JavaScript - Les fondamentaux

Pourquoi apprendre ce langage?

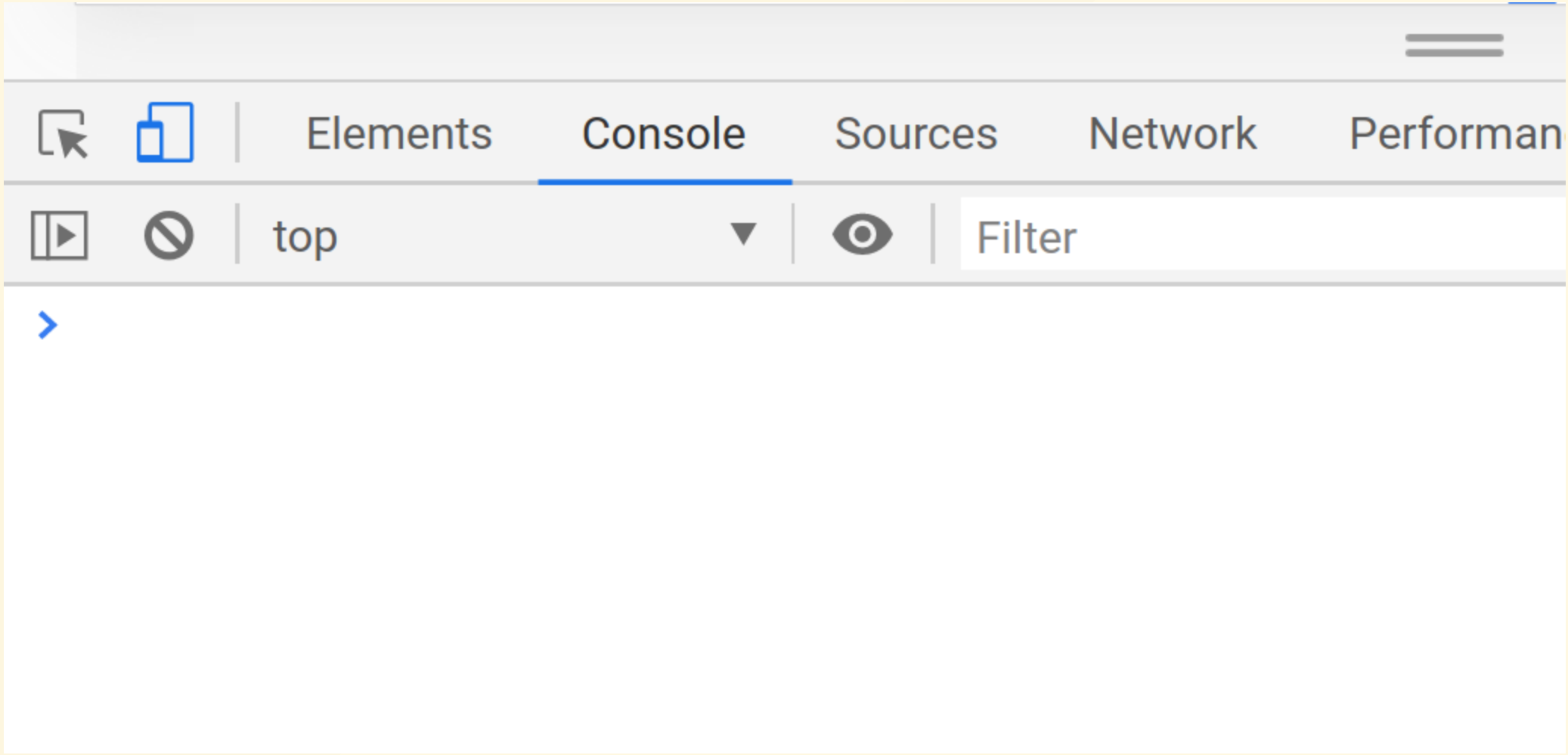
Côté client, l'exécuter sur un navigateur pour créer des:

- Interactions
- Animations
- Applications complètes (ex: gmail)
  Côté serveur web, on peut remplacer PHP par du JavaScript grâce à node.js

Attention! JavaScript et Java sont deux langages différents !

Où tester du Javascript?

Dans la console du navigateur



Les types de valeurs en JavaScript

Une donnée en JavaScript s'appelle **une valeur**.
Dans le langage JS, y en a 6:

- `Number`
- `String`
- `Boolean`
- `Object`
- Les spéciaux: `Null` et `Undefined`

En savoir plus: https://blog.lesieur.name/les-types-en-javascript-pour-tout-savoir/

Connaître le type d'une valeur

Grâce à l'opérateur `typeof`

```
typeof 3; //number
typeof "coucou"; //string
typeof false; //boolean

let color= "yellow";
typeof color; //string
```

Les variables

Pour la déclaration, on utilise le mot-clé `let`
Garde temporairement en mémoire une valeur durant l'exécution d'un script
Doit obligatoirement commencer par une lettre
Doit contenir seulement des lettres, chiffres ou `_` ou `$`. Aucun autre caractère n'est autorisé
Sensible à la casse (`let demo` différent de `let Demo`)

```
let age;
```

Où coder du JavaScript?

Dans un fichier `.js` relié à une page `.html`

```
  <body>
    ...
    <script src="app.js"></script>
  </body>
</html>
```

Fichier `app.js` :

```
console.log("Hello World");
```

Actions des messages du navigateur

- `console.log()` : affiche quelque chose dans la console du navigateur
- `prompt()` : affiche une fenêtre modale demandant l'entrée d'une donnée par l'utilisateur
- `alert()` : affiche une fenêtre modale avec un message
- `confirm()` : affiche une fenêtre modale avec un bouton valider et un bouton annuler

Les nombres (Number)

Toutes les valeurs numériques

Ca peut être des Positifs/négatifs, des entiers, des nombres à virgule

```
x=2;
y=-8;
z=3.5;
```

On peut faire de l'arithmétique avec. Les mêmes priorités de calcul que les maths sont employées

```
3 + 2 * 4; //11
(3 + 2) * 4; //20
```

Opérations sur les nombres

```
x=2+2; //2+2=4
y=3-1; // 3-1=2

mult=5*3;
div=12/2;

mod=13%3; //1
```

Modulo: Reste de la division d'un nombre par un autre

Opérations rapides

```
let x=0;
x+=2; //2
x++; //3

console.log(x); //3
console.log(x--); // Print 3 then x=2
console.log(--x); // x=1 then print 1
```

Les chaînes de caractère (ou String)

Représente du texte
Peut utiliser des guillemets (“”) ou des apostrophes (‘’)

```
welcome="Salut la compagnie!";
title='Bienvenue chez les ch\'tis';
```

Additionner des chaines de caractère => Concaténation

```
complete = welcome + " " + title;
// Salut la compagnie! Bienvenue chez les ch'tis

complete = `${welcome} ${title}`; //ES6 literals
// Salut la compagnie! Bienvenue chez les ch'tis
```

Les booléens

Opérations où seuls deux états sont possibles:

- `0` ou `1`
- `On` ou `Off`
- `Bon` ou `Pas bon`
- `vrai` ou `faux`

```
isBeautiful=true;

isYoung=false;
```

Les types spéciaux

- `Undefined` : une variable déclarée mais dont la valeur n’a pas été initialisée
- `Null` : Assigner à la variable qu’elle n’a pas de valeur actuellement

Les opérateurs

Retourne un booléen (`true` ou `false`)

Les opérateurs de comparaison

Symbole	Signification
==	Est égal à (valeur)
!=	Est différent de (valeur)
===	Est égal à (valeur + type)
!==	Est différent de (en valeur + type)

Les opérateurs de comparaison (suite)

Symbole	Signification
<	Est strictement inférieur à
<=	Est inférieur ou égal à
>	Est strictement supérieur à
>=	Est supérieur ou égal à

Les opérateurs de logique

Opérateur logique	Symbole
ET (AND)	`&&`
OU (OR)	`||`
NON (NOT)	`!`

Les conditions

Si

```
SI(condition) {
    // Programme à exécuter
}
```

```
let age=12;
if(age>7) {
    console.log("Plus de dents de lait!");
}
```

Si...Sinon

```
SI(condition) {
    // Programme à exécuter
}
SINON {
    // Autre programme à exécuter
}
```

```
let age=12;

if(age>7) {
    console.log("Plus de dents de lait!");
}
else {
     console.log("La petite souris doit encore passer!");
}
```

Si...Sinon Si...Sinon

```
let age=12;

if(age<18) {
    console.log("Je grandis!");
}
else if(age<25) {
    console.log("Je m'allonge moins qu'avant");
}
else {
    console.log("Je vais rapetisser un jour...");
}
```

Switch

```
let nbApples=1;

switch(nbApples){
    case 0:
        alert("Plus de pommes");
        break;
    case 1:
        alert("Profite de ta dernière pomme");
        break;
    case 2:
        alert("Deux valent mieux qu'une");
        break;
    default:
        alert("De quoi avoir de belles dents!");
}
```

Les boucles

for

Répète une série d'instructions.3 paramètres:

- **Le point de départ** de la boucle
- **Une condition** -> Tant qu'elle est vraie, ça continue de boucler
- **Le pas** (0.1.2 , 0.2.4, 4.3.2, 4.2.0, etc...)

```
POUR(
    valeur de départ;
    tant que condition vraie repeter boucle;
    action a faire sur valeur avant répétition boucle
)
{
    //Programme à répéter
}
```

```
for(let i=0;i<10;i++)
{
    console.log(i); //0..9
}
```

while

Pour les boucles dont la condition d'arrêt n'est pas facilement quantifiable

```
TANT QUE(condition)
{
    //Programme à répéter
}
```

```
let msg="beh";
while(msg != "behhhhhhhh")
{
    msg = msg+"h";
}
```

do while

Exécute au moins une fois la boucle, même si la condition n'est pas vraie

```
FAIRE AU MOINS UNE FOIS
{
    //Programme à répéter
}
TANT QUE(condition)
```

```
let age=12;
do {
    console.log(age);
} while(age<10);
```

Les tableaux

```
let students = ["Pierre","Paul","Jacques"];
console.log(students[0]); //Pierre
console.log(students[1]); //Paul
console.log(students[2]); //Jacques
```

`<name_array>.length` : Donne le nombre d'éléments du tableau

```
// Parcours du tableau
for(let i=0;i<students.length;i++)
{
    console.log(students[i]);
}
```