

Baptiste Gaillard
BTS SIO

Atelier JavaScript

Fondamentaux 1 :	2
Fondamentaux 2 :	3
Fondamentaux 3 :	14

Fondamentaux 1 :

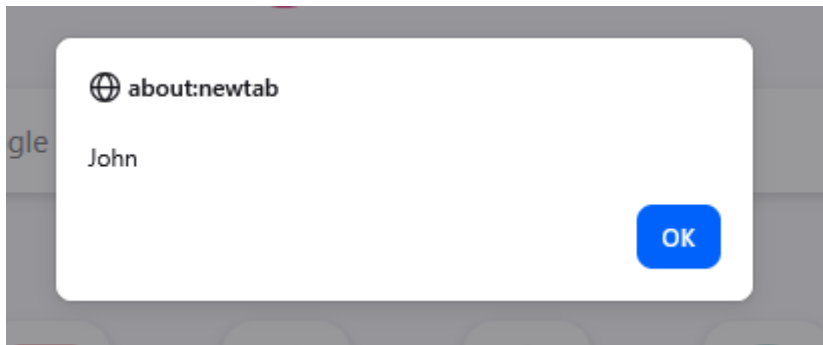
Exercice 1 :



Fondamentaux 2 :

Exercice 1 :





Exercice 2 :

```
let pplanete;  
let vvisiteur;
```

Exercice 3 :

La casse est importante donc non il n'est pas judicieux d'utiliser BIRTHDAY pour birthday car ce n'est pas les mêmes variables !

Exercise 4 :

```
<script>
  "use strict";

  let name = "Ilya";
  alert( `hello ${1}` );
  alert( `hello ${"name"}` );
  alert( `hello ${name}` ); |
</script>
```

file://

hello 1

OK

file://

hello name

☐ Don't allow this site to prompt you again

OK

file://

hello Ilya

☐ Don't allow this site to prompt you again

OK

Exercice 5 :

```
<script>

  "use strict";

  let nom = prompt("Quel est votre nom ?");
  alert( `Bonjour, ${nom}` )

</script>
```

The image shows two sequential browser dialog boxes. The first is a prompt dialog with the title 'file://' and the message 'Quel est votre nom ?'. It contains a text input field with the value 'Baptiste' and two buttons: 'OK' and 'Cancel'. The second dialog is an alert box with the title 'file://' and the message 'Bonjour, Baptiste'. It has a checkbox labeled 'Don't allow this site to prompt you again' which is unchecked, and a single 'OK' button.

file://

Quel est votre nom ?

Baptiste

OK Cancel

file://

Bonjour, Baptiste

☐ Don't allow this site to prompt you again

OK

Exercise 6 :

```
<script>

    "use strict";

    let a = 1, b = 1;
    let c = ++a;
    let d = b++;
    alert(`a = ${a}, b = ${b}, c = ${c}, d = ${d}`)

</script>
```

file://

a = 2, b = 2, c = 2, d = 1

OK

Exercise 7 :

```
<script>

    "use strict";

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

    alert(`a = ${a}, x = ${x}`)

</script>
```

file://

a = 4, x = 5

OK

Exercice 8 :

<u>Expression</u>	<u>Résultat</u>	<u>Explication</u>
<code>"" + 1 + 0</code>	<u>"10"</u>	Concaténation de chaînes : <code>"" + 1</code> → <code>"1"</code> , puis <code>"1" + 0</code> → <code>"10"</code> .
<code>"" - 1 + 0</code>	<u>-1</u>	<code>""</code> devient 0, donc <code>0 - 1 + 0</code> → -1.
<code>true + false</code>	<u>1</u>	<code>true</code> → 1, <code>false</code> → 0, donc <code>1 + 0</code> → 1.
<code>6 / "3"</code>	<u>2</u>	<code>"3"</code> est converti en nombre → <code>6 / 3</code> .
<code>"2" * "3"</code>	<u>6</u>	Les deux chaînes sont converties en nombres.
<code>4 + 5 + "px"</code>	<u>"9px"</u>	<code>4 + 5</code> → 9, puis <code>9 + "px"</code> → <code>"9px"</code> .
<code>"\$" + 4 + 5</code>	<u>"\$45"</u>	Concaténation de chaînes à partir du \$.
<code>"4" - 2</code>	<u>2</u>	<code>"4"</code> devient 4, opération arithmétique normale.
<code>"4px" - 2</code>	<u>NaN</u>	<code>"4px"</code> ne peut pas être converti en nombre.
<code>" -9 " + 5</code>	<u>" -9 5"</u>	L'espace est conservé, concaténation de chaînes.
<code>" -9 " - 5</code>	<u>-14</u>	La chaîne est convertie en -9 malgré les espaces.
<code>null + 1</code>	<u>1</u>	<code>null</code> devient 0 → <code>0 + 1</code> .
<code>undefined + 1</code>	<u>NaN</u>	<code>undefined</code> devient NaN → résultat NaN.
<code>" \t \n" - 2</code>	<u>-2</u>	String avec espaces/tab/retour ligne → converti en 0, donc <code>0 - 2</code> .

Exercice 9 :

`prompt()` **renvoie toujours une chaîne de caractères !**.

Il faut donc les convertir en nombre :

```
<script>

  "use strict";

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

  alert(a + b);

</script>
```

Exercice 10 :

Expression	Résultat	Explication
<code>5 > 4</code>	true	Comparaison numérique simple : 5 est bien plus grand que 4.
<code>"apple" > "pineapple"</code>	false	Comparaison lexicographique (ordre du dictionnaire). "a" est avant "p", donc <code>"apple" < "pineapple"</code> .
<code>"2" > "12"</code>	true	Comparaison lexicographique : elle compare caractère par caractère. "2" est plus grand que "1".
<code>undefined == null</code>	true	Règle spéciale : <code>undefined</code> n'est égal qu'à <code>null</code> dans le <code>==</code> .
<code>undefined === null</code>	false	Types différents → <code>===</code> ne convertit rien → toujours <code>false</code> .
<code>null == "\n0\n"</code>	false	<code>null</code> n'est égal qu'à <code>undefined</code> avec <code>==</code> , jamais à une string.
<code>null === +"\n0\n"</code>	false	<code>+"\n0\n"</code> → conversion en nombre → 0 . Types différents (<code>null</code> vs <code>number</code>) → <code>false</code> .

Exercice 11 :

Cela s'affiche car **"0" (string)** est **vrai** du fait qu'il représente une chaîne de caractère non vide , mais **attention 0 (nombre) est faux.**

Exercice 12 :

```
<script>

    "use strict"

    let reponse = prompt('Quel est le nom "officiel" de JavaScript?');

    if (reponse === 'ECMAScript') {
        alert('Bonne réponse !');
    } else {
        alert('Ne sait pas ? ECMAScript !');
    }

</script>
```

Exercice 13 :

```
<script>

    "use strict"

    let reponse = prompt('Choisissez un nombre');

    if (reponse >= 0) {
        alert('1');
    }

    else if (reponse == 0 ) {
        alert('0')
    }
    else {
        alert('-1');
    }

</script>
```

Exercice 14 :

```
<script>

    "use strict";

    let result = (a + b < 4) ? 'Below' : 'Over';

</script>
```

Exercice 15 :

```
<script>

    "use strict";

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

</script>
```

Exercice 16 :

2 car null = 0 , 2 = 1 et undefined = 0

Exercice 17 :

Le code exécutera la première alerte avec 1, puis l'expression alert(1) || 2 renverra 1 (puisque alert(1) retourne undefined, une valeur falsy). Ensuite, 1 est une valeur "truthy", donc l'expression entière ne va pas exécuter alert(3).

Exercice 18 :

L'opérateur && retourne la première valeur falsy dans une expression. Ici, 1 est truthy, donc il passe au suivant, qui est null (falsy), donc l'expression retourne null et l'alerte affiche null.

Exercice 19 :

L'opérateur && évalue les expressions de gauche à droite. Le premier alert(1) sera exécuté, et comme il retourne undefined (falsy), le deuxième alert(2) ne sera pas exécuté. Le code affiche uniquement 1.

Exercice 20 :

L'opérateur && a une priorité plus élevée que ||, donc le code sera évalué ainsi :

- 2 && 3 retourne 3 (puisque 2 et 3 sont tous deux truthy).
- Ensuite, null || 3 retourne 3.
- Enfin, 3 || 4 retourne 3.

Exercice 21 :

```
<script>

  "use strict";

  let age

  if (age > 13 || age < 91) {
    alert("inclus")
  } else {
    alert("non inclus")
  }

</script>
```

Exercice 22 :

1. Version avec l'opérateur NOT !

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

2. Version sans utiliser !

```
if (age < 14 || age > 90) {  
}
```

Exercice 23 :

Uniquement first et third pas second

Exercice 24 :

```
let user = prompt("Who's there?");  
  
if (user === null || user === "") {  
    alert("Canceled");  
} else if (user === "Admin") {  
    let password = prompt("Password?");  
  
    if (password === null || password === "") {  
        alert("Canceled");  
    } else if (password === "TheMaster") {  
        alert("Welcome!");  
    } else {  
        alert("Wrong password");  
    }  
} else {  
    alert("I don't know you");  
}
```

Fondamentaux 3 :

Exercice 1 :

Cela affiche chaque itération de la boucle. à chaque itération i est soustrait de 1 donc le programme affiche 3, 2, 1.

Exercice 2 :

Oui les deux affichent les mêmes valeurs . (1,2,3,4)

Exercice 3 :

Oui les deux affichent les mêmes valeurs .(1,2,3,4)

Exercice 4 :

Usage d'un pas de 2

```
"use strict";
let i
for (let i = 0; i < 12; i+=2 ) alert(i);
```

Exercice 5 :

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

Exercice 6 :

```
let i = 0;
while (i < 100) {
  i = prompt("Veuillez rentrer un nombre supérieur à 100");
}
```

Exercise 7 :

```

let n = 10;
let premiers = [];

for (let i = 2; i <= n; i++) {
    let estPremier = true;

    for (let div = 2; div < i; div++) {
        if (i % div === 0) {
            estPremier = false;
        }
    }

    if (estPremier) {
        premiers.push(i);
    }
}

alert(premiers);

```

🌐 file://

2,3,5,7

Exercise 8 :

```

if (browser === 'Edge') {
    alert("You've got the Edge!");
} else if (
    browser === 'Chrome' ||
    browser === 'Firefox' ||
    browser === 'Safari' ||
    browser === 'Opera'
) {
    alert('Okay we support these browsers too');
} else {
    alert('We hope that this page looks ok!');
}

```

Exercice 9 :

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

switch (a) {
  case 0:
    alert(0);
    break;

  case 1:
    alert(1);
    break;

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

Exercice 10 :

Non, il n'y a **aucune différence de comportement** entre les deux programmes. L'else est **inutile** ici, car le return coupe déjà l'exécution. Le comportement est identique.

Exercice 11 :

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

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

Exercice 12 :

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

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

Exercice 13 :

```
function pow(x, n) {
  let result = 1;
  for (let i = 0; i < n; i++) {
    result *= x;
  }
  return result;
}

alert(pow(3, 2))
alert(pow(3, 3))
alert(pow(1, 100))
```

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>

  <script>

    "use strict";

    function pow(x, n) {
      let result = 1;
      for (let i = 0; i < n; i++) {
        result *= x;
      }
      return result;
    }

    let x = prompt("Entrez x :");
    let n = prompt("Entrez n (entier ≥ 1 :)");

    n = Number(n);
    x = Number(x);

    if (n < 1) {
      alert("La puissance n doit être un entier naturel ≥ 1.");
    } else {
      alert( pow(x, n) );
    }

  </script>

</body>
</html>
```


Exercice 14 :

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

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

Navigateur 1 :

Exercice 1 :

- 1: Accéder au nœud <div> : `document.querySelector('div');`
- 2: Accéder au nœud : `document.querySelector('ul');`
- 3: Accéder au deuxième (Pete) : `document.querySelectorAll('li')[1];`

Exercice 2 :

- 1: Oui c'est vrai
- 2: Non c'est faux

Exercise 3 :

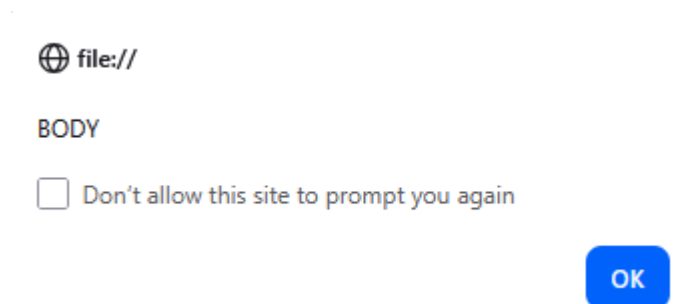
```
const rows = document.querySelectorAll('table tr');

rows.forEach((row, i) => {
  const tds = row.querySelectorAll('td');
  if (tds[i]) {
    const td = tds[i];
    td.style.backgroundColor = 'red';
  }
});
```

Exercise 4 :



Exercise 5 :



Exercise 6 :

- 1) document est une instance de HTMLDocument
- 2) EventTarget -> Node -> Document -> HTMLDocument
- 3) il hérite uniquement de node