

## PL-400 Reference Material

### Power Platform Formulas

#### Canvas Apps Formulas

##### 1. If

- **Description:** Evaluates a condition and returns one value if true, another if false.
- **Example:** If(Slider1.Value > 50, "High", "Low")

##### 2. Set

- **Description:** Assigns a value to a global variable.
- **Example:** Set(UserInput, TextInput1.Text)

##### 3. Navigate

- **Description:** Moves to a different screen.
- **Example:** Navigate(Screen2, ScreenTransition.Fade)

##### 4. Filter

- **Description:** Returns a subset of records based on a condition.
- **Example:** Filter(Products, Price > 100)

##### 5. LookUp

- **Description:** Finds the first record matching a condition.
- **Example:** LookUp(Products, ProductID = "P001", ProductName)

##### 6. Collect

- **Description:** Adds data to a collection.
- **Example:** Collect(MyCollection, {Name: TextInput1.Text, Date: Today()})

##### 7. Patch

- **Description:** Updates or creates a record in a data source.
- **Example:** Patch(Products, LookUp(Products, ProductID = "P001"), {Price: 150})

##### 8. Concatenate

- **Description:** Combines text strings.
- **Example:** Concatenate(TextInput\_FirstName.Text, " ", TextInput\_LastName.Text)

#### 9. Sum

- **Description:** Adds numeric values in a column.
- **Example:** Sum(Products, Price)

#### 10. Notify

- **Description:** Displays a temporary notification.
- **Example:** Notify("Record saved successfully", NotificationType.Success)

### Model-Driven Apps Formulas

#### 1. Calculated Field (Total Cost)

- **Description:** Computes a value from other fields.
- **Example:** Quantity \* UnitPrice

#### 2. Business Rule (Conditional Logic)

- **Description:** Applies logic to show/hide fields or set values.
- **Example:** Condition: Status = "Closed", Action: Hide Comments field

#### 3. Rollup Field (Sum of Related Records)

- **Description:** Aggregates data from related records.
- **Example:** SUM of EstimatedValue from Opportunities

#### 4. DateDiff

- **Description:** Calculates the difference between two dates.
- **Example:** DIFFINDAYS(OrderDate, DeliveryDate)

#### 5. Concat

- **Description:** Combines text fields.
- **Example:** CONCAT(FirstName, " ", LastName)

### Advanced Canvas Apps Formulas

#### 1. ForAll with Patch

- **Description:** Loops through a table to update records.
- **Example:**

ForAll(Filter(Tasks, DueDate < Today() && Status = "Pending"), Patch(Tasks, ThisRecord, {Status: "Overdue"}))

## 2. With

- **Description:** Defines temporary variables for calculations.
- **Example:**

With({DiscountedPrice: Price \* 0.9}, Patch(Orders, LookUp(Orders, OrderID = "O123"), {FinalPrice: DiscountedPrice}))

## 3. Sequence and Table

- **Description:** Generates a dynamic table from a sequence.
- **Example:**

ClearCollect(NumberTable, ForAll(Sequence(10), {ID: Value, Description: "Item " & Value}))

## 4. Match and Substitute

- **Description:** Parses and reformats text using regex.
- **Example:**

With({PhoneMatch: Match(TextInput1.Text, "\d{3}-\d{3}-\d{4}"),  
If(!IsBlank(PhoneMatch.FullMatch), Substitute(PhoneMatch.FullMatch, "-", ""),  
"Invalid"))

## Advanced Model-Driven Apps Formulas

### 1. Calculated Field with Complex Logic

- **Description:** Uses conditionals for dynamic calculations.
- **Example:** IF(priority = "High" AND impact > 50, 100, IF(priority = "Medium", 50, 25))

### 2. Rollup Field with Filtering

- **Description:** Aggregates filtered related records.
- **Example:** SUM of EstimatedValue where Status = "Open"

### 3. Business Rule with Nested Conditions

- **Description:** Applies multiple actions with nested logic.
- **Example:** Condition: If Category = "Urgent" AND Status != "Closed",  
Actions: Show UrgentDetails, set as required

## AI Builder Integration

### 1. Canvas App: Sentiment Analysis

- **Description:** Analyzes text sentiment.
- **Example:**

```
Set(SentimentResult,  
'SentimentAnalysis'.Predict(TextInput1.Text).responsev2.predictionOutput.result.senti  
ment);
```

```
If(SentimentResult = "positive", Notify("Positive!", NotificationType.Success),  
Notify("Negative!", NotificationType.Error))
```

### 2. Model-Driven App: AI Prediction Field

- **Description:** Predicts outcomes using a trained model.
- **Example:** Field: ChurnRisk, Formula: AI Builder prediction on CustomerData

### 3. Canvas App: Object Detection

- **Description:** Identifies objects in images.
- **Example:**

```
Set(DetectedObjects,  
'ObjectDetection'.Predict(Camera1.Photo).responsev2.predictionOutput.labels);  
  
Collect(ObjectList, ForAll(DetectedObjects, {ObjectName: ThisRecord.label,  
Confidence: ThisRecord.confidence})))
```

## Custom Connectors

### 1. Canvas App: Custom API Call

- **Description:** Fetches data from an external API.
- **Example:** Set(WeatherData, WeatherAPI.GetWeather({city: "Seattle"}).temperature)

### 2. Model-Driven App: Custom Action

- **Description:** Triggers an external action via a workflow.
- **Example:** SMSAPI.SendMessage({to: "1234567890", message: "Record updated"})

Canvas App Formulas Reference Complete List: [Formula reference - canvas apps - Power Platform | Microsoft Learn](#)

Dataverse Formula reference Complete List: [Dataverse formula columns formula reference](#)

Dataverse Functions reference Complete List: [Dataverse functions formula reference](#)

Model-driven apps Formulas reference Complete List: [Model-driven apps formula reference](#)

Power Fx Formulas reference Complete List: [Get started with formulas in canvas apps - Power Apps | Microsoft Learn](#)

Power Platform CLI Formulas reference Complete List: [Power Platform CLI formula reference](#)

## Consolidated List of Formulas:

App Type	Category	Formula/ Feature	Description	Example Code/Configuration
Canvas	Basic	If	Conditional logic	If(Slider1.Value > 50, "High", "Low")
Canvas	Basic	Set	Sets a global variable	Set(UserInput, TextInput1.Text)
Canvas	Basic	Navigate	Moves to another screen	Navigate(Screen2, ScreenTransition.Fade)
Canvas	Basic	Filter	Filters records	Filter(Products, Price > 100)
Canvas	Basic	LookUp	Finds a specific record	LookUp(Products, ProductID = "P001", ProductName)
Canvas	Basic	Collect	Adds to a collection	Collect(MyCollection, {Name: TextInput1.Text, Date: Today()})
Canvas	Basic	Patch	Updates a record	Patch(Products, LookUp(...), {Price: 150})
Canvas	Basic	Concatenate	Combines text	Concatenate(TextInput_FirstName.Text, " ", TextInput_LastName.Text)
Canvas	Basic	Sum	Sums a column	Sum(Products, Price)
Canvas	Basic	Notify	Shows a notification	Notify("Record saved", NotificationType.Success)
Canvas	Advanced	ForAll with Patch	Bulk updates records	ForAll(Filter(Tasks, DueDate < Today()), Patch(Tasks, ThisRecord, {Status: "Overdue"}))
Canvas	Advanced	With	Scoped calculations	With({DiscountedPrice: Price * 0.9}, Patch(Orders, LookUp(...), {FinalPrice: ...}))
Canvas	Advanced	Sequence and Table	Dynamic table creation	ClearCollect(NumberTable, ForAll(Sequence(10), {ID: Value, Description: "Item " & Value}))
Canvas	Advanced	Match and Substitute	Text parsing with regex	With({PhoneMatch: Match(...)}, Substitute(PhoneMatch.FullMatch, "-", ""))
Model-Driven	Basic	Calculated Field	Computes a value	Quantity * UnitPrice
Model-Driven	Basic	Business Rule	Conditional field logic	If Status = "Closed", hide Comments
Model-Driven	Basic	Rollup Field	Aggregates related data	SUM of EstimatedValue from Opportunities

Model-Driven	Basic	DateDiff	Date difference	DIFFINDAYS(OrderDate, DeliveryDate)
Model-Driven	Basic	Concat	Combine s text fields	CONCAT(FirstName, " ", LastName)
Model-Driven	Advanced	Calculated Field (Complex)	Complex conditional logic	IF(priority = "High" AND impact > 50, 100, IF(priority = "Medium", 50, 25))
Model-Driven	Advanced	Rollup Field (Filtered)	Filtered aggregation	SUM of EstimatedValue where Status = "Open"
Model-Driven	Advanced	Business Rule (Nested)	Nested conditional actions	If Category = "Urgent" AND Status != "Closed", show UrgentDetails, set required
AI Builder	Canvas	Sentiment Analysis	Analyzes text sentiment	'SentimentAnalysis'.Predict(TextInput1.Text).responsev2...
AI Builder	Model-Driven	AI Prediction Field	Predicts outcomes	AI Builder prediction on CustomerData
AI Builder	Canvas	Object Detection	Identifies objects in images	'ObjectDetection'.Predict(Camera1.Photo).responsev2...
Custom Connector	Canvas	Custom API Call	Fetches external API data	Set(WeatherData, WeatherAPI.GetWeather({city: "Seattle"}).temperature)
Custom Connector	Model-Driven	Custom Action	Triggers external action	SMSAPI.SendMessage({to: "1234567890", message: "Record updated"})



## Common Status Codes

### Success Codes

#### 1. 200 OK

- **Description:** Request successful.
- **Canvas Example:** `Set(WeatherData, WeatherAPI.GetWeather(...));  
If(WeatherAPI.LastResponse.StatusCode = 200, Notify("Success",  
NotificationType.Success), Notify("Error", NotificationType.Error))`
- **Model-Driven Context:** Successful record retrieval.

#### 2. 201 Created

- **Description:** Resource created.
- **Canvas Example:** `Patch(Accounts, Defaults(Accounts), {Name:  
TextInput1.Text})`
- **Model-Driven Context:** New record saved.

#### 3. 204 No Content

- **Description:** Successful, no data returned.
- **Canvas Example:** `WeatherAPI.DeleteRecord({id: "123"});  
If(WeatherAPI.LastResponse.StatusCode = 204, Notify("Deleted",  
NotificationType.Success), Notify("Error", NotificationType.Error))`
- **Model-Driven Context:** Record deleted.

### Client Error Codes

#### 4. 400 Bad Request

- **Description:** Invalid request.
- **Canvas Example:** `Set(APIResult, CustomAPI.CreateItem({name: ""}));  
If(CustomAPI.LastResponse.StatusCode = 400, Notify("Invalid input",  
NotificationType.Error), Notify("Success", NotificationType.Success))`
- **Model-Driven Context:** Missing required field.

#### 5. 401 Unauthorized

- **Description:** Authentication failed.
- **Canvas Example:** `Set(DataResult, SecureAPI.GetData());  
If(SecureAPI.LastResponse.StatusCode = 401, Notify("Log in again",  
NotificationType.Error), Notify("Success", NotificationType.Success))`
- **Model-Driven Context:** User not authenticated.

## 6. 403 Forbidden

- **Description:** Permission denied.
- **Canvas Example:** `Set(RestrictedData, AdminAPI.GetRestrictedData());  
If(AdminAPI.LastResponse.StatusCode = 403, Notify("No permission",  
NotificationType.Error), Notify("Success", NotificationType.Success))`
- **Model-Driven Context:** Insufficient security role.

## 7. 404 Not Found

- **Description:** Resource not found.
- **Canvas Example:** `Set(Record, LookUpAPI.GetRecord({id:  
"nonexistent"})); If(LookUpAPI.LastResponse.StatusCode = 404,  
Notify("Not found", NotificationType.Error), Notify("Success",  
NotificationType.Success))`
- **Model-Driven Context:** Deleted record accessed.

## 8. 429 Too Many Requests

- **Description:** Rate limit exceeded.
- **Canvas Example:** `ForAll(Sequence(100), CustomAPI.GetData());  
If(CustomAPI.LastResponse.StatusCode = 429, Notify("Rate limit",  
NotificationType.Warning), Notify("Success", NotificationType.Success))`
- **Model-Driven Context:** Heavy workflow/API usage.

## Server Error Codes

### 9. 500 Internal Server Error

- **Description:** Server error.
- **Canvas Example:** `Set(Result, DataAPI.GetItems());  
If(DataAPI.LastResponse.StatusCode = 500, Notify("Server error",  
NotificationType.Error), Notify("Success", NotificationType.Success))`
- **Model-Driven Context:** Dataverse/plugin failure.

### 10. 503 Service Unavailable

- **Description:** Server unavailable.
- **Canvas Example:** `Set(Status, ServiceAPI.CheckStatus());  
If(ServiceAPI.LastResponse.StatusCode = 503, Notify("Maintenance",  
NotificationType.Warning), Notify("Success", NotificationType.Success))`
- **Model-Driven Context:** Dataverse maintenance.

## Learning Plan for PL-400

### JavaScript Basics

- Variables, functions, arrays, objects
- Event handling and DOM basics
- Resources:
  - [JavaScript Guide - JavaScript | MDN](#)

### TypeScript Basics

- Types, interfaces, classes
- Async/await and modules
- Resources:
  - [TypeScript: Handbook - The TypeScript Handbook](#)

### C# Basics

- Syntax, OOP, LINQ
- Exception handling
- Resources:
  - [Interactive tutorials - A tour of C# | Microsoft Learn](#)
  - [C# for Beginners | Microsoft Learn](#)
  - [C# docs - get started, tutorials, reference. | Microsoft Learn](#)

### HTML/CSS and Power Platform Intro

- HTML structure, CSS styling
- Overview of Power Apps, Power Automate, Dataverse
- Resources:
  - [HTML: HyperText Markup Language | MDN](#)
  - [CSS: Cascading Style Sheets | MDN](#)

### Core Power Platform Skills

#### Dataverse Basics

- Entities, relationships, FetchXML
- Resources:
  - [Microsoft Learn: Work with Dataverse](#)

#### Client-Side Scripting

- Xrm.WebApi, form context
- Writing OnLoad/OnChange scripts

## Plugin Development Basics

- Plugin pipeline, IOrganizationService
- Deploying a simple plugin
- Resources:
  - [Microsoft Learn: Write Plugins](#)

## Intermediate Development

### PCF Development

- PCF lifecycle, building a simple control
- Using Power Platform CLI
- Resources:
  - [Microsoft Learn: PCF Overview](#)
  - [PCF Gallery](#) – Community examples

### Debugging and Tools

- Debugging PCF (VS Code), plugins (Plugin Profiler)
- Git basics for version control
- Resources:
  - [Debugging Plugins](#)

### WebAPI Integration

- Calling Dataverse WebAPI with JavaScript
- External API basics (fetch, JSON)
- Resources:
  - [Microsoft Learn: WebAPI](#)

## Advanced Integrations

### Custom Connectors

- OpenAPI, authentication, testing
- Resources:
  - [Microsoft Learn: Custom Connectors](#)

## Advanced Plugins and Project Implementation

- Async plugins, external API calls
- Performance optimization
- Resources:
  - [Advanced Plugin Concepts](#)
  - [C# Async Programming](#)

### Build a Project

- Combine PCF, plugins, WebAPI, and a custom connector
- Example: A custom form with a PCF control that integrates with an external API
- Resources:
  - [Power Apps Community Forums](#) – For inspiration and troubleshooting
  - [Sample Projects](#)

### Power Platform Developer Tools:

Tool	Release notes	Description
<a href="#">Power Platform CLI</a>	<a href="#">link</a>	Microsoft Power Platform CLI is a simple, one-stop developer CLI that empowers developers and ISVs to perform various operations in Microsoft Power Platform related to environment lifecycle, authentication, and work with Microsoft Dataverse environments, solution packages, portals, code components, and more.
<a href="#">Power Platform Tools for Visual Studio Code</a>	<a href="#">link</a>	The Power Platform extension for Visual Studio Code makes it easy to manage Power Platform environments and allows the developer to create, build and deploy Power Platform solutions, packages and portals. Installing this extension will also make the latest Power Platform CLI (aka pac) available in your VSCode terminal.
<a href="#">Power Platform Tools for Visual Studio</a>	<a href="#">link</a>	Power Platform Tools for Visual Studio supports the rapid creation, debugging, and deployment of plug-ins. Other capabilities include development of custom workflow activities, web resources, integration technologies like Azure Service endpoints and webhooks, and more.
<a href="#">Visual Studio</a>		Add Power Platform as a connected service in Visual Studio to create a custom connector.

<a href="#"><u>connected service</u></a>		
<a href="#"><u>ALM developer tools</u></a>		ALM developer tools for Power Platform enables developers to create and manage projects related to testing, deployment, maintenance, and governance of solutions.
<a href="#"><u>Dataverse developer tools</u></a>		Dataverse developer tools enable developers to extend data platform capabilities in Power Platform.
<a href="#"><u>Power Apps Test Engine (preview)</u></a>	<a href="#"><u>link</u></a>	Power Apps Test Engine is an open source project that provides developers with an automated testing platform for canvas apps.