**ERPNext DocType Guide**

**1. Role of DocTypes in ERPNext**

DocTypes define the structure of data in ERPNext. They act as database tables and manage records, fields, validations, and business logic. Every module in ERPNext is built around DocTypes, making them the foundation of the system.

**2. Key Features of a DocType & Data Management**

* **Field Definitions**: Defines the type of data stored (e.g., Text, Number, Link).
* **Permissions**: Controls access based on roles.
* **Triggers & Events**: Allows automation via scripts.
* **Customizations**: Supports adding custom fields and scripts without modifying core files.
* **Relations**: Links data between different DocTypes.

DocTypes manage data by storing records in the database, validating inputs, and enforcing relationships between fields.

**3. Structure of Student DocType & Purpose of Each Field**

A **Student** DocType might include fields such as:

* **Student Name (Data)** – Stores the full name.
* **Roll Number (Int)** – Unique identifier for students.
* **Date of Birth (Date)** – Captures birthdate.
* **Class (Link to Class DocType)** – Associates the student with a class.
* **Marks (Table, Child DocType: Student Marks)** – Stores subject-wise marks.

**4. Difference Between Client Script and Server Script**

* **Client Script**: Runs in the browser, used for UI interactions, field validations, and dynamic form behaviors.
* **Server Script**: Runs on the backend, used for complex computations, database interactions, and workflow automation.

**5. Client Script for Marks Validation**

Here’s an example script to ensure marks entered are between 0 and 100:

frappe.ui.form.on('Student', {

validate: function(frm) {

frm.doc.marks.forEach(row => {

if (row.marks < 0 || row.marks > 100) {

frappe.msgprint(\_\_('Marks should be between 0 and 100'));

frappe.validated = false;

}

});

}

});

**6. Server Script for Percentage Calculation**

This script calculates the percentage based on total marks:

frappe.whitelist()

def calculate\_percentage(doc, method):

total\_marks = sum(row.marks for row in doc.marks)

doc.percentage = (total\_marks / (len(doc.marks) \* 100)) \* 100

This script should be triggered on the validate event of the Student DocType.

**7. Triggering a DocType Event in ERPNext**

ERPNext supports **DocType Events** such as before\_insert, validate, and on\_submit. Example of triggering an event:

frappe.ui.form.on('Student', {

validate: function(frm) {

frappe.msgprint(\_\_('Validating student data'));

}

});

**8. Purpose of Print Formats in ERPNext**

Print Formats allow customized print layouts for documents. They help in generating invoices, reports, and certificates using HTML/Jinja templating.

**9. Field Types in a DocType & Explanation of Five**

ERPNext supports multiple field types, such as:

* **Data**: Stores single-line text.
* **Int**: Stores integer values.
* **Link**: Connects to another DocType.
* **Table**: Holds child records (e.g., marks in a student’s report).
* **Select**: Dropdown with predefined options.

**10. Displaying Linked Address Field in Print Format**

You can fetch address\_line1 from the linked DocType in a Print Format using Jinja:

{{ frappe.get\_value("Address", doc.address, "address\_line1") }}

This fetches address\_line1 from the referenced Address DocType.