Dictionary

Drag Assign on sequence page

(x) Assign   ⋮ ⌃

Save to	Value to Save
{§} dict ⊕	= {§} New Dictionary (Of Int32, String)

↓

(x) Assign   ⋮ ⌃

Save to	Value to Save
{§} dict(301) ⊕	= {§} "RPA" ↗ ⊕

↓

(x) Assign   ⋮ ⌃

Save to	Value to Save
{§} dict(302) ⊕	= {§} "IOT" ↗ ⊕

↓

(x) Assign   ⋮ ⌃

Save to	Value to Save
{§} dict(303) ⊕	= {§} "NLP" ↗ ⊕

↓

(x) Assign   ⋮ ⌃

Save to	Value to Save
{§} dict(304) ⊕	= {§} "ML" ↗ ⊕

↓

(x) Assign   ⌃

Save to	Value to Save
{§} dict(306) ⊕	= {§} "CLON" ↗ ⊕

↓

↓

```
[☰] For Each Current Key Value Pair Of Number And Text  ↗
    ┌──────────────────────────────────────────────────┐
    │  In *                                            │
    │  ┌────────────────────────────────────────────┐  │
    │  │ {} │ dict                            ↗  ⊕ │  │
    │  └────────────────────────────────────────────┘  │
    └──────────────────────────────────────────────────┘
    ↳ Body
    ┌──────────────────────────────────────────────────┐
    │  ✎ Write Line                            ⋮  ↗   │
    │  Text                                            │
    │  ┌────────────────────────────────────────────┐  │
    │  │ {} "the dictionary items" +                │  │
    │  │    CurrentKeyValuePairOfNumberAnd          │  │
    │  │       Text. ToString                        │  │
    │  └────────────────────────────────────────────┘  │
    └──────────────────────────────────────────────────┘
```
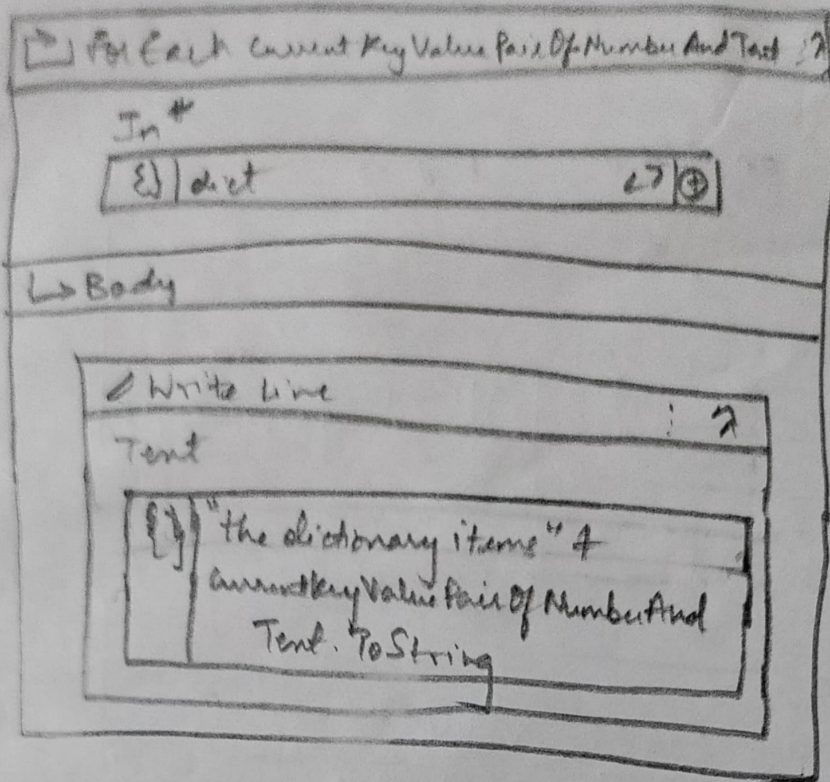
Output.

the dictionary items[301, RPA]
the dictionary items [302, TOP]
the dictionary items [303, NLP]
the dictionary items [304, AML]
the dictionary items [305, ECON]

Collections - List

- Drag Assign activity
  value as → mew list (of string)
- Drag 'Append items to collection'

```
┌─────────────────────────────────────┐
│ Append items to collection  ⋮        │
│ ┌─────────────────────────────────┐ │
│ Collection                          │
│ ┌─────────────────────────────┐     │
│ │ List                    ∠7 ⊘ │     │
│ └─────────────────────────────┘     │
│ Items*                              │
│ ┌─────────────────────────────┐     │
│ │ 2 Items in c            ∠7   │     │
│ └─────────────────────────────┘     │
└─────────────────────────────────────┘
                              ↓
                          "iot"
                          "rpa"
```

- Append few more collections
- Remove from collection

```
┌─────────────────────────────────────┐
│ Collection                          │
│ ┌─────────────────────────────────┐ │
│ │ List                            │ │
│ └─────────────────────────────────┘ │
│ Remove cle                          │
│ ☑ specific team                     │
│ Item                                │
│ ┌─────────────────────────────┐     │
│ │ "rpa"                     ⊕ │     │
│ └─────────────────────────────┘     │
└─────────────────────────────────────┘
```

- Drag exists in collection

```
┌─────────────────────────────────────┐
│ Exists in collection       ⋮ ⬈      │
│ ┌─────────────────────────────────┐ │
│ Collection*                         │
│ ┌─────────────────────────────┐     │
│ │ List                    ∠7 ⊕ │     │
│ └─────────────────────────────┘     │
│ Item                                │
│ ┌─────────────────────────────┐     │
│ │ "iot"                     ⊕ │     │
│ └─────────────────────────────┘     │
└─────────────────────────────────────┘
```

- Drag Message Box

```
┌─────────────────────────────────────┐
│ Text                                │
│ ┌─────────────────────────────┐     │
│ │ "available"          ∠7 ⊠   │     │
│ └─────────────────────────────┘     │
```

- Drag for Each current Text
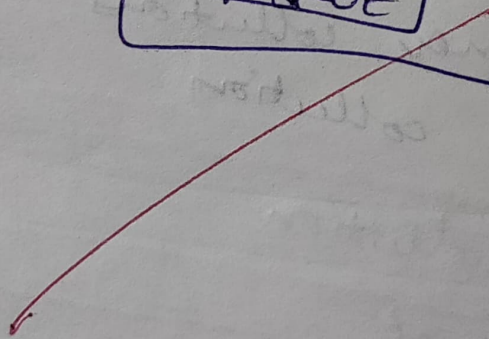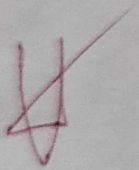

For Each
In
{} List
⬆
↳Body

Text
Current Text
⬆
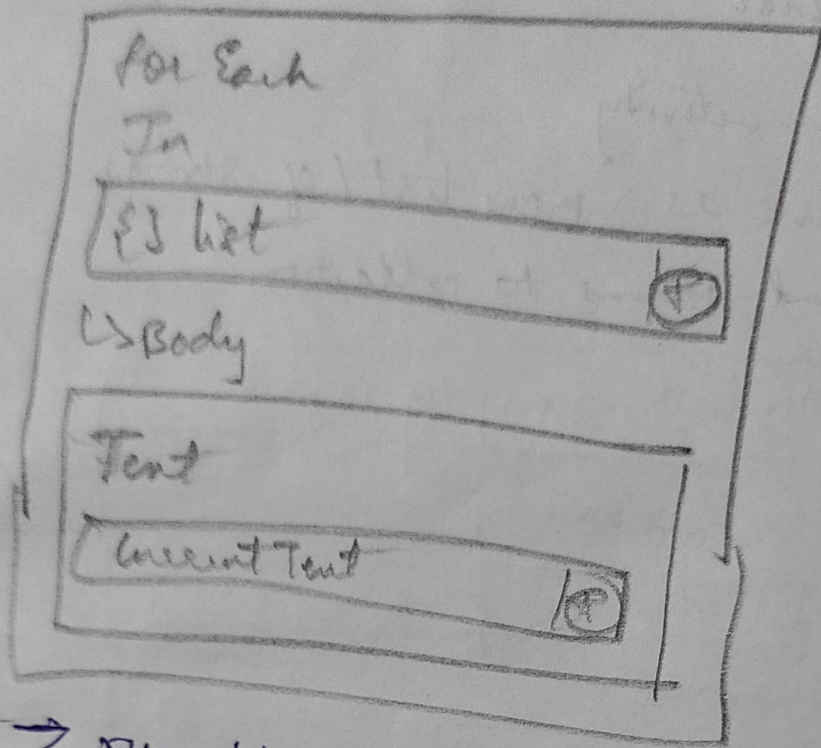
- Debug it → run fire

O/p : i/p
        n/p
        npa
        n/p

[TRUE]

Experiment-7

Date: 1|3|25

7. Perform automation for customizing selectors.

AIM

To perform automation for customizing selectors, you can follow these steps using UiExplorer and editing selectors in the properties of activities.

PROCEDURE

1. **Identify the Target Element**

- Open UiPath Studio and create a new project or open an existing one.
- Identify the UI element you want to interact with in your automation process.

2. **Use UiExplorer to Customize Selector**

- Click on the "Indicate element inside browser" or "Indicate element inside window" option in the UiPath Studio toolbar.
- Hover your mouse over the target element, and UiExplorer will highlight it.
- Click on the target element to inspect its properties in UiExplorer.

3. **Customize Selector Properties**

- In UiExplorer, review the properties of the selected element.
- Customize the selector properties as needed to ensure a robust and reliable selection.

- Adjust the selector to include unique attributes that identify the element, avoiding properties that may change dynamically.
- Use wildcards (*) where necessary to handle dynamic attribute values.
- Remove unnecessary or volatile properties from the selector to improve stability.

4. **Validate Selector**

- After customizing the selector, click on the "Validate" button in UiExplorer to ensure that the selector correctly identifies the target element.
- Verify that the selector is unique and does not match unintended elements on the screen.

5. **Update Selector in Activity Properties**

- Open the activity (e.g., Click, Type Into) that interacts with the target element.
- In the properties pane of the activity, locate the "Selector" field.
- Paste the customized selector into the "Selector" field to replace the default selector.

6. **Test the Automation**

- Test the automation process to verify that the customized selector correctly identifies and interacts with the target element. Run the workflow and observe the behavior of the activities interacting with the element

SOURCE CODE

Selectors:
In UiPath, selectors are xml strings that define the unique properties of UI elements, like id, name, title, class, etc. The selector identifies an element the screen for automation purposes.

Dynamic Selectors:
These selectors can be modified programmatically at run time to adapt to changing UI structures.

Steps to Automate Selector Customization in UiPath.

1) Basic Setup for UiPath Automation:
* Open UiPath Studio
* Create a new process
* Use the Selector Editor in activities like click, type into ....

2) Modify Selectors Dynamically:

UiPath provided a way to manipulate selectors dynamically during execution using variables or expressions. Here's an example for how you can customize 2 selector dynamically.

Ex: Automating Dynamic Selector Customization:
Step1: Capture the selector
    2: Make the Selector Dynamic
    3: Use Variables for Dynamic Selectors
    4: En.use Dynamic Selectors in workflow.
    5: Handle Dynamic Elements in multiple places.
    6: Using find Element with Dynamic Selector

Tips for Automatising Selectors Dynamically in UiPath
1) Use Wildcards.
2) Anchor Base
3) Selector Editor

4) Debugging
5) Attributes

-) By leveraging dynamic selectors in UiPath, you can automate applications where UI elements might change over time, allowing your automation to remain flexible and robust.