INTRODUCTION À L'ALGORITHMIQUE

FONCTIONS

2015-01

Glenn ROLLAND / @glenux

CONCEPT

- Sous-programme (ex: béchamel vs gratin)
- Une fonction est une valeur (objet) qu’on peut manipuler comme tout autre objet

UTILISATION

DÉCLARATION

Trois manières de créer une fonction:

- Déclaration
- Expression de fonction (fonction anonyme)
- Par l'appel de `new Function`

Les fonctions déclarées sont parsées avant l'exécution du script (on peut donc l'appeler avant sa définition)

RETURN - VALEUR DE RETOUR

Une fonction retourne toujours une valeur (par défaut c’est la valeur undefined)

EXPRESSION DE FONCTIONS

Partout où on peut mettre une valeur (un objet par exemple), on peut mettre une expression de fonction

```
var increment = function(x) { return x + 1; };
resultat = increment(3);
```

Attention : dans ce cas, la fonction n’est pas créée avant l’exécution du script

ENVIRONNEMENT D'UNE FONCTION

TRAITEMENT D'UN SCRIPT

LexicalEnvironment (voir Javascript Gagnon) window - environnement global

Le traitement d’un script suit les étapes suivantes :

1. Traitement des déclarations de fonction, qui sont ajoutées à `window`
2. Traitement des variables déclarées avec var, qui sont elles aussi ajoutées à `window`, mais avec `undefined` comme valeur
3. Le code est exécuté

ENVIRONNEMENT LEXICAL D'UNE FONCTION

Voici se qui se passe quand une fonction est exécutée :

1. Son environnement lexical est créé
2. Son environnement lexical est peuplé par les variables paramètres, les variables locales (celles déclarées avec `var`) et les fonctions imbriquées déclarées
3. Le code est exécuté
4. À la fin de l'exécution l'environnement lexical est détruit (sauf en situation de *fermeture*, comme nous le verrons plus loin)

PORTÉE

- Les blocs n'ont pas de portée
- Les variables utilisées dans une expression for existent encore à la sortie de la boucle

FERMETURE (CLOSURE)

- Une variable est d'abord cherchée dans l'environnement lexical d'une fonction
- Sinon Javascript la cherche dans le premier environnement englobant
- Une variable initialisée sans le mot-clé `var` sera toujours placée dans l'environnement lexical global : `window`
- Une fonction peut continuer d'exister une fois que l'exécution de sa fonction englobante est terminée (ex: si on retourne cette fonction)
- Dans ce cas, la fonction conserve un lien vers l'environnement lexical de la fonction englobante

FERMETURE - SUITE

```
function f(x) {
    function g(y) {
        return x + y;
    };
    return g;
}

// retourne la fonction g(),
// mais x sera liée à la valeur 3
var ajouterTrois = f(3);

ajouterTrois(4);
```

LIEN AVEC LES VARIABLES

Une variable déclarée avec le mot-clé var est une variable locale

PARAMÈTRES

VARIABLES LIBRES

VARIABLES LIÉES

RECOUVREMENT DE VARIABLE

FONCTIONS RÉCURSIVES

APPELS DE FONCTIONS VS PILE

CONDITION D'ARRÊT

EXERCICES

- fonction "calcul du prix TTC"
- fonction "calcul du prix HT"