Atelier 11 Javascript ½ :

Valentin Chaussegros.

Introduction :

Dans cette atelier nous allons apprendre à utiliser et comprendre les bases de Javascript à travers des exercices tout au long de l'atelier.

Sommaire :

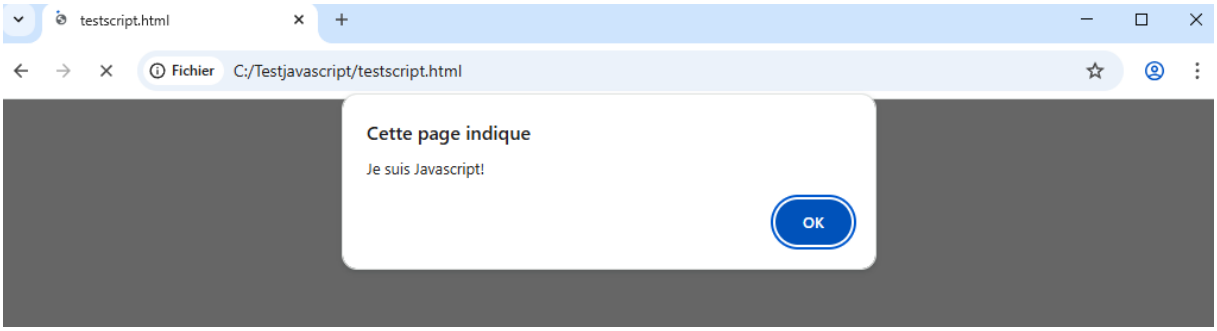
Fondamentaux 1 :

Le premier exercice nous demande afficher une alerte.

On souhaite une page qui s'affiche avec le message « Je suis Javascript ! » :

```
<!DOCTYPE HTML>
<html>

<body>

    <script>
        alert( 'Je suis Javascript!' );
    </script>

</body>
</html>
```

J'utilise donc une balise **script** pour utiliser javascript et avec « alert » j'affiche le message :



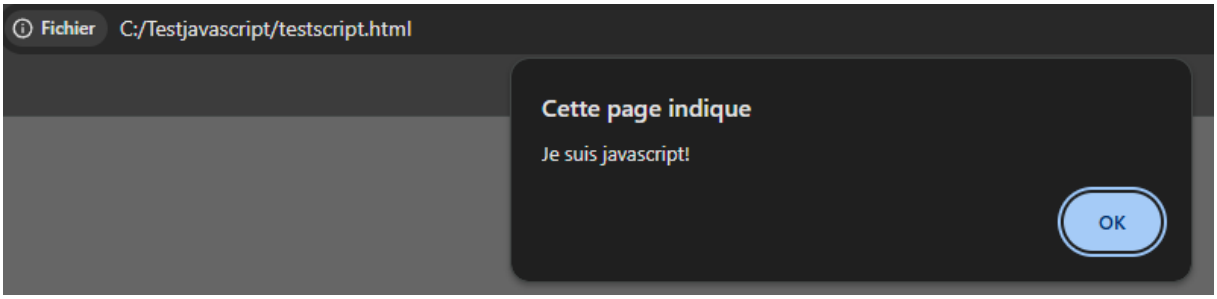
Dans la suite de l'exercice on doit afficher une alerte avec un script externe.

On va donc créer un fichier externe alert.js qui contient le message :

```
C: > Testjavascript > JS alert.js
1    alert('Je suis javascript!');
2
```

On utilise ensuite « src » pour re diriger vers ce fichier externe :

```
1   <!DOCTYPE HTML>
2   <html>
3
4   <body>
5       <script src="alert.js">
6       </script>
7   </body>
8   </html>
```

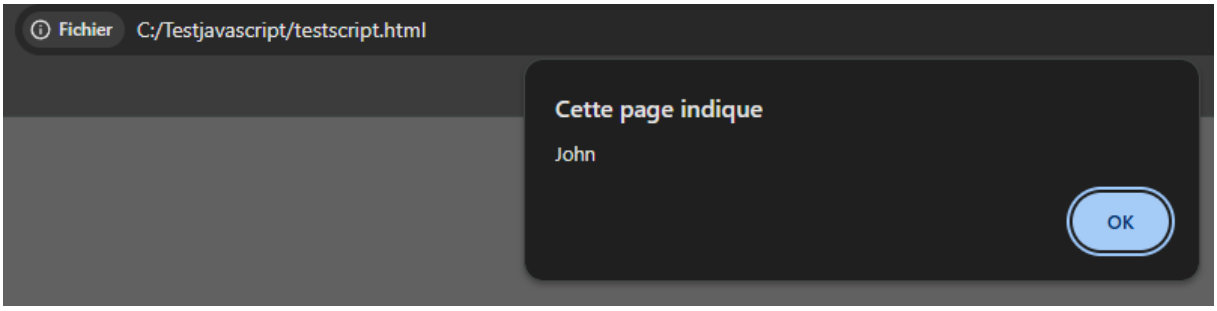


Fondamentaux 2 :

Dans cette parti on utilisera des variables.

On déclare donc nos variables et on affiche la valeur de la variable souhaiter avec « alert » :

```
<!DOCTYPE HTML>
<html>

<body>
    <script>
let admin = "John";
let name = "John";
</script>
<script>
    alert(admin);
</script>
</body>
</html>
```



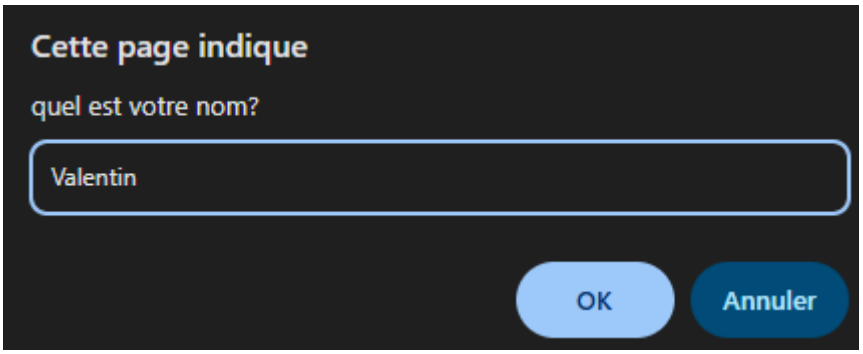
On nous demande quel est la sortie du script :

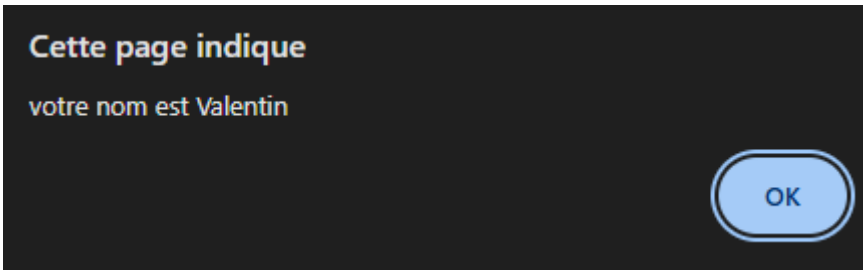
```
let name = "Ilya";

alert( `hello ${1}` ); // ?

alert( `hello ${"name"}` ); // ?

alert( `hello ${name}` ); // ?
```

Le script affiche les 3 alertes à la suite

On crée une page web qui demande un nom et l'affiche :

```html
<!DOCTYPE HTML>
<html>

<body>
    <script>
let name = prompt('quel est votre nom?', '' );
alert('votre nom est ' + name);
</script>
</body>
</html>
```





Pour commencer on nous demande quels sont les valeurs de toutes les variables :

```
let a = 1, b = 1;

let c = ++a; // ?
let d = b++; // ?
```

a = 1

b = 1

c = 2

d = 1

Les valeurs de a et x ? :

```
let a = 2;

let x = 1 + (a *= 2);
```

a = 4

x = 5

Résultats des expression suivantes :

```
"" + 1 + 0
"" - 1 + 0
true + false
6 / "3"
"2" * "3"
4 + 5 + "px"
"$" + 4 + 5
"4" - 2
"4px" - 2
"  -9  " + 5
"  -9  " - 5
null + 1
undefined + 1
" \t \n" - 2
```

1) = 10

2) = -1

3) = 1

4) = 2

5) = 6

6) = 9px

7) = $45

8) = 2

9) = NaN

10) = -9 5

11) = -14

12) = 1

13) = NaN

14) = NaN

On demande ensuite de corriger l'addition donné :

```
let a = prompt("First number?", 1);
let b = prompt("Second number?", 2);

alert(a + b); // 12
```

Je le corrige en rajoutant Number devant prompt pour précisé que c'est un calcule de nombre :

```
<body>
    <script>
let a = Number (prompt("First Number?", 1));
let b = Number (prompt("second number?", 2));

alert(a + b);
</script>
```

On cherche le résultat :

```
5 > 4
"apple" > "pineapple"
"2" > "12"
undefined == null
undefined === null
null == "\n0\n"
null === +"\n0\n"
```

1) = True

2) = false

3) = true

4) = true

5) = false

6) = false

7) = false

if (une chaîne de caractères avec zéro)

Est-ce que `alert` sera affiché ?

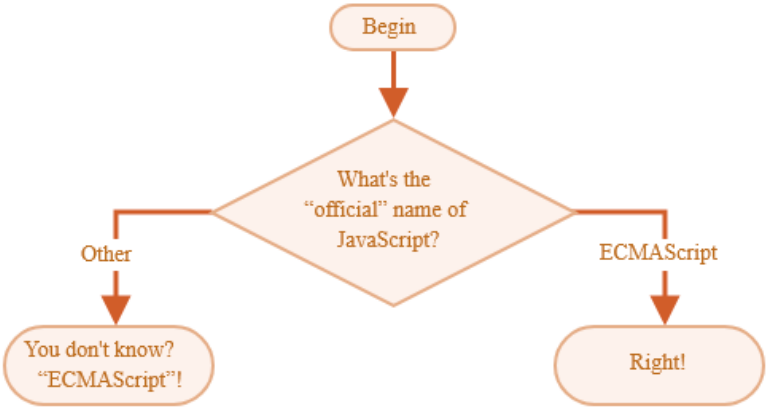
```
if ("0") {
  alert( 'Hello' );
}
```

Oui il sera affiché

Le nom de JavaScript

En utilisant la construction `if..else`, écrivez le code qui demande : 'Quel est le nom "officiel" de JavaScript?'

Si le visiteur entre "ECMAScript", alors éditez une sortie "Bonne réponse !", Sinon – retourne "Ne sait pas ? ECMAScript!"



On utilise simplement les fonctions if else :

```
    <script>
        'use strict';

let value = prompt('What the official name of javascript', '')

if (value =='ECMAScript') {
    alert('right!');
}
else {
    alert("You dont know? ECMAScript!");
}
</script>
```

Afficher le signe

En utilisant `if..else`, écrivez le code qui obtient un numéro via le `prompt`, puis l'affiche en `alert` :

- `1`, si la valeur est supérieure à zéro,
- `-1`, si inférieur à zéro,
- `0`, si est égal à zéro.

Dans cet exercice, nous supposons que l'entrée est toujours un nombre.

De même que la question précédente on utilise if ; else et else if

```
<body>
    <script>


let value = prompt('Entrer un nombre', 0);

if (value > 0) {
    alert(1);
} else if (value < 0) {
    alert(-1);
} else {
    alert(0);
}

</script>

</body>
```

Réécrire 'if' en '?'

Réécrivez ce `if` en utilisant l'opérateur conditionnel `'?'` :

```
let result;

if (a + b < 4) {
  result = 'Below';
} else {
  result = 'Over';
}
```

On récrit en utilisant « ? »

```
let result = (a + b < 4) ? 'below' : 'over';
```

Réécrire 'if..else' en '?'

Réécrire ce `if..else` en utilisant plusieurs opérateurs ternaires `'?'`.

Pour plus de lisibilité, il est recommandé de diviser le code en plusieurs lignes.

```
let message;

if (login == 'Employee') {
  message = 'Hello';
} else if (login == 'Director') {
  message = 'Greetings';
} else if (login == '') {
  message = 'No login';
} else {
  message = '';
}
```

Même chose avec ce programme on réécrit avec « ? »

```
let message = (login == 'Employee') ? 'Hello' :
   (login == 'Director') ? 'Greetings' :
   (login == '') ? 'No login' :
   '';
```

Quel est le résultat de OR ?

Qu'est-ce que le code ci-dessous va sortir ?

```
alert( null || 2 || undefined );
```

Le code affichera « 2 ».

Quel est le résultat des alertes OR ?

Qu'est-ce que le code ci-dessous va sortir ?

```
alert( alert(1) || 2 || alert(3) );
```

Le code affichera 1 puis ensuite 2.

Quel est le résultat de AND ?

Qu'est-ce que ce code va afficher ?

```
1 alert( 1 && null && 2 );
```

Le code affichera null.

Quel est le résultat des alertes AND ?

Qu'est-ce que ce code va afficher ?

```
1 alert( alert(1) && alert(2) );
```

Le code affichera 2.

Le résultat de OR AND OR

Quel sera le résultat ?

```
1 alert( null || 2 && 3 || 4 );
```

Le résultat sera « 3 ».

Vérifiez la plage entre

Ecrivez une condition `"if"` pour vérifier que l'`age` est compris entre `14` et `90` ans inclus.

"Inclus" signifie que l'`age` peut atteindre les `14` ou `90` ans.

```
if (age >= 14 && age <= 90);
```

Vérifiez à l'extérieur de la plage

Ecrivez une condition `if` pour vérifier que l'`age` n'est PAS compris entre `14` et `90` ans inclus.

Créez deux variantes: la première utilisant NOT `!`, La seconde – sans ce dernier.

Première variante :

```
if (!(age >= 14 && age <= 90));
```

Deuxième variante :

```
if (age < 14 || age > 90);
```

Une question à propos de "if"

Lesquelles de ces `alert`es vont s'exécuter ?

Quels seront les résultats des expressions à l'intérieur de `if (...)` ?

```
1 if (-1 || 0) alert( 'first' );
2 if (-1 && 0) alert( 'second' );
3 if (null || -1 && 1) alert( 'third' );
```

1. s'exécute et le résultat sera -1
2. ne s'exécute pas
3. s'exécute et le résultat sera 1

Check the login

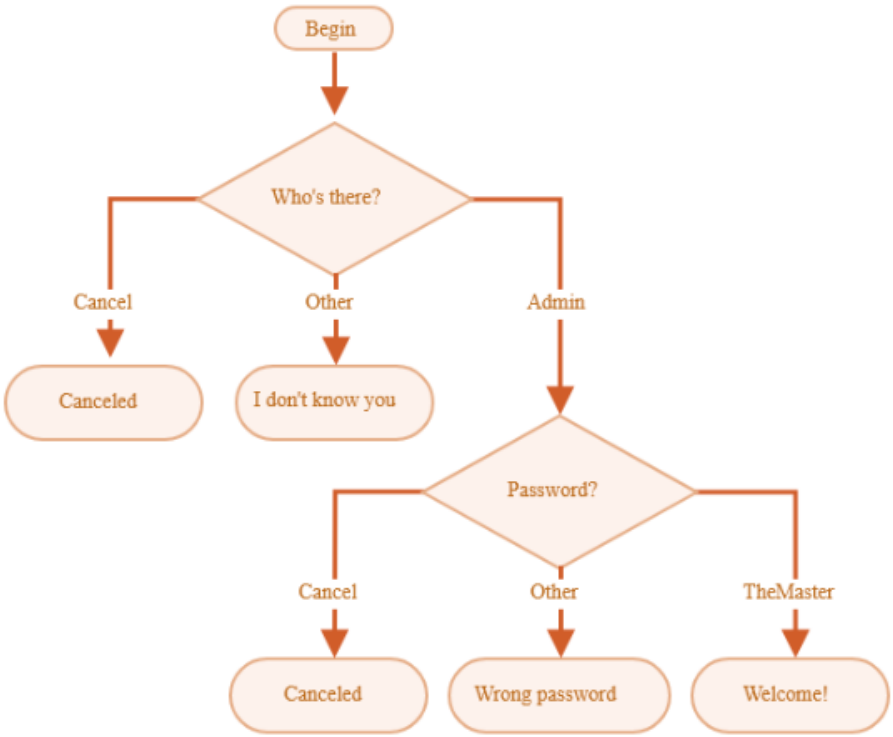
Écrivez le code qui demande une connexion avec `prompt`.

Si le visiteur entre `"Admin"`, puis `prompt` pour un mot de passe, si l'entrée est une ligne vide ou Esc – affichez "Canceled", s'il s'agit d'une autre chaîne de caractères – alors affichez "I don't know you".

Le mot de passe est vérifié comme suit :

- S'il est égal à "TheMaster", alors affichez "Welcome!",
- Une autre chaînede caractères – affichez "Wrong password",
- Pour une chaîne de caractères vide ou une entrée annulée, affichez "Canceled".

Le schéma :



Veuillez utiliser des blocs `if` imbriqués. Attention à la lisibilité globale du code.

Astuce: passer une entrée vide à un prompt renvoie une chaîne de caractères vide `''`. En pressant ESC lors d'un prompt cela retourne `null`.

Fondamentaux 3 :

Dernière valeur de boucle

Quelle est la dernière valeur affichée par ce code ? Pourquoi ?

```
let i = 3;

while (i) {
  alert( i-- );
}
```

La dernière valeur de la boucle sera 0 car à chaque fois « i » diminue de 1.

Quelles valeurs affiche la boucle while ?

A votre avis, quelles sont les valeurs affichées pour chaque boucle ? Notez-les puis comparer avec la réponse.

Les deux boucles affichent-elles les mêmes valeurs dans l'`alert` ou pas ?

1.

Le préfixe sous forme `++i` :

```
let i = 0;
while (++i < 5) alert( i );
```

2.

Le postfixe sous forme `i++` :

```
let i = 0;
while (i++ < 5) alert( i );
```

1. les valeurs de 1 à 4 s'affichent car la première valeur est i=1 en effet « ++i » renvoie la nouvelle valeur donc 1

2. les valeurs de 1 à 5 s'affichent car ici la première valeur est i=0 « i++ » renvoie l'ancienne valeur donc 0.

Quelles valeurs sont affichées par la boucle "for" ?

Pour chaque boucle, notez les valeurs qui vont s'afficher. Ensuite, comparez avec la réponse.

Les deux boucles `alert` les mêmes valeurs ou pas ?

1.

La forme postfix :

```
1  for (let i = 0; i < 5; i++) alert( i );
```

2.

La forme préfix :

```
1  for (let i = 0; i < 5; ++i) alert( i );
```

Les valeurs de 1 à 4 s'afficheront pour les 2 boucles.

Extraire les nombres pairs dans la boucle

Utilisez la boucle `for` pour afficher les nombres pairs de `2` à `10`.

On utilisera ici « % » pour vérifier avec le reste si le nombre est pair ou non :

```
<body>
    <script>
for (let i = 2; i <= 10; i++) {
    if (i % 2 == 0) {
        alert(i)
    }
}
```

Remplacer "for" par "while"

Réécrivez le code en modifiant la boucle `for` en `while` sans modifier son comportement (la sortie doit rester la même).

```
for (let i = 0; i < 3; i++) {
  alert( `number ${i}!` );
}
```

```
<body>
    <script>
let i = 0;
while (i < 3) {
    alert('number ${i}!');
    i++;
}

</script>
```

Répéter jusqu'à ce que l'entrée soit correcte

Ecrivez une boucle qui demande un nombre supérieur à `100`. Si le visiteur saisit un autre numéro, demandez-lui de le saisir à nouveau.

La boucle doit demander un numéro jusqu'à ce que le visiteur saisisse un nombre supérieur à `100` ou annule l'entrée/entre une ligne vide.

Ici, nous pouvons supposer que le visiteur ne saisit que des chiffres. Il n'est pas nécessaire de mettre en œuvre un traitement spécial pour une entrée non numérique dans cette tâche.

```
<body>
    <script>
let nombre;
do {
    nombre = prompt("Entrer un nombre plus grand que 100", 0);
} while (nombre <= 100 && nombre);
</script>
```

Ici on utilise do while pour répéter l'action jusqu'à obtenir la réponse souhaiter.

Extraire des nombres premiers

Un nombre entier supérieur à 1 est appelé un Nombre premier s'il ne peut être divisé sans reste par rien d'autre que 1 et lui-même.

En d'autres termes, `n > 1` est un nombre premier s'il ne peut être divisé de manière égale par autre chose que `1` et `n`.

Par exemple, `5` est un nombre premier, car il ne peut pas être divisé sans reste par `2`, `3` et `4`.

Écrivez un code qui produit les nombres premiers dans l'intervall e 2 à n.

Pour `n = 10`, le résultat sera `2`, `3`, `5`, `7`.

P.S. Le code devrait fonctionner pour n'importe quel `n` et aucune valeur fixe ne doit être codé en dur.

Réécrire le "switch" dans un "if"

Écrivez le code en utilisant `if..else` qui correspondrait au `switch` suivant :

```
switch (browser) {
  case 'Edge':
    alert( "You've got the Edge!" );
    break;

  case 'Chrome':
  case 'Firefox':
  case 'Safari':
  case 'Opera':
    alert( 'Okay we support these browsers too' );
    break;

  default:
    alert( 'We hope that this page looks ok!' );
}
```

```
<body>
    <script>
if(browser == 'Edge') {
    alert("You ve got the Edge!");
} else if (browser == 'Chrome' || browser == 'Firefox' || browser == 'Safari' || browser == 'Opera') {
    alert('Okay we support these browser too')
} else {
    alert('We hope this page look okay !')
}
```

On réécrit le code avec if else.

Réécrire le "if" dans un "switch"

Réécrivez le code ci-dessous en utilisant une seule instruction `switch` :

```
let a = +prompt('a?', '');

if (a == 0) {
  alert( 0 );
}
if (a == 1) {
  alert( 1 );
}

if (a == 2 || a == 3) {
  alert( '2,3' );
}
```

```
<body>
    <script>
let a = +prompt('a?', '')
switch(a) {
    case 0:
        alert(0);
        break;

    case 1:
        alert(1);
        break;

    case 2:
    case 3:
        alert('2,3');
}
```

Est-ce que "else" est requis ?

La fonction suivante renvoie `true` si le paramètre `age` est supérieur à `18`.

Sinon, il demande une confirmation et renvoie son résultat :

```
function checkAge(age) {
  if (age > 18) {
    return true;
  } else {
    // ...
    return confirm('Did parents allow you?');
  }
}
```

La fonction fonctionnera-t-elle différemment si `else` est supprimé ?

```
function checkAge(age) {
  if (age > 18) {
    return true;
  }
  // ...
  return confirm('Did parents allow you?');
}
```

Existe-t-il une différence dans le comportement de ces deux variantes ?

Il n'y a aucune différence dans les 2 cas la fonction s'exécute si « if » est faux.

Réécrivez la fonction en utilisant '?' ou '||'

La fonction suivante renvoie `true` si le paramètre `age` est supérieur à `18`.

Sinon, il demande une confirmation et renvoie le résultat.

```
function checkAge(age) {
  if (age > 18) {
    return true;
  } else {
    return confirm('Did parents allow you?');
  }
}
```

Réécrivez-le, pour effectuer la même chose, mais sans `if`, et en une seule ligne.

Faites deux variantes de `checkAge` :

1. En utilisant un opérateur point d'interrogation `?`
2. En utilisant OU `||`

1.

```
<body>
    <script>
function checkage(age) {
    return (age > 18) ? true : confirm('did parents allow you?');
}


</script>
```

2.

```
<body>
    <script>
function checkage(age) {
    return (age > 18) || confirm('did parents allow you?');
}


</script>
```

Fonction min(a, b)

Ecrivez une fonction `min(a, b)` qui renvoie le plus petit des deux nombres `a` et `b`.

Par exemple :

```
min(2, 5) == 2
min(3, -1) == -1
min(1, 1) == 1
```

```
<body>
    <script>
function min(a, b) {
    return a < b ? a : b;
}


</script>
```

Fonction pow(x,n)

Ecrivez une fonction `pow(x, n)` qui renvoie `x` à la puissance `n`. Ou, autrement dit, multiplie `x` par lui-même `n` fois et renvoie le résultat.

```
pow(3, 2) = 3 * 3 = 9
pow(3, 3) = 3 * 3 * 3 = 27
pow(1, 100) = 1 * 1 * ...* 1 = 1
```

Créez une page Web qui demande (`prompt`) `x` et `n`, puis affiche le résultat de `pow(x, n)`.

P.S. Dans cette tâche, la fonction ne doit prendre en charge que les valeurs naturelles de `n` : entiers supérieurs à `1`.

Réécrire avec les fonctions fléchées

Remplacez les expressions de fonction par des fonctions fléchées dans le code ci-dessous :

```
function ask(question, yes, no) {
  if (confirm(question)) yes();
  else no();
}

ask(
  "Do you agree?",
  function() { alert("You agreed."); },
  function() { alert("You canceled the execution."); }
);
```

On rajoute des fonctions fléchées à la place des expressions de fonction :

```
<body>
    <script>
function ask(question, yes, no) {
    if (confirm(question)) yes();
    else no();
}

ask(
    "Do you agree?",
    () => alert ("you agreed."),
    () => alert ("You canceled the execution")
);

</script>
```

Conclusion :

Dans ce TP j'ai appris à utiliser les bases fondamentales de javascript.