

A red square logo with a white border. The word "Twig" is written in white serif font in the center, and "Prattya Datta" is written in white serif font below it.

**Twig**

**Prattya Datta**

# Tasks

- Read project description
- Git Repo
- Create files necessary
- Download composer
- Install Twig
- Create 4 HTML blocks
- Create macros for the form
- Use include
- Check if News exist
- Documentation



## Twig is a modern template engine for PHP

- **Fast:** Twig *compiles* templates down to plain optimized PHP code. The overhead compared to regular PHP code was reduced to the very minimum.
- **Secure:** Twig has a *sandbox* mode to evaluate untrusted template code. This allows Twig to be used as a template language for applications where users may modify the template design.
- **Flexible:** Twig is powered by a flexible *lexer* and *parser*. This allows the developer to define its own custom tags and filters, and create its own DSL.

In essence Twig is template engine designed to distribute responsibility from PhP to render the pages. It is also useful for optimizing pages with different variables being rendered at different points. A template engine or template processor is a library designed to combine templates with a data model to produce documents. Template engines are often used to generate large amounts of emails, in source code preprocessing or producing dynamic HTML pages.

We create a template engine, where we define static parts and dynamic parts. The dynamic parts are later replaced with data. The rendering function later combines the templates with data.

Twig: Knowledge learnt

# Blocks

Blocks are used for inheritance and act as placeholders and replacements at the same time. <sup>1</sup>

```
1 <title>{% block title %}{% endblock %}</title>
2
3 <h1>{{ block('title') }}</h1>
4
5 {% block body %}{% endblock %}
```

The `block` function can also be used to display one block from another template:

```
1 {{ block("title", "common_blocks.twig") }}
```

Use the `defined` test to check if a block exists in the context of the current template:

```
1 {% if block("footer") is defined %}
2     ...
3 {% endif %}
4
5 {% if block("footer", "common_blocks.twig") is defined %}
6     ...
7 {% endif %}
```

# Filter

The filter filter filters elements of a sequence or a mapping using an arrow function.

```
1  {% set sizes = [34, 36, 38, 40, 42] %}  
2  
3  {{ sizes|filter(v => v > 38)|join(', ') }}  
4  {# output 40, 42 #}
```

Combined with the `for` tag, it allows to filter the items to iterate over:

```
1  {% for v in sizes|filter(v => v > 38) -%}  
2      {{ v }}  
3  {% endfor %}  
4  {# output 40 42 #}
```

It also works with mappings:

```
1  {% set sizes = {  
2      xs: 34,  
3      s: 36,  
4      m: 38,  
5      l: 40,  
6      xl: 42,  
7  } %}  
8  
9  {% for k, v in sizes|filter(v => v > 38) -%}  
10     {{ k }} = {{ v }}  
11 {% endfor %}  
12 {# output l = 40 xl = 42 #}
```

# include

The include function returns the rendered content of a template:

```
1 {{ include('template.html') }}
2 {{ include(some_var) }}
```

Included templates have access to the variables of the active context.

If you are using the filesystem loader, the templates are looked for in the paths defined by

The context is passed by default to the template but you can also pass additional variables:

```
1 {# template.html will have access to the variables from the current context
2 {{ include('template.html', {foo: 'bar'}) }}
```

You can disable access to the context by setting `with_context` to `false`:

```
1 {# only the foo variable will be accessible #}
2 {{ include('template.html', {foo: 'bar'}, with_context = false) }}
```

```
1 {# no variables will be accessible #}
2 {{ include('template.html', with_context = false) }}
```

And if the expression evaluates to a `\Twig\Template` OR a `\Twig\TemplateWrapper` instance, Tw

```
1 // {{ include(template) }}
2
3 $template = $twig->load('some_template.twig');
4
5 $twig->display('template.twig', ['template' => $template]);
```

# For loop

Loop over each item in a sequence. For example, to display a list of users provided in a variable called users:

```
<h1>Members</h1>
<ul>
  {% for user in users %}
    <li>{{ user.username|e }}</li>
  {% endfor %}
</ul>
```