what is template files ?

Template files are responsible for the HTML markup of every page generated by Drupal. Any file ending with the .html.twig extension is a template file. These files are composed of standard HTML markup as well as tokens used by the Twig template engine to represent dynamic content that will be substituted into the HTML markup when the template is used.

What is node ?

All content on a Drupal website is stored and treated as "nodes". A node is any piece of individual content, such as a page, poll, article, forum topic, or a blog entry. Comments are not stored as nodes but are always connected to one. Treating all content as nodes allows the flexibility to create new types of content. It also allows you to painlessly apply new features or changes to all content of one type.

**Behind the scenes, the Node module manages these nodes. This module lets you:**

1. List, sort through, and manage all the content on your site.
2. Set defaults for how all posts are displayed.
3. List and configure the "content types" for your site, and create new ones.

The Node module manages the creation, editing, deletion, settings, and display of the main site content. Content items managed by the Node module are typically displayed as pages on your site, and include a title, some meta-data (author, creation time, content type, etc.), and optional fields containing text or other data.

What is page?

**Pages** module allows you to create dynamic **pages** (nodes) from static text files. It's a simple module, and useful when working with installation profiles. You can get an archive of the **pages** module here

What is node.tpl.php?

The node.tpl.php file returns the HTML for a node.

**$title :** the (sanitized) title of the node.

**$content :** An array of node items. Use render($content) to print them all, or print a subset such as render($content['field\_example']). Use hide($content['field\_example']) to temporarily suppress the printing of a given element.

**$user\_picture** The node author's picture from user-picture.tpl.php.

**$date**

Formatted creation date. Preprocess functions can reformat it by calling format\_date() with the desired parameters on the $created variable.

**$name**

Themed username of node author output from theme\_username().

**$node\_url**

Direct url of the current node.

**$display\_submitted**

Whether submission information should be displayed.

**$submitted**

Submission information created from $name and $date during template\_preprocess\_node().

**$classes**

String of classes that can be used to style contextually through CSS. It can be manipulated through the variable $classes\_array from preprocess functions. The default values can be one or more of the following:

* **node:** The current template type, i.e., "theming hook".
* **node-[type]:** The current node type. For example, if the node is a "Blog entry" it would result in "node-blog". Note that the machine name will often be in a short form of the human readable label.
* **node-teaser:** Nodes in teaser form.
* **node-preview:** Nodes in preview mode.
* **view-mode-[mode]:** The view mode, e.g. 'full', 'teaser'…

The following are controlled through the node publishing options.

* **node-promoted:** Nodes promoted to the front page.
* **node-sticky:** Nodes ordered above other non-sticky nodes in teaser listings.
* **node-unpublished:** Unpublished nodes visible only to administrators. The following applies only to viewers who are registered users:
* **node-by-viewer:** Node is authored by the user currently viewing the page.

**$title\_prefix (array)**

An array containing additional output populated by modules, intended to be displayed in front of the main title tag that appears in the template.

**$title\_suffix (array)**

An array containing additional output populated by modules, intended to be displayed after the main title tag that appears in the template.

### **Other variables:**

**$node**

Full node object. Contains data that may not be safe.

**$type**

Node type, i.e. story, page, blog, etc.

**$comment\_count**

Number of comments attached to the node.

**$uid**

User ID of the node author.

**$created**

Time the node was published formatted in Unix timestamp.

**$pubdate**

Formatted date and time for when the node was published wrapped in a HTML5 time element.

**$classes\_array**

Array of html class attribute values. It is flattened into a string within the variable $classes.

**$zebra**

Outputs either "even" or "odd". Useful for zebra striping in teaser listings.

**$id**

Position of the node. Increments each time it's output.

**$view\_mode**

View mode, e.g. 'full', 'teaser'...

**$teaser**

Flag for the teaser state (shortcut for $view\_mode == 'teaser').

**$page**

Flag for the full page state.

**$promote**

Flag for front page promotion state.

**$sticky**

Flags for sticky post setting.

**$status**

Flag for published status.

**$comment**

State of comment settings for the node.

**$readmore**

Flags true if the teaser content of the node cannot hold the main body content. Currently broken; see <http://drupal.org/node/823380>

**$is\_front**

Flags true when presented in the front page.

**$logged\_in**

Flags true when the current user is a logged-in member.

**$is\_admin**

Flags true when the current user is an administrator.

Field variables: for each field instance attached to the node a corresponding variable is defined, e.g. $node->body becomes $body. When needing to access a field's raw values, developers/themers are strongly encouraged to use these variables. Otherwise they will have to explicitly specify the desired field language, e.g. $node->body['en'], thus overriding any language negotiation rule that was previously applied

What is page.tpl.php?