

2.4. La concaténation

Une concaténation consiste à ajouter une chaîne de caractères à la fin d'une autre, comme dans cet exemple :

<pre><script> var hi = 'Bonjour ', name = 'toi', result; result = hi + name; alert(result); // Affiche : « Bonjour toi » </script></pre>	<p>ou :</p> <pre><script> var text = 'Bonjour '; text += 'toi'; alert(text); // Affiche « Bonjour toi ». </script></pre>
--	--

On peut convertir un nombre en chaîne de caractères avec l'astuce suivante :

```
<script>
    var text, number1 = 4, number2 = 2;
    text = number1 + ' ' + number2; /* on ajoute une chaîne de caractères vide
entre les deux nombres pour permettre une concaténation sans calcul */
    alert(text); // Affiche : « 42 »
</script>
```

2.5. La fonction `prompt()`, avec concaténation et calcul

Voici la base de cette fonction :

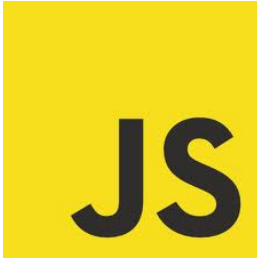
```
<script>
    var userName = prompt('Entrez votre prénom :');
    alert(userName); // Affiche le prénom entré par l'utilisateur
</script>
```

On peut demander le prénom et afficher un message avec concaténation :

```
<script>
    var start = 'Bonjour ', name, end = ' !', result;
    name = prompt('Quel est votre prénom ?');
    result = start + name + end;
    alert(result);
</script>
```

On peut aussi se servir de la fonction `prompt()` pour un calcul :

```
<script>
    var first, second, result;
    first = prompt('Entrez le premier chiffre :');
    second = prompt('Entrez le second chiffre :');
    result = parseInt(first) + parseInt(second); /* La fonction parseInt()
transforme la chaîne de caractères en nombre */
    alert(result);
</script>
```



FICHE 3. Les conditions (1/2)

Une condition (`true` ou `false`) est un test qui permet de vérifier qu'une variable contient bien une certaine valeur.

3.1. Les huit opérateurs de comparaison

Il y en a 8 :

<code>==</code> : égal à	<code>===</code> : contenu <u>et</u> type de variable égal à	<code>></code> supérieur à	<code><</code> : inférieur à
<code>!=</code> : différent de	<code>!==</code> : contenu <u>ou</u> type de variable différent de	<code>>=</code> supérieur ou égal à	<code><=</code> : inférieur ou égal à

Il suffit d'écrire deux valeurs avec l'opérateur de comparaison souhaité entre les deux et un booléen est retourné. Si celui-ci est `true` alors la condition est vérifiée, si c'est `false` alors elle ne l'est pas :

```
<script>
var number1 = 2, number2 = 2, number3 = 4, result;
  result = number1 == number2; // Au lieu d'une seule valeur, on en écrit
deux avec l'opérateur de comparaison entre elles
  alert(result); // La condition est donc vérifiée car les deux variables
contiennent bien la même valeur
  result = number1 == number3;
  alert(result); // La condition n'est pas vérifiée car 2 est différent de 4
  result = number1 < number3;
  alert(result); // La condition est vérifiée car 2 est bien inférieur à 4
</script>
```

3.2. Les opérateurs logiques

Il y en a 3 :

`&&` qui signifie ET avec par exemple : `valeur1 && valeur2`

Cet opérateur vérifie la condition lorsque toutes les valeurs qui lui sont passées valent `true`.

`||` qui signifie OU avec par exemple : `valeur1 || valeur2`

Cet opérateur est plus souple car il renvoie `true` si une des valeurs qui lui est soumise contient `true`, qu'importent les autres valeurs.

`!` qui signifie NON avec par exemple : `!valeur`

Cet opérateur se différencie des deux autres car il ne prend qu'une seule valeur à la fois. S'il se nomme « NON » c'est parce que sa fonction est d'inverser la valeur qui lui est passée, ainsi `true` deviendra `false` et inversement.