# DocField

A DocField defines a property (or a field) of a DocType. For instance, a DocType named "ToDo" might have fields such as description, status, and priority. These fields eventually become columns in the corresponding database table, like "tabToDo". Frappe comes with more than 30 different fieldtypes out-of-the-box.

**Key Insights**:

1. **Field Properties**: Each DocField has various properties, including:
   - **Label**: The value displayed to the user.
   - **Fieldname**: The property name referenced in the code and also the column name in the database.
   - **Fieldtype**: Determines how the value is stored and displayed.
   - **Options**: For select fields, a list of available choices.
   - **Default**: The default value for the field.
2. **Conditional Display**: The "depends_on" property determines if a field will be displayed based on certain conditions. For example, a field might only be shown if the status of a document is "Closed".
3. **Mandatory & Read-Only Conditions**: Two properties, "mandatory_depends_on" and "read_only_depends_on", have been introduced to make a field mandatory or read-only based on specific conditions.
4. **Variety of Fieldtypes**: Frappe offers over 30 different fieldtypes to cater to diverse use-cases. Each fieldtype serves a specific purpose, from simple text inputs to more complex selections.

Example:

```
{
    "label": "Priority",
    "fieldname": "priority",
    "fieldtype": "Select",
    "options": [              // list of options for select
        "Low",
        "Medium",
        "High"
    ],
    "default": "Low"          // the default value to be set
},
{
    "label": "Completed By",
    "fieldname": "completed_by",
    "fieldtype": "Link",
    "options": "User",
    "depends_on": "eval: doc.status == 'Closed'", // the condition on which this field's displ
},
{
    "collapsible": 1,
    "collapsible_depends_on": "eval:doc.status!='Closed'", // determines if a Section Break fi
```