

Chapitre 2-2

Conditionnement des traitements

1. Présentation de la syntaxe

La structure conditionnelle en JavaScript est très proche syntaxiquement de celle vue précédemment en langage descriptif algorithmique.

Tout à fait classiquement le bloc d'instructions à exécuter dans le cas où la condition testée est vraie est délimité par un jeu d'accolades (`{}`). Il est aussi possible de prévoir une séquence d'instructions alternative avec le mot-clé `else`. Cette séquence sera aussi encadrée par des accolades.

Le langage JavaScript est assez permissif quant au positionnement de ces accolades. Ainsi vous rencontrerez dans les scripts les constructions suivantes :

```
if (condition)
{
    Actions_1;
}
else
{
    Actions_2;
}
```

ou

```
if (condition) {
    Actions_1;
}
else {
    Actions_2;
}
```

ou

```
■ if (condition) { Actions_1; } else { Actions_2; }
```

avec :

- condition représentant un test de comparaison générant un résultat booléen `true` ou `false`,
- `Actions_1` et `Actions_2` représentant des séquences d'instructions (en général réparties sur plusieurs lignes).

■ Remarque

Attention pour effectuer un test de comparaison en égalité l'opérateur est le double égal (==) à ne pas confondre avec le simple égal (=) qui sert à effectuer les affectations.

Dans les tests de comparaison avec des constantes on préférera par exemple :

```
■ if (5 == compteur)
{
    Actions_1;
}
else
{
    Actions_2;
}
```

à

```
■ if (compteur == 5)
{
    Actions_1;
}
else
{
    Actions_2;
}
```

car l'étourderie consistant à confondre le `==` avec le `=` est plus facile à détecter dans le premier cas, JavaScript renvoyant une erreur. Dans le second cas (si vous mettez un `=` au lieu d'un `==`) la séquence `Actions_1` sera systématiquement exécutée car l'évaluation de `compteur = 5` donnerait toujours `true` (JavaScript estimant que l'affectation est réussie).

2. Exemples

■ Remarque

Pour faciliter le repérage des exercices JavaScript, la numérotation vue dans le chapitre Développement à partir d'algorithmes sera conservée.

2.1 Exercice n°6 : Polynôme du second degré

Sujet

Calculer les racines d'un polynôme du second degré Ax^2+Bx+C (avec $A \neq 0$ dans l'absolu mais ce test ne sera pas effectué ici). Les valeurs A, B et C seront saisies au clavier.

Corrigé (partiel) en JavaScript

```
/* Saisie des paramètres */
a = prompt("a :");
b = prompt("b :");
c = prompt("c :");

/* Calcul du discriminant */
delta = (parseInt(b) * parseInt(b)) - (4 * parseInt(a) * parseInt(c));

/* Affichage des paramètres */
document.write("a : " + a + "<br />");
document.write("b : " + b + "<br />");
document.write("c : " + c + "<br />");
document.write("Discriminant delta : " + delta + "<br />");

/* Détermination des racines */
if (delta < 0)
{
    document.write("Pas de solutions");
}
else
{
    if (delta == 0)
    {
        document.write("Solution unique : " + (-b / (2 * a)));
    }
    else
    {
        document.write("Solution n°1 : " + (-b + Math.sqrt(delta)) /
            (2 * a) + "<br />");
        document.write("Solution n°2 : " + (-b - Math.sqrt(delta)) /
```

```
        (2 * a));  
    }  
}
```

Commentaires du code JavaScript

Vous remarquerez qu'il a fallu imbriquer deux structures conditionnelles pour traiter le problème posé. La non-imbrication des structures était possible (équivalent de trois **Si ... Finsi** successifs au niveau algorithmique) mais cette solution n'aurait pas été optimale. Vous noterez aussi le soin particulier apporté au niveau de l'alignement des accolades et également au niveau du décalage (indentation) des blocs d'instructions.

Pour déterminer la racine carrée du discriminant (`delta`), il a fallu avoir recours à la méthode `sqrt` de l'objet JavaScript `Math`. Nous aurons l'occasion de revoir cet objet plus loin dans ce livre. Le calcul de la racine carrée aurait aussi pu être effectué par une élévation de `delta` à la puissance 0.5. L'opération de l'élévation à la puissance (exponentiation) est notée `^`.

2.2 Exercice n°8 : Impression du libellé d'un mois

Sujet

Imprimer en lettres le mois correspondant à un numéro donné au clavier (compris entre 1 et 12). Le contrôle de la saisie n'est pas à prévoir.

Corrigé (partiel) en JavaScript

```
/* Déclaration de variables locales */  
var mois;  
  
/* Saisie du numéro de mois */  
mois = parseInt(prompt("Numéro du mois (1 à 12) :"));  
  
/* Affichage du résultat */  
switch (mois)  
{  
    case 1:  
        document.write("Mois n° " + mois + " : Janvier");  
        break;  
    case 2:  
        document.write("Mois n° " + mois + " : Février");  
        break;  
    case 3:  
        document.write("Mois n° " + mois + " : Mars");  
        break;  
    case 4:  
        document.write("Mois n° " + mois + " : Avril");  
}
```

```
        break;
    case 5:
        document.write("Mois n° " + mois + " : Mai");
        break;
    case 6:
        document.write("Mois n° " + mois + " : Juin");
        break;
    case 7:
        document.write("Mois n° " + mois + " : Juillet");
        break;
    case 8:
        document.write("Mois n° " + mois + " : Août");
        break;
    case 9:
        document.write("Mois n° " + mois + " : Septembre");
        break;
    case 10:
        document.write("Mois n° " + mois + " : Octobre");
        break;
    case 11:
        document.write("Mois n° " + mois + " : Novembre");
        break;
    case 12:
        document.write("Mois n° " + mois + " : Décembre");
        break;
    default:
        document.write("Erreur de saisie sur le n° de mois");
}
```

Commentaires du code JavaScript

Plutôt que, comme dans l'exercice précédent, d'imbriquer des structures conditionnelles (fastidieux à écrire dans notre cas), la structure `switch` est ici utilisée.

Notez au niveau de la saisie du numéro du mois dans la variable `mois` (non typée au niveau de sa déclaration) qu'une conversion est demandée par l'intermédiaire de la méthode `parseInt`. La variable `mois`, une fois la saisie et la conversion effectuées, sera de type entier pour la suite du traitement.

La variable `mois` est testée par la structure conditionnelle `switch`. Dans le cas où la variable `mois` vaut 1, l'affichage de l'intitulé "Janvier" est effectué, et ainsi de suite.

Dans le cas où une saisie erronée du numéro de mois est faite alors l'instruction intégrée dans le cas `default` est exécutée.

Revenons aussi sur le rôle primordial de l'instruction `break`. En son absence en fin de chacun des cas, dès que la condition est vérifiée le traitement associé est déclenché mais ce serait également le cas pour tous les traitements suivants (même si la condition n'est pas respectée pour ces traitements). Par exemple, la saisie de la valeur 5 en tant que numéro de mois générerait l'affichage de Mai, Juin, Juillet, ..., Décembre.