

# MANUEL IRONMIND

*V1.4  
Programmes de  
Formation*

*[contact@goVan.fr](mailto:contact@goVan.fr)*

## INSTRUCTIONS

*Ce manuel est destiné aux créateurs de programmes pour l'application mobile Ironmind. Ironmind est une plate-forme d'exécution puissante et flexible, capable d'héberger tout type de programme de formation. La rédaction de programme ne nécessite aucune connaissance en informatique, le langage Ironmind est extrêmement simple d'apprentissage et d'utilisation. Les programmes sont écrits sur un ordinateur ou directement sur mobile.*

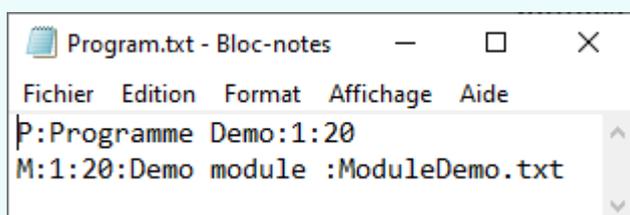
*Ironmind fonctionne uniquement sur mobile Android, et est téléchargeable sur Google Play Store. Il est recommandé d'utiliser la dernière version disponible.*

*Les programmes créés pour une utilisation personnelle nécessitent une licence Silver pour être joués sur Ironmind. Ils ne sont pas utilisables sur la version gratuite de l'application Ironmind, ni sur ceux disposant de la licence Bronze.*

*Les programmes destinés à la commercialisation nécessitent d'être signés numériquement pour être reconnus et joués sur la version gratuite Ironmind. La signature nécessite la licence Gold.*



# Principes de base pour écrire un programme Ironmind



Un programme Ironmind se compose de :

- Un fichier de description du programme (program.txt)
- Un ou plusieurs fichiers modules (ex : module1.txt, module2.txt...)

Les fichiers .txt sont des fichiers de texte, respectant la syntaxe décrite dans ce document. Sur PC, ils peuvent être écrits avec Notepad. Sur mobile, il existe de nombreux éditeurs de texte gratuits à télécharger, par exemple QuickEdit.

Un programme se définit par un titre et un nombre total de jours (entre 1 et n'importe quel nombre entier). Vous pouvez concevoir des programmes courts de quelques jours, ou de plus longs programmes de plusieurs mois.

La décomposition en modules permet de structurer le programme. Par exemple vous pouvez avoir un module d'introduction de 3 jours suivi de 3 modules de 10 jours, pour terminer par un module conclusion de 2 jours.

Un module se compose de 1 à 4 séquences, correspondant aux séances du Matin, Midi, Soir et Nuit.

Une séquence est une suite d'instructions à exécuter, chaque instruction étant écrite sur une ligne distincte et commençant par un mot-clé. Une instruction se compose de plusieurs champs séparés par le signe « : ».

Les lignes vides sont ignorées, les lignes commençant par le signe « # » sont considérées comme des commentaires, utilisables à des fins de documentation du programme.

## FICHER PROGRAMME

*Le fichier de description du programme doit obligatoirement porter le nom « program.txt ». C'est le point d'entrée du programme, là où tout commence. Ce fichier contient le titre du programme, et la liste des modules qui le composent.*

*Le fichier programme se compose des instructions suivantes, obligatoirement dans l'ordre indiqué ci-dessous. Les instructions entre crochets sont optionnelles.*

```
[E:liste des fichiers exclus du calcul de tag]
[S:Setup Module File]
P:Program Description:Nbr Modules:Nbre jours[:Min RunMode]
[M:Jour start:Jour end:Module description:Module]
```

*A l'exception de la commande M (qui peut être multiple), les autres commandes doivent être présentes au maximum une seule fois.*

### Commande E (Exclude)

*Optionnelle, une seule occurrence au maximum*

*Liste les fichiers exclus du calcul de signature, séparés par des virgules*

*Il s'agit des fichiers pouvant être modifiés par l'utilisateur du programme sans compromettre la vérification d'authenticité conditionnant l'exécution du programme.*

*Exemple :*

```
E:suggestions.txt,visualisation.ppsx,visualisation.pptx
```

### Commande S (Setup)

*Optionnelle, une seule occurrence au maximum*

*Référence un fichier contenant des commandes à exécuter une fois, à l'initialisation du programme.*

*Exemple :*

```
S:Setup.txt
```

*Le fichier contient des commandes de type « module », obligatoirement rassemblées dans une séquence n05.*

*Exemple :*

```
# Program setup
Séquence 5 Setup
F://0:Mise à jour de la licence...
# licence bronze relative pour une durée de 10 jours
VS://0:LicenceRun:R-10-00-00:1
```

### Commande P (Program)

*Obligatoire, une seule occurrence au maximum*

*Description synthétique du programme*

### **Syntaxe :**

P:Program Description:Nbr Modules:Nbre jours[:Min RunMode]

*Program Description* : description synthétique courte du programme

*Nbr Modules* : nombre total de modules composant le programme

*Nbre jours* : nombre total de jours

*Min RunMode (optionnel)* : indique le niveau minimum de licence requis pour exécuter le programme (0=free, 1=bronze, 2=silver, 3=gold)

## **Commande M (Module)**

*Optionnelle, multiples occurrences possibles*

*Référence d'un module à exécuter dans le programme*

### **Syntaxe :**

[M:Jour start:Jour end:Module description:Module File (relatif ou absolu) ]

*Start,end* : jours de début et de fin du module

*Description* : courte description textuelle du module

*Module File* : fichier module

### **Exemple :**

M:1:2:Introduction:Module1.txt

M:3:5:Médit-Affirm:Module2.txt

*Le module « Introduction », décrit dans le fichier Module1.txt, est à exécuter pendant les jours 1 à 2 du programme. Le second module, nommé « Médit-Affirm » est à exécuter les jours 3 à 5 du programme.*

*Note : les jours ne doivent pas nécessairement être consécutifs, un programme peut comporter des « jours blancs » où aucun module ne sera exécuté.*

## FICHER MODULE

Un fichier module liste les commandes à exécuter dans les différentes séquences possibles, qui sont au nombre maximum de 4 :

- 1 : Matin
- 2 : Midi
- 3 : Soir
- 4 : Nuit

Chaque séquence est optionnelle. Les séquences sont déclenchées automatiquement aux horaires configurés dans l'application, ou manuellement depuis l'écran principal.

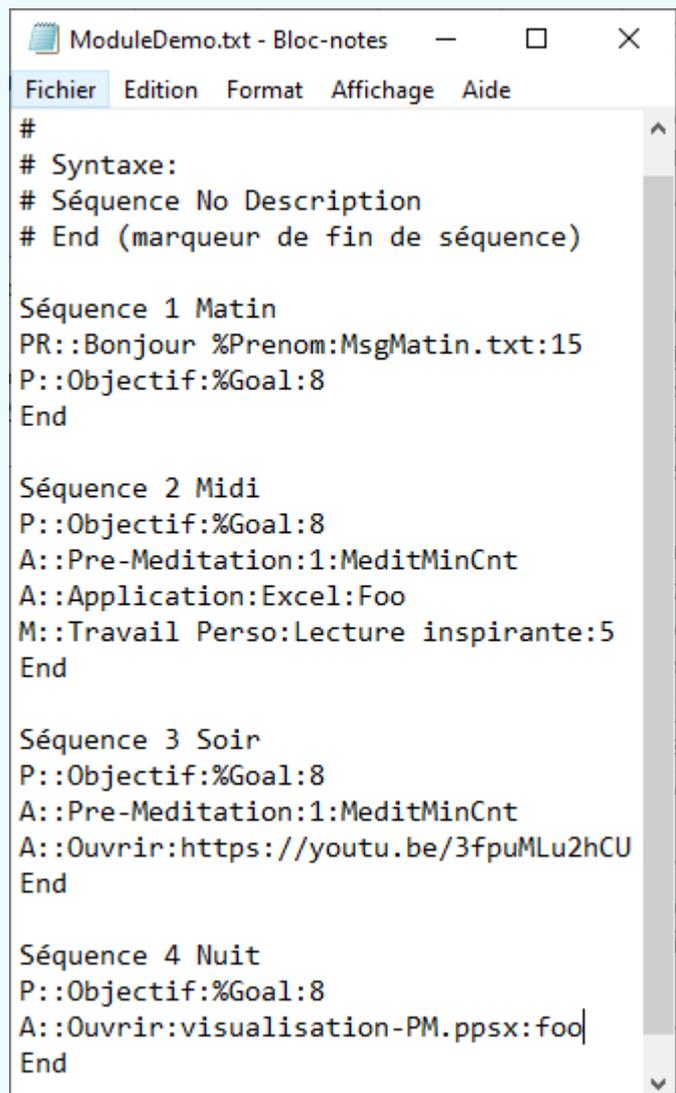
Une séquence possède la forme suivante :

```
Séquence No Description  
[liste de commandes (une par  
ligne)]  
End
```

*No* : de 1 à 4

*Description* : mot-clé affiché en début et en fin de séquence.

Les commandes sont exécutées dans l'ordre, l'une après l'autre.



```
ModuleDemo.txt - Bloc-notes
Fichier  Edition  Format  Affichage  Aide
#
# Syntaxe:
# Séquence No Description
# End (marqueur de fin de séquence)

Séquence 1 Matin
PR::Bonjour %Prenom:MsgMatin.txt:15
P::Objectif:%Goal:8
End

Séquence 2 Midi
P::Objectif:%Goal:8
A::Pre-Meditation:1:MeditMinCnt
A::Application:Excel:Foo
M::Travail Perso:Lecture inspirante:5
End

Séquence 3 Soir
P::Objectif:%Goal:8
A::Pre-Meditation:1:MeditMinCnt
A::Ouvrir:https://youtu.be/3fpuMLu2hCU
End

Séquence 4 Nuit
P::Objectif:%Goal:8
A::Ouvrir:visualisation-PM.ppsx:foo
End
```

# COMMANDES MODULE

## Résumé

*La syntaxe générale d'une commande est :*

`<mot-clé> [:<condition>] :<option1> :<option2> :<option3> :... :<optionN>`

## *Affichage de texte*

`P (popup) ::Titre:Message[:Temps-affichage]`

`PR (random-popup) ::Titre:Fichier[:Temps-affichage]`

`PC (cond-popup) ::VarName:Value:Titre:MessageYes:MessageNo[:Temps-affichage]`

`F (flash) ::Message`

`T (text) ::Titre:Fichier ou Varname[:MaxRandomIndex]`

## *Saisie de texte*

`VI (variable input) ::title:description:VarName[:timeout]`

`TI (large text variable input) ::title:VarName`

## *Actions complexes*

`A (action) ::Ouvrir:Fichier[:variable compteur de temps]`

`A (action) ::Tache:param1[:variable compteur de temps]`

`M (menu) ::Travail Perso`

## *Commandes internes*

`VS (variable set) ::VarName:Value`

`VSM (variable set with maths operation) ::VarName:Value`

`TV (lecture fichier => var) :VarName:Fichier[:MaxIndexRandom]`

`S (suggestions file read) ::Fichier`

## Condition

*Optionnel, s'applique à toutes les commandes*

*La commande est exécutée si la condition est vraie, est ignorée si elle est fausse*

*<cond> = DayStart-DayEnd/MaxRep!marker/MinRunMode-MaxRunMode/DeviceId*

*Tous les champs gauche-droit de "/" sont optionnels*

*DayStart-DayEnd (jour de début et jour de fin) : la condition est vraie si le jour actuel est situé (inclusivement) entre les jours de début et fin. Il est possible de spécifier uniquement le jour de début ou le jour de fin, ou un jour unique.*

*Exemples :*

*10-20 : condition vraie entre les jours 10 et 20 inclus*

*-10 : condition vraie entre les jours 1 et 10 inclus*

*10- : condition vraie depuis le jour 10 et jusqu'à la fin du programme*

*10 : condition vraie seulement le jour 10*

*MaxRep!marker : facteur maximal de répétition pour la journée courante  
Permet de limiter le nombre d'exécutions d'une commande quelle que soit la séquence.*

*Marker : indice du compteur permettant de mémoriser le nombre de répétitions  
La condition est vraie si le compteur de répétition est inférieur à MaxRep*

*Exemples :*

*/1!1 : commande exécutée une seule fois ce jour, indice no1*

*/2!5 : commande exécutée deux fois ce jour, indice no5*

*MinRunMode-MaxRunMode : la condition est vraie si le niveau de licence actuel est compris entre MinRunMode et MaxRunMode*

*Il est possible de spécifier uniquement MinRunMode ou MaxRunMode*

*Exemples :*

*//1-2 : condition vraie pour les licences 1 et 2*

*//-2 : condition vraie pour les licences 0, 1 et 2*

*//1 : condition vraie pour la licence 1 seulement*

*DeviceId : condition vraie seulement sur le mobile identifié par ce no de device*

## Commandes Px (PopUp)

*Affiche une fenêtre pop-up contenant du texte*

*Popup simple :*

`P::Titre:Message[:Temps-affichage]`

*Le texte indiqué comme Message est affiché dans la fenêtre identifiée par le titre Titre, et pendant la durée spécifiée comme Temps-affichage (10 secondes par défaut si non précisé)*

*Message peut être le texte lui-même, ou le nom d'une variable contenant le texte.*

*Exemples :*

`P::Bonne nuit:Je passe une bonne nuit paisible et réparatrice:20`

`P::Objectif:%Goal:8`

*(affiche le texte contenu dans la variable %Goal)*

*Popup aléatoire :*

`PR::Titre:Fichier[:Temps-affichage]`

*Affichage d'un message aléatoire lu à partir d'un fichier*

*Le fichier doit contenir comme première ligne le nombre total de lignes*

*Chaque ligne constitue ensuite un message distinct*

*A chaque exécution, un nombre aléatoire est tiré entre 1 et le nombre total*

*La ligne correspondant au numéro tiré sera lue et affichée dans la fenêtre*

*Popup conditionnel :*

`PC::VarName:Value:Titre:MessageYes:MessageNo[:Temps-affichage]`

*Affichage conditionnel d'un message*

*La variable de nom VarName est évaluée. Si sa valeur est égale à Value, le*

*message contenu dans le champ MessageYes est affiché. Dans le cas contraire c'est MessageNo qui est affiché.*

## Commande F (Flash)

`F(flash)::Message`

*Affiche un message court sous forme de bulle fugitive en partie basse de l'écran*

## Commande T (Text)

*Affiche une fenêtre de texte (pour un texte long, avec une barre de défilement verticale)*

`T::Titre:Fichier[:MaxIndexRandom]`

`T::Titre:VarName[:MaxIndexRandom]`

*Fichier* : le texte est lu à partir du fichier spécifié (doit comporter une extension .txt). Le nom de fichier peut intégrer des noms de variables à évaluer.  
*VarName* : le texte est lu directement à partir de la variable spécifiée  
La variable peut comporter des sous-variables  
Le nom de variable %random indique qu'un nombre aléatoire sera tiré, compris entre 1 et MaxIndexRandom

*Exemples :*

T::Capsule du jour:capsule%ModDay.txt

*Affiche le texte contenu dans le fichier capsuleN.txt, où N est le numéro du jour courant dans le programme*

T::Capsule du jour:capsule%random.txt:10

*Affiche le texte contenu dans le fichier capsuleN.txt, où N est un nombre aléatoire tiré entre 1 et 10*

T::Leçon:Lecon%random:10

*Affiche le texte contenu dans la variable %LeconN, où N est un nombre aléatoire tiré entre 1 et 10*

## Commande VI (Variable Input)

*Saisie d'un texte court par l'utilisateur, stocké dans une variable*

VI(variable input)::title:description:VarName[:timeout]

*Title* : titre de la fenêtre de saisie

*Description* : indications fournies à l'utilisateur

*VarName* : nom de la variable dans laquelle le texte sera stocké

*Timeout* : délai maximum autorisé pour la saisie (en secondes), le défaut est de 20 secondes s'il n'est pas spécifié

*Exemple :*

VI::Objectif:Entrez votre objectif ci-dessous :Goal:60

*Demande à l'utilisateur de saisir son objectif (texte court), qui sera stocké en variable %Goal*

## Commande TI (Text Input)

*Saisie d'un texte long par l'utilisateur, stocké dans une variable*

TI(large text variable input)::title:VarName

*Title* : titre de la fenêtre de saisie

*VarName* : nom de la variable dans laquelle le texte sera stocké

*Exemple :*

TI::Mission de Vie:Mission

*Demande à l'utilisateur de saisir sa mission de vie (texte long), qui sera stockée en variable %Mission*

## Commande A (Action)

Déclenche une action complexe, qui peut être l'ouverture d'un fichier ou d'une URL (action « Ouvrir ») ou le déclenchement d'une tâche intégrée Ironmind (action « Tache »).

*Ouverture d'un fichier ou d'une URL :*

```
A(action)::Ouvrir:Fichier[:variable compteur de temps]
```

*Déclenche l'ouverture d'un fichier ou d'une URL.*

*Fichier : nom du fichier (avec extension) ou URL Web (commence par http)*

*Variable compteur de temps (optionnel): enregistre le temps passé à visualiser le fichier ou l'URL (en cumulé) dans la variable spécifiée (en minutes)*

*Exemples :*

```
A::Ouvrir:visualisation-AM.ppsx:VisuMinCnt
```

*Le fichier de présentation PowerPoint « visualisation-AP.ppsx » est ouvert (avec l'application associée Microsoft PowerPoint mobile), le temps passé à visualiser la présentation est ajouté à la variable %VisuMinCnt*

```
A::Ouvrir:https://youtu.be/1Y3ppTWB_dk
```

*Ouvre le lien spécifié dans l'application intégrée Youtube*

*Démarrage d'une tâche intégrée Ironmind:*

```
A(action)::Tache:param1[:variable compteur de temps]
```

*Tache : nom de la tâche à déclencher. Les tâches suivantes sont supportées :*

- *Pre-Meditation : affichage du menu permettant le démarrage d'une session de méditation. Le paramètre param1 est ignoré.*
- *Pre-Affirmations : affichage du menu Oui/Non pour le démarrage d'une session d'affirmations positives. Le paramètre param1 contient le nombre d'affirmations à afficher.*
- *Pre-Sommeil : affichage du menu Oui/Non pour le démarrage de la musique de méditation destinée à l'endormissement progressif. Le paramètre param1 contient la durée en minutes pendant laquelle la musique sera jouée.*

*Variable compteur de temps (optionnel): enregistre le temps passé à effectuer l'action.*

## Commande M (Menu)

*Affichage d'un menu demandant un choix de l'utilisateur*

```
M(menu)::nom menu
```

*Nom menu :*

- *Travail Perso* : menu « travail personnel » qui propose les applications eBook installées sur le mobile (Kindle, Aldiko)

## Commandes Vx (Variables)

*Affectation d'une valeur à une variable*

```
VS(variable set)::VarName:Value
```

*VarName* : nom de la variable à affecter. Les variables globales doivent obligatoirement commencer par « *Usr* » (ex : *UsrIndicateur*).

*Value* : valeur, peut aussi contenir des noms de variables précédés du signe %

*Exemple :*

```
VS::SugMedEn:off
```

*Affecte la chaîne de caractères « off » à la variable %SugMedEn*

```
VSM(variable set with maths operation)::VarName:Value
```

*Effet identique à la commande VS, sauf que Value peut comprendre une opération mathématique à évaluer*

*Exemple :*

```
VS::compteur:%compteur+1
```

*Incrémente de 1 la variable %compteur*

## Commande TV (Text to Variable)

*Lecture d'un fichier texte dans une variable*

```
TV(lecture fichier ds var):VarName:Fichier[:MaxIndexRandom]
```

*VarName* : nom de la variable à affecter

*Fichier* : nom du fichier à lire, peut comprendre des noms de variables (précédés par %), ainsi que la variable aléatoire %random (auquel cas le paramètre supplémentaire *MaxIndexRandom* est requis)

## Commande S (Suggestions)

*Lecture du fichier des suggestions/affirmations*

```
S(suggestions file read)::Fichier
```

*Fichier* : nom du fichier à lire pour initialiser le tableau des suggestions

*Le fichier doit comprendre une suggestion par ligne, chaque suggestion se terminant par un « . ».*

*Exemple :*

```
S::suggestions.txt
```

*Initialise les suggestions à partir du contenu du fichier « suggestions.txt »*

## UTILISATION DES VARIABLES

Les variables permettent de stocker des informations de diverse nature (texte ou numérique). Elles sont de deux types :

- *Fugitives* : nom comprenant uniquement des minuscules. Elles sont définies uniquement pendant une seule séquence, disparaissent en fin de séquence.
- *Permanent*es : nom comprenant au moins un caractère majuscule. Elles sont « globales », c'est-à-dire qu'elles gardent leur valeur entre différentes séquences et différents jours du programme.

Les variables suivantes « système » sont disponibles :

- *%ModDay* : indice du jour à l'intérieur du module courant (commence à 1 le premier jour du module)
- *%AffMinCnt* : total du nombre de minutes passé en affirmations
- *%VisuMinCnt* : total du nombre de minutes passé en visualisation
- *%MeditMedCnt* : total du nombre de minutes passé en méditation

Note : par convention, *VarName* (sans %) désigne le nom de la variable alors que *%VarName* désigne le contenu de la variable.

## INSTALLATION DU PROGRAMME

Une fois terminé, le programme doit être préparé afin de pouvoir être reconnu et installé sous Ironmind.

*Pré-requis* : l'ensemble des fichiers constituant le programme doivent être rassemblés dans un répertoire (dossier) distinct de nom « *NomProg.irm* », où *NomProg* est un nom abrégé du programme à choisir.

La procédure d'installation est la suivante :

1. Création d'un fichier « archive » (zip) contenant tous les fichiers du programme. Le fichier archive s'appellera *NomProg.irm.zip*.
2. Téléchargement (upload) de l'archive programme sur le mobile, dans le répertoire « download ». Il est possible d'utiliser Google Drive pour cela.
3. Ouvrir Ironmind et sélectionner le bouton « mon coach » puis le bouton « Activer ». Le nom de votre nouveau programme doit figurer dans la liste.
4. Choisir le nouveau programme dans la liste, il sera alors initialisé et prêt à être exécuté.

*Rappel* : un programme personnel non signé ne peut être activé que sur une application Ironmind disposant de la licence Silver ou Gold.

Pour supprimer définitivement un programme sous Ironmind, il est nécessaire d'effacer le sous-répertoire programme *NomProg.irm* sur le mobile, en dessous du répertoire download. Il faut aussi supprimer le fichier *NomProg.irm.zip* dans download.

**Bonne Utilisation !**