

TP - Javascript



PARTIE 1

Exercice 1

Afficher une alerte

Afficher une alerte avec un script externe

PARTIE 2

Exercice 1

Travailler avec des variables

Assigner le bon nom

Exercice 2

String Quotes

Exercice 3

Une simple page

Exercice 4

Les formes postfixes et préfixes

Résultat d'affectation

Les conversions de type

Comparaison

Exercice 5

if (une chaîne de caractères avec zéro)

Le nom de Javascript

Afficher le signe

Réécrire 'if' en '?

Réécrire 'if...else' en '?

Exercice 5

Quel est le résultat de OR ?

Quel est le résultat des alertes OR ?

Quel est le résultat de AND ?

Quel est le résultat des alertes AND ?

PARTIE 1

Exercice 1

Exercices

Afficher une alerte

Créez une page qui affiche un message "Je suis JavaScript!".

Assurez-vous simplement que cela fonctionne.

Afficher une alerte avec un script externe

Prendre la solution de l'exercice précédent Afficher une alerte. Modifiez-le en extrayant le contenu du script dans un fichier externe `alert.js`, résidant dans le même dossier.

Ouvrez la page, assurez-vous que l'alerte fonctionne.

Afficher une alerte

```
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
</head>
<body>
    <script>
        alert("Je suis JavaScript!");
    </script>
</body>
</html>
```

Cette page indique

Je suis JavaScript!

OK

Afficher une alerte avec un script externe

```
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
</head>
<body>
    <script src="alert.js"></script>
</body>
</html>
```

```
alert("Je suis JavaScript!");
```



Exercice 1

Exercices

Travailler avec des variables

1. Déclarez deux variables : `admin` and `name`.
2. Assignez la valeur `"John"` à `name`.
3. Copiez la valeur de `name` à `admin`.
4. Afficher la valeur de `admin` en utilisant `alert` (devrait afficher "John").

Assigner le bon nom

1. Créez la variable avec le nom de notre planète. Comment nommeriez-vous une telle variable ?
2. Créez la variable pour stocker le nom du visiteur actuel. Comment nommeriez-vous cette variable ?

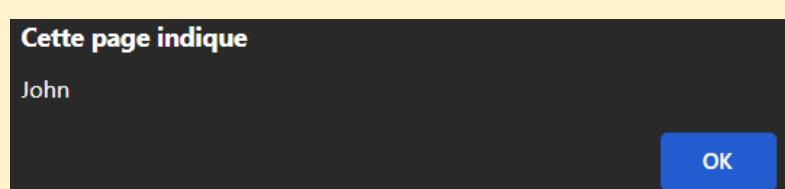
Travailler avec des variables

```
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
</head>
<body>
    <script>
        let name = "John";
        let admin = name;

        alert(admin); // Affiche "John"

    </script>

```



Assigner le bon nom

```
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
</head>
<body>
    <script>

        let planetNom = "Earth";
        let nomUtilisateur = "John";

    </script>
</body>
</html>
```

Exercice 2

String Quotes

Exercices

String quotes

Quelle est la sortie du script ?

```
1 let name = "Ilya";
2
3 alert(`hello ${1}`); // ?
4
5 alert(`hello ${"name"}`); // ?
6
7 alert(`hello ${name}`); // ?
```

```
script.html > html
1  <!DOCTYPE html>
2  <html lang="fr">
3  <head>
4      <meta charset="UTF-8">
5
6 </head>
7 <body>
8
9
10 <script>
11     let name = "Ilya";
12
13     alert(`hello ${1}`); // 1. Affiche "hello 1"
14
15     alert(`hello ${"name"}`); // 2. Affiche "hello name"
16
17     alert(`hello ${name}`); // 3. Affiche "hello Ilya"
18
19 </script>
20 </body>
```

Cette page indique

hello 1

OK

Cette page indique

hello name

•

OK

Cette page indique

hello Ilya

OK

Exercice 3

Une simple page

Exercices

Une simple page

Créez une page Web qui demande un nom et l'affiche

Cette page indique

Quel est votre nom ?

OK

Annuler

Cette page indique

Votre nom est : lilou

OK

Exercice 4

Exercices

Les formes postfixes et préfixes

Quelles sont les valeurs finales de toutes les variables `a`, `b`, `c` et `d` après le code ci-dessous ?

```
1 let a = 1, b = 1;
2
3 let c = ++a; // ?
4 let d = b++; // ?
```

Les formes postfixes et préfixes

```
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <title>Préfixes et postfixes</title>
</head>
<body>
    <script>
        let a = 1, b = 1;

        let c = ++a; // a est incrémenté avant son affectation
        // a devient 2 et c reçoit alors 2

        let d = b++; // b est incrémenté après affectation
        // d reçoit 1 qui est la valeur avant incrémantation et b devient 2
    </script>
</body>
</html>
```

Résultat d'affectation

Résultat d'affectation

Quelles sont les valeurs de `a` et `x` après le code ci-dessous ?

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

```
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
</head>
<body>
    <script>
        let a = 2;

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

        // a *= 2 équivaut à a = a * 2
        // Donc a devient 4
        // L'expression (a *= 2) retourne 4

        // Étape par étape :
        // a *= 2 à a = 2 * 2 = 4
        // (a *= 2) retourne 4
        // 1 + 4 = 5
        // x = 5

    </script>
</body>
</html>
```

Les conversions de type

Les conversions de type

Quels sont les résultats de ces expressions ?

```
1  "" + 1 + 0
2  "" - 1 + 0
3  true + false
4  6 / "3"
5  "2" * "3"
6  4 + 5 + "px"
7  $" + 4 + 5
8  "4" - 2
9  "4px" - 2
10 " -9 " + 5
11 " -9 " - 5
12 null + 1
13 undefined + 1
14 " \t \n" - 2
```

```
// 1. "" + 1 + 0
// "" + 1 = "1" (concatenation)
// "1" + 0 = "10" (concatenation)

// 2. "" - 1 + 0
// "" devient 0 en conversion numérique
// 0 - 1 = -1
// -1 + 0 = -1

// 3. true + false
// true = 1, false = 0
// 1 + 0 = 1

// 4. 6 / "3"
// "3" devient 3 en conversion numérique
// 6 / 3 = 2

// 5. "2" * "3"
// "2" devient 2, "3" devient 3
// 2 * 3 = 6

// 6. 4 + 5 + "px"
// 4 + 5 = 9
// 9 + "px" = "9px" (concatenation)
```

```
// 7. $" + 4 + 5
// $" + 4 = "$4"
// "$4" + 5 = "$45"

// 8. "4" - 2
// "4" devient 4
// 4 - 2 = 2

// 9. "4px" - 2
// "4px" n'est pas un nombre valide → NaN
// NaN - 2 = NaN

// 10. " -9 " + 5
// " -9 " + 5 = " -9 5" (concatenation)

// 11. " -9 " - 5
// " -9 " devient -9 (les espaces sont ignorés)
// -9 - 5 = -14

// 12. null + 1
// null devient 0 en conversion numérique
// 0 + 1 = 1

// 13. undefined + 1
// undefined devient NaN en conversion numérique
// NaN + 1 = NaN
```

```
// 14. " \t \n" - 2
// Chaîne vide (espaces, tabulations, sauts de ligne) devient 0
// 0 - 2 = -2
```

```
body>
<script>
  // CORRECTION : Convertir en nombres avec Number()
  let a = Number(prompt("Premier nombre ?", 1));
  let b = Number(prompt("Deuxième nombre ?", 2));

  alert(a + b); // Maintenant ça affiche 3 !

  // Explication dans la console
  // le Code original aurait donné : '1' + '2' = '12'
  // le Code corrigé donne : 1 + 2 = , a + b
</script>
</body>
</html>
```

Comparaison

Comparaisons

Quel sera le résultat pour les expressions suivantes :

```
1 5 > 4
2 "apple" > "pineapple"
3 "2" > "12"
4 undefined == null
5 undefined === null
6 null == "\n0\n"
7 null === +"\n0\n"
```

```
<!DOCTYPE html>
<html>
<body>
  <script>
    //1. 5 > 4 =", 5 > 4) le résultat sera TRUE
    //2. "apple" > "pineapple" =', "apple" > "pineapple") Le résultat False
    //3. "2" > "12" =', "2" > "12") Le résultat True
    //4. undefined == null =", undefined == null)Le résultat True
    //5. undefined === null =", undefined === null)Le résultat False
    //6. null == "\n0\n" =', null == "\n0\n")Le résultat False
    //7. null === +"\n0\n" =', null === +"\n0\n")Le résultat False
```

Exercice 5

Exercices

if (une chaîne de caractères avec zéro)

Est-ce que `alert` sera affiché ?

```
1 if ("0") {
2   alert( 'Hello' );
3 }
```

if (une chaîne de caractères avec zéro)

```
<!DOCTYPE html>
<html>
<body>
    <script>
        // Ce code :
if ("0") {
    alert('Hello');
}

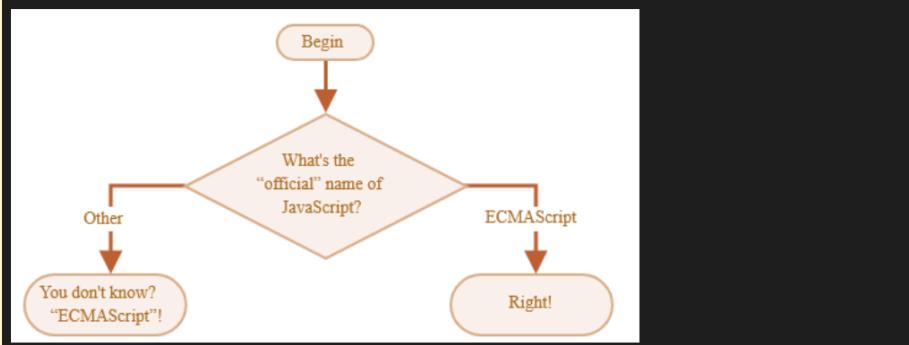
// S'exécute parce que :
// 1. "0" est une string non-vide
// 2. Toutes les strings non-vides sont truthy en JavaScript
// 3. Donc if("0") équivaut à if(true)

    </script>
</body>
</html>
```

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!"



Cette page indique

Quel est le nom "officiel" de JavaScript?

OK Annuler

```
<!DOCTYPE html>
<html>
<body>
    <script>
        // Demander le nom officiel
        let reponse = prompt('Quel est le nom "officiel" de JavaScript?', '');

        // Vérifier la réponse
        if (reponse === 'ECMAScript') {
            alert('Bonne réponse !');
        } else {
            alert('Ne saviez-vous pas ? ECMAScript!');
        }
    </script>
</body>
</html>
```

Cette page indique

Ne saviez-vous pas ? ECMAScript!

OK

Afficher le signe

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.

```
<!DOCTYPE html>
<html>
<body>
<script>
    // Obtenir un nombre via prompt
    let nombre = prompt("Entrez un nombre : ", "0");

    // Convertir en nombre (puisque prompt retourne une string)
    nombre = Number(nombre);

    // Vérifier le signe avec if...else
    if (nombre > 0) {
        alert("1");
    } else if (nombre < 0) {
        alert("-1");
    } else {
        alert("0");
    }
</script>
</body>
</html>
```

Cette page indique

Entrez un nombre :

10

OK

Annuler

Cette page indique

1

OK

Réécrire 'if' en '?

Réécrire 'if' en '?)'

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

```
1 let result;
2
3 if (a + b < 4) {
4     result = 'Below';
5 } else {
6     result = 'Over';
7 }
```

```

<!DOCTYPE html>
<html>
<body>
    <script>
        let result = (a + b < 4) ? 'Below' : 'Over';
        alert(result);
    </script>
</body>
</html>

```

Réécrire 'if...else' en '?"

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.

```

1 let message;
2
3 if (login == 'Employee') {
4     message = 'Hello';
5 } else if (login == 'Director') {
6     message = 'Greetings';
7 } else if (login == '') {
8     message = 'No login';
9 } else {
10     message = '';
11 }

```

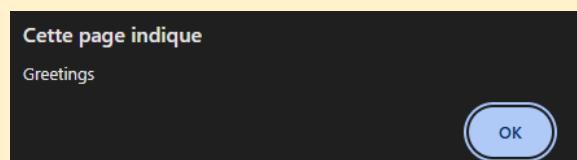
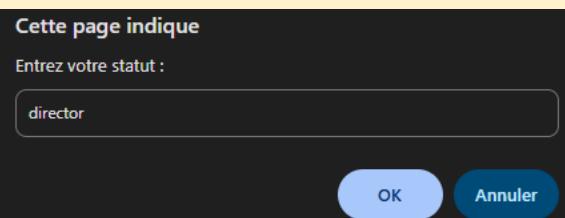
```

<!DOCTYPE html>
<html>
<body>
    <script>
        let login = prompt("Entrez votre statut : ", "");
        let message;

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

        alert(message);
    </script>
</body>
</html>

```



Exercice 5

Quel est le résultat de OR ?

Exercices

Quel est le résultat de OR ?

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

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

```
<!DOCTYPE html>
<html>
<body>
  <script>
    | alert( null || 2 || undefined );
  </script>
</body>
</html>
```

Cette page indique

2

Quel est le résultat des alertes OR ?

Quel est le résultat des alertes OR ?

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

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

```
<!DOCTYPE html>
<html>
<body>
  <script>
    | alert( alert(1) || 2 || alert(3) );
  </script>
</body>
</html>
```

Cette page indique

2

Quel est le résultat de AND ?

Quel est le résultat de AND ?

Qu'est-ce que ce code va afficher ?

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

```
<!DOCTYPE html>
<html>
<body>
  <script>
    alert( 1 && null && 2 );
  </script>
</body>
</html>
```

Cette page indique

null

OK

Quel est le résultat des alertes AND ?

```
cript/html > html > body > script
<!DOCTYPE html>
<html>
<body>
  <script>
    alert[ ( 1 ) && alert ( 2 ) ];
  </script>
</body>
</html>
```

Cette page indique

2

Quel est le résultat de OR AND OR ?

Le résultat de OR AND OR

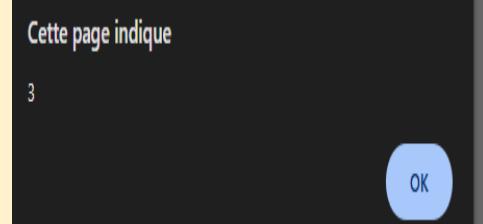
Quel sera le résultat ?

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

```

<!DOCTYPE html>
<html>
<body>
    <script>
        alert( null || 2 && 3 || 4 );
    </script>
</body>
</html>

```



Vérifier la plage entre

Vérifiez la plage entre

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

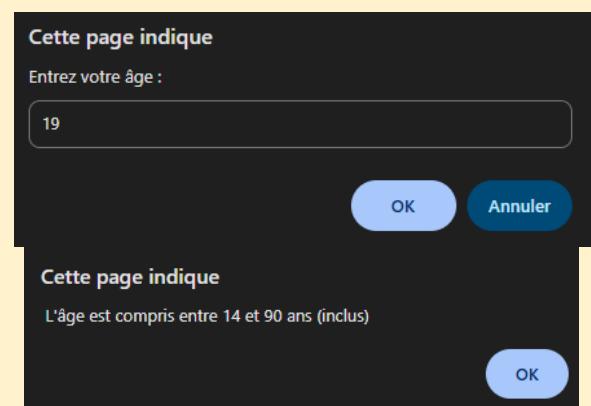
"Inclus" signifie que l'`age` peut atteindre les `14` ou `90` ans.

```

<!DOCTYPE html>
<html>
<body>
    <script>
        let age = prompt("Entrez votre âge :", "25");
        age = Number(age);

        if (age >= 14 && age <= 90) {
            alert("L'âge est compris entre 14 et 90 ans (inclus)");
        } else {
            alert("L'âge n'est PAS compris entre 14 et 90 ans");
        }
    </script>

```

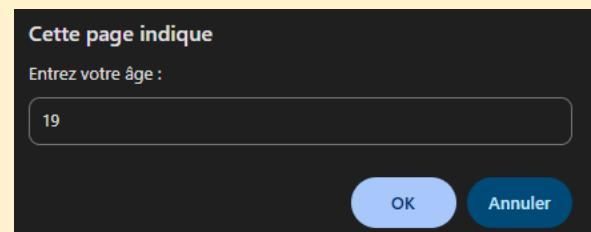


```

<!DOCTYPE html>
<html>
<body>
    <script>
        let age = Number(prompt("Entrez votre âge :", "25"));

        // Variante 2 : SANS NOT
        if (age < 14 || age > 90) {
            alert("L'âge n'est PAS entre 14 et 90 ans (sans NOT)");
        } else {
            alert("L'âge est entre 14 et 90 ans");
        }
    </script>
</body>
</html>

```



Une question à propos de ||

Lesquelles de ces `alert`s 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' );
```

```
<!DOCTYPE HTML>
<html> .
<body>
  <script>
    -1 || 0
    // -1 est truthy (nombre non-zéro)
    // Retourne -1 (premier truthy)
    // if(-1) → true
  </script>
</body>
</html>
```

```
<!DOCTYPE HTML>
<html>
<body>
  <script>
    -1 && 0
    // -1 est truthy, on continue
    // 0 est falsy (premier falsy trouvé)
    // Retourne 0
    // if(0) → false
  </script>
</body>
```

```
<!DOCTYPE HTML>
<html>
<body>
  <script>
    null || -1 && 1
    // Priorité : && avant ||
    // Donc : null || (-1 && 1)
    // -1 && 1 = 1 (dernier truthy)
    // null || 1 = 1 (premier truthy)
    // if(1) → true
  </script>
</body>
</html>
```

Check the login

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".

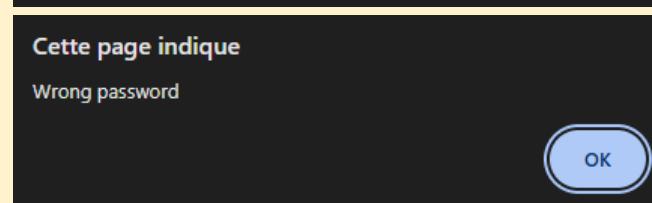
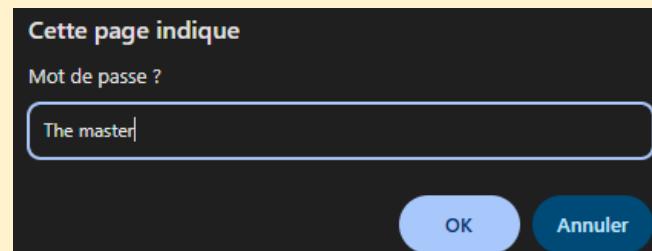
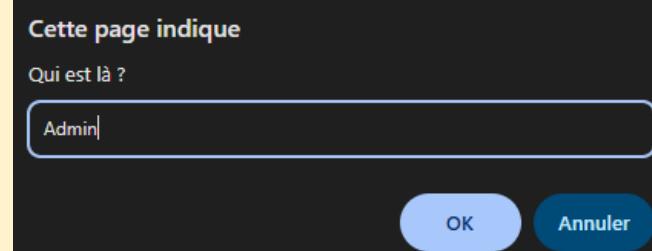
Le mot de passe est vérifié comme suit :

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

```
<!DOCTYPE html>
<html>
<body>
    <script>
        // Demander le login
        let login = prompt("Qui est là ?", "");

        // Vérifier le login
        if (login === null || login === "") {
            // Annulé ou vide
            alert("Canceled");
        } else if (login !== "Admin") {
            // Pas Admin
            alert("I don't know you");
        } else {
            // C'est Admin, demander le mot de passe
            let password = prompt("Mot de passe ?", "");

            // Vérifier le mot de passe
            if (password === null || password === "") {
                // Annulé ou vide
                alert("Canceled");
            } else if (password !== "TheMaster") {
                // Mauvais mot de passe
                alert("Wrong password");
            } else {
                // Bon mot de passe
                alert("Welcome!");
            }
        }
    </script>
</body>
</html>
```



PARTIE 3

Dernière valeur de boucle

Dernière valeur de boucle

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

```
1 let i = 3;
2
3 while (i) {
4     alert( i-- );
5 }
```

```
<!DOCTYPE html>
<html>
<body>
    <script>
        let i = 3; // i = 3 au début
        while (i) { // Tant que i est truthy (# 0)
            alert(i--); // Affiche i, PUIS décrémente i de 1
        }
    </script>
</body>
</html>
```

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` :

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

2.

Le postfixe sous forme `i++` :

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

```
<!DOCTYPE html>
<html>
<body>
    <script>
        let i1 = 0;
        while (++i1 < 5) { // ++i : compare d'abord (0 < 5), puis incrémente (i devient 1), puis compare
            alert("Boucle 1 (++i) : " + i1); // Affiche : 1, 2, 3, 4
        } // Dernière itération : quand i1 = 4 # 4 < 5 = true # entre dans la boucle
        // i2++ fait i2 = 5 # on affiche 5
        // Ensuite i2 = 5 # 5 < 5 = false # sortie

        let i2 = 0;
        while (i2++ < 5) { // i++ : compare d'abord (0 < 5), puis incrémente (i devient 1)
            alert("Boucle 2 (i++) : " + i2); // Affiche : 1, 2, 3, 4, 5
        } // Dernière itération : i2 = 4 # 4 < 5 = true # entre dans la boucle
        // i2++ fait i2 = 5 # on affiche 5
        // Ensuite i2 = 5 # 5 < 5 = false # sortie

        // REPONSE :
        // Boucle 1 (++i) affiche : 1, 2, 3, 4
        // Boucle 2 (i++) affiche : 1, 2, 3, 4, 5
        // Les deux boucles n'affichent PAS les mêmes valeurs !
    </script>
</body>
```

Extraire les nombres pairs dans la boucle

Extraire les nombres pairs dans la boucle

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

```
<!DOCTYPE html>
<html>
<body>
    <script>
        console.log("==> Méthode 2 : Départ à 2, incrément de 2 ==>");
        for (let i = 2; i <= 10; i += 2) {
            alert(i + " est pair");
        }
    </script>
</body>
```

Remplacer "for" par "while"

Remplacer "for" par "while"

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

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



```
<!DOCTYPE html>  
<html>  
<body>  
  <script>  
    let i = 0; // Initialisation  
    while (i < 3) { // Condition  
      alert(`number ${i}!`);  
      i++; // Incrémentation  
    }  
  </script>  
</body>  
</html>
```

Répéter jusqu'à ce que l'entrée soit correcte

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.

```

<!DOCTYPE html>
<html>
<body>
    <script>
        let nombre = 0;
        while (nombre <= 100) {
            nombre = prompt("Entrez un nombre supérieur à 100 : ", "");
            if (nombre === null || nombre === "") {
                alert("Opération annulée.");
                break;
            }
            nombre = Number(nombre);
            if (nombre <= 100) {
                alert("Le nombre doit être supérieur à 100 !");
            }
        }
        // Si on a un nombre valide
        if (nombre > 100) {
            alert(`Merci ! Vous avez entré : ${nombre}`);
        }
    </script>
</body>
</html>

```

Réécrire le "if" dans un "switch"

Réécrire le "if" dans un "switch"

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

```

1 let a = +prompt('a?', '');
2
3 if (a == 0) {
4     alert( 0 );
5 }
6 if (a == 1) {
7     alert( 1 );
8 }
9
10 if (a == 2 || a == 3) {
11     alert( '2,3' );
12 }

```

```

<!DOCTYPE html>
<html>
<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');
                break;
        }
    </script>
</body>
</html>

```

Est-ce que "else" est requis ?

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 :

```
1 function checkAge(age) {  
2   if (age > 18) {  
3     return true;  
4   } else {  
5     // ...  
6     return confirm('Did parents allow you?');  
7   }  
8 }
```

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

```
1 function checkAge(age) {  
2   if (age > 18) {  
3     return true;  
4   }  
5   // ...  
6   return confirm('Did parents allow you?');  
7 }
```

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

Réponse :

Non, il n'y a aucune différence dans le comportement entre les deux versions. Les deux fonctions font exactement la même chose.

Pourquoi c'est identique :

1. Si `age > 18` :

- Dans les deux versions, `return true` est exécuté
- Le `return` termine immédiatement la fonction
- Le code qui suit n'est jamais atteint

2. Si `age <= 18` :

- La condition `if` est fausse
- Dans la version 1 : on entre dans le bloc `else` → `return confirm(...)`
- Dans la version 2 : on saute le bloc `if` et on exécute directement `return confirm(...)`

Le résultat final est le même dans tous les cas. Le `else` est redondant ici à cause de l'instruction `return` dans le `if`.

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

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.

```
1 function checkAge(age) {  
2     if (age > 18) {  
3         return true;  
4     } else {  
5         return confirm('Did parents allow you?');  
6     }  
7 }
```

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 `||`

```
<!DOCTYPE html>  
<html>  
<body>  
    <script>  
        function checkAge(age) {  
            return age > 18 ? true : confirm('Did parents allow you?');  
        }  
    </script>  
</body>  
</html>
```

```
<!DOCTYPE html>  
<html>  
<body>  
    <script>  
        function checkAge(age) {  
            return age > 18 || confirm('Did parents allow you?');  
        }  
    </script>
```

Fonction min(a,b)

Fonction min(a, b)

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

Par exemple :

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

```
<!DOCTYPE html>
<html>
<body>
|   <script>
function min(a, b) {
|     if (a < b) {
|       return a;
|     } else {
|       return b;
|     }
|   </script>
</body>
</html>
```

Fonction pow(x,n)

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.

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

```
<!DOCTYPE html>
<html>
<body>
|   <script>
function pow(x, n) {
|     return x ** n; // Utilise l'opérateur d'exponentiation ES7
}
|   </script>
</body>
</html>
```

Réécrire avec les fonctions fléchées

Réécrire avec les fonctions fléchées

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

```
1 function ask(question, yes, no) {  
2     if (confirm(question)) yes();  
3     else no();  
4 }  
5  
6 ask(  
7     "Do you agree?",  
8     function() { alert("You agreed."); },  
9     function() { alert("You canceled the execution."); }  
10 );
```

```
<!DOCTYPE html>  
<html>  
<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>  
</body>  
</html>
```

