

# Power Automate - Fondations de l'automatisation

---

# Objectifs de la Formation

## Compétences visées

- Comprendre les concepts fondamentaux de Power Automate
- Créer et gérer des flux automatisés pour différents scénarios
- Maîtriser les connecteurs et les intégrations Microsoft 365
- Implémenter des logiques conditionnelles et des boucles
- Gérer les erreurs et optimiser les performances des flux

## Approche pédagogique

- Théorie avec exemples concrets
- Exercices pratiques guidés
- Cas d'usage réels
- Projet final intégrant tous les concepts

# Programme de la Journée

## **Matin (9h00 - 12h30)**

- Introduction à Power Automate
- Écosystème Power Platform
- Types de flux
- Interface et navigation
- Licences et versions
- Déclencheurs et actions
- Connecteurs Microsoft 365
- Expressions

## **Après-midi (13h30 - 17h00)**

- Logique conditionnelle
- Boucles et contrôles
- Approbations
- Gestion des données
- Variables et JSON
- Gestion des erreurs
- Optimisation
- AI Builder
- Cas pratiques et projet

# Introduction à Power Automate

# Qu'est-ce que Power Automate ?

## Définition

Service cloud qui permet de créer des workflows automatisés entre applications et services pour synchroniser des fichiers, recevoir des notifications, collecter des données et plus encore.

## Anciennement connu sous le nom

Microsoft Flow (renommé en 2019)

## Cas d'usage courants

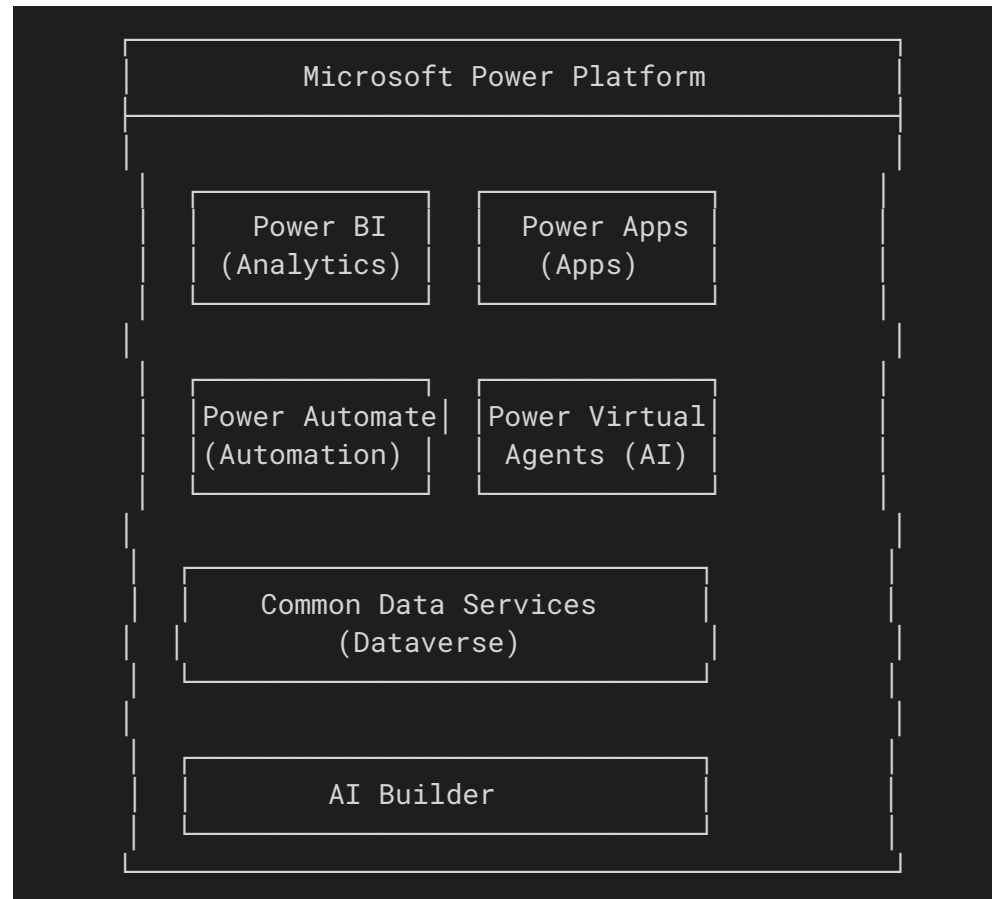
- Automatisation de tâches répétitives
- Notifications et alertes
- Synchronisation de données entre applications
- Workflows d'approbation
- Collecte et traitement de données
- Intégration d'applications tierces

# Qu'est-ce que Power Automate ?

## Avantages

- Low-code / No-code
- Intégration native Microsoft 365
- Centaines de connecteurs disponibles
- Scalable et fiable

# L'Écosystème Power Platform



# Positionnement de Power Automate

## Intégration avec Power Platform

### Power Apps

- Déclenchement de flux depuis applications
- Collecte et traitement de données
- Actions personnalisées

### Power BI

- Alertes sur données
- Export et distribution de rapports
- Refresh automatique de datasets

### Power Virtual Agents

- Actions dans les chatbots
- Intégration de processus métier

### Dataverse

- Déclencheurs sur modifications d'entités
- CRUD operations
- Logique métier centralisée

### AI Builder

- Traitement de documents
- Analyse de sentiments
- Extraction d'informations



# Types de Flux Power Automate

## 1. Flux Automatisés (Automated Cloud Flows)

Déclenchés automatiquement par un événement

- Nouvel email reçu
- Fichier créé dans SharePoint
- Tweet mentionnant votre entreprise
- Élément ajouté dans une liste

## 2. Flux Instantanés (Instant Cloud Flows)

Déclenchés manuellement par un utilisateur

- Bouton dans Power Apps
- Bouton mobile
- Raccourci depuis SharePoint

## 3. Flux Planifiés (Scheduled Cloud Flows)

Exécutés à intervalles réguliers

- Tous les jours à 8h00
- Chaque lundi
- Toutes les 15 minutes

## 4. Flux de Processus Métier (Business Process Flows)

Guider les utilisateurs à travers un processus défini

- Processus de vente
- Onboarding employés
- Gestion d'incidents

# Types de Flux

## 5. Flux de Bureau (Desktop Flows / RPA)

Automatiser des tâches sur ordinateur local

- Automatisation d'applications legacy
- Web scraping
- Saisie de données répétitive
- Intégration avec applications sans API

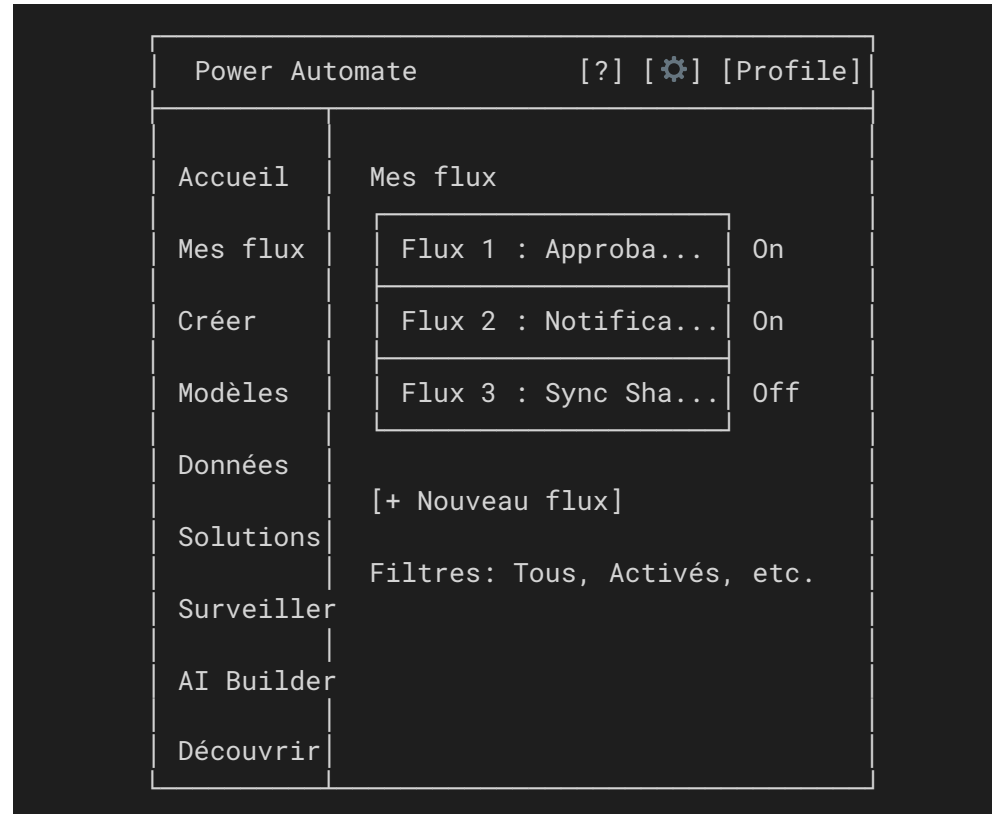
## 6. Flux de Solutions

Flux packagés pour déploiement ALM

- Développement en équipe
- Déploiement entre environnements
- Versioning

# Interface Power Automate

Portail [make.powerautomate.com](https://make.powerautomate.com)



# Navigation dans l'Interface

## **Menu Principal**

### **Accueil**

Vue d'ensemble, modèles recommandés, activité récente

### **Mes flux**

Liste de tous vos flux avec statut, dernière exécution, créateur

### **Créer**

Démarrer un nouveau flux : automatisé, instantané, planifié

### **Modèles**

Bibliothèque de flux pré-configurés

### **Données**

Connexions, tables personnalisées, entités Dataverse

### **Solutions**

Gestion de solutions pour ALM (Application Lifecycle Management)

### **Surveiller**

Historique d'exécution, erreurs, analytics

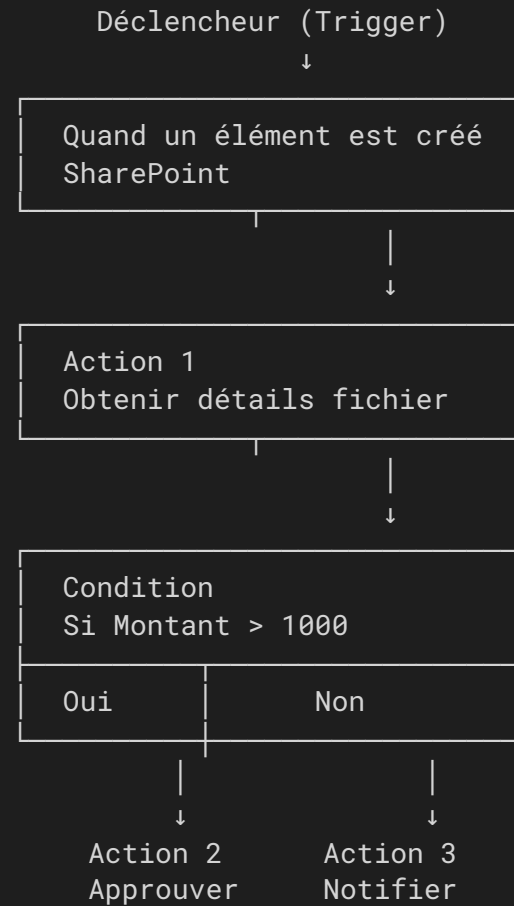
# Créer un Premier Flux Simple

**Exercice : Email de notification sur nouvel élément SharePoint**

## Étapes

1. **Créer** → Flux automatisé
2. **Nom** : "Notification nouvel élément SharePoint"
3. **Déclencheur** : "Quand un élément est créé - SharePoint"
4. **Configuration déclencheur**
  - Site Address : Votre site SharePoint
  - List Name : Votre liste
5. **Ajouter action** : "Envoyer un e-mail (V2) - Office 365 Outlook"
6. **Configuration email**
  - To : [votre.email@company.com](mailto:votre.email@company.com)
  - Subject : "Nouvel élément : [Titre]"
  - Body : Contenu dynamique
7. **Enregistrer**
8. **Tester**

# Anatomie d'un Flux



# Licences Power Automate

## Types de Licences

### Power Automate Free

- Inclus dans Microsoft 365 (selon plan)
- 750 exécutions/mois
- Connecteurs Standard
- Flows cloud automatisés, instantanés, planifiés

### Power Automate Premium (par utilisateur)

- 20 USD/utilisateur/mois
- 40 000 exécutions/mois
- Connecteurs Premium
- RPA (Desktop Flows) illimité
- AI Builder (5 000 crédits/mois)
- Dataverse

### Power Automate Process (par flux)

- 150 USD/flux/mois
- Capacité réservée pour flux spécifique
- Pour flux partagés (bots, services)

# Standard vs Premium : Tableau Comparatif

Fonctionnalité	Standard	Premium
Exécutions/mois	2 000	40 000
Connecteurs Microsoft 365	Oui	Oui
Connecteurs Premium	Non	Oui (SQL, HTTP, Salesforce)
Desktop Flows (RPA)	Non	Oui (illimité)
AI Builder	Non	Oui (5 000 crédits)
Dataverse	Limité	Complet
Business Process Flows	Non	Oui
Environnements personnalisés	Non	Oui
DLP Policies	Limité	Complet
Prix	Inclus M365	20 USD/user/mois



# Connecteurs : Standard vs Premium

## Connecteurs Standard (Inclus)

- SharePoint
- OneDrive for Business
- Outlook Office 365
- Microsoft Teams
- Excel Online
- Microsoft Forms
- Planner
- Notifications (mobile)
- Approvals

## Connecteurs Premium (Licence Premium requise)

- SQL Server
- HTTP / HTTP with Azure AD
- Salesforce
- Dynamics 365
- Oracle Database
- SAP
- Azure services avancés
- Connecteurs personnalisés

# Quand Utiliser Quelle Version ?

## Utilisez Standard si

- Automatisation interne Microsoft 365
- Volumes modérés (< 2 000 exécutions/mois)
- Budget limité
- Pas besoin RPA
- Workflows simples

## Passez à Premium si

- Intégration systèmes tiers (SQL, Salesforce)
- Volumes importants (> 2 000 exécutions/mois)
- Besoin RPA (automatisation desktop)
- AI Builder requis
- Processus métier complexes
- Dataverse nécessaire
- Développement multi-environnements

## Calcul ROI

20 USD/mois Premium = économie combien d'heures manuelles ?

# Déclencheurs (Triggers)

## Types de Déclencheurs

### Polling (Interrogation)

Vérifie périodiquement si événement s'est produit

- Fréquence configurable (1 min à 1 mois)
- Consomme 1 exécution par vérification même si pas de nouvelle donnée
- Exemple : "Quand un élément est créé - SharePoint"

### Push (Événementiel)

Notification immédiate lors d'événement

- Temps réel
- Ne consomme exécution que si événement réel
- Exemple : "Quand un email arrive (V3) - Outlook" (avec webhook)

### Récurrance

Déclenche à intervalles réguliers

- Cron-like scheduling
- Exemple : Tous les jours à 8h00

# Configuration des Déclencheurs

## Exemple : Déclencheur SharePoint

```
Quand un élément est créé
├─ Site Address : https://contoso.sharepoint.com/sites/HR
├─ List Name : Demandes de congés
└─ Paramètres avancés (optionnel)
    ├─ Limit Columns : Titre, Dates, Employé
    └─ Trigger Conditions : Status eq 'Nouveau'
```

## Best Practice

- Limiter colonnes retournées
- Utiliser filtres OData si possible
- Privilégier déclencheurs push quand disponibles

## Trigger Conditions

Éviter exécutions inutiles

```
@equals(triggerOutputs()?['body/Status'],  
        'Nouveau')
```

# Actions Essentielles

## Actions Microsoft 365 Courantes

### SharePoint

- Créer élément
- Mettre à jour élément
- Obtenir pièces jointes
- Copier fichier

### Outlook

- Envoyer email
- Créer événement calendrier
- Marquer email comme lu

### Teams

- Publier message dans canal
- Créer équipe
- Ajouter membre

### Excel Online

- Ajouter ligne
- Mettre à jour ligne
- Lister lignes

# Configuration d'Actions

## Exemple : Envoyer un Email

Envoyer un e-mail (V2) - Office 365 Outlook

```
├ To : manager@company.com
├ Subject : Nouvelle demande de congés
├ Body :
│   Bonjour,
│
│   Une nouvelle demande a été soumise :
│   - Employé : [Contenu dynamique: Employé]
│   - Dates : [DateDébut] - [DateFin]
│   - Motif : [Contenu dynamique: Motif]
│
│   Veuillez approuver ou rejeter.
└ Importance : Haute (optionnel)
```

## Contenu Dynamique

Données du déclencheur ou actions précédentes

## Bonnes Pratiques

- Utiliser version V2 des actions
- Formater email en HTML pour meilleure présentation
- Inclure liens pertinents
- Personnaliser messages

# Connecteurs Microsoft 365 - SharePoint

## Actions SharePoint Principales

### Lecture

- Obtenir élément
- Obtenir éléments (avec filtres OData)
- Obtenir pièces jointes de l'élément
- Lister éléments de liste

### Écriture

- Créer un élément
- Mettre à jour l'élément
- Supprimer l'élément

### Fichiers

- Copier fichier
- Créer fichier
- Obtenir contenu de fichier
- Obtenir métadonnées de fichier

### Best Practices

- Utiliser filtres OData pour limiter données
- Gérer pagination pour grandes listes (> 5 000 éléments)
- Éviter appels dans boucles si possible

# Connecteurs Microsoft 365 - Outlook

## Emails

- Envoyer un e-mail (V2)
- Répondre à un e-mail
- Transférer un e-mail
- Marquer comme lu/non lu
- Déplacer vers dossier

## Calendrier

- Créer un événement
- Mettre à jour événement
- Obtenir événements de calendrier

## Contacts

- Créer un contact
- Obtenir contacts

## Pièces Jointes

- Télécharger pièce jointe
- Extraire contenu

## Astuces

- V2 des actions recommandée (plus de paramètres)
- Utiliser HTML pour emails formatés



# Connecteurs Microsoft 365 - Teams

## Messages

- Publier un message dans un canal
- Publier message avec choix adaptatifs
- Répondre à message
- Obtenir messages

## Équipes et Canaux

- Créer une équipe
- Ajouter membre à équipe
- Créer canal

## Appels et Réunions

- Créer une réunion en ligne

## Adaptive Cards

Cartes interactives avec boutons, inputs

- Collecter feedback
- Approbations
- Sondages

## Use Cases

- Notifications interactives
- Approbations rapides
- Collecte de données
- Sondages et votes

# Connecteurs Microsoft 365 - Excel

## Actions Excel Online Principales

### Tables

- Lister lignes présentes dans un tableau
- Ajouter une ligne dans un tableau
- Mettre à jour une ligne
- Supprimer une ligne

### Fichiers

- Créer tableau
- Obtenir un tableau

## Limitations

- Fichier doit être dans OneDrive ou SharePoint
- Données doivent être dans Table Excel (pas simple plage)
- Performance limitée pour gros volumes

## Best Practices

- Utiliser filtre OData pour réduire lignes récupérées
- Éviter modifications concurrentes
- Considérer SharePoint Lists ou Dataverse pour volumes importants

# Connecteurs Premium - Aperçu

## SQL Server

- Obtenir lignes
- Insérer ligne
- Mettre à jour ligne
- Exécuter requête SQL

## HTTP / HTTP with Azure AD

- Appeler APIs REST externes
- Intégration services web
- Webhooks

## Salesforce

- Créer enregistrement
- Mettre à jour enregistrement
- Obtenir enregistrements

## Connecteurs Personnalisés

Créer vos propres connecteurs pour APIs propriétaires

# Manipulation de Données - Expressions

## Qu'est-ce qu'une Expression ?

Formules pour transformer, calculer ou manipuler des données

### Syntaxe

Similaire à Excel, mais avec fonctions spécifiques Power Automate

### String (Chaînes)

- concat() : Concaténer
- substring() : Extraire sous-chaîne
- replace() : Remplacer
- length() : Longueur
- toLower() / toUpper() : Casse

## Date et Heure

- utcNow() : Date/heure actuelle UTC
- addDays() : Ajouter jours
- formatDateTime() : Formater
- addHours() : Ajouter heures

### Logique

- if() : Condition
- and() / or() : Opérateurs logiques
- equals() : Égalité
- greater() / less() : Comparaisons

# Expressions - Exemples Pratiques

## Concaténation

```
concat('Bonjour ', triggerOutputs()?['body/Nom'])
```

## Formater Date

```
formatDateTime(utcNow(), 'dd/MM/yyyy')
```

## Conditionnel

```
if(greater(triggerOutputs()?['body/Montant'], 1000), 'Élevé', 'Normal')
```

## Extraire Domaine Email

```
split(triggerOutputs()?['body/Email'], '@')[1]
```

## Calculer Différence Jours

```
div(sub(ticks(item()?['DateFin']), ticks(item()?['DateDebut'])), 864000000000)
```

## Longueur Texte

```
length(triggerOutputs()?['body/Description'])
```

# Exercice Pratique 1

## Automatisation : Email avec pièces jointes SharePoint

### Objectif

Lorsqu'un nouveau fichier est ajouté dans un dossier SharePoint, envoyer un email avec le fichier en pièce jointe.

### Étapes

1. Déclencheur : "Quand un fichier est créé (propriétés uniquement) - SharePoint"
2. Action : "Obtenir contenu de fichier - SharePoint"
3. Action : "Envoyer un e-mail (V2) - Outlook"
  - Utiliser contenu dynamique pour pièce jointe
4. Tester en uploadant un fichier

### Bonus

Ajouter une condition : seulement si le fichier est un PDF

# Logique Conditionnelle et Contrôles

# Conditions (If/Else)

## Structure de Base

```
Condition
├─ Si oui (If)
│   └─ Actions si condition vraie
└─ Si non (Else)
    └─ Actions si condition fausse
```

## Configuration

```
Condition
├─ Choisir valeur : [Contenu dynamique: Montant]
├─ Opérateur : est supérieur à
└─ Valeur : 1000
```

## Opérateurs Disponibles

- est égal à / n'est pas égal à
- est supérieur à / inférieur à
- contient / ne contient pas
- commence par / se termine par
- est vide / n'est pas vide



# Conditions - Exemple Pratique

## Cas : Approbation selon montant

```
Déclencheur : Nouvelle demande de dépense (SharePoint)
↓
Condition : Montant > 500
├─ Si oui
│   └─ Envoyer email au directeur pour approbation
└─ Si non
    └─ Approuver automatiquement et notifier employé
```

## Configuration Condition

```
Condition
├─ Choisir valeur : @triggerOutputs()?['body/Montant']
├─ Opérateur : est supérieur à
└─ Valeur : 500
```

## Actions dans branches

Branch "Si oui" : Demander approbation manager

Branch "Si non" : Mettre à jour statut + notifier

# Conditions Avancées - ET/OU

## Mode Avancé

Combiner plusieurs conditions avec ET (AND) / OU (OR)

### Exemple : Condition ET

```
Montant > 1000 ET Catégorie = 'Matériel'
```

### Configuration :

```
Mode : Avancé  
Expression :  
@and(  
  greater(triggerOutputs()?['body/Montant'], 1000),  
  equals(triggerOutputs()?['body/Categorie'], 'Matériel')  
)
```

### Exemple : Condition OU

```
Priorité = 'Urgent' OU Montant > 5000
```

```
@or(  
  equals(triggerOutputs()?['body/Priorite'], 'Urgent'),  
  greater(triggerOutputs()?['body/Montant'], 5000)  
)
```

# Switch (Instruction de Basculement)

## Quand Utiliser ?

Lorsque plusieurs branches basées sur valeur d'une variable

## Structure

```
Switch
├─ On : [Contenu dynamique: Statut]
├─ Case : Nouveau
│   └─ Actions pour statut Nouveau
├─ Case : En cours
│   └─ Actions pour statut En cours
├─ Case : Terminé
│   └─ Actions pour statut Terminé
└─ Par défaut (Default)
    └─ Actions pour autres statuts
```

## Avantage vs Conditions Imbriquées

- Plus lisible
- Plus performant
- Plus facile à maintenir

# Switch - Exemple

## Cas : Notification selon priorité ticket

```
Déclencheur : Nouveau ticket support (SharePoint)
↓
Switch sur Priorité
├─ Case : Critique
│   └─ SMS + Email urgent + Créer incident major
├─ Case : Élevé
│   └─ Email urgent + Assigner technicien senior
├─ Case : Moyen
│   └─ Email standard + Assigner technicien disponible
├─ Case : Faible
│   └─ Ajouter à file d'attente
└─ Default
    └─ Email erreur + Demander clarification priorité
```

# Boucles - Apply to Each

## Itérer sur Collection

```
Apply to each
├ Sélectionner sortie : [Tableau d'éléments]
└ Actions
  └ Action sur chaque élément (item())
```

## Exemple : Traiter pièces jointes email

```
Déclencheur : Réception email avec pièces jointes
↓
Apply to each : Pièces jointes
├ Obtenir contenu pièce jointe
└ Sauvegarder dans SharePoint
```

## Accès Élément Courant

```
item()?['NomPropriété']
```

## Attention

Peut être lent pour gros volumes (> 100 éléments)

# Boucles - Do Until

## Répéter jusqu'à Condition Vraie

```
Do until
├─ Condition : Statut égal à 'Approuvé'
├─ Limite : 10 itérations max (sécurité)
└─ Actions
    └─ Vérifier statut, attendre, etc.
```

## Exemple : Attendre approbation

```
Do until
├─ Condition : @equals(variables('Statut'), 'Approuvé')
└─ Actions
    ├── Obtenir élément SharePoint (statut)
    ├── Définir variable Statut
    └─ Délai (attendre 5 minutes)
```

## Best Practices

- Toujours définir limite d'itérations
- Inclure délai pour éviter throttling
- Gérer time-out

# Actions Parallèles

## Exécuter Actions Simultanément

```
Parallel Branch
├─ Branche 1
│   └─ Envoyer email équipe A
└─ Branche 2
    └─ Envoyer email équipe B
```

## Quand Utiliser ?

Actions indépendantes pouvant s'exécuter en même temps

## Exemple : Notifications Multiples

```
Nouvelle demande de congés
↓
Parallel Branch
├─ Branche 1 : Notifier manager
├─ Branche 2 : Notifier RH
└─ Branche 3 : Créer événement calendrier
```

## Avantage

Réduction temps d'exécution total

# Scope (Portée)

## Grouper Actions Logiquement

```
Scope : Traitement des données
```

```
|— Action 1  
|— Action 2  
└— Action 3
```

## Utilité

- Organisation visuelle
- Gestion d'erreurs groupée
- Contrôle d'exécution (skip, timeout)

## Exemple : Traitement avec Gestion Erreurs

```
Scope : Envoi notifications
```

```
|— Configure run after : Toujours exécuter  
└— Actions  
    |— Envoyer email manager  
    |— Envoyer SMS  
    └— Publier dans Teams
```

```
Condition : Vérifier si Scope a échoué
```

```
└— Si oui : Logger erreur + Notifier admin
```



# Approbations

## Action : Démarrer et attendre une approbation

Intégrée nativement dans Power Automate

### Types d'Approbation

- Approuver/Rejeter - Première réponse
- Approuver/Rejeter - Tout le monde doit approuver
- Approuver/Rejeter - Personne dans liste
- Réponse personnalisée - Attendre une réponse
- Réponse personnalisée - Attendre toutes les réponses

### Configuration

```
Démarrer et attendre une approbation
├─ Type d'approbation : Approuver/Rejeter - Première réponse
├─ Titre : "Approuver demande de congés"
├─ Affecté à : manager@company.com
├─ Détails : Corps de l'approbation avec infos
└─ Lien d'élément : Lien vers demande
```

# Workflow d'Approbation Complet

## Exemple : Approbation Demande de Dépense

```
1. Déclencheur : Nouvel élément SharePoint "Demandes Dépenses"
  ↓
2. Obtenir détails du fichier justificatif
  ↓
3. Démarrer et attendre une approbation
  - Affecté à : Manager (colonne SharePoint)
  - Détails : Montant, Description, Justificatif
  ↓
4. Condition : Réponse = "Approve"
  └─ Si Approuvé
     └─ Mettre à jour élément SharePoint (Statut = Approuvé)
        └─ Envoyer email confirmé au demandeur
  └─ Si Rejeté
     └─ Mettre à jour élément SharePoint (Statut = Rejeté)
        └─ Envoyer email rejet avec commentaires
           └─ Logger dans liste historique
```

# Exercice Pratique 2

## Workflow d'Approbation de Documents

### Objectif

Créer un flux d'approbation multi-niveaux pour documents

### Spécifications

1. Déclencheur : Fichier ajouté dans dossier SharePoint "À approuver"
2. Si taille fichier > 10 MB : Rejeter automatiquement (trop volumineux)
3. Sinon : Demander approbation au manager
4. Si approuvé : Déplacer vers "Approuvés" et notifier créateur
5. Si rejeté : Notifier créateur avec raison

### Bonus

- Ajouter timeout : si pas de réponse sous 48h, escalader au directeur
- Logger toutes approbations dans Excel

# Variables

## Initialiser une variable

```
Initialiser la variable
├─ Nom : MontantTotal
├─ Type : Entier (Integer)
└─ Valeur : 0
```

## Types Disponibles

- Chaîne (String)
- Entier (Integer)
- Nombre à virgule flottante (Float)
- Booléen (Boolean)
- Tableau (Array)
- Objet (Object)

## Actions sur Variables

- Définir la variable (Set)
- Incrémenter la variable
- Décrémenter la variable
- Ajouter à la variable tableau (Append)

## Bonnes Pratiques

- Initialiser toutes les variables au début du flux
- Nommer clairement (camelCase)
- Utiliser le bon type
- Documenter l'usage

# Variables - Exemples

## Compteur dans Boucle

```
Initialiser variable : Compteur = 0

Apply to each : Liste éléments
├ Condition : Statut = 'Actif'
├   └ Si oui : Incréments variable Compteur
└ ...

Envoyer email : "Nombre d'éléments actifs : @{variables('Compteur')}}"
```

## Tableau d'Emails

```
Initialiser variable tableau : ListeEmails = []

Apply to each : Employés
├ Ajouter à la variable tableau : Email de l'employé

Envoyer email groupé : To = @{join(variables('ListeEmails'), ';')}
```

# Manipulation de JSON

## Parser JSON

```
Analyser JSON
├─ Contenu : @triggerOutputs()?['body']
└─ Schéma : {...}
```

## Générer Schéma

Use sample payload pour auto-générer schéma

## Accéder Propriétés

Après parsing :

```
@body('Analyser_JSON')?['propriete']
@body('Analyser_JSON')?['tableau'][0]
@body('Analyser_JSON')?['objet']?['sousPropriete']
```

## Créer JSON

```
@json('{"nom": "John", "age": 30}')
```

## Fonctions Utiles

```
json() : Convertir string en JSON
string() : Convertir JSON en string
xpath() : Requête XML/JSON
```

## Tips

- Toujours valider le schéma
- Gérer les valeurs nulles avec ?

# Tableaux et Collections

## Filtrer Tableau

```
Filtrer le tableau
├─ De : [Tableau d'éléments]
└─ Condition : item()?['Statut'] eq 'Actif'
```

## Sélectionner (Mapper)

Transformer tableau

```
Sélectionner
├─ De : [Tableau source]
└─ Mapper
    ├─ Nom : item()?['FullName']
    └─ Email : item()?['EmailAddress']
```

## Fonctions Utiles

```
join() : Joindre éléments
length() : Nombre d'éléments
first() : Premier élément
last() : Dernier élément
skip() : Ignorer N éléments
take() : Prendre N éléments
contains() : Vérifier présence
```

## Best Practice

Filtrer avant de boucler pour améliorer les performances

# Gestion des Erreurs - Try-Catch

## Configure Run After

Définir comportement après action précédente

### Options

- est réussie (Success)
- a échoué (Failed)
- est ignorée (Skipped)
- a expiré (Timed out)

## Pattern Try-Catch

```
Scope : Try
├─ Actions à risque
└─ ...

Scope : Catch (Run after : a échoué)
├─ Composer message d'erreur
├─ Envoyer email admin
└─ Logger dans SharePoint
```



# Gestion des Erreurs - Exemple

## Exemple : Appel API avec Retry

```
Scope : Appel API
├─ HTTP : GET https://api.example.com/data
└─ Analyser JSON : Réponse

Scope : Gestion Erreur (Configure run after : a échoué)
├─ Condition : Erreur est 429 (Too Many Requests)
│   ├── Si oui
│   │   ├── Délai : 60 secondes
│   │   └─ HTTP : Réessayer appel
│   └─ Si non
│       ├── Composer : Message erreur détaillé
│       └─ Envoyer email : Notifier admin
└─ ...

Condition : Vérifier si données obtenues
└─ Continuer traitement ou abandonner
```

# Retry Policy

## Configuration Automatique

Dans paramètres de l'action :

```
Paramètres
└─ Stratégie de nouvelle tentative
    └─ Type : Par défaut / Exponentiel /
        Fixe / Aucun
    └─ Nombre de tentatives : 4
    └─ Intervalle : PT10S (10 secondes)
```

## Types de Stratégies

### Par défaut (Exponentiel)

Intervalle augmente : 10s, 20s, 40s, 80s

### Fixe

Même intervalle : 10s, 10s, 10s, 10s

### Use Cases

- Appels HTTP/API
- Actions avec services externes
- Opérations réseau

# Optimisation des Flux

## Réduire Appels API

- Utiliser filtres OData
- Limiter colonnes retournées
- Batching quand possible

## Éviter Boucles Inutiles

- Filtrer données avant Apply to each
- Utiliser actions natives (ex: Filter array)

## Parallélisation

- Actions indépendantes en parallèle
- Réduire temps d'exécution

## Caching

- Stocker résultats intermédiaires en variables
- Éviter appels répétés mêmes données

## Pagination

- Gérer correctement grandes collections (> 5 000 éléments)

## Trigger Conditions

- Filtrer au niveau du déclencheur
- Éviter exécutions inutiles

# Limites et Quotas

## Limites Techniques

Limite	Standard	Premium
Exécutions/mois	2 000	40 000
Durée max exécution	30 jours	30 jours
Actions par exécution	100 000	100 000
Taille max contenu	100 MB	100 MB
Boucle iterations	100 000	100 000
Variables max	250	250
Taille variable	100 MB	100 MB

### Throttling

Limites par connecteur (ex: SharePoint : 600 appels/min)

### Best Practice

Monitorer consommation, optimiser si approche limites

# Sécurité - Connexions

## Types de Connexions

### Connexions avec Informations d'Identification

Utilisateur s'authentifie avec son compte

- Permissions = permissions de l'utilisateur

### Connexions Partagées

Service account ou compte de service

- Pour flux automatiques sans interaction utilisateur

## Best Practices

### Principe du Moindre Privilège

Compte avec seulement permissions nécessaires

### Comptes de Service

Pour flux production, ne pas utiliser comptes personnels

### Rotation des Secrets

Renouveler périodiquement authentifications

### Audit

Surveiller utilisation connexions

# DLP (Data Loss Prevention)

## Groupes de Connecteurs

- Business : SharePoint, Teams, Outlook
- Non-Business : Twitter, Gmail personnel
- Bloqués : Connecteurs interdits

## Règles

- Connecteurs Business ne peuvent pas communiquer avec Non-Business dans même flux
- Bloqués complètement interdits

## Configuration

Centre d'administration Power Platform

## Exemple

Empêcher données SharePoint d'être envoyées vers Gmail personnel

## Impact

Flux violant DLP ne pourront pas être enregistrés/exécutés

## Bonne Pratique

Définir politiques DLP dès le départ

# Introduction à AI Builder

## Modèles Pré-construits

- Extraction de texte (OCR)
- Traitement de formulaires
- Détection d'objets dans images
- Analyse de sentiments
- Extraction d'informations (cartes de visite, factures, reçus)
- Catégorie de texte

## Modèles Personnalisés

- Classification de texte custom
- Extraction d'informations custom
- Détection d'objets custom

## Crédits AI Builder

Licence Premium : 5 000 crédits/mois  
Consommation selon modèle utilisé

## Cas d'Usage

- Automatiser traitement factures
- Analyser feedback clients
- Classifier emails/tickets

# AI Builder - Exemple : Traitement Factures

## Use Case : Automatiser Extraction Données Factures

```
1. Déclencheur : Email reçu avec pièce jointe (facture PDF)
  ↓
2. Obtenir pièce jointe
  ↓
3. AI Builder : Extraire informations de factures
  - Détecte automatiquement : Fournisseur, Date, Montant, etc.
  ↓
4. Créer élément SharePoint avec données extraites
  ↓
5. Démarrer approbation si montant > seuil
  ↓
6. Si approuvé : Envoyer vers système comptable (via SQL/API)
```

### Avantage

Zéro saisie manuelle, 90%+ précision



# Exercice Pratique 3

## Traitement de Données Excel avec Gestion d'Erreurs

### Objectif

Lire fichier Excel, traiter données avec gestion d'erreurs

### Spécifications

1. Déclencheur planifié : Tous les jours à 9h00
2. Lister lignes d'un tableau Excel (OneDrive)
3. Pour chaque ligne :
  - Valider données (montant > 0, date valide)
  - Si valide : Créer élément SharePoint
  - Si invalide : Ajouter à tableau erreurs
4. À la fin : Envoyer email récapitulatif
  - Nombre lignes traitées
  - Nombre erreurs
  - Liste erreurs (si applicable)
5. Gestion erreurs : Si échec lecture Excel, notifier admin

# Bonnes Pratiques - Nommage

## Flux

- Descriptif et clair
- Inclure action principale
- Exemples :
  - "Approbation demandes de congés"
  - "Notification nouveaux tickets support"
  - "Sync utilisateurs AD vers SharePoint"

## Actions dans Flux

- Renommer actions par défaut
- Descriptif de ce que fait l'action
- Exemples :
  - "Obtenir détails employé" au lieu de "Obtenir un élément"
  - "Envoyer email approbation manager" au lieu de "Envoyer un e-mail"

## Variables

- CamelCase ou snake\_case
- Descriptif
- Exemples : MontantTotal, ListeErreurs, compteur\_elements

## Scopes

- Nommer selon fonction logique
- "Validation", "Traitement", "Notifications"

## Connexions

- Identifier environnement
- Exemple : "SharePoint-Prod", "SQL-Dev"

# Bonnes Pratiques - Organisation

## Structuration

### Scopes

Grouper actions logiquement

- Scope : "Validation données"
- Scope : "Traitement principal"
- Scope : "Gestion erreurs"
- Scope : "Notifications"

### Commentaires

Utiliser action "Composer" pour documenter

```
Composer : "Cette section traite les approbations hiérarchiques..."
```

## Solutions

Pour projets complexes ou multi-environnements

- Développement
- Test
- Production

## Versioning

Exporter flux régulièrement (backups)

# Bonnes Pratiques - Maintenance

## Monitoring

- Vérifier historique régulièrement
- Configurer alertes échecs

## Analytics

- Temps d'exécution moyen
- Taux de réussite
- Actions les plus longues

## Documentation

- Documenter logique métier
- Diagrammes de flux
- Contacts et ownership

## Tests

- Tester flux après modifications
- Environnements de test séparés
- Données de test

## Archivage

Désactiver flux obsolètes (ne pas supprimer immédiatement)

## Backup

- Exporter flux régulièrement
- Versionner dans source control
- Documenter changements

# Tests et Débogage

## Outils de Test

### Test de Flux

Bouton "Tester" dans éditeur

- Mode manuel : Déclencher manuellement
- Mode automatique : Attendre déclencheur réel

### Historique d'Exécution

- Voir toutes exécutions passées
- Détails de chaque action
- Entrées et sorties
- Durées d'exécution

## Débogage

### Action Compose

Afficher valeurs intermédiaires

```
Compose : @variables('MaVariable')
```

### Vérifier Expressions

Tester expressions dans actions Compose avant utilisation réelle

### Tips

- Vérifier les entrées/sorties de chaque action
- Utiliser Peek Code pour voir le JSON

# Cas Pratique 1 : Onboarding Employés

## Scénario

Automatiser processus d'onboarding nouvel employé

## Workflow

```
1. Déclencheur : Nouvel employé ajouté dans SharePoint
  ↓
2. Parallel Branch
  ├── Branche 1 : Créer compte Microsoft 365
  ├── Branche 2 : Créer boîte email
  ├── Branche 3 : Ajouter à groupes Teams
  └── Branche 4 : Créer ticket IT matériel
  ↓
3. Envoyer email bienvenue avec :
  - Identifiants
  - Planning première semaine
  - Documents RH
  ↓
4. Créer tâches Planner pour manager :
  - Formation J1, J2, J3
  - Point hebdomadaire
  ↓
5. Planifier rappel J-1 : Email au buddy assigné
```

# Cas Pratique 2 : Gestion Tickets Support

## Scénario

Automatiser traitement et escalade tickets support

## Workflow

```
1. Déclencheur : Nouveau ticket (Microsoft Forms ou Email)
  ↓
2. Créer élément SharePoint avec détails ticket
  ↓
3. AI Builder : Analyser sentiment description
  ↓
4. Switch sur Priorité déterminée
  | Critique : Créer incident + SMS équipe + Email direction
  | Élevé : Assigner technicien senior + Email
  | Moyen : Assigner technicien disponible
  | Faible : Ajouter à file d'attente
  ↓
5. Do Until : Statut = "Résolu" ou 24h écoulées
  | Vérifier statut toutes les 2h
  | Si toujours ouvert après 24h : Escalader
  ↓
6. Enquête satisfaction (Forms) envoyée automatiquement
```

# Cas Pratique 3 : Workflow Approbation Achats

## Scénario

Approbation multi-niveaux selon montant

## Workflow

```
1. Déclencheur : Demande d'achat (SharePoint)
  ↓
2. Obtenir détails + pièces justificatives
  ↓
3. Condition selon montant
  | < 500€ : Approbation manager direct uniquement
  | 500-5000€ : Manager + Directeur département
  | > 5000€ : Manager + Directeur + Finance
  ↓
4. Boucle approbations (séquentiel ou parallèle)
  ↓
5. Condition : Toutes approbations reçues ?
  | Oui approuvé
  |   | Créer bon de commande
  |   | Envoyer vers ERP (SQL/API)
  |   | Notifier demandeur + achats
  | Au moins un rejet
  |   | Mettre à jour statut
  |   | Notifier demandeur avec raisons
```



# Projet Final : Automatisation Complète

## Brief Projet

Créer un système complet de gestion de demandes de formation

## Fonctionnalités Requises

1. **Soumission** : Formulaire Forms → SharePoint
2. **Validation Budget** : Vérifier budget disponible département
3. **Approbation Manager** : Workflow approbation
4. **Traitement RH** : Si approuvé, RH contacte organisme
5. **Inscription** : Créer événement calendrier employé
6. **Rappels** : J-7 et J-1 avant formation
7. **Feedback** : Enquête post-formation automatique
8. **Reporting** : Email mensuel récap formations

## Contraintes

- Gestion erreurs robuste
- Optimisation performance
- Documentation complète

# Architecture du Projet

Forms → SharePoint "Demandes Formation"

↓

Power Automate : Flux Principal

- └ Validation données
- └ Vérification budget (Excel ou SQL)
  - └ Si insuffisant : Rejet automatique + notification
- └ Approbation Manager
  - └ Timeout 5 jours → Escalade
- └ Approbation RH (si > 1000€)
- └ Si approuvé
  - └ Créer événement Outlook
  - └ Notification employé
  - └ Déclencher flux rappels (flux planifié séparé)
- └ Si rejeté : Notification + archivage

Flux Planifié : Rappels Formations

- └ Quotidien : Vérifier formations J-7 et J-1
  - └ Envoyer emails rappel

Flux Planifié : Enquêtes Post-Formation

- └ Quotidien : Vérifier formations terminées hier
  - └ Envoyer Forms satisfaction

Flux Planifié : Reporting Mensuel

- └ Premier du mois : Compiler stats et envoyer

# Ressources et Documentation

## Documentation Officielle

- [docs.microsoft.com/power-automate](https://docs.microsoft.com/power-automate)
- Tutoriels guidés
- Référence expressions
- Limites et quotas

## Communauté

- Power Automate Community (forums)
- Power Users (groupe utilisateurs)
- YouTube : Shane Young, Reza Dorrani

## Templates

- Bibliothèque de modèles intégrée
- GitHub : Exemples communauté

## Certifications Microsoft

- PL-900 : Power Platform Fundamentals
- PL-200 : Power Platform Functional Consultant
- PL-400 : Power Platform Developer

## Blogs Recommandés

- Microsoft Power Platform Blog
- Power Automate Blog officiel

# Dépannage Problèmes Courants

## Problème : Flux ne se déclenche pas

- Vérifier permissions connexions
- Vérifier conditions déclencheur
- Vérifier statut flux (activé ?)
- Vérifier quotas pas atteints

## Problème : Action échoue

- Vérifier historique d'exécution (erreur exacte)
- Vérifier connexion valide
- Vérifier expressions correctes
- Retry avec données test

## Problème : Performance lente

- Optimiser boucles
- Réduire appels API
- Utiliser filtres
- Paralléliser actions indépendantes

## Problème : Erreur d'expression

- Vérifier syntaxe
- Tester dans Compose avant utilisation
- Vérifier types de données
- Vérifier les valeurs nulles

# Checklist Avant Mise en Production

## Validation Technique

- Tests avec données réelles validés
- Gestion erreurs implémentée
- Performance acceptable
- Quotas et limites respectés
- Connexions utilisent comptes de service

## Validation Métier

- Workflow approuvé par stakeholders
- Documentation complète
- Formation utilisateurs effectuée
- Plan de support défini

## Sécurité

- Principe moindre privilège respecté
- DLP politiques conformes
- Audit logging activé
- Données sensibles protégées

## Monitoring

- Alertes configurées
- Propriétaires identifiés
- Procédure escalade définie
- Backup du flux effectué

# Évolutions et Nouveautés Power Automate

## **Copilot in Power Automate**

Assistance IA pour création flux

- Décrire besoin en langage naturel
- Génération automatique flux
- Suggestions optimisations

## **Process Mining**

Découvrir opportunités automatisation

- Analyser processus existants
- Identifier inefficacités
- ROI potentiel

## **Desktop Flows (RPA)**

Automatisation applications legacy

- Enregistrement d'actions
- Scripting avancé
- Intégration avec cloud flows

## **Intégration Croissante IA**

AI Builder de plus en plus puissant

## **Cloud Flows v2**

Nouvelle architecture plus performante

# Récapitulatif de la Journée

## Ce que nous avons couvert

### **Matin**

- Fondamentaux Power Automate et Power Platform
- Types de flux et use cases
- Déclencheurs et actions
- Connecteurs Microsoft 365
- Expressions et manipulation données
- Exercice : Automatisation emails

### **Après-midi**

- Logique conditionnelle (if/else, switch)
- Boucles (apply to each, do until)
- Workflows d'approbation
- Variables et JSON
- Gestion d'erreurs robuste
- Optimisation et best practices
- AI Builder

