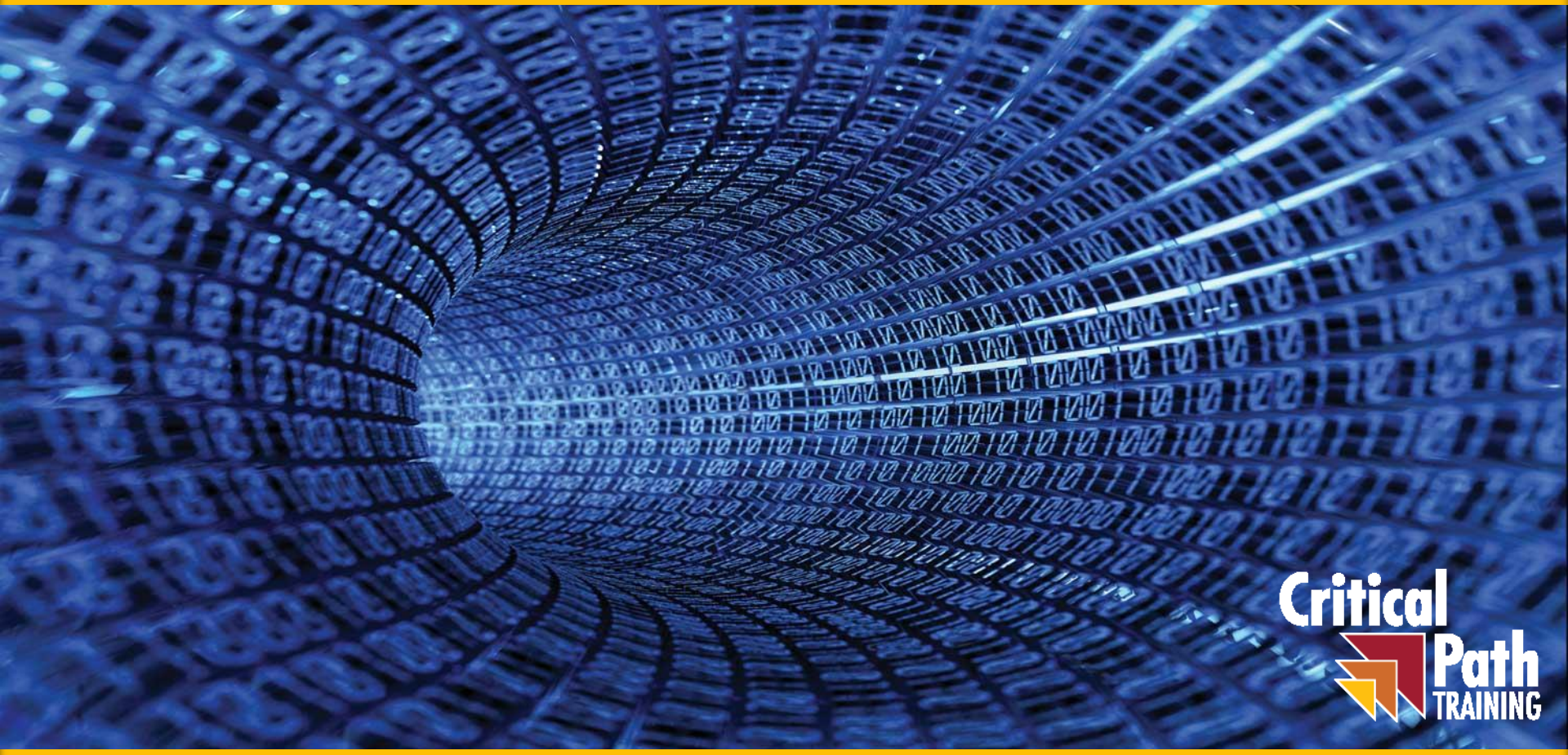
Designing PowerApps using Advanced Techniques



Agenda

- Working with Variables and Collections
- Managing Application State using Records and Tables
- Using Table Functions for Filtering, Sorting and Grouping
- Using Delegates to Filter, Sort and Aggregate Data
- Writing Imperative Logic in PowerApps Event Handlers
- Creating Custom Connectors

Input and Output Properties

- Input only
  - Value based on formula which cannot be used in other formulas
  - `Textinputbox1.Default` *(initial value)*
- Input/Output
  - Value based on formula which can be used in other formulas
  - `Textinputbox1.BackgroundColor`
- Output only
  - Value cannot be set by formula but can be used on other formulas
  - `Textinputbox1.Text` *(value always controlled by user input)*

Primitive Data Types

- Number: 3.141592
- Text: "Hello World”
- Boolean: True or False
- DateTime: 3/27/2018 12:00PM
- Date: 3/27/2018

- Any type can contain blank (i.e. null) values
  - Test for null value using IsBlank() function
  - Set null value using Blank() function

State Variables

- Collections
  - Created as tables at app scope
  - Managed using Collect, Clear and ClearCollect
  - Can be stored to local device using SaveData & LoadData
- Context variables
  - Created as primitive, record or table at screen scope
  - Managed using UpdateContext and Navigate
- Global variables
  - Created as primitive, record or table at app scope
  - Created and managed using Set function

Agenda

- ✓ Working with Variables and Collections
- ➢ Managing Application State using Records and Tables
- Using Table Functions for Filtering, Sorting and Grouping
- Writing Imperative Logic in PowerApps Event Handlers
- Using Delegates to Filter, Sort and Aggregate Data

Compound Data Types

- Record

```
{ FirstName: "Chuck", LastName: "Sterling" }
```

- Table

```
Table( { FirstName: "Chuck", LastName: "Sterling" },
       { FirstName: "Ted", LastName: "Pattison" } )
```

- Shorthand for Table with one column named value

```
[ "Moe", "Curly", Larry" ]
```

- Records and tables can be nested
  - Table can contain records of tables of records…
  - Record can contain tables of records of tables …

Agenda

- ✓ Working with Variables and Collections
- ✓ Managing Application State using Records and Tables
- ➢ Using Table Functions for Filtering, Sorting and Grouping
- • Writing Imperative Logic in PowerApps Event Handlers
- • Using Delegates to Filter, Sort and Aggregate Data

Table Functions

- Each table represented as a value
  - Table can be passed as argument
  - Table can be returned by function

- Functions that return tables
  - Filtering: `Filter`, `Search`
  - Sorting: `Sort`, `SortByColumns`
  - Shaping: `AddColumns`, `DropColumns`, `RenameColumns`
  - Grouping: `GroupBy`, `Ungroup`


DEMO
Filtering Data in a Table

Agenda

- ✓ Working with Variables and Collections
- ✓ Managing Application State using Records and Tables
- ✓ Using Table Functions for Filtering, Sorting and Grouping
- ➢ Writing Imperative Logic in PowerApps Event Handlers
- • Using Delegates to Filter, Sort and Aggregate Data

Events and State Changes

- Formulas for event properties can contain imperative logic
  - `OnSelect`, `OnVisible`, `OnStart`, etc.
- Imperative logic is used to take action
  - Set value of global variable or context variable
  - Add item to a collection
  - Navigate between screens
  - Submit data to server
- You can chain actions together with chaining operator (;)

```
Collect(Customers, {ID: NextCustomerId, FirstName: FirstNameTextInput.Text, LastName: LastNameTextInput.Text});
Reset(FirstNameTextInput);
Reset(LastNameTextInput);
Set(NextCustomerId, NextCustomerId+1);
```

App OnStart – Only on First Form

- Screen OnStart used to initialize app
  - Event only supported on startup screen for app
  - Can be used to initialize app-level variables

Imperative Functions

- Highlighted formulas can be used to perform actions

Abs	Collect	Day	HashTags	Max	Rand	Shuffle	TrimEnds
Acceleration	Color	Defaults	Hour	Mid	Refresh	Sin	Ungroup
Acos	ColorFade	Degrees	If	Min	Remove	Sort	Update
Acot	ColorValue	Disable	IfError	Minute	RemoveIf	SortByColumns	UpdateContext
AddColumns	Compass	Distinct	IsBlank	Mod	RenameColumns	Split	UpdateIf
And	Concat	Download	IsEmpty	Month	Replace	Sqrt	Upper
App	Concatenate	DropColumns	IsMatch	Navigate	Reset	StartsWith	User
Asin	Connection	EditForm	IsNumeric	NewForm	ResetForm	StdevP	Validate
Atan	Count	Enable	IsToday	Not	Revert	Substitute	Value
Atan2	Cos	EndsWith	Language	Notify	RGBA	SubmitForm	VarP
Average	Cot	Errors	Last	Now	Right	Sum	ViewForm
Back	CountA	EncodeUrl	LastN	Operators	Round	Switch	Weekday
Blank	CountIf	Exit	Launch	Or	RoundDown	Table	Year
Calendar	CountRows	Exp	Left	Param	RoundUp	Tan	
Char	DataSourceInfo	Filter	Len	Patch	SaveData	Text	
Choices	Date	Find	Ln	Pi	Search	Time	
Clear	DateAdd	First	LoadData	PlainText	Second	TimeValue	
ClearCollect	DateDiff	FirstN	Location	Power	Select	TimeZoneOffset	
Clock	DateTimeValue	ForAll	LookUp	Proper	Set	Today	
Coalesce	DateValue	GroupBy	Lower	Radians	ShowColumns	Trim	

Agenda

- ✓ Working with Variables and Collections
- ✓ Managing Application State using Records and Tables
- ✓ Using Table Functions for Filtering, Sorting and Grouping
- ✓ Writing Imperative Logic in PowerApps Event Handlers
- ➢ Using Delegates to Filter, Sort and Aggregate Data

Understanding Delegation

- **Delegation** is where you can delegate the processing of data to a data source before the data is pulled in an app
  - Key to building efficient apps
  - Minimizes the amount of data that needs to be brought into the app
- PowerApps includes a powerful set of functions to filter, sort, and shape tables of data
  - Equivalent to writing database queries

Types of Delegate Functions

- **Filter functions**
  - *Filter*, *Search*, and *LookUp* can be delegated
- **Sorting functions**
  - *Sort* and *SortByColumns* can be delegated
- **Aggregate functions**
  - *Sum*, *Average*, *Min*, and *Max* can be delegated
  - Not all data sources support this delegation

https://powerapps.microsoft.com/en-us/tutorials/delegation-list/

Small and Large Data Sets

- **Large data sets (over 500 records)**
  - Requires using data sources and formulas that can be delegated
  - Only way to keep your app performing well and ensures users can access all the information they need
- **Small data sets (less than 500 records)**
  - Can use any data source and formula
  - Processing can be done locally if the formula cannot be delegated

Summary

- ✓ Working with Variables and Collections
- ✓ Managing Application State using Records and Tables
- ✓ Using Table Functions for Filtering, Sorting and Grouping
- ✓ Writing Imperative Logic in PowerApps Event Handlers
- ✓ Using Delegates to Filter, Sort and Aggregate Data