

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for contents of text-input area

Checker Input

Show ☒ source ☐ outline ☐ image report

Options...

Check by

text input

☐ css

```
{% extends "base.html" %}
{% block structure %}

    <!-- Static image for the background taken from
https://unsplash.com/photos/08b0YnH_r_E-->
    

    <h3 class="center-header">Add Meal Type</h3>

    {% block content %}
    <div class="row add-recipe-card card-panel">
        <!-- using the POST Method and calling on the functin 'add_meal'-->
        <form class="col s12" method="POST" action="{{ url_for('add_meal') }}">
            <!--Placeholder recipe name (mandatory field)-->
            <div class="row">
```

Check

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Message Filtering

1.

Warning

Consider adding a `lang` attribute to the `html` start tag to declare the language of this document.

[From line 1, column 1; to line 1, column 25](#)

```
{% extends "base.html" %}↵{% b1
```

For further guidance, consult [Declaring the overall language of a page](#) and [Choosing language tags](#).

If the HTML checker has misidentified the language of this document, please [file an issue report](#) or [send e-mail to report the problem](#).
2.

Error

Non-space characters found without seeing a doctype first. Expected `<!DOCTYPE html>`.

[From line 1, column 1; to line 1, column 25](#)

```
{% extends "base.html" %}↵{% b1
```
3.

Error

Element `head` is missing a required instance of child element `title`.

[From line 1, column 1; to line 1, column 25](#)

```
{% extends "base.html" %}↵{% b1
```

Content model for element [head](#):

If the document is [an iframe srcdoc document](#) or if title information is available from a higher-level protocol: Zero or more elements of [metadata content](#), of which no more than one is a [title](#) element and no more than one is a [base](#) element.

Otherwise: One or more elements of [metadata content](#), of which exactly one is a [title](#) element and no more than one is a [base](#) element.

4. **Error** Bad value `{{ url_for('static', filename='img/bg.jpg') }}` for attribute `src` on element `img`: Illegal character in path segment: `{` is not allowed.

[From line 5, column 5; to line 5, column 79](#)

```
--> 
```

5. **Error** An `img` element must have an `alt` attribute, except under certain conditions. For details, consult [guidance on providing text alternatives for images](#).

[From line 5, column 5; to line 5, column 79](#)

```
--> 
```

6. **Error** Bad value `{{ url_for('add_meal') }}` for attribute `action` on element `form`: Illegal character in path segment: `{` is not allowed.

[From line 12, column 9; to line 12, column 79](#)

```
> <form class="col s12" method="POST" action="{{ url_for('add_meal') }}">
```

7. **Error** Bad value `row flashes` for attribute `id` on element `div`: An ID must not contain whitespace.

[From line 29, column 29; to line 29, column 50](#)

```
<div id="row flashes">
```

8. **Error** Stray end tag `form`.

[From line 53, column 9; to line 53, column 15](#)

```
</form> <
```

Document checking completed.

Source

```
1. {% extends "base.html" %}
2. {% block structure %}
3.     <!-- Static image for the background taken from
4.     https://unsplash.com/photos/08b0YnH r E-->
5.     
6.
7.     <h3 class="center-header">Add Meal Type</h3>
8.
9.     {% block content %}
10.     <div class="row add-recipe-card card-panel">
11.         <!-- using the POST Method and calling on the function 'add_meal'-->
```

```

12. <form class="col s12" method="POST" action="{ { url_for('add_meal') } }">↵
13. <!--Placeholder recipe name (mandatory field)-->↵
14. <div class="row">↵
15. <div class="input-field col m10 offset-m1 s12">↵
16. <i class="fas fa-pencil-alt prefix"></i>↵
17. <!-- input requirement of min length of 3 characters and max
length of 25-->↵
18. <input id="meal_name" name="meal_name" minlength="3"
maxlength="25" ↵
19. type="text" class="validate" required>↵
20. <label for="meal_name">Meal Name*</label>↵
21. </div>↵
22. </div> ↵
23. ↵
24. <section>↵
25. <!-- Flash message to delivery the said message from the app.py
file -->↵
26. {% with messages = get_flashed_messages() %}↵
27. {% if messages %}↵
28. {% for message in messages %}↵
29. <div id="row flashes">↵
30. <h4 class="red lighten-4 center-align"
id="snackbar">{{ message }}</h4>↵
31. </div>↵
32. {% endfor %}↵
33. {% endif %}↵
34. {% endwith %}↵
35. </section>↵
36. ↵
37. <div class="row button-set"> ↵
38. <!-- To create the back button, i used the following website to
assist https://www.computerhope.com/issues/ch000317.htm -->↵
39. <button class="col 14 m4 s12 btn-large grey accent-1 black-text
text-shadow" ↵
40. value="Go back!" onclick="history.back()">↵
41. <i class="fas fa-arrow-circle-left"></i> Go Back↵
42. </button>↵
43. <!-- Clear the form button -->↵
44. <button type="reset" class="col 14 m4 s12 middle-button btn-
large red accent-1 black-text text-shadow">↵
45. Clear <i class="fas fa-times-circle"></i>↵
46. </button>↵
47. <!-- Submit the form button -->↵
48. <button type="submit" class="col 14 m4 s12 center-align btn-
large orange accent-1 black-text text-shadow">↵
49. Add Meal Type <i class="fas fa-plus-circle"></i>↵
50. </button>↵
51. </div>↵
52. </form> ↵
53. </form>↵
54. </div>↵
55. {% endblock %}↵
56. ↵
57. {% endblock %}

```

Used the HTML parser.

Total execution time 8 milliseconds.