



TYPO3 Fundamentals

Core Concepts

Page Tree:	The hierarchical structure that organizes website content. Each page in the tree represents a page on the website.
Content Elements (CEs):	Individual pieces of content (text, images, etc.) that are added to pages. Also known as 'Elements'.
TypoScript:	The configuration language used to control how content is rendered. Allows for dynamic generation of HTML, CSS, and JavaScript.
Fluid:	The templating engine used in TYPO3. It separates the presentation layer from the logic.
Extensions:	Reusable components that extend TYPO3's functionality. Can be installed and configured to add features like blogs, forms, or e-commerce capabilities.
Backend Layouts:	Define the structure of the content area in the TYPO3 backend, dictating where content elements can be placed.

Key Directories

<code>/typo3conf/</code>	Contains configuration files, including <code>LocalConfiguration.php</code> and <code>AdditionalConfiguration.php</code> .
<code>/typo3temp/</code>	Temporary files and cache. Clear this directory to resolve many common issues.
<code>/fileadmin/</code>	Default location for uploaded files (images, documents, etc.). Can be configured.
<code>/typo3/</code>	
<code>/typo3/extensionmanager/</code>	Location for extensions installed via composer.
<code>/typo3/public/</code>	Default web root. Where <code>index.php</code> and other public files are located. (May vary depending on server setup)

Backend Navigation

Page Module:	Used to manage the page tree and content elements on a specific page.
List Module:	Provides a detailed view of all records in a table, allowing for filtering and editing.
Filelist Module:	Manages files and folders in the file system.
Template Module:	Edit TypoScript templates.
Extension Manager:	Install, configure, and manage extensions.
Admin Tools:	Various admin tools like clearing cache, system information, install tool etc.

TypoScript Essentials

Basic Syntax

<code>page = PAGE</code>	Assigns a new PAGE object. <code>PAGE</code> is a TypoScript object type.
<code>page.10 = TEXT</code>	Assigns a TEXT object to the property <code>10</code> of the <code>page</code> object.
<code>page.10.value = Hello World</code>	Sets the <code>value</code> property of the TEXT object to 'Hello World'.
<code><INCLUDE_TYPOSCRIPT: source="FILE:fileadmin/template/setup.typoscript"></code>	Includes TypoScript from an external file.
<code>[globalVar = TSFE:id = 1]</code>	Conditional statement: Checks if the current page ID is 1.
<code>lib.myMenu = HMENU</code>	Creates a menu object called 'myMenu'.

Common Objects

PAGE:	Defines the overall structure of the HTML page.
TEXT:	Outputs plain text.
IMAGE:	Renders an image. Can be configured to resize, crop, etc.
CONTENT:	Renders content elements from the database.
HMENU:	Generates a navigation menu.
USER_INT:	Allows calling a user-defined function within TypoScript (cached). <code>USER</code> is similar but uncached.

Example Configuration

<pre>page = PAGE page.10 = TEXT page.10.value = Hello World page.10.wrap = <h1> </h1> page.includeCSS { main = fileadmin/css/style.css }</pre>	<p>This TypoScript configuration creates a basic HTML page with a heading 'Hello World' and includes a CSS file.</p>
---	--

Fluid Templating

Basic Syntax

<code><f:if condition="{variable}">...</f:if></code>
<code><f:for each="{items}" as="item">...</f:for></code>
<code><f:link.page pageUid="123">Link to Page</f:link.page></code>
<code><f:translate key="LLL:EXT:my_extension/Resources/Private/Language/local ></code>
<code><f:format.date format="%d.%m.%Y">{date}</f:format.date></code>
<code>{variable -> f:format.html() }</code>

ViewHelpers

f:if: Conditional rendering of a conditional statement.
f:for: Looping through data. Renders
f:link.page: Creating internal links.
f:link.external: Creating external links. only if the condition
f:translate: Localizing text. if true.
f.format.*: Various formatting ViewHelpers (date, number, html, etc.). Looping construct. Iterates over an array of

Example Template

```
<f:layout name="Default" />

<f:section name="Main">
  <h1>{page.title}</h1>
  <f:for each="{contentElements}"
  as="contentElement">
    <f:cObject
  typoscriptObjectPath="lib.contentElement"
  data="{contentElement}" />
  </f:for>
</f:section>
```

This example demonstrates a basic Fluid template with a layout, a section for the main content, and a loop that renders content elements using `f:cObject`.

Formats a variable to HTML, escaping potentially dangerous characters.

Common Tasks

Clearing Cache

<p>TYPO3 caches content and configuration for performance. Clearing the cache is often necessary after making changes.</p>
<p>Backend: Use the 'clear cache' menu in the top bar of the backend.</p>
<p>Install Tool: Navigate to the Install Tool and use the 'Clear all caches' functionality.</p>
<p>TypoScript: <code>config.disableAll = 1</code> (for development, disables caching)</p>
<p>Command-line: Use the <code>typo3 cache:flush</code> command.</p>

Working with Content Elements

Adding a CE	In the Page module, click the '+' icon in the content area to add a new content element.
Editing a CE	Click on a content element in the Page module to edit its properties.
Moving a CE	Drag and drop content elements within the content area to change their order. Alternatively use the up/down arrows.
Deleting a CE	Click the delete icon (trash can) on a content element to remove it.
Hiding/Disabling a CE	Use the enable/disable icon (eye icon) to hide a CE.

Extension Management

<p>Installing an extension: Use the Extension Manager module to search for and install extensions from the TYPO3 Extension Repository (TER) or via composer.</p>
<p>Configuring an extension: Many extensions have configuration options that can be set in the Extension Manager or via TypoScript.</p>
<p>Updating an extension: The Extension Manager can be used to update extensions to the latest versions.</p>
<p>Uninstalling an extension: Use the Extension Manager to uninstall extensions that are no longer needed.</p>