Power Form Benutzerhandbuch

Version 2.0

Die Informationen in diesem Handbuch haben keinen Vertragscharakter und können ohne vorherige Ankündigung geändert werden.

Die in diesem Handbuch beschriebene Software wird im Rahmen einer Lizenzvereinbarung verkauft. Die Software darf nur gemäß den Bedingungen der Vereinbarung verwendet, kopiert oder reproduziert werden.

Dieses Handbuch darf ohne schriftliche Genehmigung von TEKLYNX Corporation SAS weder ganz noch teilweise in irgendeiner Form, egal mit welchen Mitteln oder zu welchen Zwecken, außer zur eigenen Verwendung des Käufers kopiert, reproduziert oder übertragen werden.

©2026 TEKLYNX Corporation SAS,
Alle Rechte vorbehalten.

Inhaltsverzeichnis

Einführung 4
Die Oberfläche 4
Die Menüleiste 5
Die Symbolleiste 5
Der Projektbaum 7
Das Fenster für Eigenschaften 8
Das Hauptfenster 9
Die Statusleiste 9
Einführung 10
Was sind Aktionen? 10
Skriptaktion 15
Fehlerbehebung 23
Arbeiten mit Formularvariablen 24
Aktion Objekt aktualisieren 24

Einführung

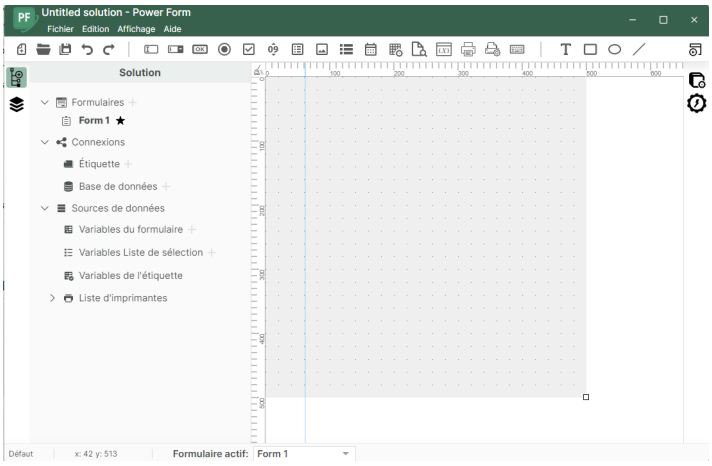
Power Form ist ein visuelles Gestaltungswerkzeug, mit dem Sie schnell Frontend-Anwendungen für die Eingabe von Daten und den Druck von Etiketten erstellen können.

Es beinhaltet unter anderem folgende Funktionen:

- Auswahl von Etiketten und Druckertyp
- Druckvorschau
- Seiten- und Druckereinstellungen
- Datenbankverbindungen
- Nutzung mehrerer Bildschirme/Formulare
- Vordefinierte Werte und Datensteuerung

Die Oberfläche

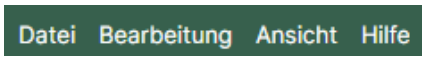
Die Oberfläche von Power Form ist benutzerfreundlich gestaltet, sodass Sie schnell mit der Software vertraut werden können. Sie umfasst eine Menüleiste, eine Symbolleiste, einen Projektbaum, ein Fenster für die Eigenschaften, das Hauptfenster und eine Statusleiste.



Die Menüleiste

Die Menüleiste besteht aus 4 Elementen: Datei, Bearbeiten, Ansicht und Hilfe. Wenn Sie auf ein Element klicken, wird dieses erweitert und zeigt eine neue Liste von Elementen an, über die der Benutzer verschiedene Funktionen ausführen kann. Um schneller auf die Menüpunkte zuzugreifen, drücken Sie die Alt-Taste zusammen mit dem Buchstaben des gewünschten Elements:

- Mit Alt+F öffnen Sie das Menü „Datei“.
- Mit Alt+E öffnen Sie das Menü „Bearbeitung“.
- Mit Alt+V öffnen Sie das Menü „Ansicht“.
- Mit Alt+H öffnen Sie das Menü „Hilfe“.





Die Symbolleiste

Die Symbolleiste wird normalerweise direkt unterhalb der Menüleiste angezeigt. Sie besteht aus Symbolen, die die am häufigsten verwendeten Softwarefunktionen und Objekte zur Bearbeitung eines Formulars darstellen. Das Anklicken dieser Symbole ermöglicht den schnellen Zugriff auf diese Funktionen.

Die Objekte bestehen aus 16 Steuerelementen und 4 Formen, die einem Formular hinzugefügt werden können: Texteingabefeld, Kombinationsfeld, Schaltfläche, Optionsfeld, Kontrollkästchen, numerisches Auf-/Ab-Eingabefeld, Gruppenfeld, Bildfeld, Listenfeld, Datumsauswahl, Datentabellenraster, Etikettenvorschau, Datenraster für Etikettenvariablen, Druckerauswahl, Druckeinstellungen, Text, Rechteck, Ellipse und Linie.

Jedes Steuerelement erfüllt einen bestimmten Zweck. Sie können mit vorgegebenen oder variablen Daten verknüpft und mit Aktionen verbunden werden. Um ein Steuerelement in Ihr Formular einzufügen, klicken Sie einfach auf das Steuerelement und ziehen Sie es per Drag & Drop in das Formular.

Steuerelement	Beschreibung
Texteingabe	Feld zur Dateneingabe
Kombinationsfeld	Zeigt eine Dropdown-Liste mit Daten an, die in den Eigenschaften definiert wurden
Schaltfläche	Ein aktives Steuerelement, das zur Ausführung bestimmter Funktionen/Aktionen dient
Optionsfeld	Zeigt eine Liste von Daten als Auswahlfeld-Ansicht an
Kontrollkästchen	Ein Steuerelement zum Aktivieren/Deaktivieren oder Auswählen/Abwählen vordefinierter Funktionen
Numerisch auf/ab	Eingabefeld nur für numerische Werte
Gruppenfeld	Ein nicht aktives Feld, das zur besseren Übersicht um eine Gruppe von Steuerelementen herum angeordnet ist
Bildfeld	Ein Feld, in das ein Bild importiert werden kann
Listenfeld	Zeigt eine Liste von Daten an, wie in den Eigenschaften definiert
Datumsauswahl	Eingabefeld für das in den Eigenschaften definierte Datumsformat
Datentabellenraster	Zeigt die in den Eigenschaften definierte Datenbanktabelle zusammen mit einigen grundlegenden Funktionen wie Filtern und Suchfeldern an
Etikettenvorschau	Ein Feld, in dem eine Vorschau des mit dem Formular verknüpften Etiketts angezeigt wird
Datenraster für Etikettenvariablen	Zeigt die Etikettenvariable „Druckzeit“ als Ansicht Name, Wert an
Druckerauswahl	Ermöglicht dem Benutzer, den aktiven Drucker auszuwählen und die Einstellungen des ausgewählten Druckers zu ändern

Steuerelement	Beschreibung
Druckeinstellungen	Ermöglicht es dem Benutzer, das ausgewählte Etikett zu drucken und die Etikettenanzahl, die Anzahl der Etikettenkopien, die Anzahl der Seitenkopien sowie die Schnittparameter festzulegen
Virtuelle Tastatur	Eine Bildschirmtastatur zur Dateneingabe.

Steuerelement	Beschreibung
Text	Inaktiver Text, der zur Anzeige des Namens eines Eingabefeldes verwendet wird
Rechteck	Zeigt ein Rechteck an
Ellipse	Zeigt eine Ellipse an
Linie	Zeigt eine Linie an

Der Projektbaum

Der **Projektbaum** ist eine Reihe von Registerkarten, die auf der linken Seite der Anwendung angezeigt wird. Er besteht aus Bereichen, die dazu dienen, allgemeine Konfigurationen auf das Projekt anzuwenden.



Die **Registerkarte Lösung** beinhaltet drei verschiedene Bereiche, die konfiguriert und auf das gesamte Formular angewendet werden können: Formulare, Verbindungen und Datenquellen.

Bereich	Beschreibung
Formulare	Eine Liste der in der Lösung verwendeten Formulare zur Darstellung verschiedener Arbeitsbereiche
Verbindungen	Eine Liste der verfügbaren Verbindungen, darunter API-Verbindung, Etikettenverknüpfung und Datenbankverbindung.
Datenquellen	Enthält unterschiedliche Arten von Variablen, die in der Lösung verwendet werden können. Es gibt 4 Arten von

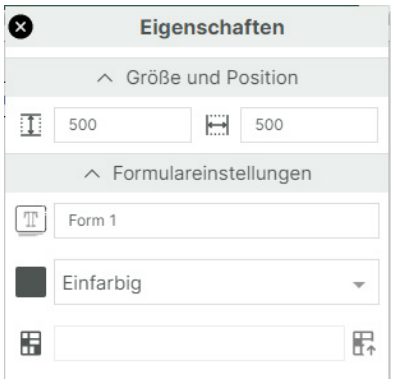
Bereich	Beschreibung
	Datenquellen: Formularvariablen, Auswahllistenvariablen, Etikettenvariablen und Druckerliste.

Die **Registerkarte Objekte** enthält eine Liste der Objekte, die auf dem aktiven Formular platziert sind.



Das Fenster für Eigenschaften

Im **Fenster für Eigenschaften** werden alle Eigenschaften des ausgewählten Steuerelements aufgelistet. Hier können Sie die Eigenschaften Ihrer Steuerelemente ändern, wie beispielsweise den Namen, die Farbe, die Größe oder die Aktionen. Es ist kontextsensitiv, sodass sich die Liste der Eigenschaften je nach ausgewähltem Steuerelement ändert. Wenn das Fenster angezeigt wird, ersetzt dies den üblichen Befehl im Kontextmenü.

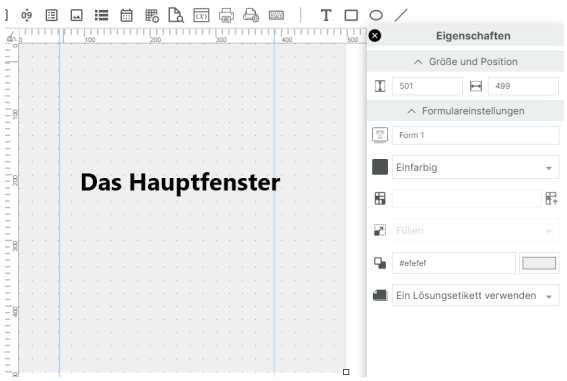


Einige Steuerelemente können einen Bereich Datentyp enthalten, in dem Benutzer dem Steuerelement eine Variable zuweisen oder eine Datenbank damit verknüpfen können. Sobald der Steuerwert zugewiesen wurde, wird er an den ausgewählten Datentyp gebunden, und alle Änderungen, die entweder am Steuerwert oder am Datentyp vorgenommen werden, werden synchronisiert.



Das Hauptfenster

Das in der Mitte angezeigte **Hauptfenster** ist Ihr Arbeitsbereich. Hier fügen Sie Steuerelemente hinzu und erstellen die Formulare, die Sie zur Eingabe von Daten für den Etikettendruck verwenden. Sobald ein Formular erstellt und konfiguriert wurde, können Sie über **Datei > Ausführen** das Formular ausführen und sich das Endergebnis anzeigen lassen.



Die Statusleiste

Die **Statusleiste** zeigt Ihnen Informationen zum aktiven Drucker, die Position Ihres Mauszeigers auf der x- und y-Achse sowie ein Dropdown-Menü für die Auswahl des aktiven Formulars an.



Einführung

Was sind Aktionen?

Aktionen sind Funktionen, die Schaltflächen zugewiesen werden sowie dem Laden und Entladen von Formularen dienen. In Power Form wurden 14 vordefinierte Aktionen in das Menü **Aktionen** aufgenommen, um Ihnen Zeit zu sparen und Programmieraufwand zu vermeiden. In einigen Fällen, in denen bestimmte Parameter konfiguriert werden müssen, wurden diese ebenfalls in der Konfiguration der Aktion vordefiniert.

Folgende vordefinierte Aktionen gibt es:

Aktion	**Popup für Bestätigung**
	Zeigt ein Dialogfeld zur Bestätigung an, in dem der Benutzer eine vordefinierte Mitteilung sowie Schaltflächen zum Bestätigen oder Abbrechen der Mitteilung sieht.
Parameter	**Titel**
	Ermöglicht dem Benutzer, den Text der Überschrift festzulegen.
	Mitteilung
	Hier kann der Benutzer den Haupttext der Nachricht festlegen.
	Schaltflächen
	Ein Dropdown-Menü mit verschiedenen Schaltflächenoptionen, die im Popup der Mitteilung angezeigt werden sollen.
	Symbol
	Ein Dropdown-Menü mit verschiedenen Symbolen, die im Popup der Mitteilung angezeigt werden sollen.

Aktion	**Variablenwert aktualisieren**
	Ermöglicht dem Benutzer die Aktualisierung von Variablen.
Parameter	**Variablenname**
	Der Benutzer kann den Namen der zu aktualisierenden Variablen auswählen.
	Neuer Wertetyp

	Dropdown-Liste der Wertetypen, die der zu aktualisierenden Variablen zugewiesen werden können.
	Option
	Ermöglicht dem Benutzer, einen neuen Wert zum bestehenden Wert der ausgewählten Variablen hinzuzufügen.
	Neuer Wert
	Feld, in dem der Benutzer einen neuen Wert für die ausgewählte Variable auswählen kann.

Aktion	**Formular anzeigen**
	Ermöglicht dem Benutzer, ein neues Formular zu öffnen.
Parameter	**Formular für Anzeige**
	Liste der Formulare, aus der der Benutzer auswählen kann, welches Formular angezeigt werden soll.
	Anzeigeverhalten
	Öffnet das ausgewählte Formular in einem neuen Fenster oder im selben Fenster.
	Option
	Ermöglicht dem Benutzer, ein zuvor geöffnetes Formular zu öffnen, sofern vorhanden.

Aktion	**Formular verlassen**
	Ermöglicht dem Benutzer, das Fenster zum Ausführen des Formulars zu schließen.

Aktion	**Drucker einrichten**
	Ermöglicht dem Benutzer, auf dem ausgewählten Drucker zu drucken und etwaige dem Dokument zugewiesene Drucker zu überschreiben.
Parameter	**Drucker einrichten**
	Ermöglicht es, einen Drucker als Standarddrucker festzulegen, der beim Anklicken der Schaltfläche verwendet wird, oder den aus einer Liste ausgewählten Drucker zu verwenden – sofern eine Variable zur Druckerauswahl verwendet und in den Formulareigenschaften definiert wurde.

Aktion	**Etikett drucken**
	Ermöglicht dem Benutzer zu drucken.
Parameter	**Etikettenanzahl**
	Sendet die Anzahl der zu druckenden Etiketten an den Drucker. Wenn das Dokument einen Zähler enthält, erhöht sich dieser entsprechend der Anzahl der Etiketten.
	Etikettenkopie
	Teilt dem Drucker mit, wie viele identische Exemplare jedes Etiketts gedruckt werden sollen.
	Seitenkopie
	Teilt dem Drucker mit, wie viele Exemplare jeder Seite gedruckt werden sollen.
	Zwischenschnitt
	Teilt dem Drucker mit, wie viele Etiketten vor jedem Schnitt gedruckt werden sollen.

Aktion	**Daten aus Datei lesen**
	Ermöglicht dem Benutzer, Daten aus Dateien zu lesen und deren Werte auf Variablen oder Steuerelemente anzuwenden.
Parameter	**Datei auswählen**
	Ermöglicht die Auswahl der zu lesenden Datei.
	Maximale Größe
	Legt die maximale Anzahl der Zeilen fest, die aus der Datei gelesen werden.
	Wertetyp anwenden
	Ermöglicht dem Benutzer auszuwählen, wofür die Daten aus der Datei verwendet werden sollen; dies kann eine Variable oder ein Steuerelement sein.
	Variable/Steuerelement auswählen
	Ermöglicht dem Benutzer auszuwählen, auf welches Steuerelement die Daten angewendet werden sollen.

Aktion	**Dialog öffnen**
	Ermöglicht dem Benutzer, Dialogfenster des Label Designers zu

	öffnen.
Parameter	**Dialog öffnen**
	Dropdown-Menü mit einer Liste der verfügbaren Dialogfenster, die geöffnet werden können.

Aktion	**Weblink öffnen**
	Ermöglicht dem Benutzer, den ausgewählten Internetlink zu öffnen.
Parameter	**Linkadresse**
	Eingabefeld für den Link, der geöffnet werden soll.

Aktion	**SQL-Abfrage ausführen**
	Ermöglicht dem Benutzer, eine SQL-Abfrage zur Ausführung an eine verbundene Datenbank zu senden.
Parameter	**Datenbank**
	Zur Auswahl der Datenbank, in der die SQL-Abfrage ausgeführt wird
Parameter	**SQL-Anweisung**
	Textbereich für den SQL-Code

Aktion	**Skriptanfrage ausführen**
	Ermöglicht dem Benutzer, nativen JavaScript-Code auszuführen, um Werte zu vergleichen, Variablen zu aktualisieren, Daten in eine Datei zu schreiben usw.
Parameter	**Skript**
	Textbereich für den Code
Parameter	**Kategorie**
	Dropdown-Menü mit einer Liste von Kategorienamen für Methoden, Konstanten, Operatoren usw.
Parameter	**Codeschnipsel**
	Liste der Codeschnipsel in der ausgewählten Kategorie
Parameter	**Hinzufügen**
	Schaltfläche zum Hinzufügen des ausgewählten Codeschnipsels zum Code-Textfeld
Parameter	**Validieren**
	Schaltfläche zum Überprüfen der Syntax des im Textfeld

	eingegebenen Codes

Aktion	**Datenbank aktualisieren**
	Ermöglicht dem Benutzer, eine Aktualisierung der Datenbank zu erzwingen
Parameter	**Daten nehmen aus**
	Ein Dropdown-Menü mit der Option „Datenbank“, um die Aktualisierung der Datenbank zu erzwingen, und „Vom Ergebnis der vorherigen Aktion“, um die Datenbank mithilfe der Filterung von SQL-Aktionen zu aktualisieren.

Aktion	**Weblink öffnen**
	Ermöglicht dem Benutzer, den angegebenen Internetlink zu öffnen.
Parameter	**Linkadresse**
	Feld für die Eingabe der zu öffnenden Internet-Adresse.

Aktion	**Fokus setzen**
	Ermöglicht dem Benutzer, den Fokus auf das ausgewählte Steuerelement zu setzen.
Parameter	**Steuerelement auswählen**
	Wählt das Steuerelement aus, auf das der Fokus gesetzt werden soll.

Aktion	**Objekt aktualisieren**
	Ermöglicht dem Benutzer, das ausgewählte Steuerelement zu aktualisieren.
Parameter	**Steuerelement auswählen**
	Wählt das zu aktualisierende Steuerelement aus.
Parameter	**Liste von Eigenschaften**
	Zeigt die Liste der Eigenschaften an, die für das ausgewählte Steuerelement aktualisiert werden können.

Skriptaktion

Mit der Aktion „Skript ausführen“ können Benutzer JavaScript-Code an UI-Elemente (wie Schaltflächen) in einer Power-Form-Anwendung binden. Auf diese Weise können Sie die Logik der App erweitern, indem Sie Skripte schreiben, die auf Benutzeraktionen reagieren.

Schritt-für-Schritt-Anleitung:

1. Element der Oberfläche auswählen
    1. Klicken Sie auf das Element der Oberfläche, dem Sie ein Skript zuweisen möchten (z. B. eine Schaltfläche).
    2. Tipp: Sie können Schaltflächen, Eingabefelder, Raster oder alles andere auswählen, was Ereignisse unterstützt.

2. Fenster Ereignisse öffnen
    1. Klicken Sie auf das Zahnrad-Symbol .
    2. Suchen Sie im Fenster „Ereignisse“ das Ereignis, mit dem Sie arbeiten möchten – z. B. „Klick“.
    3. Betätigen Sie die Schaltfläche „...“ neben dem Namen des Ereignisses, um es zu konfigurieren.

3. Skriptaktion hinzufügen
    1. Im sich öffnenden Dialogfeld „Ereignisaktionen konfigurieren“:
        a. Klicken Sie auf das Symbol „Skriptaktion ausführen“ .
        b. Die neue Aktion erscheint in der Liste.

4. Ihr Skript schreiben

    Geben Sie im Fenster „Eigenschaften“ Ihre benutzerdefinierte JavaScript-Logik ein.

    Beispiel:

    ```
    const a = 12;
    const b = 32;
    return b * a;
    ```

    * Dieser JavaScript-Codeschnipsel definiert zwei Konstanten: `a` mit dem Wert 12 und `b` mit dem Wert 32. Anschließend wird das Ergebnis der Multiplikation von `b` mit `a` zurückgegeben.

5. Überprüfen & Anwenden
    1. Klicken Sie auf „Validieren“, um Ihr Skript auf Syntaxfehler zu überprüfen.
    2. Wenn dies erfolgreich war, klicken Sie auf „Anwenden“, um die Aktion zu speichern.

Codeschnipsel verwenden

Die Skriptschnittstelle bietet eine Liste vordefinierter Codeschnipsel, die nach Kategorien geordnet sind. Diese Fragmente helfen Ihnen dabei, häufig verwendete Ausdrücke, Konstanten, Strukturen und Funktionen einzufügen, ohne dass Sie diese von Grund auf neu schreiben müssen.

So verwenden Sie ein Codeschnipsel:

1. Wählen Sie eine Kategorie aus dem Dropdown-Menü.
2. Wähen Sie einen Codeschnipsel aus dem zweiten Dropdown-Menü.
3. Klicken Sie auf „Hinzufügen“, um es in den Skript-Editor einzufügen.

Alle verfügbaren Codeschnipsel:

Snippet	**Description**
Math	
Math.sqrt(number)	Returns the square root of a number
Math.abs(number)	Returns the absolute value
Math.cos(number)	Returns the cosine of a number (radians)
Math.sin(number)	Returns the sine of a number (radians)
Math.ceil(number)	Rounds up to the nearest integer
Math.floor(number)	Rounds down to the nearest integer
Math.round(number)	Rounds to the nearest integer
Math.random()	Returns a random number between 0 and 1

Snippet	Description
Math.pow(a, b)	Raises a to the power of b
Math.max(a, b)	Returns the maximum of two values
Math.min(a, b)	Returns the minimum of two values
Math.log(x)	Natural logarithm (base e) of x
Math.exp(x)	Returns e to the power of x
JSON	
JSON.stringify(object)	Converts an object to a JSON string
JSON.parse(jsonString)	Parses a JSON string into an object
Variables	
let x = value;	Declares a variable with block scope
const PI = 3.14;	Declares a constant
let result;	Declares an uninitialized variable
var total = 0;	Declares a function-scoped variable
Constants	
Number.MAX_VALUE	Maximum numeric value
Number.MIN_VALUE	Smallest positive numeric value
undefined	Primitive indicating no value
null	Intentional absence of any object value
NaN	Not-a-Number value
Math.PI	The ratio of circumference to diameter

Snippet	Description
Math.E	Euler's number
Math.LN2	Natural log of 2
Math.LN10	Natural log of 10
Math.LOG2E	Base 2 log of e
Math.LOG10E	Base 10 log of e
Math.SQRT1_2	Square root of 0.5
Math.SQRT2	Square root of 2
Logical	
a > b	Greater than comparison
a < b	Less than comparison
a >= b	Greater than or equal to
a <= b	Less than or equal to
a && b	Logical AND
a || b	Logical OR
!a	Logical NOT
a != b	Inequality comparison
a !== b	Strict inequality comparison
TRUE	Boolean true
FALSE	Boolean false
Operators	

Snippet	Description
a + b	Addition
a - b	Subtraction
a * b	Multiplication
a / b	Division
a % b	Modulo
a ** b	Exponentiation
a += b	Add and assign
a -= b	Subtract and assign
a *= b	Multiply and assign
a /= b	Divide and assign
a == b	Equality
a === b	Strict equality
++a	Pre-increment
--a	Pre-decrement
Statements	
if (condition) { }	Conditional execution
else { }	Executes if condition is false
else if (condition) { }	Checks another condition
for (let i = 0; i < n; i++) { }	For loop
while (condition) { }	While loop

Snippet	Description
do { } while (condition);	Do-while loop
switch (expression) { case: break;}	Switch-case conditional
try { } catch (e) { }	Try-catch block
return value;	Returns a value
break;	Breaks out of loop
continue;	Skips to next loop iteration
Array	
array.push(item)	Adds item to end of array
array.pop()	Removes last item from array
array.map(x => x)	Transforms each array element
array.filter(x => x > 0)	Filters elements based on condition
array.find(x => x === 0)	Finds first match
array.reduce(...)	Reduces array to a single value
array.includes(value)	Checks if value exists in array
array.slice(start, end)	Returns a shallow copy of array
array.indexOf(value)	Returns index of value in array
Object	
Object.keys(obj)	Returns keys of object
Object.values(obj)	Returns values of object
Object.entries(obj)	Returns key-value pairs

Snippet	Description
Object.assign({}, source)	Copies values into a new object
obj.hasOwnProperty(key)	Checks if object has property
String	
str.toUpperCase()	Converts to uppercase
str.toLowerCase()	Converts to lowercase
str.includes("text")	Checks for substring
str.replace("", "")	Replaces text
str.split(",")	Splits string into array
str.trim()	Removes whitespace
str.slice(start, end)	Extracts part of string
Number	
Number(value)	Converts to number
num.toFixed(2)	Formats number to fixed decimals
num.toString()	Converts number to string
Boolean	
Boolean(value)	Converts to boolean
!value	Negation
!!value	Double negation (truth check)
MapSet	
const map = new Map();	Creates new map

Snippet	Description
map.set("key", value);	Sets key-value pair
map.get("key");	Gets value by key
map.has("key");	Checks key existence
map.delete("key");	Deletes key
const set = new Set();	Creates new set
set.add(value);	Adds value to set
set.has(value);	Checks value existence
set.delete(value);	Removes value from set
DateTime	
new Date("")	Creates date from string
dateVar.toISOString()	Converts to ISO string
dateVar.getTime()	Gets time in ms
Date.now()	Gets current timestamp
Timers	
setTimeout(() => {}, 1000);	Delays function execution
setInterval(() => {}, 1000);	Repeats function execution
clearTimeout(timeoutId);	Clears timeout
clearInterval(intervalId);	Clears interval
FileSystem	
fs.readFileSync("", "utf8")	Reads file content

Snippet	Description
fs.writeFileSync("", "")	Writes to file
fs.appendFileSync("", "")	Appends to file
const fd = fs.openSync("", "rw")	Opens file
fs.fsyncSync(fd);	Flushes file content
fs.closeSync(fd);	Closes file
fs.existsSync("")	Checks file existence
fs.readdirSync("")	Reads directory content
Functions	
function myFunction() { }	Declares a function
const add = () => null;	Declares an arrow function
myFunction();	Calls a function

Fehlerbehebung

Hinweis: `console.log()` ist in dieser Skriptumgebung nicht verfügbar. Um Fehler zu identifizieren, überprüfen Sie bitte die Fehlerprotokolle unter `%APPDATA%/TekLynx/web-forms/logs`

Problem	Lösung
Skript läuft nicht	Überprüfen Sie, ob Sie auf „Anwenden“ geklickt und die Aktion dem richtigen Ereignis zugeordnet haben
Überprüfung schlägt fehl	Überprüfen Sie die Syntax und stellen Sie sicher, dass alle Variablen deklariert sind

Tipps

1. Verwenden Sie *return*, um das Ergebnis an das Formularsystem zurückzugeben.
2. Kombinieren Sie mehrere Logikzeilen – alle JavaScript-Funktionen werden unterstützt.

Arbeiten mit Formularvariablen

Sie können Formularvariablen direkt in Ihrem Skript aktualisieren, indem Sie ihnen unter Angabe ihres Namens einen neuen Wert zuweisen. Wenn eine Formularvariable beispielsweise `candy_amount` heißt, können Sie ihren Wert wie folgt festlegen:

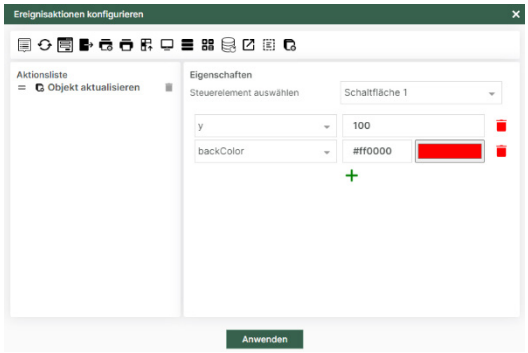
```
candy_amount = 12;
```

Die Skript-Engine erkennt diese Zuweisung automatisch und weist den Wert der entsprechenden Formularvariablen in der Datenquelle nach Abschluss der Skriptausführung zu.

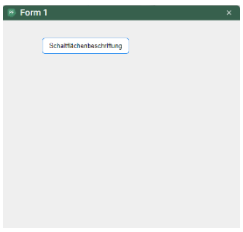
Sie können das Ergebnis einer `return`-Anweisung aus einer Skriptaktion auch in der nachfolgenden Aktion verwenden, indem Sie in der Konfiguration der nächsten Aktion auf das „Ergebnis der vorherigen Aktion“ verweisen.

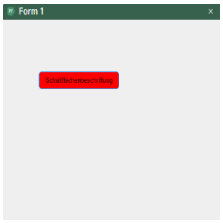
Aktion Objekt aktualisieren

Mit der Aktion „Objekt aktualisieren“ kann der Benutzer die Eigenschaften eines ausgewählten Steuerelements zur Laufzeit ändern, wenn ein Ereignis ausgelöst wird. Nachdem diese Aktion im Dialogfeld Ereignisaktionen konfigurieren hinzugefügt wurde, wählt der Benutzer unter „Ausgewähltes Steuerelement“ das Zielsteuerelement aus und legt anschließend im Bereich „Liste der Eigenschaften“ eine oder mehrere Eigenschaften fest, die aktualisiert werden sollen.



Dies kann genutzt werden, um das Verhalten oder das Erscheinungsbild von Steuerelementen dynamisch anzupassen, beispielsweise durch die Änderung von Position, Größe, Text, Farbe, Sichtbarkeit oder anderen unterstützten Einzelwert-Eigenschaften auf Grundlage von Benutzerinteraktionen oder dem Ergebnis einer vorherigen Aktion.





Einschränkungen: Diese Aktion kann nicht zum Festlegen von Eigenschaften verwendet werden, die mehrere Optionen oder komplexe Sammlungen erfordern, wie beispielsweise Elemente für ein Listenfeld oder eine Optionsgruppe. Außerdem kann damit der aktive Drucker nicht festgelegt oder geändert werden.



France
+33 (0) 562 601 080

Germany
+49 (0) 2103 2526 0

Singapore
+65 6908 0960

United States
+1 (414) 837 4800

Copyright 2026 TEKLYNX Corporation SAS. All rights reserved. LABEL MATRIX, LABELVIEW, CODESOFT, LABEL ARCHIVE, SENTINEL, PRINT MODULE, BACKTRACK, TEKLYNX CENTRAL, TEKLYNX, and Barcode Better are trademarks or registered trademarks of TEKLYNX Corporation SAS or its affiliated companies. All other brands and product names are trademarks and/or copyrights of their respective owners.

www.teklynx.com