

Rappel FRONT-END

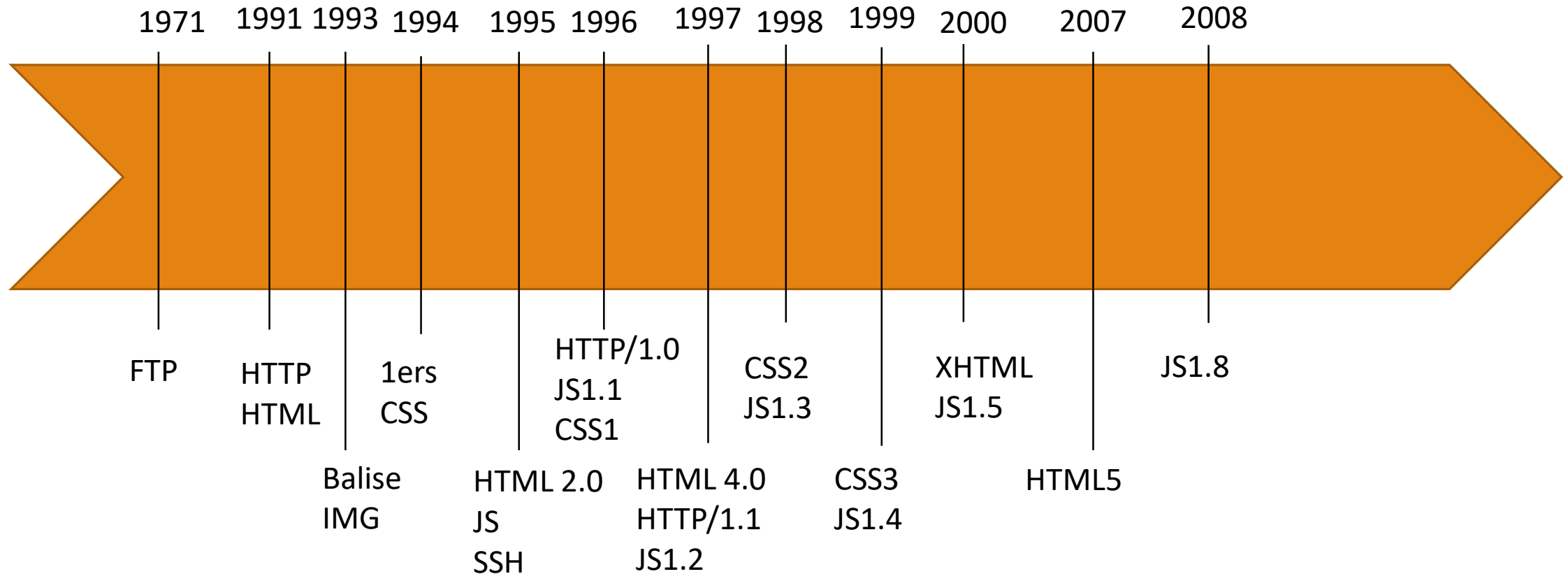
HTML – CSS – JAVASCRIPT – SUR BASE DU PACMAN

Objectifs du cours

Compréhension et maîtrise du web client

Réalisation d'un jeu PacMan

Historique du WEB



Architecture N-TIER

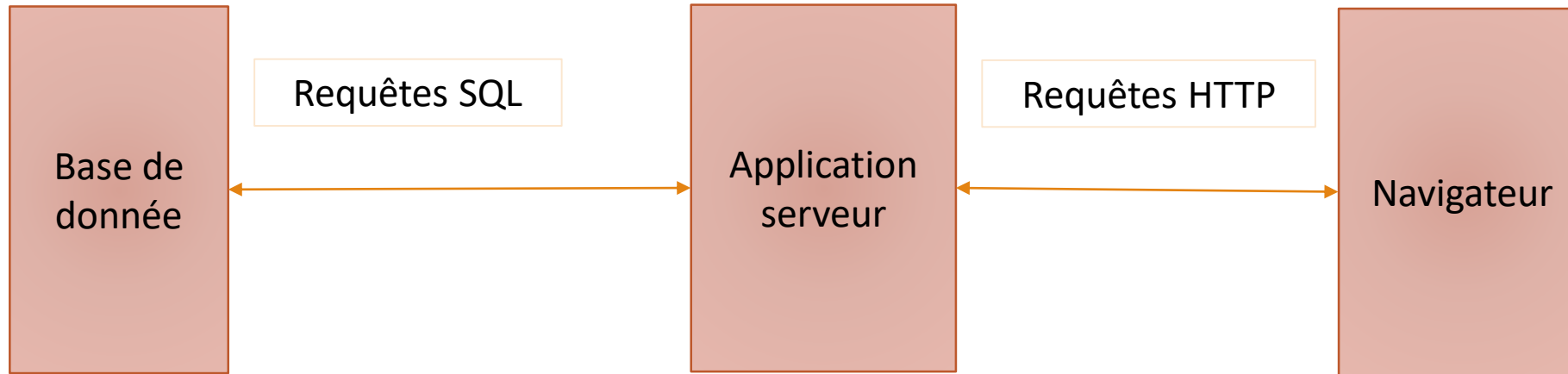


Chaque couche est indépendante

Chaque couche ne dialogue qu'avec celle du dessus ou du dessous

Intérêt : modularité, travail en équipe, stabilité, évolutivité

WEB Architecture 3-TIERs



Les applications Web sont donc par défaut 3-tiers

Les bases de données peuvent être de plusieurs types (Mysql, PostgreSQL, Oracle, Sql Serveur, MongoDB, MariaDB, etc...)

Les applications serveurs peuvent être dans plusieurs langages (PHP, Perl, Python, Java, .Net, Javascript, etc...)

Le Navigateur pour sa part interprète : Les langages à balises (HTML, XML), les feuilles de styles (CSS), Le code Javascript

Le métier de développeur

Compétences / caractère

Curiosité, Evolutivité, Logique, Imagination, Réflexion, Travail, Autonomie

Méthodes de travail

Agile, Kaban, Roadmap, principe de boy scout, notion de rapidité/qualité/prix

HTML éditeurs / concept

Il existe de nombreux éditeurs HTML en mode texte :

Notepad++, visual studio code, eclipse, netbean, etc...

N'utilisez pas d'éditeurs visuels, ils sont fait pour les non connaisseur et le code n'est pas propre !

Il y a trois grandes parties en front end, il est essentiel de bien les séparer et de gérer une architecture de site (dossiers séparés) :

- contenu / structure : HTML
- présentation : CSS
- logique / programmation : JS

Structure de base HTML

HTML un langage à balise :

Balise : `<section>`

Attribut : `<section num="3">`

Élément : `<section>voici un exemple de section</section>`

HTML structure de base

```
<!doctype html>  
<html lang="fr">  
  <head>  
    ...  
  </head>  
  <body>  
    ...  
  </body>  
</html>
```

HTML encodages et caractères spéciaux

Les types d'encodages :

UTF8, ISO8859-1, etc...

Les caractères spéciaux en HTML :

structure : &...;

Ex : & < > " etc....

Gestion du Dom

Récupération des éléments au travers de CSS/JS :

| type | HTML | CSS | Javascript | JQuery |
|--------------------|---|-----------------------|---|---|
| Balise standard | <code><div></div></code> | <code>div {}</code> | <code>Let tabElem = document.getElementsByTagName('div');</code> | <code>Let objTabElem = \$('div')</code> |
| Balise avec classe | <code><div class="gras"></div></code> | <code>.gras {}</code> | <code>Let tabElem = document.getElementsByClassName('gras');</code> | <code>Let objTabElem = \$('.gras')</code> |
| Balise avec id | <code><div id="grille"></div></code> | <code>#grille</code> | <code>Let monElem = document.getElementById('grille');</code> | <code>Let objTabElem = \$('#grille')</code> |

En Javascript natif, la fonction `document.querySelector('.gras')` récupère le premier élément et non un tableau correspondant au sélecteur CSS fournis en paramètre.

Objet document – Modifier un élément

| Méthode | Description |
|--|-----------------------------------|
| <i>element.innerHTML</i> = "nouveau contenu" | Modifier le contenu d'un élément |
| <i>element.attribut</i> = « nouvelle valeur » | Modifier la valeur d'un attribut |
| <i>element.setAttribute(attribut, valeur)</i> | Modifier la valeur d'un attribut |
| <i>element.style.propriete</i> = « nouveau style » | Modifier le style d'une balise |
| <i>element.onclick</i> = function(){code} | Ajouter un évènement à un élément |

Objet document–Ajout/Suppression d'éléments

| Méthode | Description |
|--|---|
| <code>document.createElement(« <i>NomBalise</i> »)</code> | Créer un nouvel élément HTML |
| <code>document.removeChild(<i>element</i>)</code> | Supprimer un élément HTML |
| <code>document.appendChild(<i>element</i>)</code> | Ajouter un élément HTML |
| <code>document.replaceChild(NouveauElement,AncienElement)</code> | Remplace l'ancien élément par le nouveau |
| <code>document.write(« <i>contenu</i> »)</code> | Ecrit un contenu directement à la fin de la page HTML |