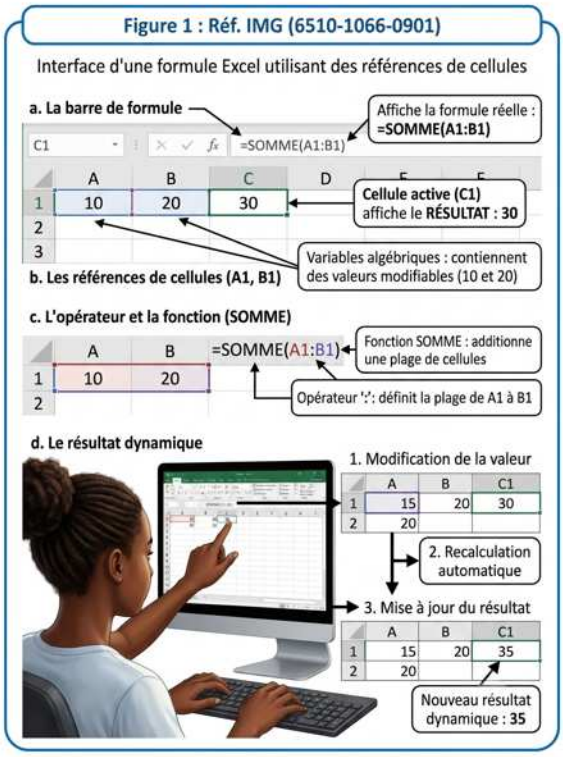


Figure 1 — Interface d'une formule Excel utilisant des références de cellules

Cet espace, souvent noté `fx`, affiche la formule mathématique réelle contenue dans la cellule active (C1). La cellule, elle, affiche uniquement le résultat final du calcul (30), pas la logique derrière. Les cellules A1 et B1 agissent comme des variables algébriques. Elles contiennent des valeurs (10 et 20) qui peuvent être modifiées. Le tableur utilise leurs adresses pour effectuer le

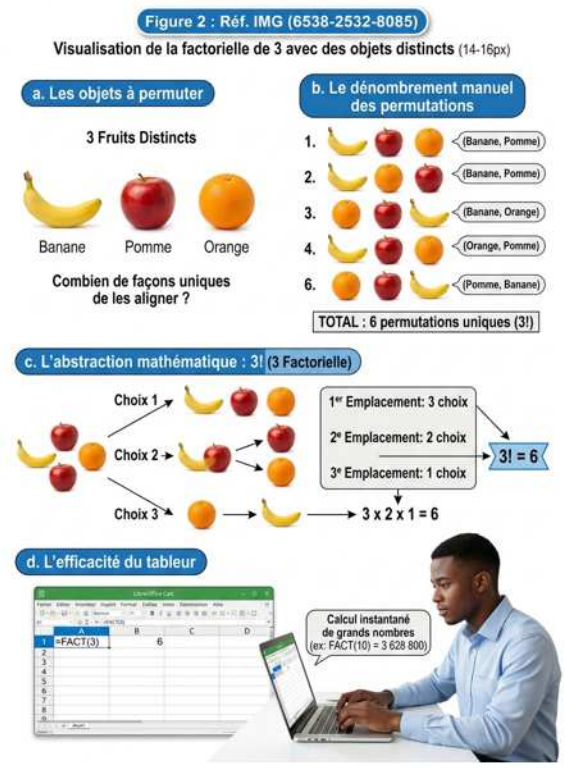


Figure 2 — Visualisation de la factorielle de 3 avec des objets distincts

L'image montre trois fruits différents : une banane, une pomme et une orange. L'objectif est de trouver toutes les manières uniques de les aligner. Chaque arrangement est une permutation. L'analyse visuelle de ces arrangements prépare à la compréhension du calcul factoriel. En listant toutes les possibilités, on trouve six combinaisons : (Banane, Pomme, Orange), (Banane, Orange,

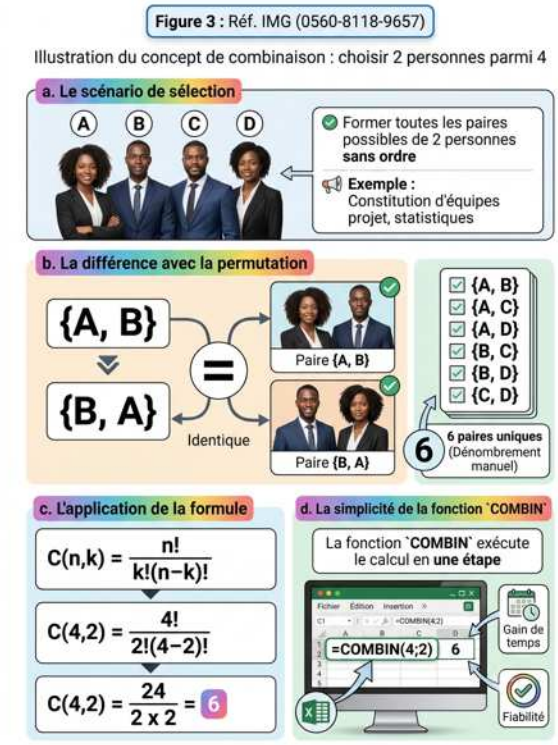


Figure 3 — Illustration du concept de combinaison : choisir 2 personnes parmi 4

L'illustration présente un groupe de quatre individus (A, B, C, D). La tâche est de former toutes les paires possibles (groupes de 2) sans se soucier de l'ordre de sélection. Ceci est un problème de combinaison classique, fréquent en gestion de projet ou en statistiques. Contrairement à la permutation, l'ordre n'a pas d'importance ici. Le duo formé par A et B est exactement le même que

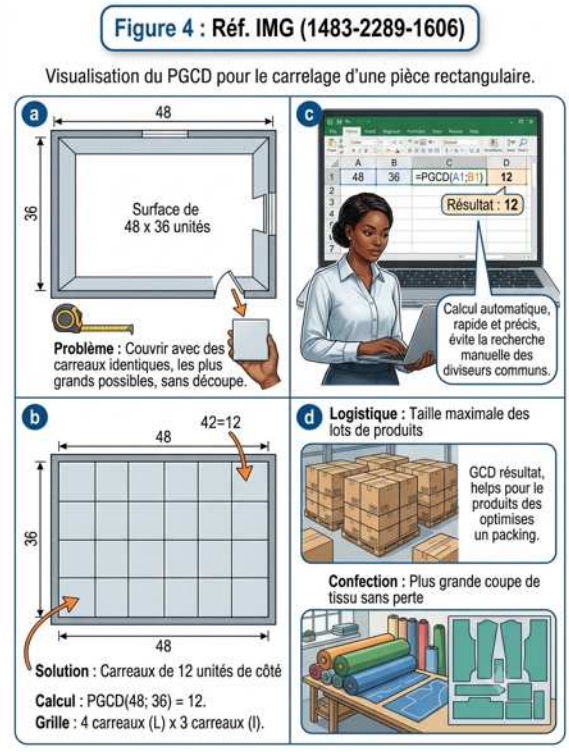


Figure 4 — Visualisation du PGCD pour le carrelage d'une pièce rectangulaire.

Une surface de 48 unités par 36 unités doit être couverte par des carrés identiques, les plus grands possibles. La taille du côté de ces carrés correspond au PGCD des dimensions de la surface. Ceci évite toute découpe et optimise l'utilisation du matériau. Le calcul `=PGCD(48; 36)` donne `12`. La solution est donc d'utiliser des carreaux de 12 unités de côté. Ces carreaux s'ajusteront

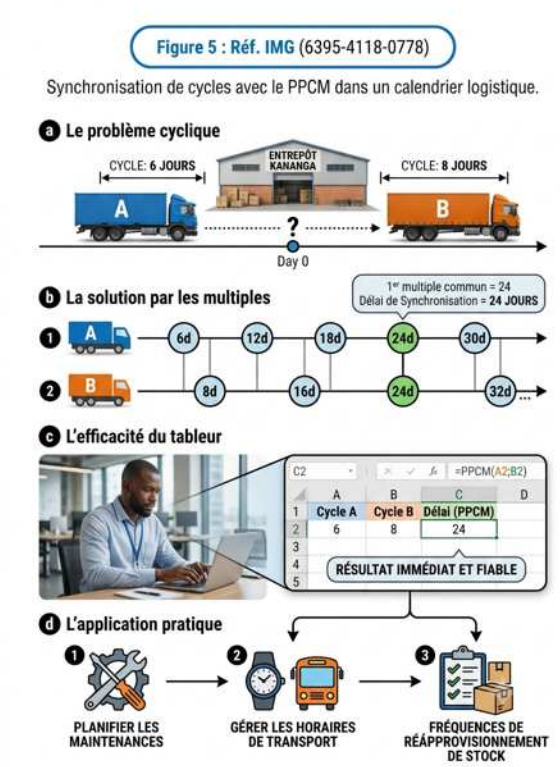


Figure 5 — Synchronisation de cycles avec le PPCM dans un calendrier logistique.

Deux camions partent d'un entrepôt à Kananga. L'un effectue sa tournée en 6 jours, l'autre en 8 jours. S'ils partent le même jour, quand repartiront-ils de nouveau ensemble ? Ce point de rencontre temporel correspond au PPCM de leurs cycles. Les départs du premier camion se font aux jours 6, 12, 18, 24, 30... Les départs du second camion se font aux jours 8, 16, 24, 32... Le premier multiple

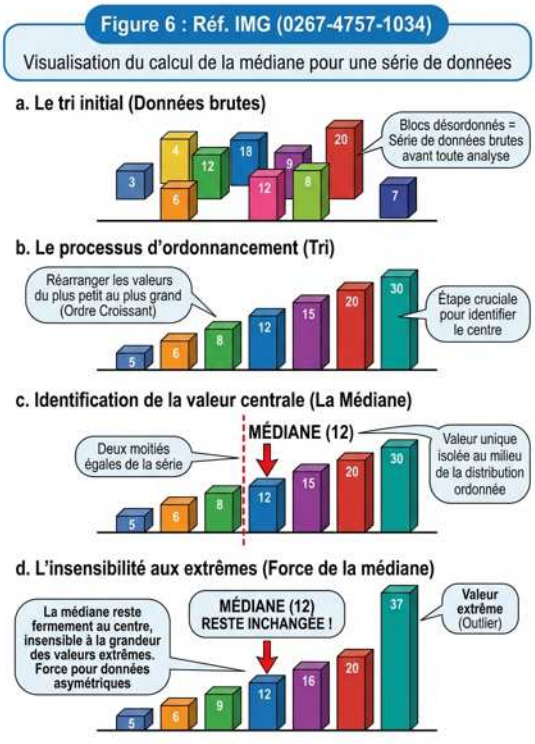


Figure 6 — Visualisation du calcul de la médiane pour une série de données

L'image montre des blocs de hauteurs différentes, symbolisant des valeurs numériques. Ils sont d'abord désordonnés, représentant une série de données brutes avant toute analyse. C'est le point de départ de nombreuses analyses statistiques. La première étape cruciale pour trouver la médiane est de trier ces valeurs. L'illustration montre les blocs réarrangés du plus petit au plus grand. Cette

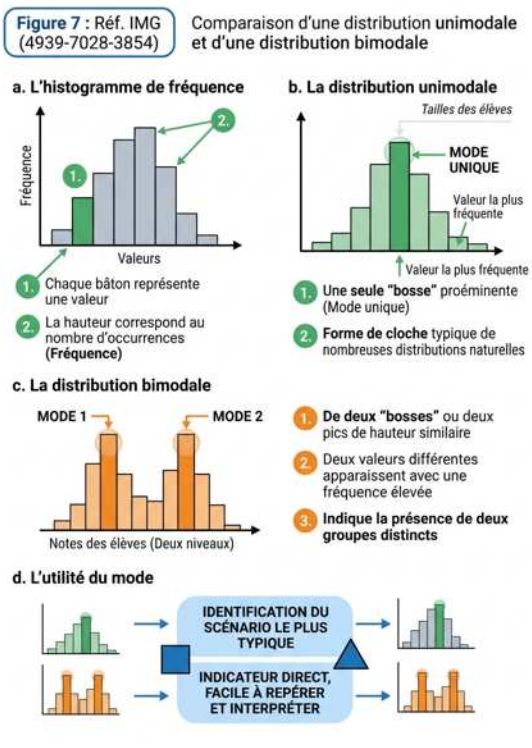


Figure 7 — Comparaison d'une distribution unimodale et d'une distribution bimodale

L'image présente deux graphiques en bâtons, ou histogrammes. Chaque bâton représente une valeur et sa hauteur correspond au nombre de fois où cette valeur apparaît dans la série. C'est une représentation visuelle directe de la fréquence. Le premier graphique montre une seule "bosse" proéminente. Le bâton le plus haut correspond à une valeur unique qui est beaucoup plus fréquente

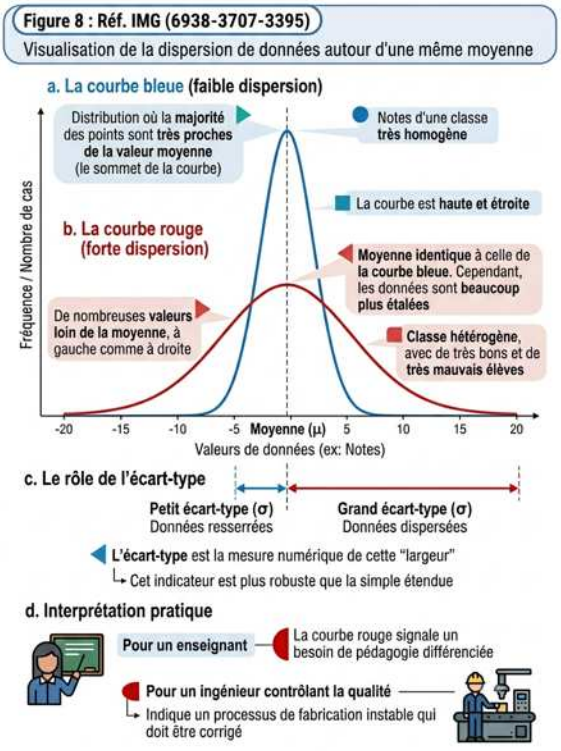


Figure 8 — Visualisation de la dispersion de données autour d'une même moyenne

Ce graphique montre une distribution où la majorité des points de données sont très proches de la valeur moyenne (le sommet de la courbe). Les notes d'une classe très homogène suivraient cette forme. La courbe est haute et étroite. Ici, la moyenne est identique à celle de la courbe bleue. Cependant, les données sont beaucoup plus étalées. On trouve de nombreuses valeurs loin de la

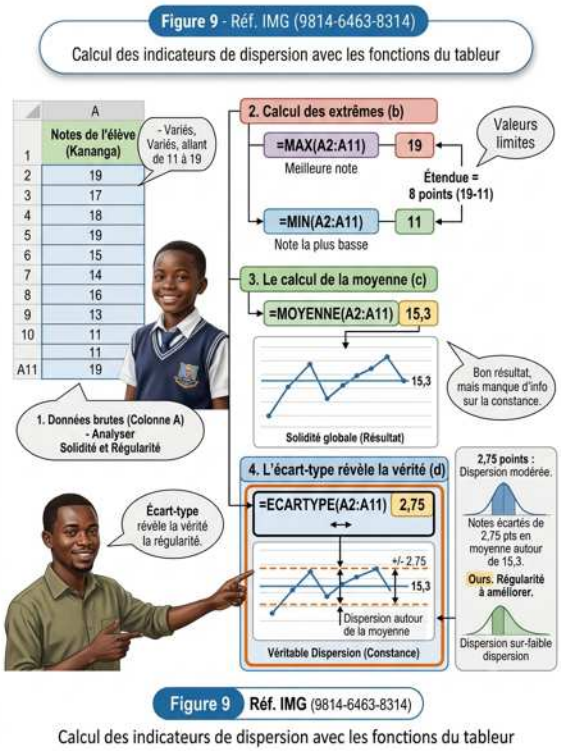


Figure 9 — Calcul des indicateurs de dispersion avec les fonctions du tableur

La colonne A contient les notes d'un élève de Kananga sur dix interrogations. Les valeurs sont variées, allant de 11 à 19. L'objectif est d'analyser non seulement sa moyenne, mais aussi la régularité de ses performances. La cellule D2 utilise la fonction `=MAX(A2:A11)` pour trouver la meilleure note (19). La cellule D3 utilise `=MIN(A2:A11)` pour trouver la note la plus basse (11).

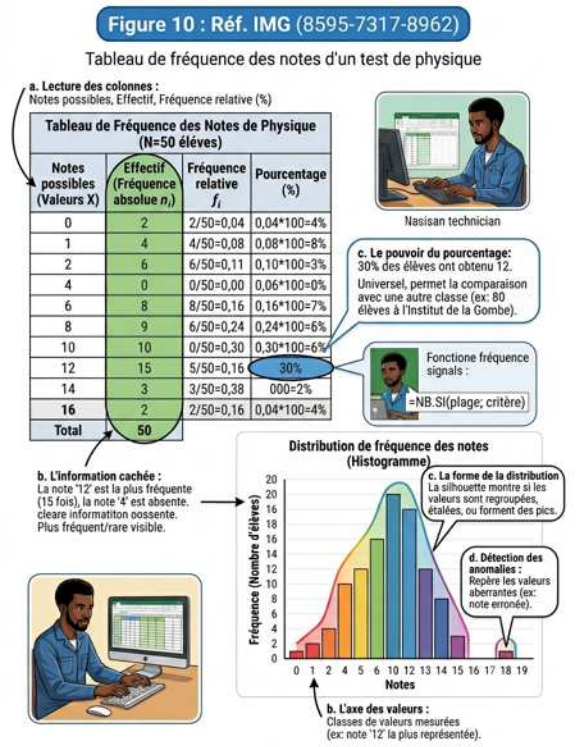


Figure 10 — Tableau de fréquence des notes d'un test de physique

Ce tableau organise les données d'un test. La première colonne liste les notes possibles (les "valeurs"). La deuxième, "Effectif", indique la fréquence absolue : combien d'élèves ont eu cette note. La dernière colonne calcule la fréquence relative en pourcentage. Observer la colonne "Effectif" révèle immédiatement les notes les plus courantes et les plus rares. Ici, la note "12"

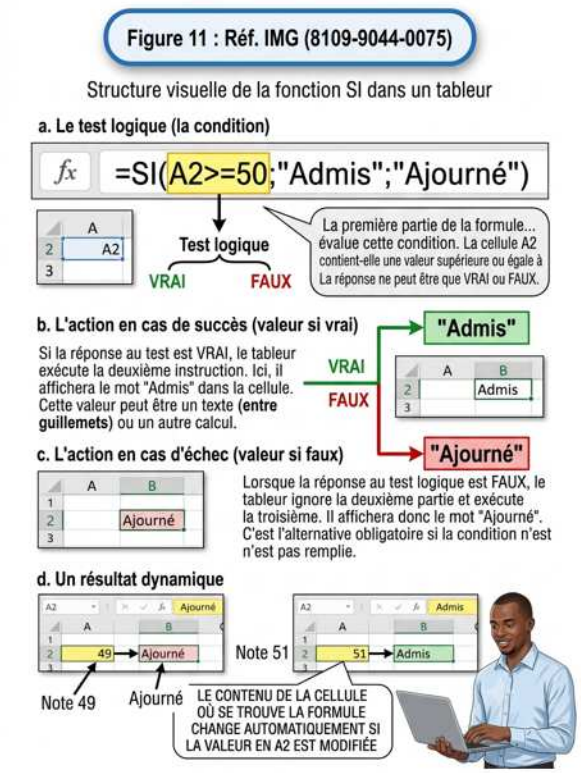


Figure 11 — Structure visuelle de la fonction SI dans un tableur

La première partie de la formule, ici `A2>=50`, est une question. Le tableur évalue cette condition. La cellule A2 contient-elle une valeur supérieure ou égale à 50 ? La réponse ne peut être que VRAI ou FAUX. Si la réponse au test est VRAI, le tableur exécute la deuxième instruction. Ici, il affichera le mot "Admis" dans la cellule. Cette valeur peut être un texte (entre guillemets) ou un

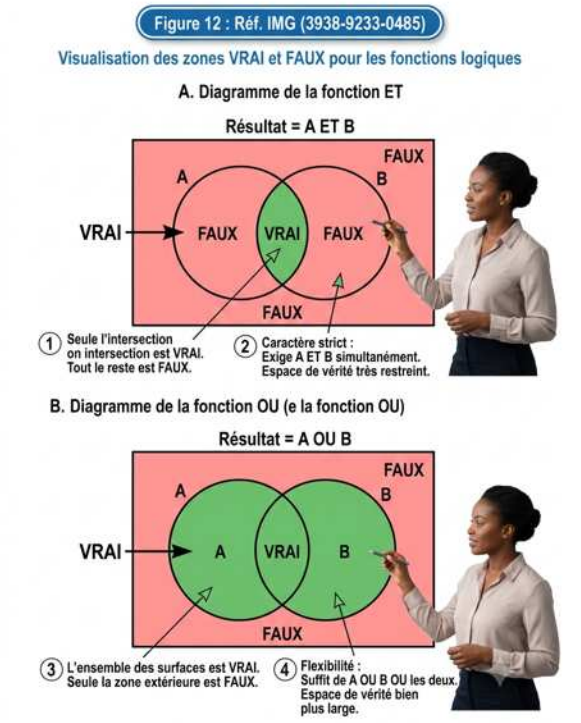


Figure 12 — Visualisation des zones VRAI et FAUX pour les fonctions logiques

L'illustration montre deux cercles qui se croisent. Pour la fonction `ET`, seule la petite zone de superposition, où les deux conditions sont remplies simultanément, correspond à un résultat `VRAI`. Tout ce qui est en dehors de cette intersection est `FAUX`. Cette représentation visuelle confirme le caractère strict de la fonction `ET`. Elle exige que l'on se trouve dans le cercle A `ET` dans le

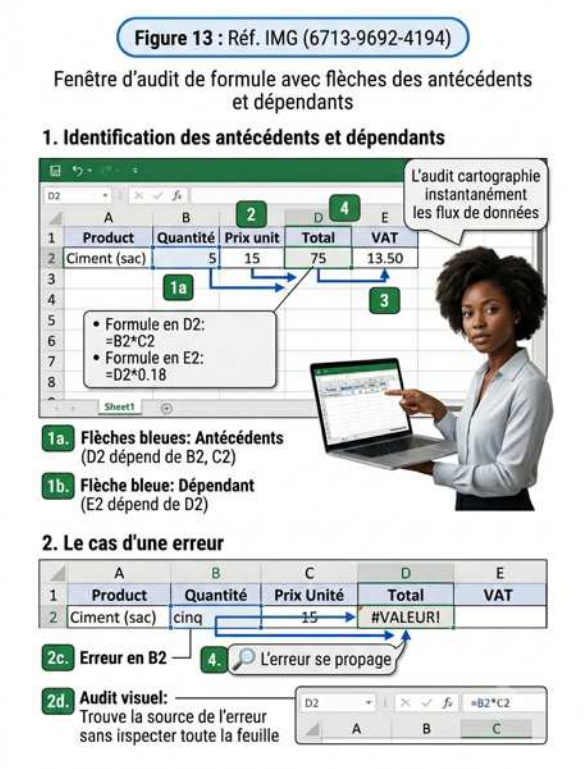


Figure 13 — Fenêtre d'audit de formule avec flèches des antécédents et dépendants.

Des flèches bleues partent des cellules B2 (Quantité) et C2 (Prix) pour pointer vers la cellule D2 (Total). Ces flèches matérialisent les antécédents. Elles montrent que le calcul en D2 dépend directement des valeurs présentes en B2 et C2. Une flèche part de D2 vers la cellule E2 (TVA). Cela signifie que la cellule E2 est une dépendante. Son propre calcul utilise le résultat de la formule

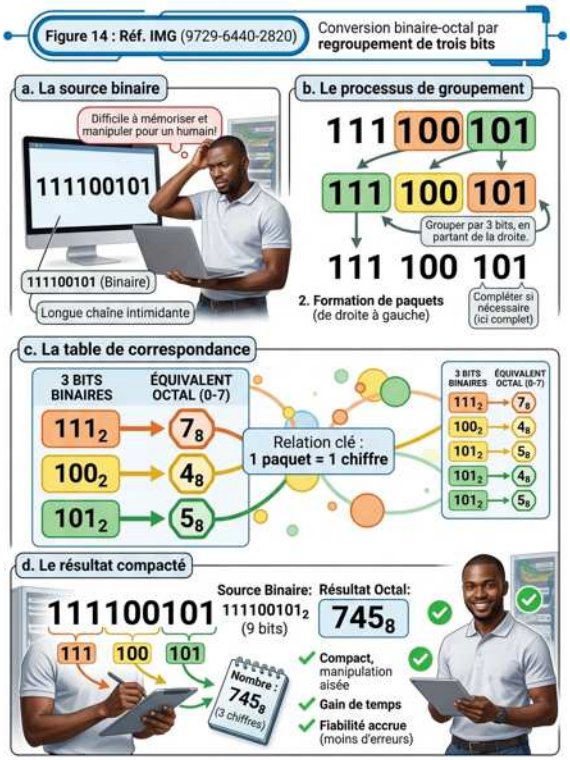


Figure 14 — Conversion binaire-octal par regroupement de trois bits

Le point de départ est une longue chaîne de bits. Ici, `111100101`. Elle est difficile à lire et à mémoriser pour un opérateur humain, par exemple un technicien réseau à Mbuji-Mayi qui configure des droits d'accès. En partant de la droite, on forme des paquets de trois bits. Si le dernier paquet à gauche est incomplet, on le complète avec des zéros. `111 100 101` se groupe naturellement ici. Ce

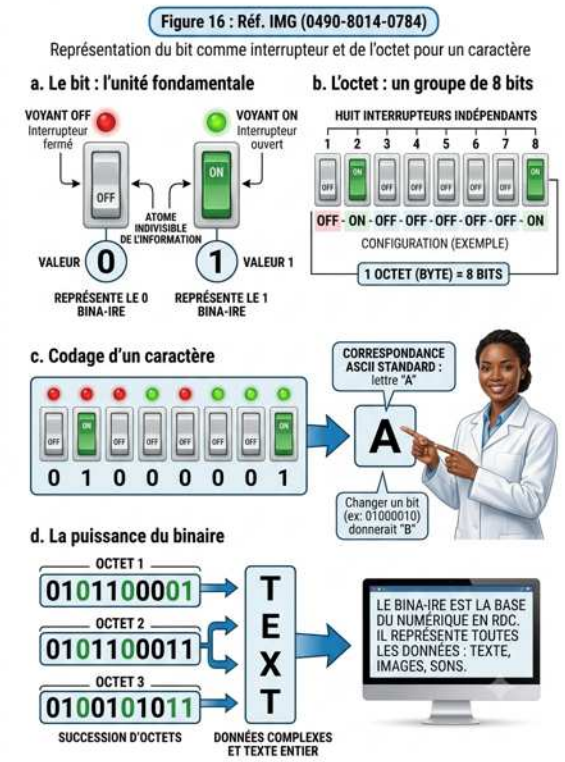


Figure 16 — Représentation du bit comme interrupteur et de l'octet pour un caractère

L'image illustre un bit comme un simple interrupteur. En position "ON", il représente la valeur 1. En position "OFF", il représente la valeur 0. Cette simplicité est la base de toute la complexité du monde numérique. C'est l'atome indivisible de l'information. L'illustration montre ensuite une rangée de 8 interrupteurs. Cet ensemble forme un octet. Chaque interrupteur peut être ON ou OFF

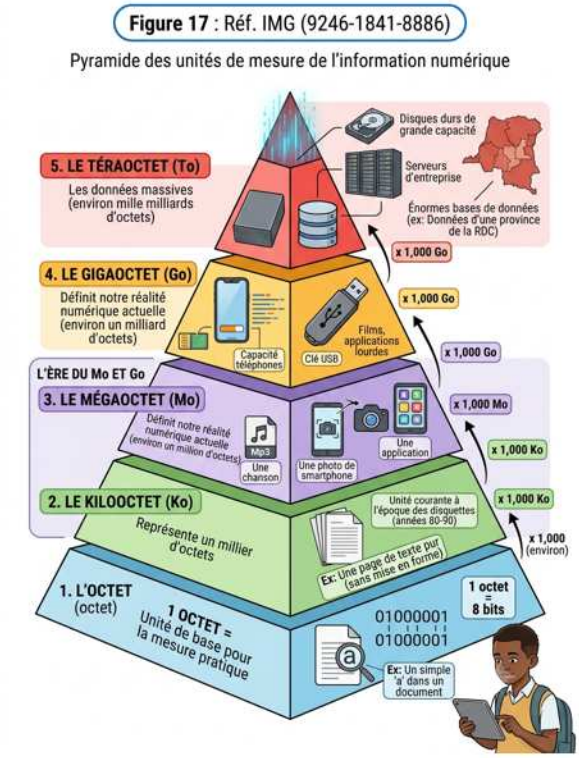


Figure 17 — Pyramide des unités de mesure de l'information numérique

Au bas de la pyramide se trouve l'octet. C'est l'unité de base pour la mesure pratique, capable de stocker un seul caractère. Tout ce qui est plus grand est construit sur cette fondation. Un simple "a" dans un document pèse un octet. Le premier étage est le Kilooctet. Il représente un millier d'octets. Une page de texte pur, sans mise en forme, pèse quelques Kilooctets. C'était une unité de

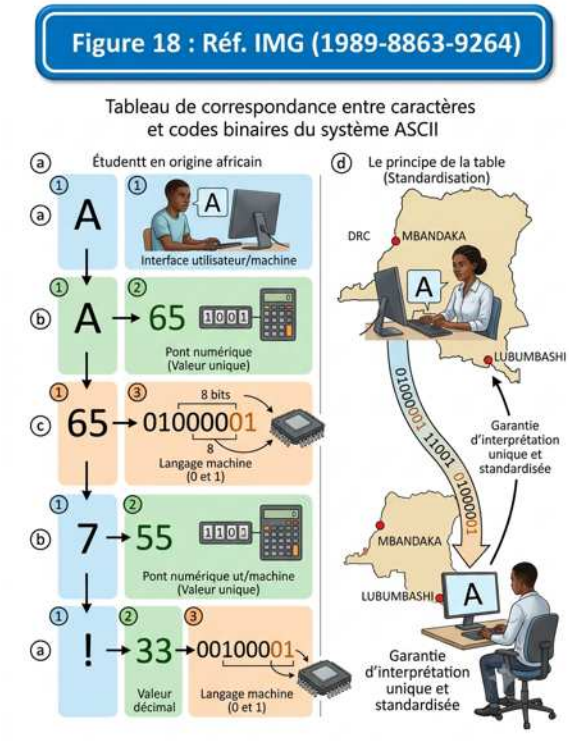


Figure 18 — Tableau de correspondance entre caractères et codes binaires du système

La colonne de gauche affiche des symboles que l'humain lit et écrit : la lettre 'A', le chiffre '7', le symbole '!'. Ils sont l'interface entre l'utilisateur et la machine. La colonne du milieu attribue une valeur numérique unique à chaque caractère. 'A' vaut 65, '7' vaut 55, et '!' vaut 33. Ce nombre sert de pont entre le symbole et son code binaire. La colonne de droite montre la

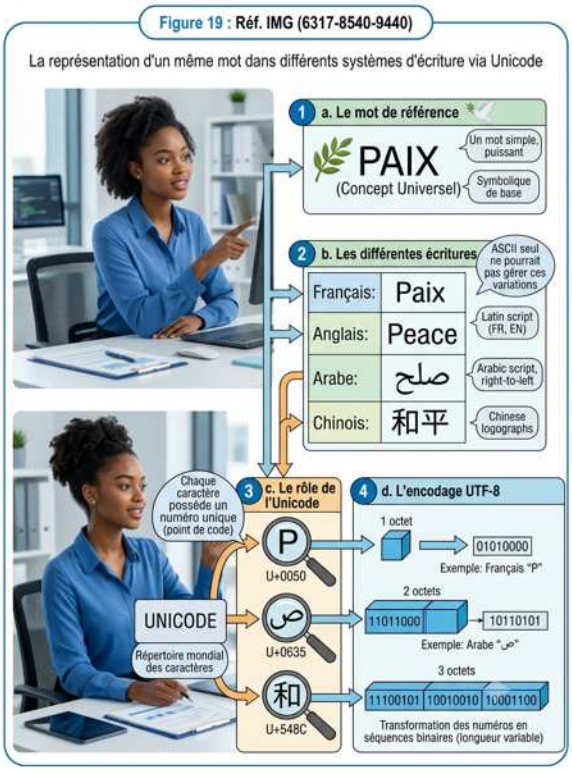


Figure 19 — La représentation d'un même mot dans différents systèmes d'écriture via

Le concept universel de "Paix" est ici utilisé comme exemple. Ce mot, simple en apparence, possède des représentations graphiques radicalement différentes selon les cultures et les langues. Le tableau montre "Paix" (Français), "Peace" (Anglais), "صلح" (Arabe), et "□□" (Chinois). Chaque système d'écriture possède ses propres symboles, formes et règles de composition. ASCII seul ne

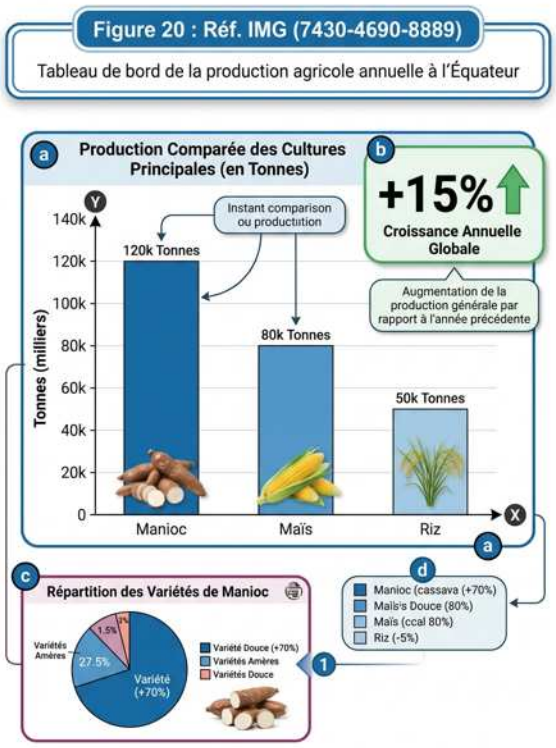


Figure 20 — Tableau de bord de la production agricole annuelle à l'Équateur

Le visuel central est un diagramme en barres. Il compare la production de manioc, maïs et riz en tonnes. L'œil identifie immédiatement que le manioc est la culture dominante, sa barre étant la plus haute. La comparaison entre les cultures est instantanée et ne demande aucun calcul. En haut à droite, un grand chiffre (ex: +15%) peut indiquer la croissance annuelle. Cet élément, un indicateur

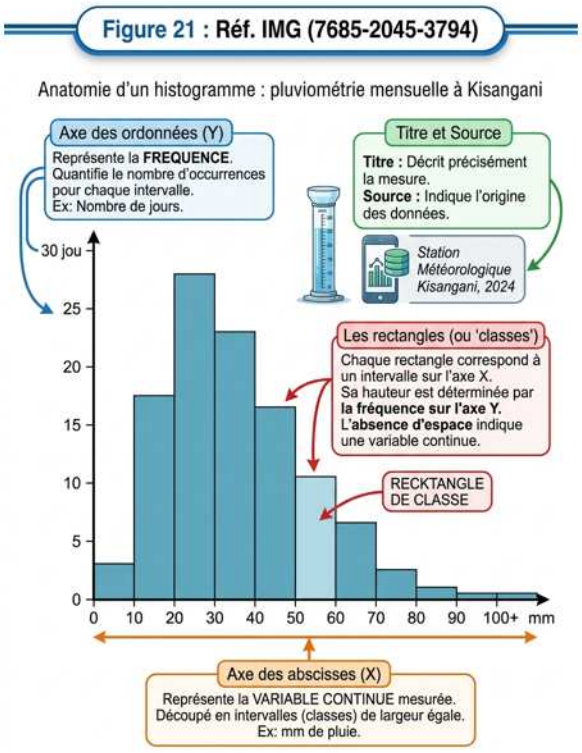


Figure 21 — Anatomie d'un histogramme : pluviométrie mensuelle à Kisangani

Cet axe vertical représente la fréquence. Il quantifie le nombre d'occurrences pour chaque intervalle. Par exemple, il peut montrer le nombre de jours où la pluviométrie était comprise entre 10 et 20 mm. Cet axe horizontal représente la variable continue mesurée. Il est gradué et découpé en intervalles (classes) de largeur égale. Les valeurs affichées sont les bornes de ces intervalles,

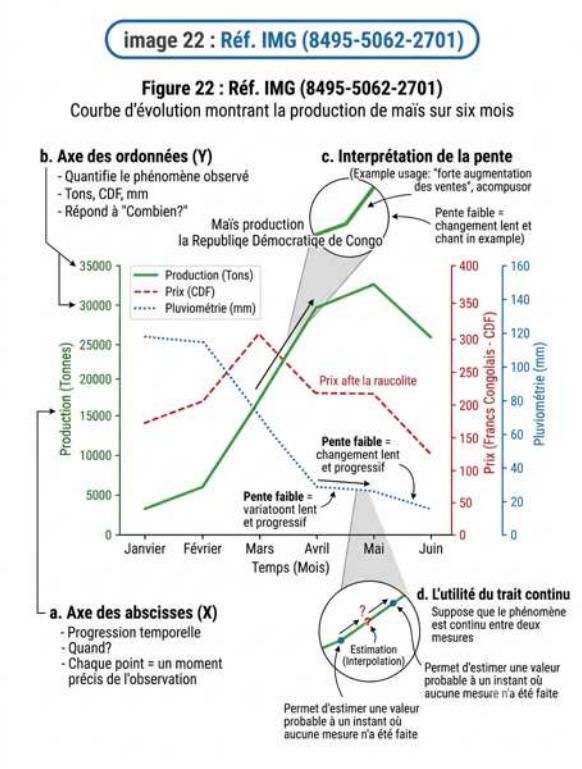


Figure 22 — Courbe d'évolution montrant la production de maïs sur six mois

Cet axe horizontal représente une progression temporelle. Ici, les mois de janvier à juin. Il donne le "quand". Chaque point sur cet axe est un moment précis de l'observation. Cet axe vertical quantifie le phénomène observé. Il peut s'agir de tonnes de maïs, du prix en Francs Congolais ou de la pluviométrie en millimètres. Il répond à la question "combien ?". La pente de la ligne entre deux

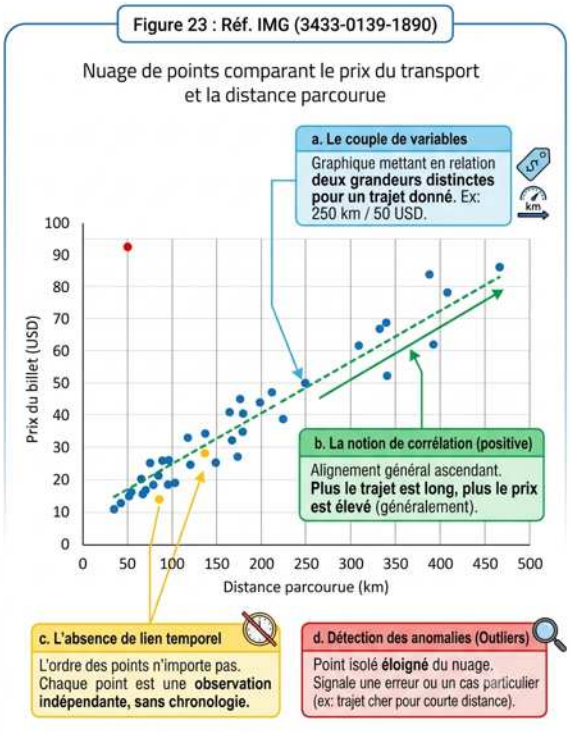


Figure 23 — Nuage de points comparant le prix du transport et la distance parcourue

Le graphique met en relation deux grandeurs distinctes. Par exemple, la distance d'un trajet en kilomètres (axe X) et le prix du billet en dollars (axe Y). Il n'y a pas de notion de temps ici. L'alignement général des points révèle la nature du lien. Si les points s'organisent globalement le long d'une droite imaginaire ascendante, la corrélation est positive. Plus le trajet est long, plus

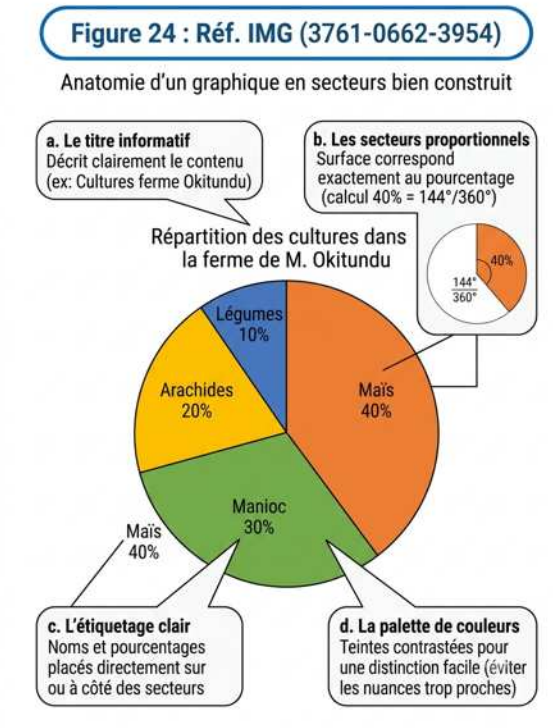


Figure 24 — Anatomie d'un graphique en secteurs bien construit

Un bon graphique possède un titre qui décrit clairement ce qu'il représente. Par exemple : "Répartition des cultures dans la ferme de M. Okitundu". Sans titre, le graphique perd tout son sens et devient une simple figure colorée sans information exploitable. La surface de chaque part du "gâteau" doit correspondre exactement à son pourcentage. Un secteur de 50% doit occuper la moitié du

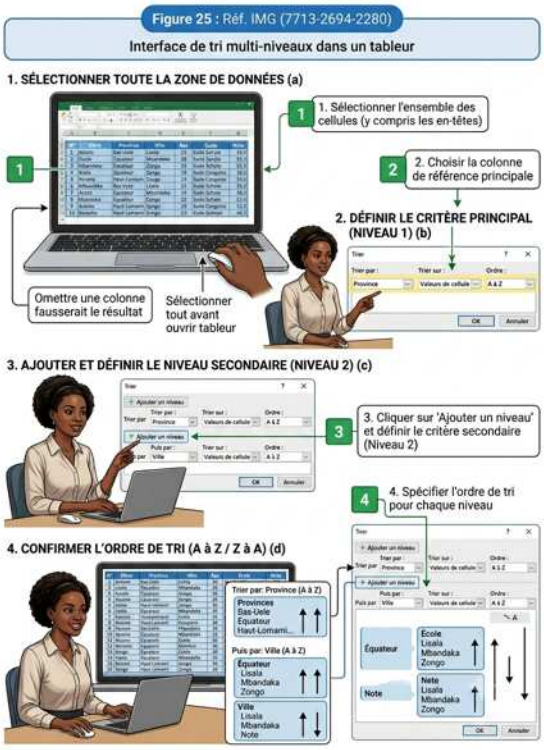


Figure 25 — Interface de tri multi-niveaux dans un tableur

Avant toute chose, l'utilisateur sélectionne l'ensemble des cellules contenant les données à trier, y compris les en-têtes de colonnes. Omettre une colonne ou une ligne fausserait complètement le résultat, car les informations ne correspondraient plus les unes aux autres. Dans la fenêtre de dialogue, la première ligne définit le tri principal. L'utilisateur choisit la colonne de référence

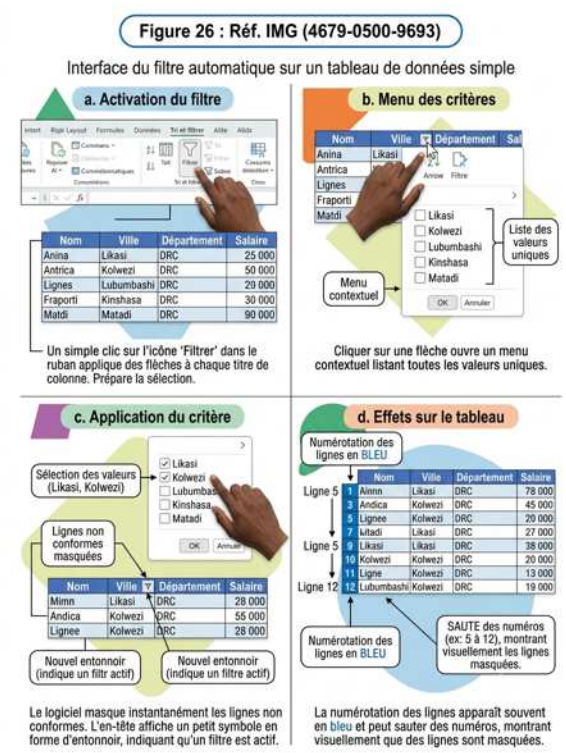


Figure 26 — Interface du filtre automatique sur un tableau de données simple

Un simple clic sur l'icône "Filtrer" dans le ruban du logiciel applique des flèches à chaque titre de colonne. Cette action prépare le tableau pour la sélection. Le tableau original reste intact, mais il est maintenant interactif. Cliquer sur une flèche ouvre un menu contextuel. Ce menu liste toutes les valeurs uniques de la colonne. L'utilisateur peut sélectionner une ou plusieurs de ces

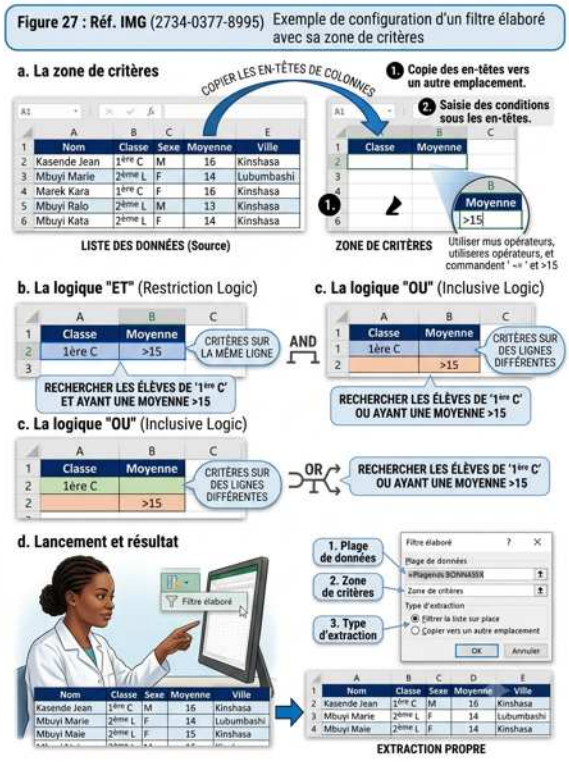


Figure 27 — Exemple de configuration d'un filtre élaboré avec sa zone de critères

L'utilisateur doit d'abord copier les en-têtes des colonnes à filtrer dans une autre partie de la feuille. Sous ces en-têtes, il écrit les conditions. Par exemple, sous l'en-tête "Moyenne", on peut écrire `>15`. Pour une condition restrictive, les critères sont saisis sur la même ligne. Pour trouver les élèves de "1ère C" avec une moyenne supérieure à 15, on place "1ère C" sous la colonne

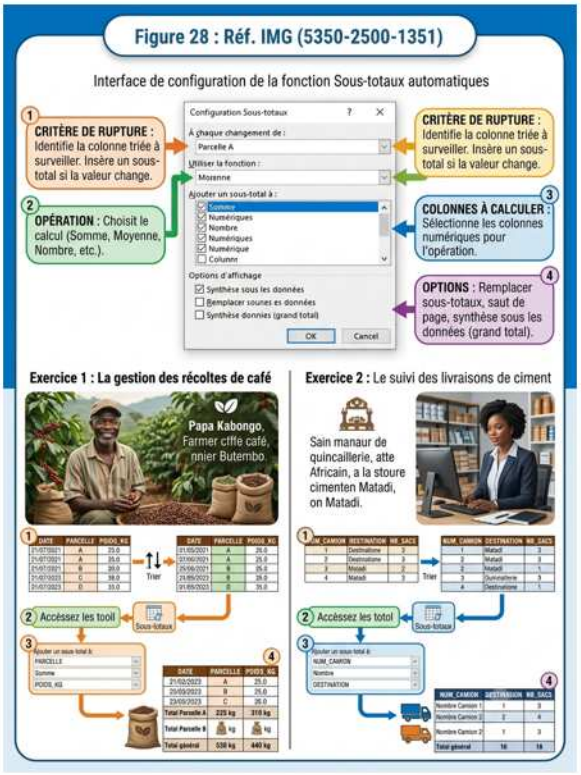


Figure 28 — Interface de configuration de la fonction Sous-totaux automatiques

Le champ "À chaque changement de" est le plus important. Il indique au logiciel la colonne à surveiller. Le programme insérera une ligne de sous-total chaque fois qu'il détectera une nouvelle valeur dans cette colonne. C'est la raison pour laquelle le tri préalable est vital. La liste "Utiliser la fonction" permet de choisir le type de calcul à appliquer pour chaque groupe. La

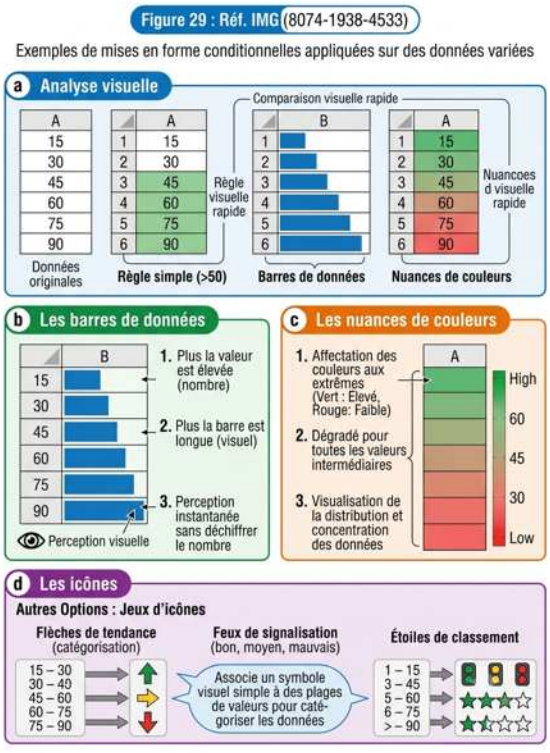


Figure 29 — Exemples de mises en forme conditionnelles appliquées sur des données v

Cette illustration compare plusieurs résultats de mise en forme conditionnelle sur une même colonne de données. Chaque méthode révèle une facette différente de l'information. À gauche, une règle simple. Au centre, des barres. À droite, une échelle de couleurs. L'outil s'adapte à l'objectif d'analyse. Les barres de données, visibles au centre, remplissent proportionnellement chaque

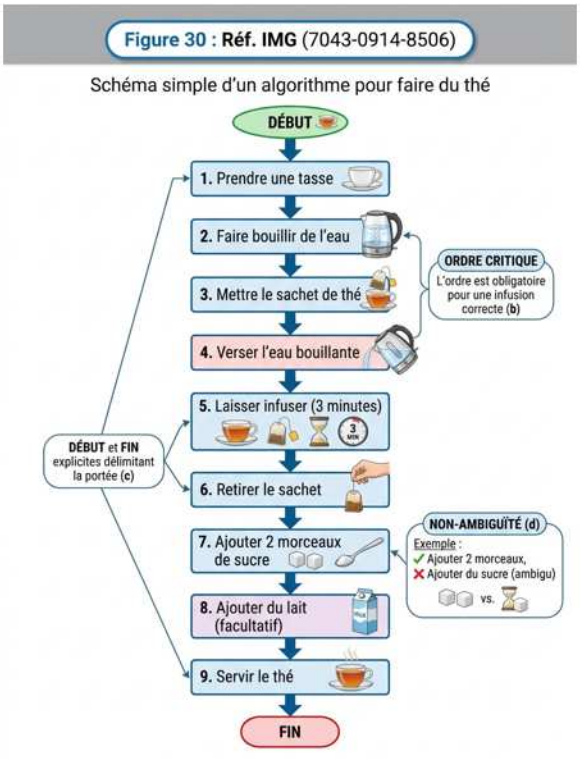


Figure 30 — Schéma simple d'un algorithme pour faire du thé

Ce schéma illustre une séquence d'actions. Chaque boîte représente une instruction claire, comme "Faire bouillir de l'eau". Les flèches indiquent l'ordre obligatoire des opérations. C'est la structure de base d'un algorithme : une suite logique et orientée d'actions simples menant à un but précis. L'ordre des instructions est critique. Mettre le sachet de thé avant de faire bouillir l'eau

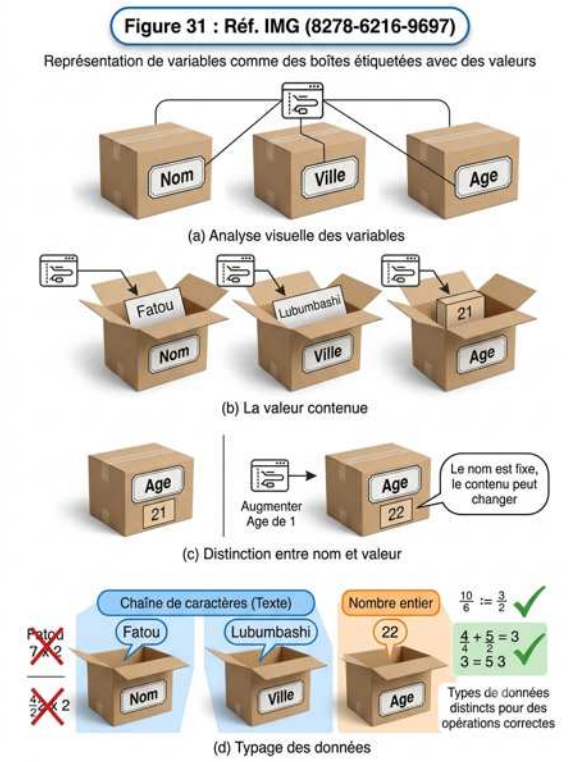


Figure 31 — Représentation de variables comme des boîtes étiquetées avec des valeur

L'image montre trois boîtes distinctes. Chaque boîte possède une étiquette unique : `Nom`, `Ville` et `Age`. Ces étiquettes sont les noms des variables. Elles permettent à l'algorithme de savoir exactement où chercher ou stocker une information spécifique, sans confusion possible entre les différentes données. À l'intérieur de chaque boîte, une information est stockée. La boîte `Nom`

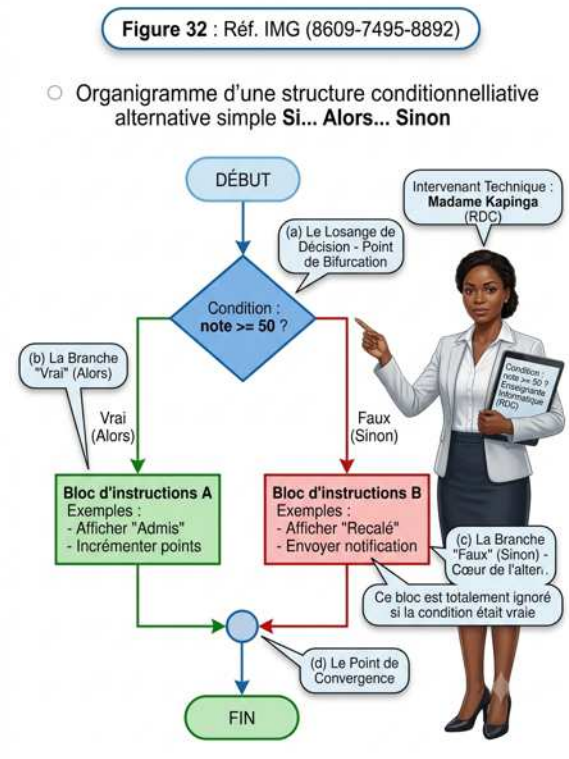


Figure 32 — Organigramme d'une structure conditionnelle alternative simple Si... Alors

Cet organigramme débute par le losange. Il contient la condition à évaluer, comme "note >= 50 ?". C'est le point de départ de la bifurcation. Le flux d'exécution arrive à ce point et doit choisir une direction. Si la réponse à la question dans le losange est "Vrai", le programme suit la branche correspondante. Cette branche mène à un ou plusieurs rectangles représentant le "Bloc d'instructions



Figure 33 — Un programme face à une décision binaire, comme une bifurcation routière

Le point de départ est la condition, matérialisée ici par le panneau de signalisation. Le programme arrive à ce "carrefour" et doit prendre une décision basée sur une information (la valeur d'une variable, une entrée utilisateur). La route principale, bien goudronnée, représente le chemin "ALORS". C'est souvent le cas attendu ou le plus fréquent. Le programme l'emprunte si la condition

Figure 34 : Réf. IMG (9858-3048-8582)

Algorigramme illustrant une structure SI...ALORS...SINON imbriquée

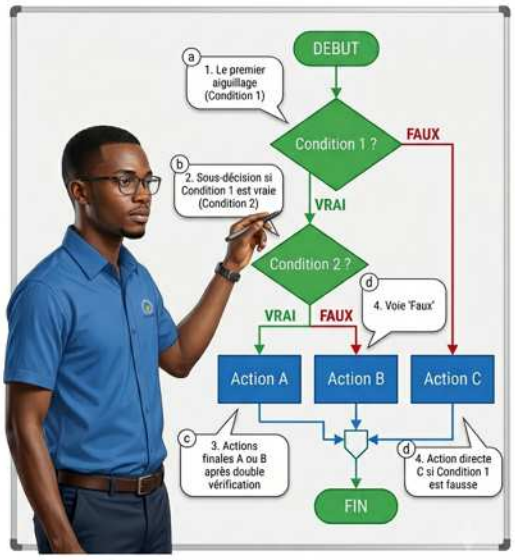


Figure 34 — Algorigramme illustrant une structure SI...ALORS...SINON imbriquée

L'algorigramme montre un premier losange représentant la `Condition 1`. Ce bloc divise le flux d'exécution en deux chemins distincts : un chemin "Vrai" et un chemin "Faux". Ceci représente la décision principale que le programme doit prendre. En suivant la branche "Vrai", le flux n'aboutit pas directement à une action. Il rencontre un second losange, représentant la `Condition 2`. Cette

Figure 35 : Réf. IMG (5962-6851-7623)

Arbre de décision pour un choix de transport à Matadi

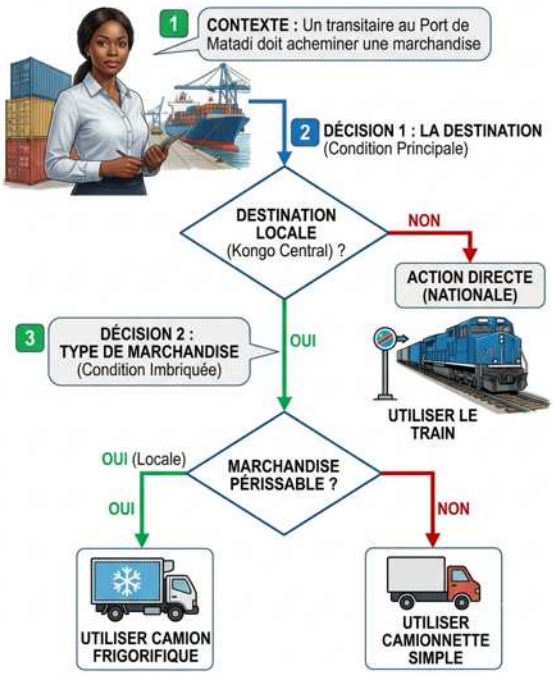


Figure 35 — Arbre de décision pour un choix de transport à Matadi

Un transitaire au port de Matadi doit décider comment acheminer une marchandise. La première question est la destination. Est-elle locale (dans le Kongo Central) ou nationale ? Cet arbre illustre ce processus de décision. Le premier niveau de l'arbre est "Destination : Locale ?". Si la réponse est "Oui", on suit une branche. Si c'est "Non", on en suit une autre. C'est la condition

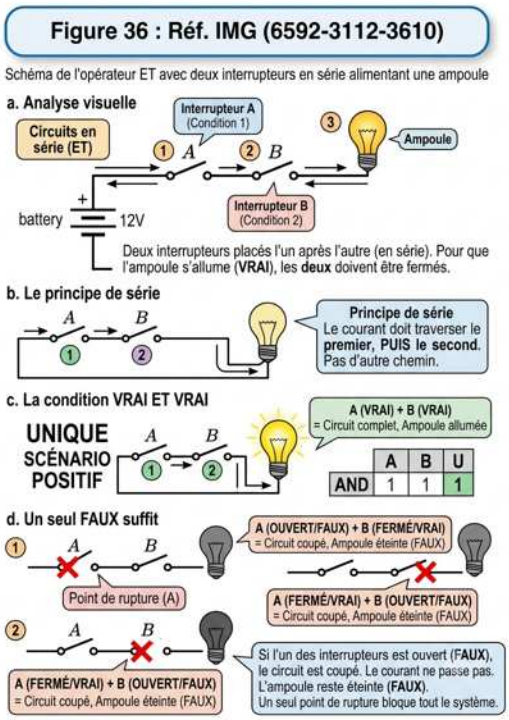


Figure 36 — Schéma de l'opérateur ET avec deux interrupteurs en série alimentant une

Le dessin montre deux interrupteurs (A et B) placés l'un après l'autre (en série) sur un circuit électrique. Pour que le courant passe et allume l'ampoule (résultat VRAI), les deux interrupteurs doivent être fermés. C'est la représentation physique parfaite de l'opérateur ET. La disposition en série signifie que le courant doit traverser le premier composant, PUIS le second. Il n'y a pas

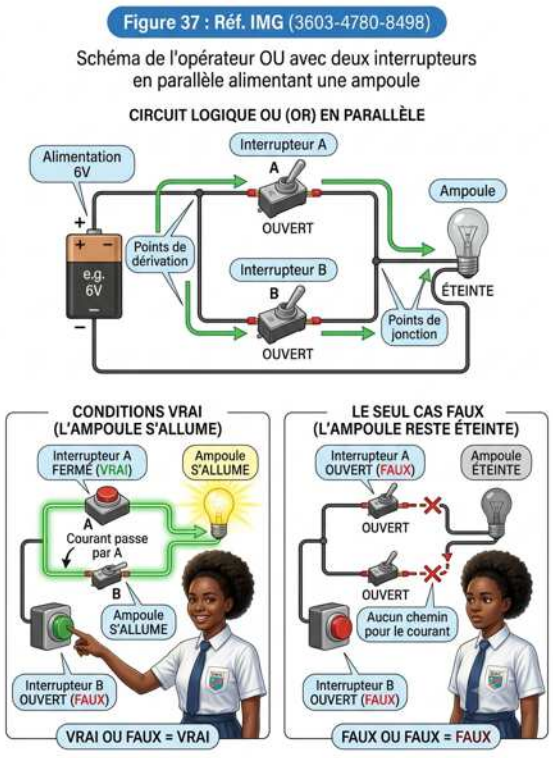


Figure 37 — Schéma de l'opérateur OU avec deux interrupteurs en parallèle alimentant

Ce schéma présente deux interrupteurs (A et B) montés en parallèle. Le courant a deux chemins possibles pour atteindre l'ampoule. Il peut passer par l'interrupteur A OU par l'interrupteur B. Cette configuration illustre l'opérateur logique OU. L'ampoule s'allume si au moins un chemin est disponible. Le montage en parallèle offre plusieurs voies. Le système fonctionne si la voie A est

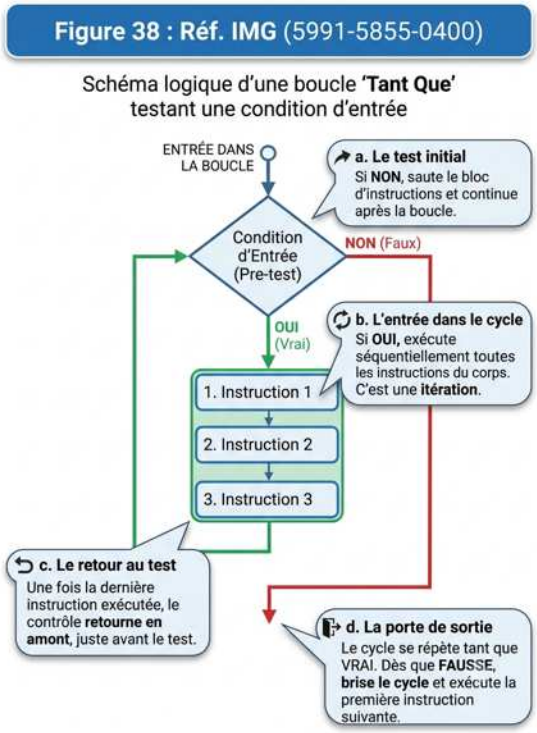


Figure 38 — Schéma logique d'une boucle 'Tant Que' testant une condition d'entrée.

Le diagramme montre une forme de losange représentant la condition. Avant toute chose, le programme évalue si cette condition est vraie. Si la réponse est NON, le programme saute entièrement le bloc d'instructions et continue son chemin après la boucle. Si la condition est VRAIE, le programme entre dans le corps de la boucle. Il exécute séquentiellement toutes les instructions qui s'y trouvent, de

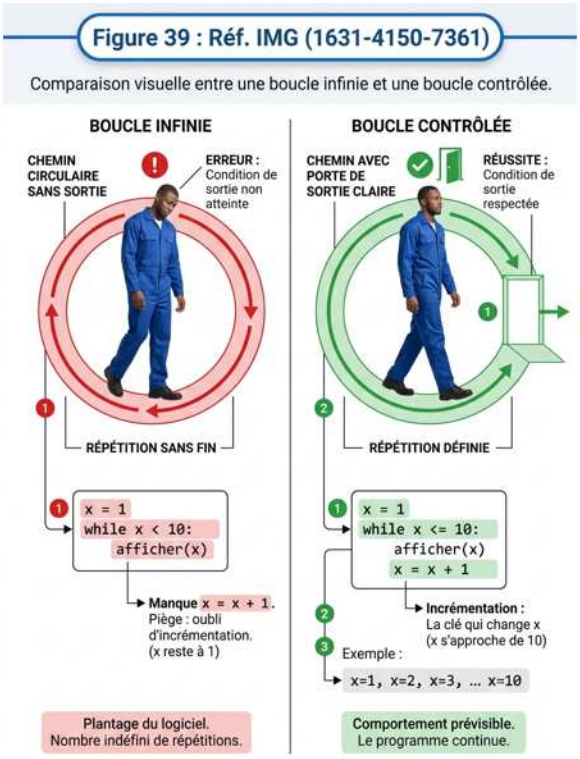


Figure 39 — Comparaison visuelle entre une boucle infinie et une boucle contrôlée.

Le dessin de gauche illustre une boucle infinie : un chemin circulaire sans aucune sortie. À droite, le chemin possède une porte de sortie claire, symbolisant une boucle bien conçue qui s'arrêtera lorsque la condition de sortie sera atteinte. Une boucle devient infinie si le programmeur oublie d'inclure l'instruction qui fait évoluer la condition. Par exemple, si on compte de 1 à 10 mais

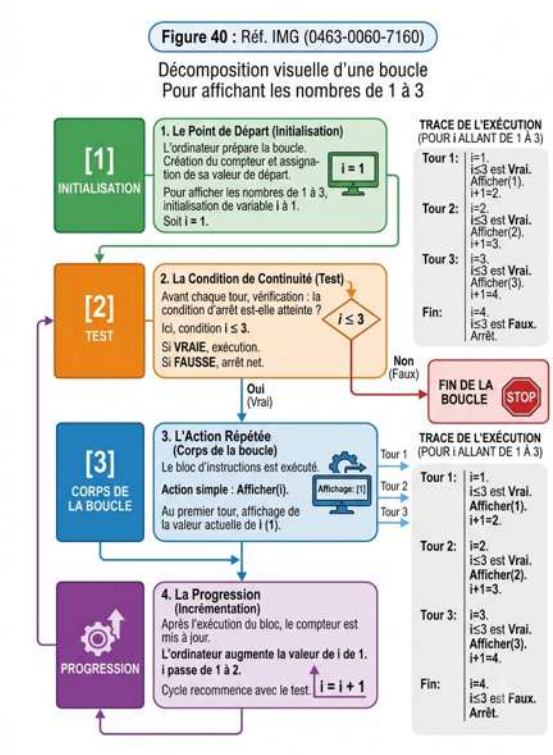


Figure 40 — Décomposition visuelle d'une boucle Pour affichant les nombres de 1 à 3

L'ordinateur prépare la boucle. Il crée un compteur et lui assigne sa valeur de départ. Pour afficher les nombres de 1 à 3, il commence par initialiser une variable, par exemple i, à 1. Soit $i = 1$. Avant chaque tour, une question est posée : la condition d'arrêt est-elle atteinte ? Ici, la condition est $i \le 3$. Si la réponse est vraie, la boucle exécute son bloc d'instructions. Si

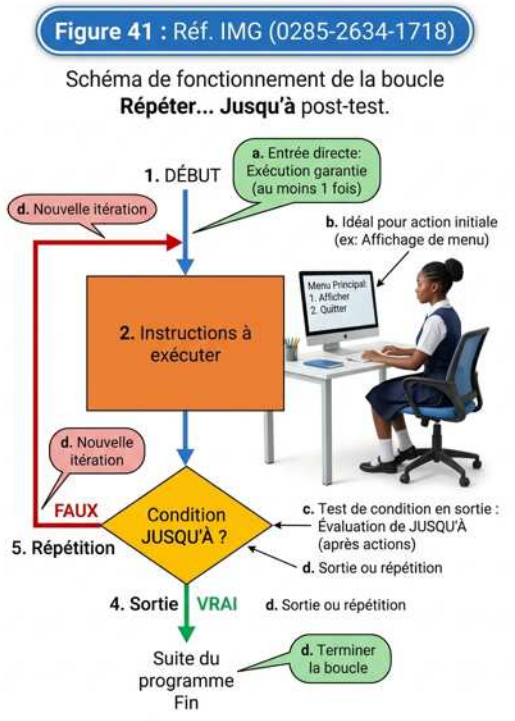


Figure 41 — Schéma de fonctionnement de la boucle Répéter... Jusqu'à post-test.

Un organigramme montre une flèche entrant directement dans un rectangle symbolisant les "Instructions à exécuter". Il n'y a aucune porte ou condition à l'entrée. Ceci garantit que le code à l'intérieur est exécuté au moins une fois, sans vérification préalable. Cette entrée directe est la signature de la boucle `Répéter... Jusqu'à`. Contrairement à la boucle `Tant que`, qui peut ne

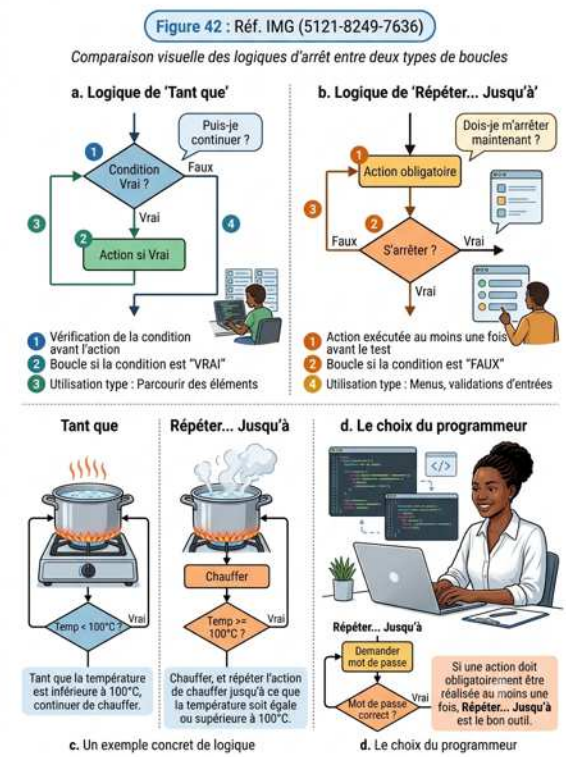


Figure 42 — Comparaison visuelle des logiques d'arrêt entre deux types de boucles.

Un schéma montre un test de condition (losange) placé *avant* le bloc d'action. La boucle tourne si la condition est "VRAI". Elle est souvent utilisée pour parcourir des éléments tant qu'il en reste. La question est : "Puis-je continuer ?". Le schéma montre le bloc d'action placé *avant* le test de condition. La boucle tourne si la condition est "FAUX". Elle est typique des menus ou des

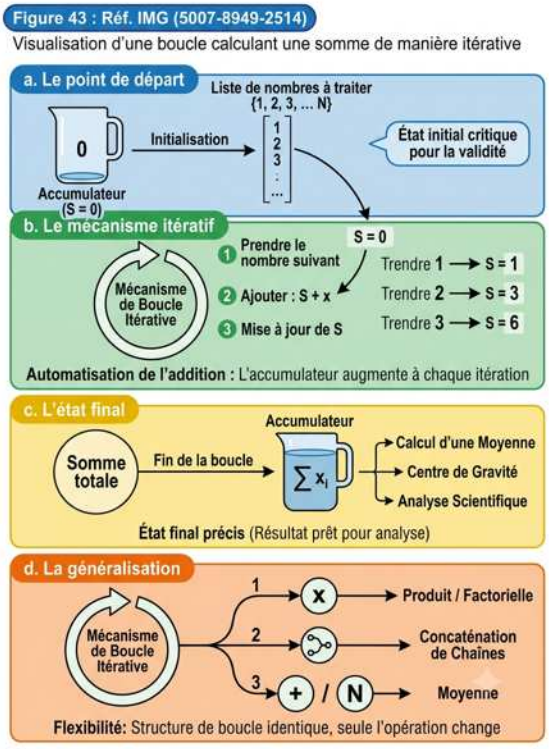


Figure 43 — Visualisation d'une boucle calculant une somme de manière itérative

L'image représente un accumulateur, initialisé à zéro. C'est le conteneur vide avant le début du processus. À côté, une liste de nombres (1, 2, 3... N) attend d'être traitée. L'état initial est crucial pour garantir la validité du résultat final et éviter les erreurs de calcul. Une flèche symbolise la boucle. À chaque "tour", elle prend le prochain nombre de la liste et l'ajoute à

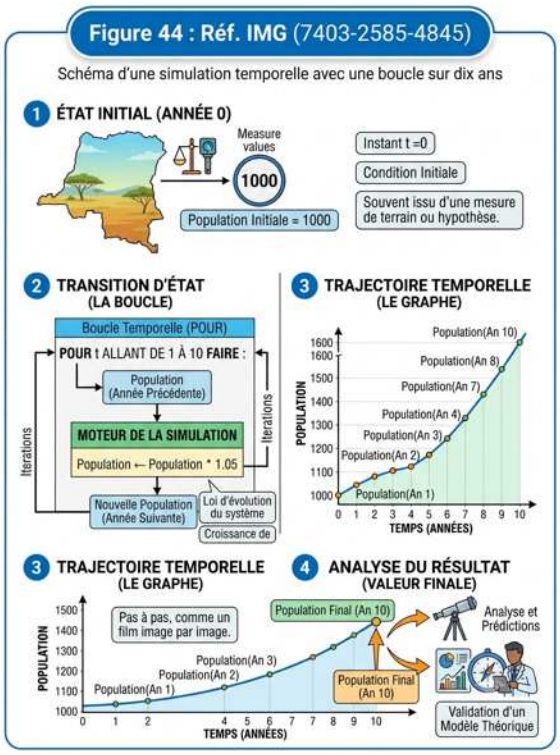


Figure 44 — Schéma d'une simulation temporelle avec une boucle sur dix ans

Le diagramme montre une valeur de départ, par exemple une population de 1000 individus. Cet instant $t=0$ est la condition initiale de la simulation. Sans cette donnée, le modèle ne peut pas démarrer. Elle est souvent issue d'une mesure de terrain ou d'une hypothèse de travail précise. Une boucle `POUR` encapsule la simulation. À chaque itération, qui représente une année, une formule

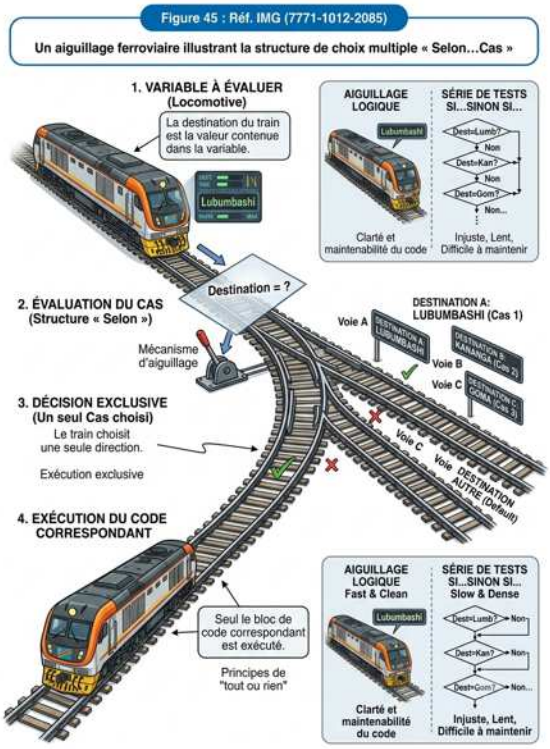


Figure 45 — Un aiguillage ferroviaire illustrant la structure de choix multiple « Selon...C

L'image montre un aiguillage ferroviaire. La locomotive représente la variable à évaluer. Chaque voie (Voie A, Voie B, Voie C) correspond à une valeur possible de la variable. Le programme choisit une seule direction, un seul "Cas", et ignore les autres. C'est un principe d'exécution exclusive. La destination du train (par exemple, Lubumbashi, Kananga, ou Goma) est la valeur contenue dans la

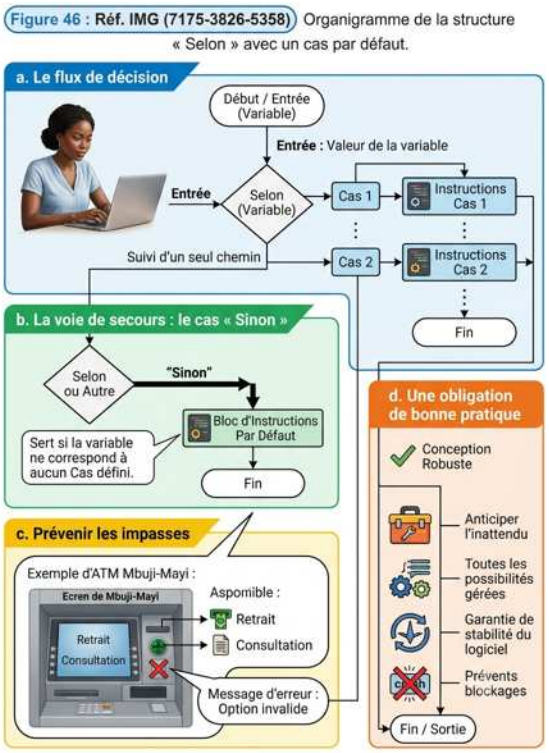


Figure 46 — Organigramme de la structure « Selon » avec un cas par défaut.

Cet organigramme représente le flux logique. L'entrée est la variable. Le losange central "Selon (Variable)" est le point de décision. Les flèches partent vers les différents cas possibles (Cas 1, Cas 2). L'exécution suit un seul chemin, rendant le processus clair et prévisible. La flèche étiquetée "Sinon" (ou "Autre") mène à un bloc d'instructions séparé. C'est la voie de secours. Si la

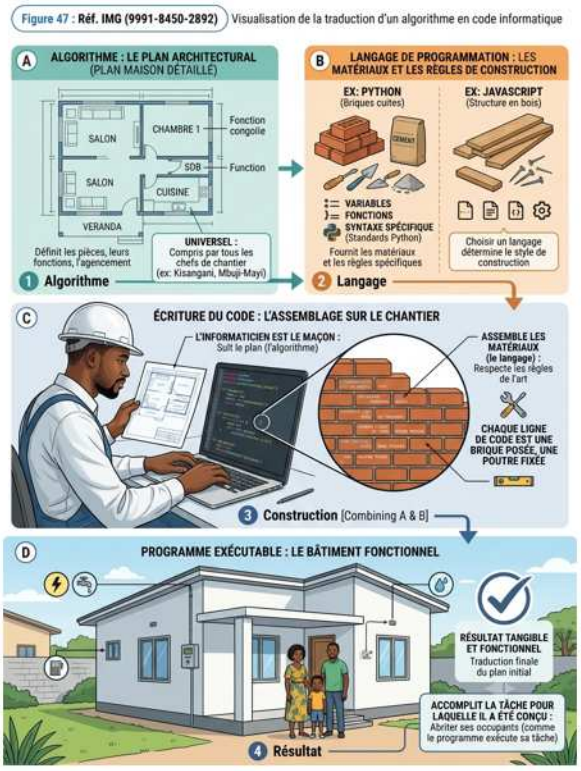


Figure 47 — Visualisation de la traduction d'un algorithme en code informatique

L'algorithme représente le plan détaillé d'une maison. Il définit les pièces, leurs fonctions et leurs agencements. Ce plan est universel. N'importe quel chef de chantier peut le comprendre, qu'il construise à Kisangani ou à Mbuji-Mayi. Un langage de programmation fournit les matériaux (variables, fonctions) et les règles (la syntaxe). Choisir Python, c'est comme choisir de construire