

# Programmation

JavaScript : principes de base

JS

# Plan

- Introduction :
  - JavaScript
  - Algorithme vs programme
  - Notion de fonction
- Mise en place et exécution du code JavaScript
- Environnement de développement
  - Écriture du code
  - Débogage avec l'outil "console" du navigateur
- Fonctions JavaScript de base pour l'affichage et la lecture au clavier

# Plan

- **Introduction :**
  - JavaScript
  - Algorithme vs programme
  - Notion de fonction
- Mise en place et exécution du code JavaScript
- Environnement de développement
  - Écriture du code
  - Débogage avec l'outil "console" du navigateur
- Fonctions JavaScript de base pour l'affichage et la lecture au clavier

# JavaScript, un langage de programmation pour le web

- JavaScript permet notamment de :
  - manipuler par programmation les codes HTML et CSS d'une page
  - gérer des événements déclenchés par l'utilisateur sur la page (ex : click de souris, appuis sur les touches du claviers, etc...)
  - lire des données en ligne ou dans un fichier, les manipuler, les afficher...
- JavaScript fournit de nombreuses bibliothèques logicielles ("ajouts" par rapport au JavaScript de base), par exemple :
  - [jQuery](#) pour faciliter la manipulation du HTML/CSS et la gestion des événements
  - [P5](#) (code créatif), [Leaflet](#) (cartographie), [D3](#) & [Plotly](#) (visualisation de données)

# Algorithme vs programme

- La **description séquentielle** des traitements effectués par un programme s'appelle un **algorithme** (souvent écrit en pseudo-code)
- L'écriture dans un **langage de programmation** d'un algorithme, enregistré dans un **fichier**, est un **programme informatique** (directement exécutable dans le navigateur dans le cas de JavaScript)
- Les **instructions** sont les actions ou traitements effectués par un programme et sont exécutées **l'une après l'autre dans l'ordre d'écriture du code** (i.e. de haut en bas)
- Les données manipulées par le programme vont être mémorisées dans des **variables**

# Algorithme vs programme

- Exemple d'algorithme

*Variables*

quantite, prixUnitaire, prixTotal : nombre

*Début*

afficher("Entrer le prix unitaire du produit :")

lire(prixUnitaire)

afficher("Entrer la quantité de produits :")

lire(quantite)

prixTotal = prixUnitaire \* quantite

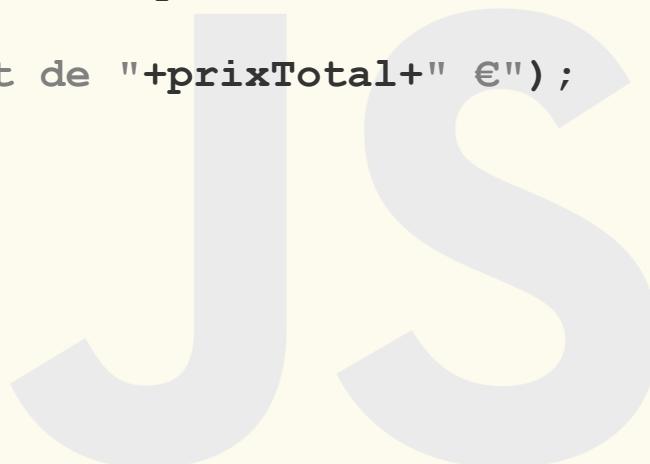
afficher("Le prix total est de ", prixTotal, " €")

*Fin*

# Algorithme vs programme

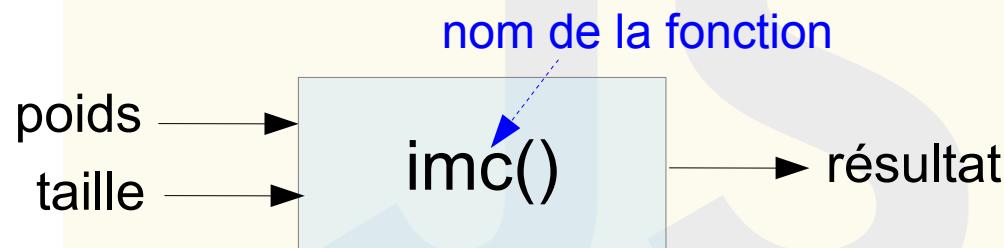
- Exemple de programme JavaScript

```
var prixUnitaire = prompt("Entrer le prix unitaire du produit");  
  
var quantite = prompt("Entrer la quantité de produit");  
  
var prixTotal = prixUnitaire*quantite;  
  
alert("Le prix total est de "+prixTotal+" €");
```



# Notion de fonction

- Une fonction est un **sous-algorithme** qui contient une **série d'instructions** exécutées lorsqu'on appelle la fonction
  - On appelle la fonction grâce à son **nom** suivi de **parenthèses**
  - Elle peut *éventuellement* utiliser des données qu'on lui passe entre les parenthèses (**entrée** de la fonction)
  - Elle peut *éventuellement* retourner un résultat (**sortie** de la fonction) qui devra être récupéré dans une **variable**



- Appel en JavaScript et récupération du retour dans une variable `monImc1` :

```
monImc1 = imc(55, 1.69);
```

# Notion de fonction

- Il existe de nombreuses fonctions **prédéfinies** qu'on utilise telle quelle
  - dans JavaScript (affichage, saisie clavier,...cf ci-après)
  - dans des bibliothèques logicielles JavaScript (graphisme, cartographie, visualisation de données, etc.)
- Il est possible de définir ses propres fonctions
  - Cf suite du cours "Algorithmique et JavaScript"

# Plan

- Introduction :
  - JavaScript
  - Algorithme vs programme
  - Notion de fonction
- **Mise en place et exécution du code JavaScript**
- Environnement de développement
  - Écriture du code
  - Débogage avec l'outil "console" du navigateur
- Fonctions JavaScript de base pour l'affichage et la lecture au clavier

# Mise en place et exécution du code JavaScript

- Le JavaScript est exécuté **dans le navigateur** et lancé par **une page web** au moment du chargement de celle-ci
- Pour faire exécuter du code JavaScript depuis une page web, il faut placer des balises HTML **<script>** et **</script>** dans cette page
- Où placer le code JavaScript ?
  - 2 solutions :
    - 1) Placer le code JavaScript directement entre les balises **<script>** et **</script>**
    - 2) Placer le code JavaScript dans un **fichier indépendant** (d'extension **.js**) auquel on fait référence dans la balise **<script>** grâce à l'attribut « **src** » :
      - Ex : **<script src="js/monCode.js"></script>**

# Mise en place et exécution du code JavaScript

- Où placer les balises `<script>` et `</script>` ?
  - Elles peuvent a priori être placées n'importe où dans le HTML (avec quelques différences de comportement dues au chargement et à l'exécution du code)
    - Dans l'élément `<head>`
    - Dans l'élément `<body>`
  - On adoptera les pratiques suivantes :
    - placer les balises `<script>` et `</script>` à la fin du `<body>` (juste avant `</body>`)
      - Cela permet de charger la page HTML/CSS avant d'exécuter le JavaScript.

# Mise en place et exécution du code JavaScript

- Comment charger des bibliothèques/modules externes ?
  - Il est souvent utile voire nécessaire d'utiliser des **bibliothèques externes** (ex : jQuery)
  - Ces bibliothèques externes sont elles aussi chargées chacune par une balise `<script>`, à placer avant celle qui appelle notre propre programme
    - 2 solutions :

1) **Télécharger le fichier de la bibliothèque** depuis le site officiel, le placer dans le répertoire de notre projet et utiliser un lien local :

```
<script src="js/jquery.min.js"></script>
```

2) **Utiliser un lien en ligne** vers la bibliothèque

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
```

# Plan

- Introduction :
  - JavaScript
  - Algorithme vs programme
  - Notion de fonction
- Mise en place et exécution du code JavaScript
- **Environnement de développement**
  - Écriture du code
  - Débogage avec l'outil "console" du navigateur
- Fonctions JavaScript de base pour l'affichage et la lecture au clavier

# Environnement de développement

- Écriture du code :
  - Dans un éditeur de texte avec coloration (ex : Notepad++)
  - Dans un logiciel de développement type IDE (ex : Brackets, SublimeText,...)
  - En respectant la **syntaxe du langage**
    - Ex : chaque ligne d'instruction en JavaScript doit se terminer par un **point virgule " ; "**
    - La coloration vous aide à repérer les erreurs de syntaxe !
- Débogage : affichage d'erreurs et de messages de contrôle dans le programme
  - Dans l'outil **console** du navigateur (**à utiliser absolument!!!**)
    - Chercher la console dans les Outils de développement
  - Dans la **console** de l'IDE

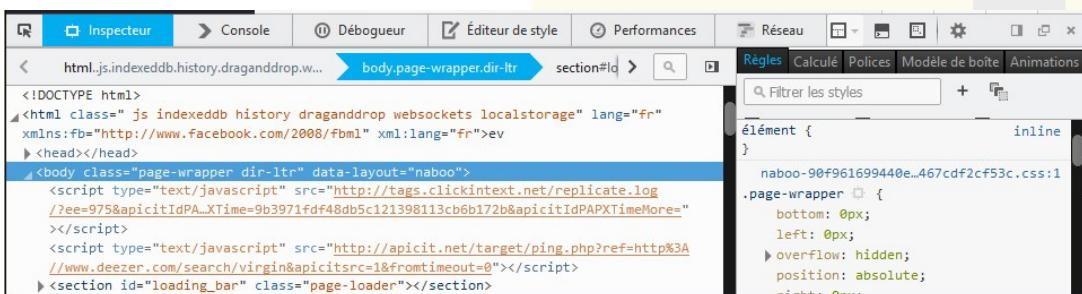
# Environnement de développement

- Les outils de développement web du navigateur
  - Permettent de voir et éventuellement modifier :
    - La structure d'une page HTML
    - Les feuilles CSS
    - Les sorties console du JavaScript
    - Les requêtes/réponses réseaux
    - Les performances

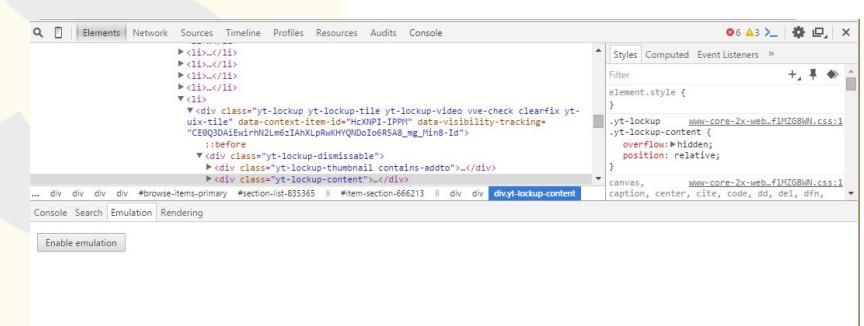
# Environnement de développement

- Accès simple dans de nombreux navigateurs
  - Menu « Développement web » ou « Plus d'outils »/« Outils de développement »
  - Ou, click droit sur la page > « Examiner/Inspecter l'élément »
  - Ou selon le navigateur, raccourci clavier : F12, Ctrl+Maj+I, Ctrl+Maj+C,...

Mozilla Firefox

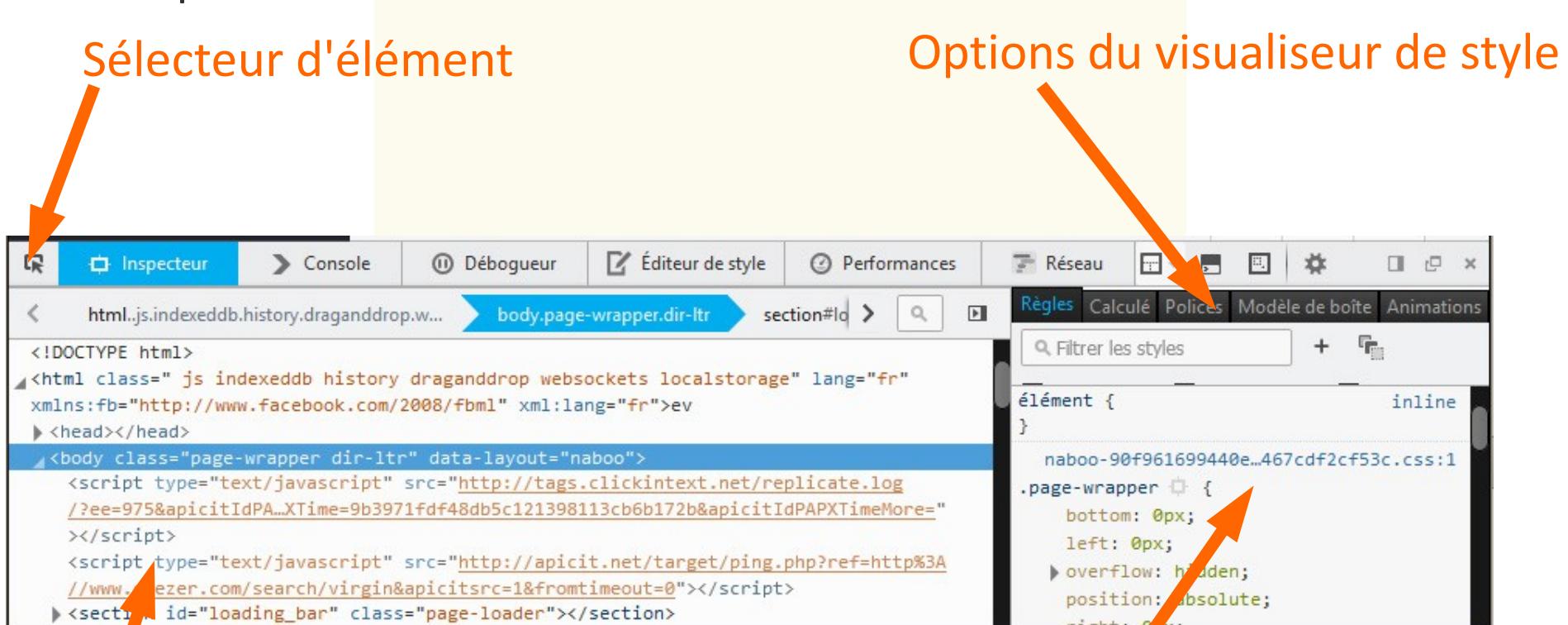


Google Chrome



# Environnement de développement

- Inspecteur d'élément HTML CSS



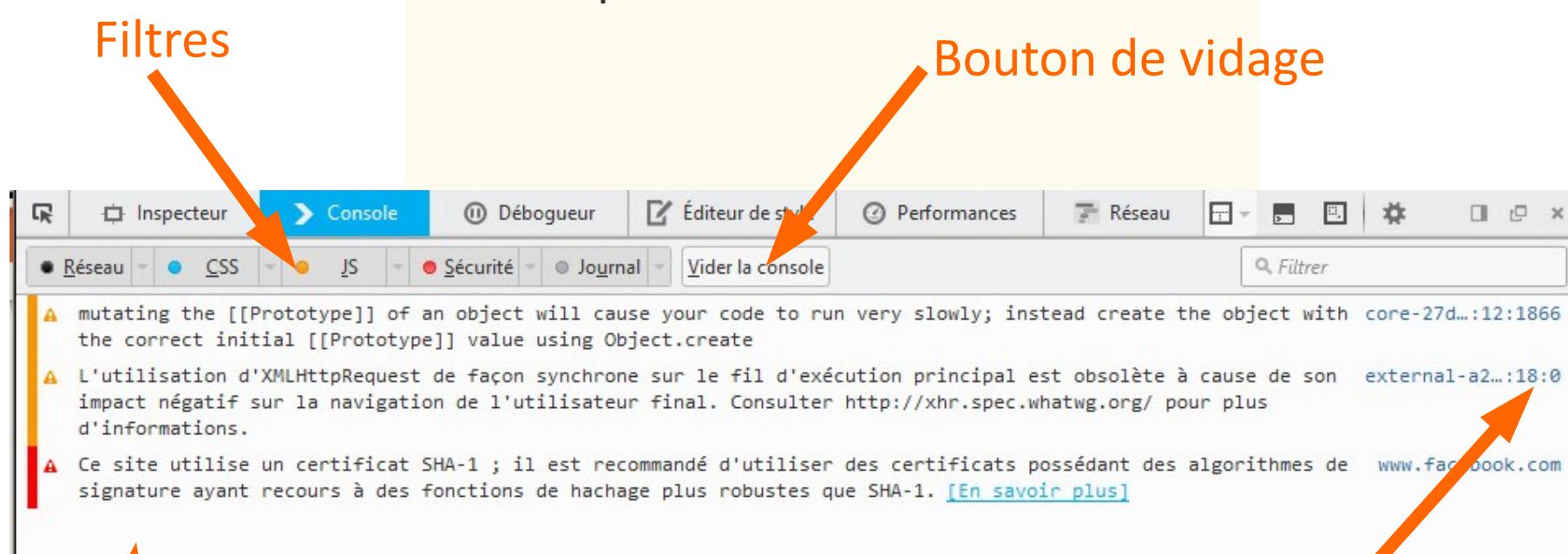
Page HTML en mémoire

Options du visualiseur de style

Liste des règles CSS appliquées à l'élément sélectionné

# Environnement de développement

- Console JavaScript



Intitulé de l'erreur

Emplacement de l'erreur sous la forme fichier:ligne:colonne

# Plan

- Introduction :
  - JavaScript
  - Algorithme vs programme
  - Notion de fonction
- Mise en place et exécution du code JavaScript
- Environnement de développement
  - Écriture du code
  - Débogage avec l'outil "console" du navigateur
- **Fonctions JavaScript de base pour l'affichage et la lecture au clavier**

# Fonctions JavaScript de base pour l'affichage et la lecture au clavier

- Affichage de type "popup"

```
alert("Hello world! I am an alert box!!");
```

- Affiche à l'utilisateur une popup avec un message et un bouton OK

- Saisie au clavier

```
var prenom; // déclaration d'une variable prenom
```

```
prenom = prompt("Entrer votre prénom :");
```

- Affiche un message avec une zone de saisie de texte et un bouton OK
- Renvoie la valeur saisie par l'utilisateur (qui peut être récupérée dans une variable, ici **prenom**) **sous forme d'une chaîne de caractères** (qu'il faudra parfois convertir en nombre pour pouvoir effectuer des calculs avec...)

- Affichage dans la console (pour débogage par exemple)

```
console.log("Valeur de la variable prenom :" + prenom);
```

- Affiche un message dans la console du navigateur
- Permet d'afficher des valeurs de variables