

# The ultimate Salesforce Flow Cheatsheet

Salesforce Flows can be hard to master. This cheatsheet has all the information you need to know before you can master flows

Salesforce flows can be invoked by a number of different events. Choose the correct flow type based on the event that should invoke the flow.

Flow Type				
Type	Invoked when	Runs In	Use when	Example
<b>Screen Flow</b>	Launched by a user (via a button)	The user interface	You need users to follow a UI-based business process	Guide users through the steps required to complete a booking on behalf of a customer
<b>Record-Triggered Flow</b>	A record is created, updated or deleted	The background	You need to automate something in response of a record being modified by a user	Update the opportunity name whenever the amount field is updated
<b>Autolaunched Flow</b>	Called by another Flow, Apex, or the REST API	The background	You want to reuse flow logic in other areas of your org	Create a flow that defines how a child opportunity is created. Re-use this logic across flows, apex, and Workato (example of REST API)
<b>Scheduled-Triggered Flow</b>	The current time matches a specified time and frequency	The background	You want automation to run based on a defined schedule, i.e every Friday at 10:00PM	Automatically go through open cases every Friday night and update the "On-call manager"
<b>Platform Event-Triggered Flow</b>	A platform event message is received	The background	Your org is already using platform events and an event-driven architecture, and you want plug flows into the mix	A platform event is created when an opportunity is closed-won. You want a flow to run in the background that is going to email the account manager.

Record-triggered flows can execute logic on the record that triggered the flow or on related records. Choose the correct record-triggered flow type based on what records should be modified.

Record-Triggered Flow Type			
Type	Use when	Example	Considerations
<b>Before save</b>	You need to update the same record that triggered the flow	When an opportunity amount is updated, update the opportunity name to reflect the new amount.	<ul style="list-style-type: none"> <li>The flow can't perform actions other than updating the triggering record's field values.</li> <li>The flow can't update values in records that are related to the triggering record.</li> <li>Only these elements are supported: Assignment, Decision, Get Records and Loop.</li> </ul>
<b>After save</b>	You need to update or create other records related to the record that triggered the flow	When an opportunity is closed-won, update all the opportunity line items and send an email to the account manager.	It can be used to update the same record that triggered the flow, but it's not efficient and will consume governor limits.

The core logic of a flow is defined by Flow Elements.

Flow Elements		
Name	Use when	Example
<b>Assignment</b>	You need to assign values to a variable on the flow. It could be a record variable or a user-defined variable	The flow creates a new opportunity. Use the Assignment element to define the values of the new opportunity (name, stage, amount, etc.)
<b>Decision</b>	You have multiple execution paths based on the conditions of the record. Also known as "if/else" logic.	<p>If the opportunity stage is Closed Won, execute X,Y and Z actions.</p> <p>If the opportunity stage is Closed Lost, send an email to the account manager.</p>
<b>Loop</b>	You have a collection of records and you want to execute some action on each individual record.	The flow has a list of opportunity line items. You want to go through all of them, and for each, calculate the amount based on the currency selected on the opportunity.
<b>Collection Sort</b>	You have a collection of records and you want to sort them based on their attributes.	<p>You want to sort a list of opportunities by their amount. After sorting, the opportunity with the highest amount will be the first one on the list.</p> <p>The opportunity with the lowest amount will be the last one on the list.</p>

## Flow Elements

Name	Use when	Example
<b>Collection Filter</b>	You have a collection of records and you want to create another collection that only includes certain records from the original collection, based on some conditions.	You have a collection of all opportunities for a given account. You want to create another collection that only has those opportunities where the amount is greater than 100.
<b>Create Records</b>	You want to create records when the flow is executed	You have a flow that fires when an opportunity is Closed-won. Use a "Create Records" element to automatically <b>create</b> a new Contract record.
<b>Update Records</b>	You want to update records when the flow is executed. It should not be used to update the record that triggered the flow (you should use before-save flows for that).	You have a flow that fires when an opportunity is Closed-won. You want to now <b>update</b> the parent account to reflect the amount of the opportunity.
<b>Get Records</b>	You want to query or get records that are somehow related to the record that triggered the flow.	<p>You have a flow that fires when an opportunity is Closed-won.</p> <p>You use the "Get Records" element to <b>get</b> all the opportunity line items related to the opportunity.</p> <p>You then use this collection of records for further processing.</p>
<b>Delete Records</b>	You want to delete records when the flow is executed	You have a scheduled flow that runs every Friday night looking for duplicate records. Once the duplicate records are found, you use a "Delete Records" element to <b>delete</b> them.
<b>Action</b>	You want the flow to execute an action that exists outside of the flow, typically used to invoke apex code.	<p>You have an apex class that has the logic to create an upsell opportunity. This logic must be reused across all areas of your org.</p> <p>When an opportunity is Closed-won, you call an action to invoke the apex logic that will create an upsell opportunity.</p>
<b>Subflow</b>	<p>You have an existing flow that contains logic for a well-defined action, you want to reuse this logic in other flows rather than writing it all over again.</p> <p>It can also be used to simplify your flows by breaking them down into logic units.</p>	<p>You have a flow that has the logic to create an upsell opportunity.</p> <p>When an opportunity is Closed-won, you call this sub-flow, rather than creating the same logic all over again.</p>

Resources are the building blocks of the logic in your flow.

Flow Resources		
Type	Use when	Example
<b>Variable</b>	<p>You need to store a value that can be read and updated throughout the flow.</p> <p>Typically used to store the record that triggered the flow, or a set of related records.</p> <p>Can also be used to store user-defined variables.</p>	<p>You have a "Get Records" element that gets all the opportunity line items of an opportunity.</p> <p>The records are stored in a collection variable of type Record.</p>
<b>Constant</b>	<p>You need to store a value that can be read throughout the flow but should not be changed once set.</p>	<p>You create a constant to store the maximum opportunity amount that should trigger the creation of an upsell opportunity.</p> <p>This value can be read throughout the flow, but it should not be changed.</p> <p>As the name implies, it's a variable that remains constant throughout the flow execution.</p>
<b>Formula</b>	<p>You need to calculate a value based on the values of a record. Similar to a formula field.</p> <p>The resulting value can then be used throughout the flow.</p>	<p>You create a formula that returns a "Yes" or "No" value based on the opportunity amount.</p> <p>Later in the flow, you can use this formula in a "Decision" element to execute different paths of logic.</p>
<b>Text Template</b>	<p>You need to create text that can be used later in the flow.</p>	<p>You create a text template that organizes the opportunity details in a simplified format.</p> <p>You later use this text template as the text for an email alert that goes out to the account manager.</p>



Visualize your Salesforce flows like never before

[Try our forever-free product now](#)