

SOMMAIRE

[Règles Générales 2](#_Toc338680080)

[Les types de notations 2](#_Toc338680081)

[Règles générales 2](#_Toc338680082)

[Normes de développement Javascript 3](#_Toc338680083)

[Fonctions/Méthodes 3](#_Toc338680084)

[Variables 3](#_Toc338680085)

[Paramètres 3](#_Toc338680086)

[Normes de développement SQL 4](#_Toc338680087)

[Nommage des Bases 4](#_Toc338680088)

[Nommage des Tables 4](#_Toc338680089)

# Règles Générales

## Les types de notations

|  |  |  |
| --- | --- | --- |
| **Nom** | **Descriptif** | **Exemple** |
| **Pascal** | La 1ère lettre de chaque mot est en lettre capitale. | BackColor, BackGround |
| **Camel** | La 1ère lettre est en minuscule et toutes les 1ères lettres des mots suivants sont en majuscules. | backColor, backGround |
| **Uppercase** | Toutes les lettres sont en majuscules. | BACKCOLOR, BACKGROUND |
| **Notation hongroise** | La 1ère lettre est le type de la variable et est en minuscule puis le reste de la notation est en Pascal. | intBackcolor |

## Règles générales

* Les noms doivent décrire au mieux l’élément qu’ils désignent et être les plus claires et parlants possible.
* Eviter le plus possible les abréviations.
* Les underscores « \_ » sont interdits sauf pour les méthodes évènementielles. (cf. normes de développement Javascript)
* Tout acronyme de 3 lettres ou plus doit être dans la notation d’origine et non « Uppercase » (par exemple « Xml » en Pascal au lieu de « XML »).
* Ne pas donner le même nom avec une casse différente à deux éléments différents. Afin d’éviter toutes confusions.

# Normes de développement Javascript

## Fonctions/Méthodes

**Type de notation** : Pascal

Pas d’underscore excepté pour les gestionnaires d’évènements (Event Handlers).

**Exemple** :

***JS:*** *Function* ***SupprimerFlamme****(...)*

## Variables

**Type de notation** : Hongroise

**Exemple** :

***JS:*** *var* ***intNbExplosions*** *= 0;*

## Paramètres

**Type de notation** : Hongroise

**Exemple** :

***JS:*** *function EnvoyerMessage(****strMessage****)*

# Normes de développement SQL

Comme pour le développement, les règles de nommage des objets SQL permettent une lisibilité du code et une meilleure compréhension des objets utilisés.

## Nommage des Bases

Le nom des bases doit respecter la notation « Pascal » et être précédé du préfixe « Db ».

**Exemple** :

* *DbMonApplication*

## Nommage des Tables

Le nom de la table doit respecter la notation « Pascal ». Ce nom doit être suffisamment précis et parlant afin d’exprimer clairement l’objectif fonctionnel de la table, et afin de pouvoir identifier chaque table de la base.

En plus du nom il faut ajouter deux préfixes :

* « Code Environnement » : préfixe sur trois lettres suivi d’un « underscore »
* « Code Table» : préfixe sur trois lettres suivi d’un « underscore »

Le « Code Environnement » correspond à un trigramme permettant d’identifier les éléments faisant partie d’un même ensemble fonctionnel ou technique, en général cela correspond à une application. Ce préfixe sera repris dans les tables, procédures stockées, fonctions, vues.

**Exemple** :

* Ensemble des tables de l’application « MonApplication » : *MAP\_*

Le « Code Table » est un code permettant d’identifier la table sur 3 caractères. Ce préfixe sera repris dans les champs de la table.

**Exemple** :

* Table contenant les utilisateurs de l’application « MonApplication » : *MAP\_USR\_Utilisateurs*

## Nommage des Champs des tables

Les champs des tables doivent identifier clairement la valeur qu’ils contiennent. Le nommage à respecter est la notation Pascal.

En plus du nom, le champ doit être préfixé du « Code Table » de la table à laquelle il appartient.

**Exemple** :

* Champ pour les prénoms des utilisateurs : *USR\_Prenom*

Les champs de type « IDENTITY » doivent porter le nom ID. Ainsi l’identifiant de la table MAP\_USR\_Utilisateurs doit être : *USR\_ID*.

## Nommage des Procédures stockées

Le nommage des procédures stockées doit respecter la notation Pascal. Le nom des procédures stockées débutent toujours par un verbe représentant son action puis elle est complétée par l’objet sur lequel elle porte.

Chaque procédure stockée doit être préfixée par le « Code Base » puis le préfixe « sp ».

**Exemple** :

* Procédure stockée renvoyant les utilisateurs de l’application « MonApplication » : *MAP\_spGetUtilisateurs()*

## Nommage des Vues

Le nommage des vues doit respecter la notation Pascal. Comme pour les procédures stockées elles débutent toujours par un verbe représentant son action puis elles sont complétées par l’objet sur lequel elle porte.

Chaque vue doit être préfixée par le « Code Base » puis le préfixe « vw ».

**Exemple** :

* Vue renvoyant l’ensemble des utilisateurs actifs de l’application « MonApplication » : *MAP\_vwUtilisateursActifs ()*

## Nommage des fonctions

Le nommage des fonctions doit respecter la notation Pascal. Comme pour les procédures stockées elles débutent toujours par un verbe représentant son action puis elles sont complétées par l’objet sur lequel elle porte.

Chaque fonction doit être préfixée par le « Code Base » puis le préfixe « fn ».

**Exemple** :

* Vue renvoyant l’utilisateur correspondant à l’Identifiant fourni en paramètre dans la base « MonApplication » : *MAP\_fnGetUtilisateur (@intIdUtilisateur INTEGER)*

# Typages personnalisés

## Tableau de correspondance

|  |  |  |
| --- | --- | --- |
| **Type** | **Acronyme** | **Exemple** |
| **Perso** | PRS | prsNomVariable |
| **Bombe** | BMB | bmbNomVariable |
| **Decor** | DEC | decNomVariable |
| **Bonus** | BNS | bnsNomVariable |
| **XMLHttpRequest** | XHR | xhrNomVariable |
| **Image** | IMG | imgNomVariable |
| **Array** | TAB | tabNomVariable |
| **Sprite** | SPR | sprNomVariable |