

Working with Twig and HTML

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.

Twig syntax

There are two forms of twig syntax you will use in your html

1. `{% %}` -- This says to something, ex(Loop, If, or set Var), and will not output anything on the page. It's like a function
2. `{{ }}` -- This allows you to add dynamic content to your page.

Example (in the list tab)

```
This is a twig variable that we are setting equal to our custom collection  
{% set recipes = su.collection("recipes").meal('Breakfast').sort('name') %}
```

```
<div class="item-container">
```

Twig for loop. This loops through a list of items and lays out each item how we want it to.

```
{% for recipes in recipes.find() %}  
  <div class="wrapper-tumb">
```

`{{ su.page.urlForCollectionItem(recipes) }}` --- This allows us to generate a URL for our **details page** and then link to it.

```
  <a style="text-decoration: none; color:black;" href='{{ su.page.urlForCollectionItem(recipes) }}'>  
    
```

`{{ recipes.backgroundColor }}` --- We can also use color or even a short answer field to do inline styling

```
    <h3 class='recipe-title' style='color:white; background:{{ recipes.backgroundColor }}; margin:0;'><strong>{{  
recipes.recipeName }}</strong></h3>  
    <p class="recipe-description">{{ recipes.description }}</p>  
    <div class='cat'>  
      <p style="font-weight:bold;">Category: {{ recipes.categories }}</p>  
    </div>  
  </a>  
  
  </div>  
{% endfor %}  
</div>
```

Everything the **<style>** tag is for styling your content on the page. You will want to use id and class names for styling to make sure it doesn't effect the rest of your site.

Note: Why do we do inline style and a style tag? We could put our CSS in the themes > custom css section if we wanted but this keeps everything in one location and easier to track.

```
<style>
  .wrapper-tumb{
    margin: 5px;
    background: white;
    max-width:32%;
    border-radius: 10px;
  }
  .wrapper-tumb:hover {
    transform:scale(1.05);
    transition:1.5s;
  }

  .wrapper-tumb img{
    width:100%;
    height:250px;
    border-radius:10px 10px 0 0;
    object-fit:cover;
  }

  .item-container {
    max-width:100%;
    display:flex;
    flex-flow: row wrap;
    justify-content: center;
  }
  .recipe-title{
    text-align:center;
    margin:0 !important;
    padding:5px;
  }
}
```

```
.cat {
    width:100%;
    height:20px;
    padding:0 10px;
}
.cat p {
    float:left;
}
.recipe-description {
    padding: 3px 10px;
}
</style>
```

Details Page

The details page allows for us to display content for a single item when it has been clicked on. In order to get the details page to work you need to following two pieces of code.

List Tab

```
<a href='{{ su.page.urlForCollectionItem(recipes) }}'>
... Content here
</a>
```

Details Tab

```
{% set recipes = su.page.collectionItem %}
```

Example:

Styles

```
<style>
  #detail-container {
    width:100%;
  }
  .detail-title-content {
    width:50%;
    position:relative;
    height:30vh;
  }
  #detail-title {
    font-weight:bold;
    background:{{ recipes.backgroundColor }};
    color:white;
    text-align:center;
    padding:1%;
    margin:0 0 10px 0;
```

```
}  
.detail-image {  
    width:50%;  
}  
#detail-type {  
    position:absolute;  
    bottom:0;  
    left:0;  
}  
#detail-description {  
    padding:0 1%;  
}  
#detail-meal, #detail-cat {  
    font-weight:bold;  
    width:50%;  
}  
#detail-meal {  
    float:right;  
}  
#detail-cat {  
    float:left;  
}
```

```
</style>
```

HTML

```

{% set recipes = su.page.collectionItem %}

<div class="detail-container row">
  <div class="detail-title-content col-md-6 col-sm-12 col-xs-12 col-lg-6">
    <h1 id="detail-title">{{ recipes.recipeName }}</h1>
    <p id="detail-description">{{ recipes.description }}</p>
    <div id="detail-type">
      <p class='col-md-6 col-sm-12' id="detail-cat">Categories :{{ recipes.categories }}</p>
      <p class='col-md-6 col-sm-12' id="detail-meal">Meal: {{ recipes.meal }}</p>
    </div>
  </div>

  <div class="detail-image col-md-6 col-sm-12 col-xs-12 col-lg-6">
    
  </div>
  <hr style='clear:both;' />
  <div>
    <h3 style="font-weight:bold;">Ingredients</h3>
    <p>{{ recipes.ingredients }}</p>

  </div>
  <hr >
  <div>
    <h3 style="font-weight:bold;">Directions</h3>
    <p> {{ recipes.directions }}</p>

  </div>
</div>

```

