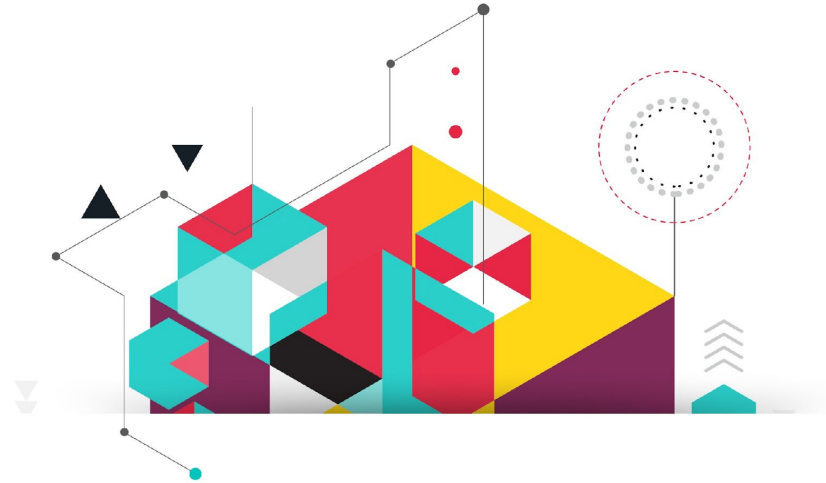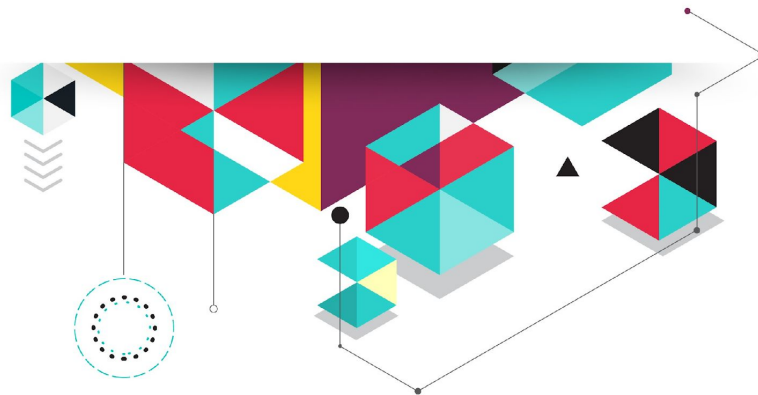# LESSON 12 – DEBUGGING AND EXCEPTION HANDLING - RECAP

Overview

In this module we dived into 2 of the most important aspects of software development in general and the tools UiPath Studio offers in these regards. You were introduced to the **Debug tab** and the **Try Catch Activity**, along with some other useful techniques.

Takeaways

You can start debugging by pressing the **Start Debug button** from the **Execute tab**.

When debugging, you can notice 3 things that happen:

-There is a yellow highlight for the action that is currently executed, and, if suitable, an optional red highlight for the element that is affected by this action
-The **Locals panel**, where you can check the current value of all the variables
-A detailed log of all the actions that took place in the workflow

If you need to slow things down, use the **Slow Step button**, or you can completely pause the execution by using the **Toggle Breakpoint button** and continue the execution step by step by using the **Step Over button**.

When waiting for an application to load, an activity's default timeout value is 30 seconds, but you can also use activities like **Element Exists**, **Find Element** or **Wait Element Vanish** and their image counterparts.

Takeaways

- **Element Exists** doesn't affect your workflow, it just returns a boolean value, whereas the other 2 activities will stop execution until an element is found or it disappeared.

- The **Try Catch** activity should contain the actions that might throw an error inside the **Try** block, the actions to take when an error occurs inside the **Catch** block and, optionally, the actions to always be performed after the other 2 blocks inside the **Finally** block.

- You can have multiple **Catches** for different types of exceptions.

- Even though you caught an exception, you sometimes might want to make sure the workflow actually stops, in that case you can use the **Rethrow** activity.

- You can separate individual components of your automation into different workflow files and then call them using **Invoke Workflow**.

Best practices

- It's very important to use relevant names for actions and flowcharts, and it pays off on the long term.

- A good strategy if you want to avoid the problems generated by windows that might be on top of the one you want to use is to keep away from the default input method.

- When you are having trouble with the selectors use the **Indicate On Screen** and **Attach to live element** options to "refresh" it.

  .

Useful links

[Enabling Tracing](#)
[How to use Try Catch Activity?](#)