PowerStack

CHEAT SHEET - POWER FX

Power Apps (Power Fx) cheat sheet

The Power Fx functions you reach for daily, with one-line examples and the delegation notes that bite in production.

119	13	Power Fx
FUNCTIONS	CATEGORIES	LANGUAGE

v2026.07.26 - generated 26 July 2026
Latest version, searchable, always free: https://www.powerstack.live/cheat-sheets

App & navigation 11 functions

Set	Set a global variable `Set(gblUser, User().FullName)`
UpdateContext	Set screen-scoped variables `UpdateContext({locShowPanel: true})`
Navigate	Move to another screen `Navigate(scrDetail, ScreenTransition.Cover)`
Navigate (with context)	Pass data to the next screen `Navigate(scrDetail, None, {locId: ThisItem.ID})`
Back	Return to the previous screen `Back()`
Launch	Open a URL or app `Launch("https://learn.microsoft.com")`
Notify	Show a banner message `Notify("Saved", NotificationType.Success)`
Reset	Reset a control to default `Reset(txtSearch)`
Select	Trigger another control's OnSelect `Select(btnSave)`
Param	Read a launch parameter `Param("recordId")`
Exit	Close the running app `Exit()`

Creating & changing records 8 functions

Patch (update)	Update an existing record `Patch(Tasks, ThisItem, {Status: "Done"})`
Patch (create)	Create a new record `Patch(Tasks, Defaults(Tasks), {Title: txtTitle.Text})`
Patch (bulk)	Update many records at once `Patch(Tasks, colEdits)`
Defaults	Get a blank record for a source `Defaults(Tasks)`
Remove	Delete a specific record `Remove(Tasks, galTasks.Selected)`
RemoveIf	Delete records matching a condition `RemoveIf(Tasks, Status = "Archived")`
Update	Replace a record wholesale `Update(Tasks, oldRec, newRec)`

`UpdateIf`	Change fields on matching records `UpdateIf(Tasks, Priority = "Low", {Status: "Closed"})`

Forms 8 functions

Form behavior lives in functions; mode and errors live in properties on the form control.

`SubmitForm`	Save the form to its source `SubmitForm(frmEdit)`
`NewForm`	Put a form into new-record mode `NewForm(frmEdit)`
`EditForm`	Put a form into edit mode `EditForm(frmEdit)`
`ViewForm`	Put a form into read-only mode `ViewForm(frmEdit)`
`ResetForm`	Discard unsaved changes `ResetForm(frmEdit)`
`frm.Mode`	Read the current form mode `frmEdit.Mode = FormMode.New`
`frm.Error`	Read the last submit error `frmEdit.Error`
`frm.LastSubmit`	Get the record just saved `frmEdit.LastSubmit.ID`

Collections & local storage 5 functions

`Collect`	Add rows to a collection `Collect(colCart, {Item: "Pen", Qty: 2})`
`ClearCollect`	Clear, then refill a collection `ClearCollect(colStaff, Filter(Staff, Active))`
`Clear`	Empty a collection `Clear(colCart)`
`SaveData`	Persist a collection on the device `SaveData(colCart, "cart")`
`LoadData`	Reload a saved collection `LoadData(colCart, "cart", true)`

Filtering, search & sort 10 functions

This is where delegation matters most. Non-delegable functions silently cap at the row limit (default 500).

`Filter`	Return rows matching a condition `Filter(Staff, Department = "IT")`
`Search`	Substring search across columns `Search(Staff, txtFind.Text, "Name", "Email")`
`LookUp`	Return the first matching record `LookUp(Staff, ID = locId)`

Sort	Order by a single formula `Sort(Staff, LastName)`
SortByColumns	Order by named columns `SortByColumns(Staff, "LastName", SortOrder.Ascending)`
First / Last	Get the first or last record `First(Sort(Orders, Created, SortOrder.Descending))`
FirstN / LastN	Get the first or last N records `FirstN(Orders, 10)`
Distinct	Unique values from a column `Distinct(Staff, Department)`
CountRows	Count rows in a table `CountRows(Filter(Tasks, !Done))`
CountIf	Count rows matching a condition `CountIf(Tasks, Status = "Open")`

Conditional logic 10 functions

If	Return a value by condition `If(Score >= 50, "Pass", "Fail")`
Switch	Match one value against many `Switch(Status, "A", "Active", "I", "Inactive", "Unknown")`
IsBlank	Test for blank `IsBlank(txtName.Text)`
IsEmpty	Test a table for zero rows `IsEmpty(colCart)`
IsBlankOrError	Test for blank or error `IsBlankOrError(LookUp(Staff, ID = locId))`
Coalesce	First non-blank value `Coalesce(txtNick.Text, txtName.Text, "Guest")`
IfError	Catch and replace an error `IfError(Value(txtQty.Text), 0)`
And &&	All conditions true `And(Active, Approved)`
Or ||	Any condition true `Or(IsAdmin, IsOwner)`
Not !	Invert a condition `Not(IsBlank(txtName.Text))`

Text 16 functions

Concatenate &	Join text values `"Hi, " & txtName.Text`

Left / Right	Characters from one end `Left("PowerStack", 5)`
Mid	Characters from a position `Mid("PowerStack", 6, 5)`
Len	Count characters `Len(txtCode.Text)`
Upper / Lower	Change case `Upper("po-123")`
Proper	Title-case each word `Proper("jane doe")`
Trim	Trim ends + collapse inner spaces `Trim("  a   b  ")`
TrimEnds	Trim leading/trailing spaces only `TrimEnds("  a   b  ")`
Substitute	Replace text by match `Substitute("a-b-c", "-", "/")`
Replace	Replace text by position `Replace("2026", 1, 2, "19")`
StartsWith / EndsWith	Test a prefix or suffix `StartsWith(txtFile.Text, "PO-")`
Find	Position of one string in another `Find("@", txtEmail.Text)`
Split	Break text into a table `Split("a,b,c", ",")`
Concat	Join a table column into text `Concat(colTags, Value, ", ")`
IsMatch	Test against a regex pattern `IsMatch(txtZip.Text, "\d{5}")`
Char	Character from a code point `Char(10)`

Date & time 11 functions

Now	Current date and time `Now()`
Today	Current date at midnight `Today()`
DateAdd	Add a span to a date `DateAdd(Today(), 7, TimeUnit.Days)`
DateDiff	Span between two dates `DateDiff(StartDate, EndDate, TimeUnit.Days)`
DateValue	Parse text into a date `DateValue("2026-06-13")`

DateTimeValue	Parse text into a datetime `DateTimeValue("2026-06-13 09:30")`
Text (format)	Format a date as text `Text(Now(), "dd mmm yyyy")`
Date	Build a date from parts `Date(2026, 6, 13)`
Time	Build a time from parts `Time(9, 30, 0)`
Weekday	Day-of-week number `Weekday(Today())`
Year / Month / Day	Pull a part from a date `Month(Today())`

Math & aggregation 11 functions

Aggregations over large sources are mostly non-delegable - they run on the rows already pulled, not the whole table.

Sum	Total a column `Sum(Orders, Amount)`
Average	Mean of a column `Average(Orders, Amount)`
Min / Max	Smallest or largest value `Max(Orders, Amount)`
Count	Count numeric values in a column `Count(Orders.Amount)`
CountA	Count non-blank values in a column `CountA(Orders.Notes)`
Round	Round to N decimals `Round(12.345, 1)`
RoundUp / RoundDown	Round away from / toward zero `RoundUp(12.31, 1)`
Mod	Remainder after division `Mod(17, 5)`
Abs	Absolute value `Abs(-8)`
Rand	Random decimal 0-1 `Rand()`
RandBetween	Random whole number in range `RandBetween(1, 100)`

Table shaping 11 functions

AddColumns	Add a calculated column `AddColumns(Staff, "Full", First & " " & Last)`

DropColumns	Remove columns `DropColumns(Staff, "Email", "Phone")`
RenameColumns	Rename columns `RenameColumns(Staff, "First", "FirstName")`
ShowColumns	Keep only named columns `ShowColumns(Staff, "Name", "Department")`
GroupBy	Group rows by columns `GroupBy(Sales, "Region", "Rows")`
Ungroup	Flatten a grouped table `Ungroup(grouped, "Rows")`
ForAll	Evaluate a formula per row `ForAll(colEdits, Patch(Tasks, ThisRecord))`
With	Name temporary values inline `With({tax: 0.1}, total + total * tax)`
Sequence	Generate a numeric table `Sequence(5)`
Concurrent	Run actions in parallel `Concurrent(Refresh(A), Refresh(B))`
Table	Build an inline table `Table({n: 1}, {n: 2})`

Data conversion

7 functions

Value	Text to number `Value("123.45")`
Text	Number to formatted text `Text(1234.5, "#,##0.00")`
JSON	Serialize a value to JSON `JSON(colCart, JSONFormat.Compact)`
ParseJSON	Parse JSON into an untyped object `ParseJSON(txtPayload.Text)`
Boolean	Text to boolean `Boolean("true")`
GUID	Generate or parse a GUID `GUID()`
ColorValue	Hex string to a color `ColorValue("#742774")`

Records & scope references

5 functions

These are operators and references, not functions - but you use them constantly.

ThisItem	Current row inside a gallery `ThisItem.Title`

`ThisRecord`	Current row in ForAll / Filter `Filter(Orders, ThisRecord.Amount > 0)`
`Self`	The current control `Self.Fill`
`Parent`	The containing control `Parent.TemplateHeight`
`As`	Alias a record scope `ForAll(Orders As O, O.Amount * 1.1)`

Connectors (real patterns) 6 functions

SharePoint and Dataverse are added as data sources, then queried with the normal table functions. There is no SharePoint.GetItems() or Dataverse.Retrieve() in Power Fx - those are Power Automate connector actions.

`Office365Users.MyProfileV2`	Rich profile of the current user `Office365Users.MyProfileV2().jobTitle`
`Office365Users.SearchUserV2`	Search the directory `Office365Users.SearchUserV2({searchTerm: txtFind.Text}).value`
`Office365Outlook.SendEmailV2`	Send an email as the user `Office365Outlook.SendEmailV2("a@x.com", "Hi", "Body")`
`SharePoint list`	Query a list like any table `Filter('My List', Status.Value = "Open")`
`Dataverse table`	Query a table like any table `LookUp(Accounts, accountid = locId)`
`User`	Basic current-user info `User().Email`