From Robot Framework 2.9 to 6.0

- This slide set explains biggest enhancements and other changes between Robot Framework 2.9 and 6.0
- The biggest backwards incompatible changes are related to FOR loops and variable item access
- The provided `rf32to40.py` script can update loop and variable syntax automatically

General upgrading strategy

1. Make sure all tests succeed with the current Robot Framework version
2. Upgrade to the next feature release (`x.y` level)
3. Use the last bug fix release (`x.y.z` level) to get fixes to possible regressions (2.9.2, 3.0.4, 3.1.2, 3.2.2, 4.0.3, 4.1.3, 5.0.1)
4. Run tests again
5. Use `--dryrun` first if execution takes lot of time
6. Fix possible broken tests and resolve deprecation warnings
7. Go to 1

Python version support

- Python 3 is supported since RF 3.0
- Python 2 support was removed in RF 5.0
- Jython and IronPython support was removed in RF 5.0
- Robot Framework 6.0 supports Python 3.6 and newer
- Robot Framework 7.0 will require Python 3.7 or newer

FOR loop changes

```
*** Test Cases ***
Old FOR loop
    :FOR    ${animal}    IN    cat    dog    cow
    \    Log    ${animal}

New FOR loop
    FOR    ${animal}    IN    cat    dog    cow
        Log    ${animal}
    END
```

- New syntax added in RF 3.1
- Old syntax deprecated in RF 3.2 and removed in RF 4.0
- Provided `rf32to40.py` script can update loop syntax

IF/ELSE

```
*** Test Cases ***
IF only
    IF      ${condition}
        Keyword
    END

IF/ELSE IF/ELSE
    IF      ${var} > 0
        Positive Keyword
    ELSE IF    ${var} < 0
        Negative Keyword
    ELSE
        Zero Keyword
    END
```

Added in RF 4.0

Nested control structures

```
*** Test Cases ***
Example
    FOR    ${row}    IN    @{rows}
        FOR    ${cell}    IN    @{row}
            IF    "${cell}" != "IGNORE"
                Process Cell    ${cell}
            END
        END
    END
```

Added in RF 4.0

TRY/EXCEPT

```
*** Test Cases ***
TRY/EXCEPT
    TRY
        Some Keyword
    EXCEPT    Error message
        Error Handler Keyword
    END

TRY/FINALLY
    TRY
        Use Connection
    FINALLY
        Close Connection
    END
```

Added in RF 5.0

WHILE

```
*** Test Cases ***
Loop while condition is true
    WHILE    ${x} > 0
        Log    ${x}
        ${x} =    Evaluate    ${x} - 1
    END

Loop until BREAK
    WHILE    True
        Log    ${x}
        ${x} =    Evaluate    ${x} - 1
        IF    ${x} == 0    BREAK    # Inline IF
    END
```

Added in RF 5.0

Other control structures

- Inline `IF` in RF 5.0
- `RETURN` for returning from user keywords in RF 5.0
- `BREAK` and `CONTINUE` for loop control in RF 5.0
- Loop control keywords `Exit For Loop` and `Continue For Loop` must be used directly inside loops starting from RF 5.0
- Loop control keywords are considered deprecated since RF 5.0

Changes to variable items access

- List and dictionary variable items can be accessed using generic `${var}[item]` syntax instead of `@{list}[0]` and `&{dict}[key]` starting from RF 3.1
- Nested access like `${var}[x][y]` is supported since RF 3.1 as well
- Accessing list and dictionary items like `@{list}[0]` and `&{dict}[key]` was deprecated in RF 3.2
- `@{var}[0]` and `&{var}[key]` mean accessing the specified item and *unpacking* it as list or dictionary since RF 4.0
- Provided `rf32to40.py` script can update item access syntax

Status changes

- `SKIP` status was added in RF 4.0
- Criticality concept was removed in RF 4.0
- New `--skiponfailure` works similarly as old `--noncritical`
- `--critical` and `--noncritical` were deprecated in RF 4.0 and removed in RF 5.0
- `NOT RUN` status is used with un-executed keywords and control structures starting from RF 4.0

`pybot` replaced with `robot` and `python -m robot`

- `robot` and `python -m robot` were added in RF 3.0
- `pybot`, `jybot` and `ipybot` were removed in RF 3.1
- `robot.bat` was changed to `robot.exe` on Windows in RF 3.1

Data format changes

- Support for HTML and TSV formats was removed in RF 3.2
- Only `*.robot` files are parsed by default starting from RF 3.2
- `--extension` option can be used if other files need to be parsed
- New `*.resource` extension for resource files was added in RF 3.1

Changes to section and setting names

- Aliases like `*** Metadata ***`, `Document` and `Pre Condition` were removed in RF 3.1
- Names are space sensitive (e.g. `*** TestCases ***` or `Set Up` won't work) starting from RF 3.2
- Singular headers like `*** Test Case ***` are considered deprecated starting from RF 6.0
- `Test Tags` and `Keyword Tags` were added in RF 6.0
- `Force Tags` and `Default Tags` are considered deprecated since RF 6.0

Evaluating expressions

- Modules are imported automatically in expressions like `platform.system() == 'Linux'` with `Evaluate`, `Run Keyword If`, `IF/ELSE`, etc. starting from RF 3.2
- Inline Python evaluation (e.g. `${{[1, 2, 3, 4]}}` and `${{datetime.date(2022, 11, 22)}}` was added in RF 3.2

Listener changes

- Old listeners API v1 was removed in RF 3.0
- Powerful listener API v3 was added in RF 3.0
- Listener API v3 can add and remove tests since RF 3.2
- Small changes to information passed to listener API v2 in different releases

Libdoc changes

- New and enhanced HTML styles were added in RF 4.0
- Support for JSON spec files was added in RF 4.0
- Some changes to XML spec files in different releases

Automatic argument conversion

```
def keyword(count: int, case_insensitive: bool = True):
    ...
```

```
*** Test Cases ***
Example
    Keyword    42
    Keyword    -1    case_insensitive=False
```

- Added in RF 3.1 and enhanced in subsequent releases
- Support for custom argument converters added in RF 5.0

Other library API enhancements

- `@library` and `@not_keyword` decorators added in RF 3.2
- Named-only argument support added in RF 3.1
- Positional-only argument support (Python 3.8+ only) added in RF 4.0

Controlling continue-on-failure mode

- Controlled using tags on test and keyword level
- `robot:continue-on-failure` and `robot:recursive-continue-on-failure` added in RF 4.1
- `robot:stop-on-failure` and `robot:recursive-stop-on-failure` added in RF 6.0

Log file enhancements

- Relative order of messages and keywords preserved since RF 4.0
- Un-executed keywords (after failures, in un-executed `IF/ELSE` branches, etc.) shown in log using `NOT RUN` status (gray) since RF 4.0

output.xml changes

- New `SKIP` and `NOT RUN` statuses added
- Criticality information removed
- Container elements `<metadata>`, `<tags>`, `<arguments>` and `<assign>` around items have been removed
- `FOR` loops are represented as `<for>` elements instead of `<kw type='for'>`
- New elements for other control structures
- Boolean values in attributes changed from `yes/no` to `true/false`
- All these changes were done in RF 4.0

xUnit format changes

- `skip` attribute changed to `skipped` in RF 3.1
- Each suite gets its own `<testsuite>` element starting from RF 5.0
- `<testsuite>` elements have `timestamp` attribute since RF 5.0
- Suite documentation and metadata are added under each `<testsuite>` as properties since RF 5.0

New external RoboTidy

- https://robotidy.readthedocs.io
- Much more powerful and easier to customize than old built-in Tidy
- Can convert `Run Keyword If` to `IF/ELSE`, update deprecated settings, and so on
- Cannot handle old `FOR` loops
- Built-in Tidy deprecated in RF 4.1 and removed in RF 5.0

Other changes

- Support for process automation (RPA) in RF 3.1
- New parser in RF 3.2
- Performance enhancements to remote API in RF 4.0
- Localization in RF 6.0
- Better conflict resolution with keywords accepting embedded arguments in RF 6.0
- Enhancements to keyword namespaces (e.g. `robot:private`) in RF 6.0