

JavaScri

Table of Contents

| | |
|---|---|
| Objectifs | 2 |
| Un langage pour les navigateurs | 2 |
| PlantUML C'est quoi ? | 2 |
| A quoi sert un contenu interactif ? | 2 |
| Et quoi d'autre encore ? | 3 |
| Le langage JavaScript | 3 |
| Les variables | 3 |
| Syntaxe | 3 |
| Démonstration | 4 |
| Les tableaux | 4 |
| Syntaxe | 4 |
| Démonstration | 4 |
| Les opérateurs de comparaison | 5 |
| Syntaxe | 5 |
| Démonstration | 5 |
| Les opérateurs logiques | 5 |
| Syntaxe | 5 |
| Les conditionnelles | 6 |
| Syntaxe | 6 |
| Démonstration | 6 |
| Les boucles | 6 |
| Syntaxe de la boucle for | 6 |
| Syntaxe de la boucle do while | 7 |
| Syntaxe de la boucle while | 7 |
| Démonstration | 7 |
| Les fonctions | 7 |
| Syntaxe | 8 |
| Démonstration | 8 |
| Les principaux objets disponibles | 8 |
| Démonstration | 9 |
| Conclusion | 9 |
| Objectif / Niveau | 9 |
| A toi de jouer | 9 |

Module 01 - Les fondamentaux

Discours formateur

Bienvenue dans ce module traitant des fondamentaux du JavaScript

Objectifs

- Connaître l'utilité du langage JavaScript
- Connaître les fondamentaux du langage JavaScript

Un langage pour les navigateurs

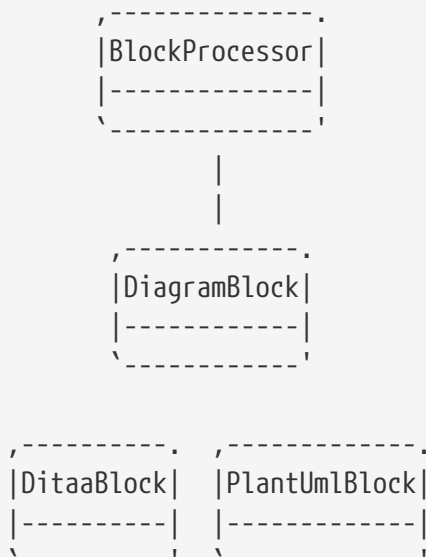
Les navigateurs savent interpréter trois langages :

- Le **HTML** : permet de structurer le contenu
- Le **CSS** : permet de mettre en forme le contenu
- Le **JavaScript** : permet de rendre le contenu **interactif**

NOTE

Le PHP est interprété sur le serveur web, les navigateurs sont incapables d'interpréter le PHP.

PlantUML C'est quoi ?



A quoi sert un contenu interactif ?

- A faire les slides que vous lisez
- A récupérer le flux webcam où vous souriez

- A envoyer & recevoir les messages du chat que vous écrivez
- A faire tout le cockpit dans lequel vous êtes installés !

Et quoi d'autre encore ?

- Faire une action sans recharger la page
 - supprimer un produit du panier
- Valider le contenu d'un formulaire
 - afficher erreur ou succès au moment où l'on écrit
- Animer les éléments de la page
 - faire un carrousel d'images

Bref, JavaScript dynamise les pages web !

Le langage JavaScript

- JavaScript est un langage qui peut être employé en tant que langage procédural ou orienté objet.

Discours formateur

Indiquer que nous verrons le paradigme du développement Orienté Objet en S04. Le procédural c'est comme une recette de cuisine, on fait les actions les unes après les autres comme c'est écrit dans la recette.

Les variables

Une variable permet de stocker une valeur.

Discours formateur

Exactement comme en PHP vu en S02.

Syntaxe

```
let unEntier = 10; //number
let uneString = "I'm happy d'être avec vous today"; //string
let unBoulean = true; // false
let uneVariable; //undefined
let unTrucNull = null; //Null
```

NOTE

Deux caractéristiques du langage JavaScript :

- Typage dynamique = c'est la valeur d'affectation qui définit le type de la variable.
- Non fortement typé = Une variable peut changée de type lors de l'exécution du programme.

Démonstration

Les variables

Discours formateur

Faire la démonstration en live.

Les tableaux

Un tableau permet de stocker un ensemble de valeurs.

Discours formateur

Comme en PHP vu lors de la saison 02.

Syntaxe

```
let tableau = ['0clock', 'Rennes', 14540, 'Truc', 'Muche']  
tableau[6] = 'Wololo'  
console.log(tableau[0])
```

Démonstration

Les tableaux

Discours formateur

Faire la démonstration en live.

Les opérateurs de comparaison

- Comme en PHP

Syntaxe

| Opérateur | Description |
|-------------------------|--|
| Egalité (==) | Renvoie true si les opérandes sont égaux. |
| Inégalité (!=) | Renvoie true si les opérandes sont différents. |
| Égalité stricte (===) | Renvoie true si les opérandes sont égaux et du même type. |
| Inégalité stricte (!==) | Renvoie true si les opérandes sont du même type et différents ou s'ils ne sont pas du même type. |
| Supériorité stricte (>) | Renvoie true si l'opérande gauche est strictement supérieur à l'opérande droit. |
| Supériorité (>=) | Renvoie true si l'opérande gauche est supérieur ou égal à l'opérande droit. |
| Infériorité stricte (<) | Renvoie true si l'opérande gauche est strictement inférieur à l'opérande droit. |
| Infériorité (<=) | Renvoie true si l'opérande gauche est inférieur ou égal à l'opérande droit. |

Démonstration

Les opérateurs de comparaison

Discours formateur

Faire la démonstration en live.

Les opérateurs logiques

- Comme en PHP

Syntaxe

| Opérateur | Utilisation | Description |
|-----------------|--------------------------|---|
| ET logique (&&) | condition1 && condition2 | Renvoie true si les deux conditions sont égales à true. |

| Opérateur | Utilisation | Description |
|------------------|---------------------------|--|
| OU logique () | condition1 condition2 | Renvoie true si au moins une des deux conditions est égale à true. |
| NON Logique | !condition | inverse la valeur de la condition |

Les conditionnelles

Instructions permettant de tester si une condition est vraie ou fausse.

Syntaxe

```
if (condition) {
    //liste d'instructions
}
else {
    //autre série d'instructions
}

(condition) ? instruction si vrai : instruction si faux
```

Démonstration

Les conditionnelles

Discours formateur

Faire la démonstration en live.

Les boucles

Les boucles

- while
- do ... while
- for, for ... in, for ... of

Syntaxe de la boucle for

```
for (let pas = 0; pas < 5; pas++) {  
  // Ceci sera exécuté 5 fois  
  // À chaque exécution, la variable "pas" augmentera de 1  
  // Lorsque'elle sera arrivée à 5, le boucle se terminera.  
  console.log('Faire ' + pas + ' pas vers l\'est');  
}
```

Syntaxe de la boucle do while

Le bloc de code est exécuté au moins une fois.

```
let i = 0;  
do {  
  i += 1;  
  console.log(i);  
} while (i < 5);
```

Syntaxe de la boucle while

Le bloc de code est exécuté tant que la condition est vraie.

```
let n = 0;  
let x = 0;  
while (n < 3) {  
  n++;  
  x += n;  
}
```

Démonstration

Les boucles

Discours formateur

Faire la démonstration en live.

Les fonctions

Même fonctionnement qu'en PHP.

Syntaxe

```
function additionner(valeur1, valeur2) {  
  let resultat = valeur1 + valeur2  
  return resultat  
}
```

Démonstration

Les fonctions

Discours formateur

Faire la démonstration en live.

Les principaux objets disponibles

- String
 - `charAt()`, `concat()`, `indexOf()`, `substring()`...
- Number
 - `MAX_VALUE`, `MIN_VALUE`, `isInteger()`, `isNaN()`...
- Math
 - `random()`, `floor()`, `ceil()`...
- Date
 - `getDate()`, `getMonth()+1`, `getFullYear()`, `getHours()`... `setMinutes()`, `setSeconds()`...
`toLocaleDateString()`...
- Error
 - `name`, `message`
- Propriétés et variables globales
 - `Infinity`, `NaN`, `undefined`, `isNaN()`, `Number()`, `parseFloat()`, `parseInt()`...

Discours formateur

Ces objets et méthodes devront être connus, en pratiquant vous y arriverez.

Démonstration

Les principaux objets disponibles

Discours formateur

Faire la démonstration en live.

Conclusion

- Vous avez les connaissances pour écrire vos premiers programmes en JS.

Objectif / Niveau

- **Essentiel** : Savoir utiliser les variables, les conditionnelles et les boucles.
- **Attendu** : Savoir utiliser les variables, les tableaux, les conditionnelles et les boucles.
- **Avancé** : Savoir utiliser les variables, les tableaux, les conditionnelles, les boucles et tous les principaux objets disponibles.

A toi de jouer

Analyse d'un code JavaScript