

Les procédures

Rappelons brièvement l'intérêt des procédures en programmation (vous pouvez revenir au chapitre Développement à partir d'algorithmes pour de plus amples explications). Vouloir coder en JavaScript des traitements trop complexes, mettant en œuvre un nombre important de variables, d'objets et comportant par exemple des jeux de boucles et des structures conditionnelles imbriquées, peut être un frein sérieux à la maintenance du script. Ceci est encore plus vrai quand le code est mal agencé (absence de commentaires, indentations aléatoires, noms des variables et objets peu significatifs).

Une autre motivation importante à coder par l'intermédiaire de procédures est la réutilisation possible de celles-ci dans des contextes différents. Il sera possible de passer des paramètres à ces procédures. Si ces séquences de code appelées depuis un traitement principal restituent des valeurs en retour, nous parlerons dans ce cas de fonctions. Outre le fait que cette factorisation réduit la longueur globale du code source, elle permet surtout d'isoler (y compris dans un script physiquement différent) une séquence d'instructions fiabilisée pour laquelle nous n'aurons plus de questions à nous poser quant au fonctionnement. Ces procédures (ou fonctions) pourront donc être considérées comme des briques logicielles utilisables simplement dans la construction d'une architecture logicielle.

1. Syntaxe

```
function maProcédure(paramFormel1[, paramFormel2, ...])
{
    /* Séquence de code sans valeur retournée au traitement appelant */
}
```

où

```
function maProcédure(paramFormel1[, paramFormel2, ...]) {
    /* Séquence de code sans valeur retournée au traitement appelant */
}
```

Après le mot clé `function`, il faut placer un nom de procédure suivi d'une série de paramètres formels notés entre parenthèses. En cas de paramètres multiples, il convient de les séparer par des virgules.

Il est surprenant que le mot clé `function` soit utilisé alors que nous parlons de procédure dans ce paragraphe. Nous verrons ultérieurement que `function` est un mot clé servant indifféremment aux procédures et aux fonctions en JavaScript.

La particularité, déjà évoquée, est qu'une procédure ne renvoie pas de valeur en retour.

L'appel de la procédure se fait selon la syntaxe suivante :

```
maProcédure(paramEffectif1[, paramEffectif2, ...]);
```

2. Exercice n°16 : Appel d'une procédure avec passage de paramètres

Sujet

Appeler une procédure affichant le double de deux paramètres effectifs passés en paramètres (les deux paramètres seront à saisir au clavier)

Corrigé (partiel) en JavaScript

```
/* Procédure doublement */
function doublement(x, y)
{
    /* Doublement */
    x = x * 2;
    y = y * 2;
    /* Affichage du couplet de valeurs après doublement */
    document.write("Le couplet doublé est (" + x + ", " + y + "<br />");
}

/* Affichage du nom du script */
alert("Exercice Procedure");

/* Déclaration de variables locales */
var a, b, c, d;

/* Saisie des paramètres a et b */
a = parseInt(prompt("Paramètre a :"));
b = parseInt(prompt("Paramètre b :"));

/* Appel de la procédure doublement */
doublement(a, b);

/* Saisie des paramètres c et d */
c = parseInt(prompt("Paramètre c :"));
d = parseInt(prompt("Paramètre d :"));

/* Appel de la procédure doublement */
doublement(c, d);
```

Commentaires du code JavaScript

La procédure doublement est à positionner si possible avant les appels réalisés (doublement(a, b); puis doublement (c, d);) mais ceci n'est pas obligatoire.

Dans un premier temps deux valeurs entières a et b sont saisies au clavier. Ces deux valeurs seront les paramètres effectifs de la procédure doublement. Les valeurs de a et de b sont respectivement imputées aux paramètres x et y (qui de ce fait sont aussi considérés comme des entiers). La procédure assure ensuite le doublement de chacune des valeurs et affiche elle-même le couplet des valeurs doublées.

Cas particulier de procédure sans paramètres

Les procédures sont également utilisables sans paramètre comme le montre le script ci-après :

```
/* Procédure petitBonjour */
function petitBonjour()

    /* Affichage d'un petit bonjour de Bretagne */
```

```
        document.write("Un petit bonjour de Bretagne<br />");
    }

    /* Affichage du nom du script */
    alert("Exercice Procedure_sans_param");

    /* Premier appel de la procédure petitBonjour */
    petitBonjour();

    /* Deuxième appel de la procédure petitBonjour */
    petitBonjour();
```

Les parenthèses après le nom de la procédure sont quand même requises en l'absence de paramètres.