d'échanger avec un serveur web (l'ordinateur sur lequel est stocké le site qu'on visite)
	HTTPS	Hyper Text Transfert Protocol Secured est la version de l'HTTP sécurisée qui permet l'échange de données cryptées (grâce aux protocoles SSL et TLS).
Echanger des courriels	POP3	Post Office Protocol version 3 (Protocole du bureau de poste) est le protocole qui permet la réception de méls
	SMTP	Simple Mail Transfer Protocol (Protocole simple de transfert de courriel) est le protocole qui permet l'envoi de méls
	IMAP	Internet Message Access Protocol est un protocole qui, comme POP3, permet de récupérer des messages. L'IMAP a l'avantage de synchroniser votre messagerie avec le serveur sur lequel sont stockés vos messages. Toutes les actions faites sur votre messagerie (suppression de message, création de dossier...) sont reportées sur le serveur. De plus L'IMAP laisse le message sur le serveur. Résultat, vous pouvez consulter vos messages avec plusieurs terminaux (ordinateurs, smartphones) car vous avez toujours accès aux mêmes données
Echanger des fichiers	FTP	Filles Transfert Protocol (Protocole de transfert de fichier) est un protocole qui permet d'échanger des fichiers entre deux ordinateurs. Ce protocole est très utilisé pour envoyer les fichiers d'un site web sur le serveur qui l'héberge.

- Le modèle TCP/IP et le modèle OSI



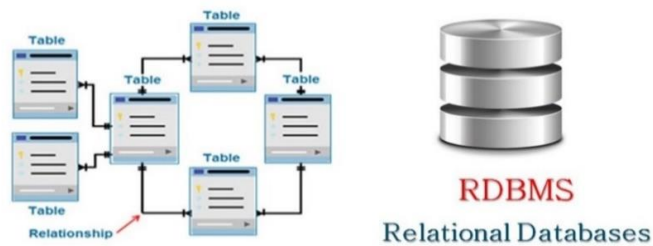
- Duplex (canal de communication)



- Les Websockets sont une spécification full-duplex du protocole HTTP
 - Utilisé pour le développement de messagerie instantanée

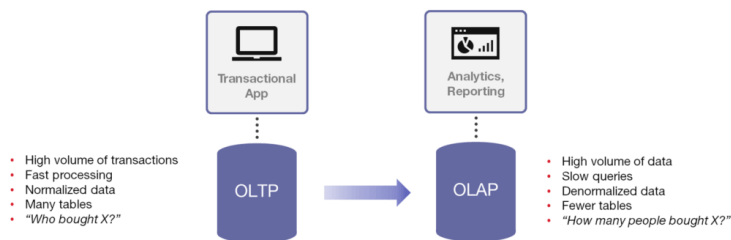
⇒ Base de données :

- RDBMS / SGBDR :

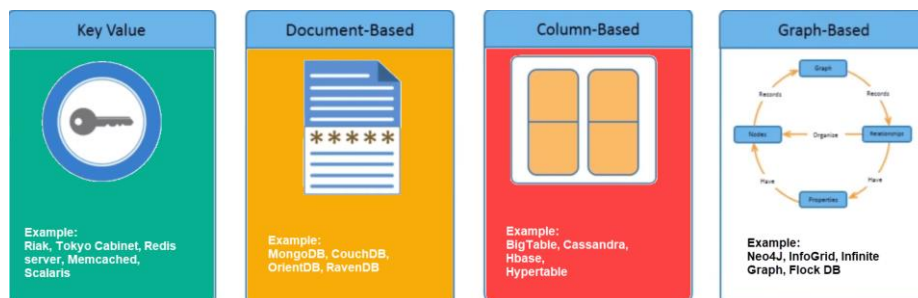


- Propriétaire : #1 Oracle, #2 Microsoft Server #3 MySQL,
- Libre : #1 PostgreSQL #2 MariaDB, #3 SQLite
- OLTP (Online Transactional Processing) / OLAP (Online analytical processing)

OLTP vs OLAP



- NoSQL databases :



- Les services cloud : #1 Azure, #2 AWS, #3 Google cloud, #4 Alibaba cloud
- ORM (Object-Relational Mapping)

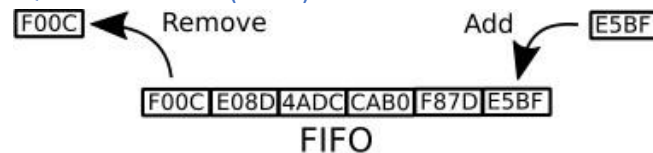
⇒ La mémoire :

- Allocation statique vs dynamique

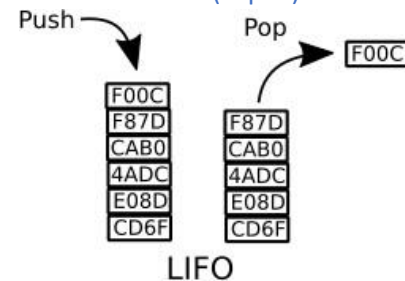
Mode d'allocation			Région mémoire utilisée	Avantage
statique			text, data ou BSS	utilisation plus efficace du temps d'exécution
dynamique	sur la pile (stack)	(lexicale ou automatique)	stack	utilisation plus efficace du temps d'exécution
	sur le tas (heap)	(allocation dynamique de mémoire)	heap	utilisation plus efficace de l'espace mémoire

- Stockage de la mémoire

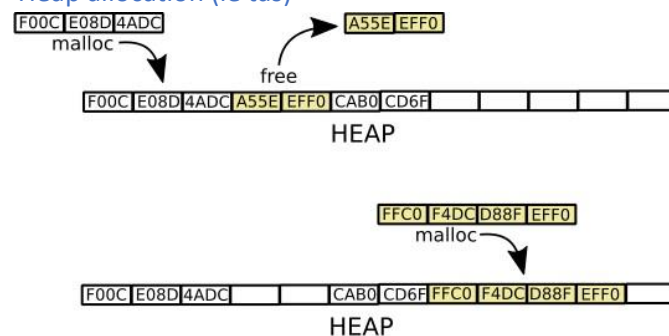
- Queue allocation (la file)



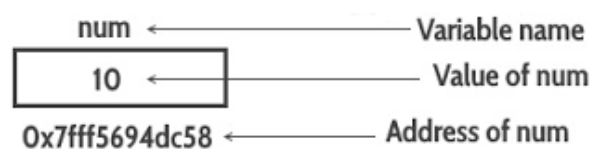
- Stack allocation (la pile)



- Heap allocation (le tas)



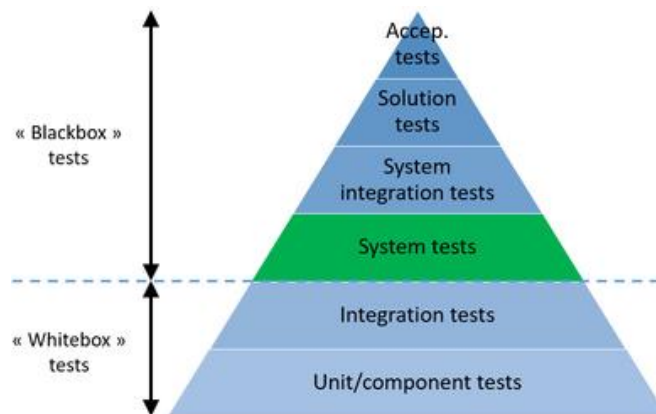
- Les pointeurs



- ⇒ File management (read, write, append, create, etc.)
 - Les types de fichiers (TXT, CSV, JSON, PDF, etc.)
- ⇒ Les algorithmes (voir : [project euler](#))
 - Sort Algorithms (Merge Sort, Quick Sort, Bucket Sort, Heap Sort, Counting Sort)
 - Search Algorithms (Binary search, Depth/Breadth First Search)
 - Hashing
 - Dynamic Programming
 - Exponentiation by squaring
 - String Matching and Parsing (KMP Algorithm, Regular Expression)
 - Primality Testing Algorithms

⇒ Les tests :

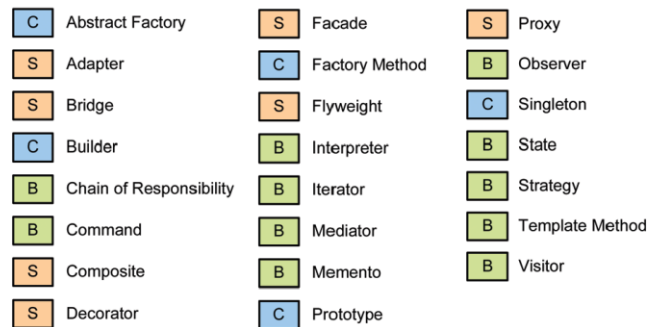
- Classification :



- Autres :
 - End-to-End testing (e2e)
 - Tests de non-régression
 - Test fonctionnels et non fonctionnels
 - Smoke test and sanity check
 - Proof of Concept (PoC)
- ⇒ Exception Handling :
 - **try... raise/catch/except...finally/except**
 - **throws...throw**
- ⇒ Les designs patterns (outil apporte des solutions aux problèmes communs de développement logiciel en rendant le code flexible, réutilisable et maintenable)
 - Il existe 3 grand type de design pattern
 - Creational « factory » : création
 - Behavioral: comportemental

- Structural : structurante

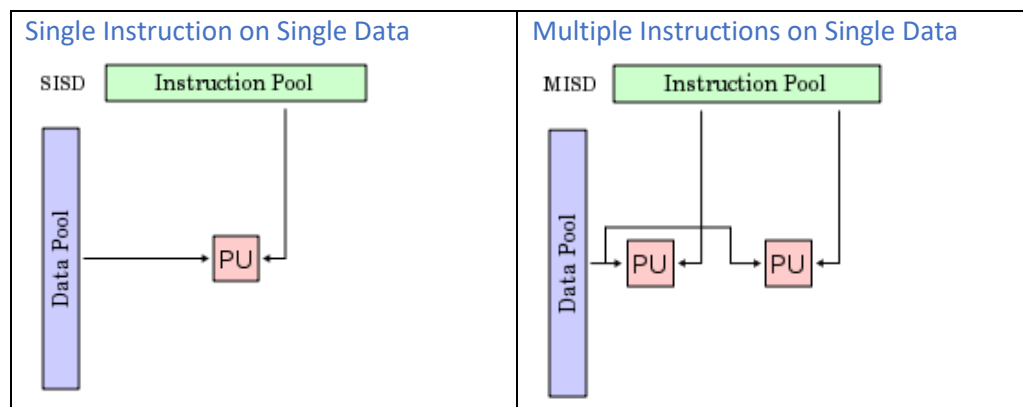
THE 23 GANG OF FOUR DESIGN PATTERNS

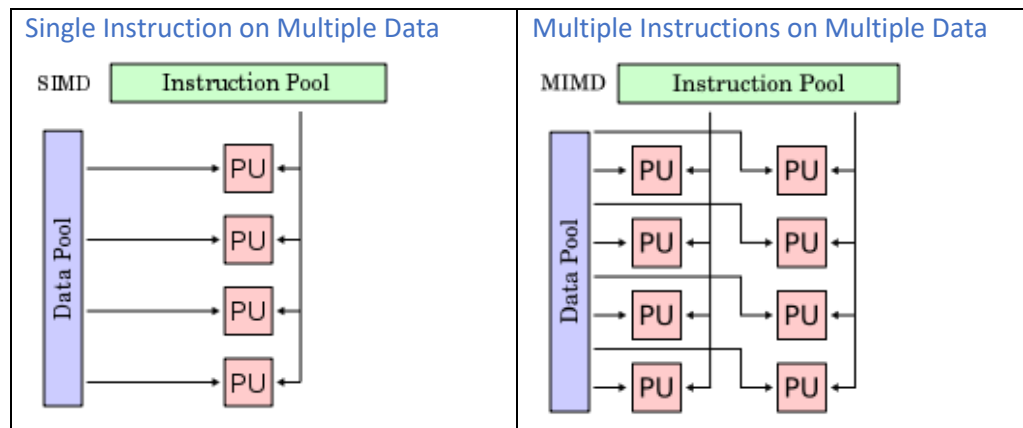


⇒ Documentation logicielle

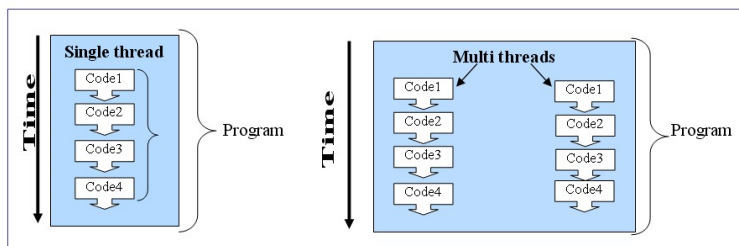
- L'expression de besoin
 - Définit le besoin du métier lors d'une demande.
- Architecture / Conception : diagrammes UML
 - Vue d'ensemble sur le logiciel. Elle inclut les relations à l'environnement et les principes à utiliser dans la conception et la réalisation des composants logiciels.
- Technique
 - Documentation du code, algorithmes, interfaces, et interfaces de programmation (API).
- Utilisateur
 - Manuels pour les utilisateurs, administrateurs systèmes et personnel de support.
- Marketing
 - Instructions sur le produit et garantie promotionnelle.

⇒ L'architecture des ordinateurs (Taxonomie de Flynn)

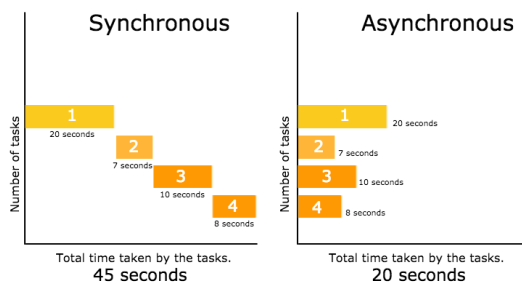




⇒ Single-Thread vs Multi-Thread programming

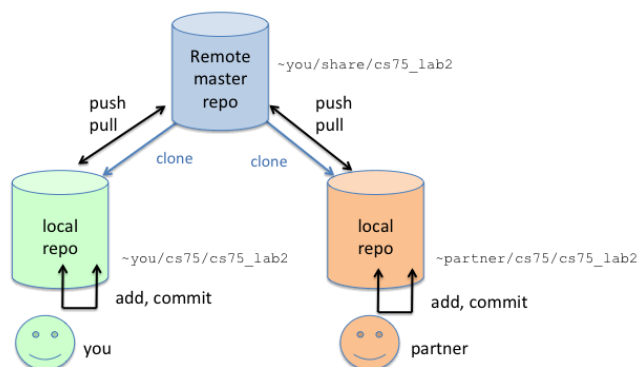


⇒ Synchronous vs Asynchronous programming



⇒ DevOps

- Git & Git CLI (#1 GitHub, #2 Gitlab, #3 Bitbucket)



- Semantic versioning

Semantic Versioning

La numérotation de type **XYZ** est utilisée par de nombreux logiciels.

La convention **Semantic Versioning** propose la signification suivante :

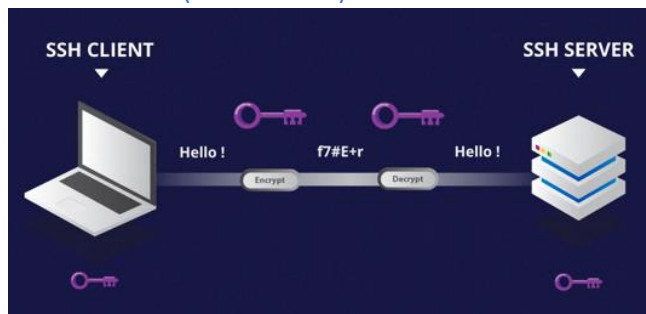
- X correspond à la **version majeure** : changements non rétrocompatibles. Les évolutions majeures apportent de nouvelles fonctionnalités, en changeant radicalement l'apparence ou l'architecture du logiciel.
- Y correspond à la **version mineure** : ajouts de fonctionnalités rétrocompatibles, principalement des corrections de bugs, ajouts de quelques fonctionnalités.
- Z correspond au **correctif** : corrections d'anomalies rétrocompatibles, failles de sécurité.

Cette convention est de plus en plus adoptée, notamment dans le milieu de l'*open-source*, comme Python.

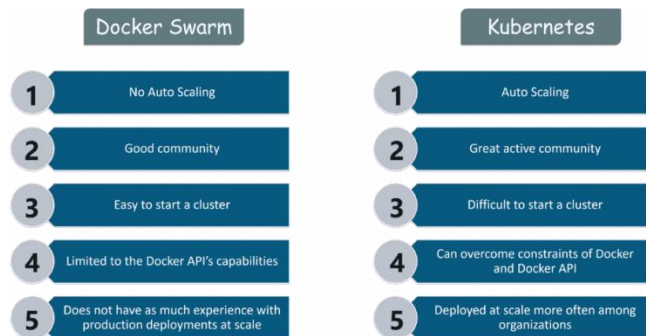
- Linux : le shell (bash) et les scripts



- Protocole SSH (Secure SHell)



- Docker vs Kubernetes



⇒ Data Science



- Big Data
- Intelligence artificielle
 - Les réseaux neuronaux
- La blockchain
- Machine learning :
 - Classification VS Régression

- Apprentissage : supervisé, non/semi-supervisé, par renforcement et incrémental
 - Algorithmes (k-moyennes / proches voisins, etc.)
- Python libraries
 - Machine Learning : scikit-learn, orange, keras
 - Web Scraping : beautifulsoup, selenium
 - Deep Learning : tensorflow, pytorch
 - Data Visualization : matplotlib, seaborn
 - Data Management : pandas, numpy, scipy
 - Image Processing : pillow
- JavaScript libraries
 - Web Scraping : puppeteer.js
 - Data Visualization : D3.js
 - Machine learning : ml.js
- R package
 - Data Visualization : ggplot2, plotly, esquisse
 - Machine Learning : mlr3
 - Data Management : dplyr
 - Mics : bioconductor, shiny
- Java
 - Hadoop
 - Apache Spark
- ⇒ Mobile development
 - Android
 - Plateforme : Android Studio
 - Langage : Java, Kotlin
 - iOS
 - Plateforme: Xcode
 - Language : Swift, Objective-C
 - Cross-platform
 - Flutter & Dart
 - JavaScript & React Native
 - Xamarin & C#
 - HTML5 & Apache Cordova
- ⇒ ERP VS CRM