



### Use of web forms when creating apps with Flask and Python

- Our example is how to calculate [Summation](#) or [Product of a sequence](#). This example gives results just in range from 1 to N, as an example:

$$\sum_{i=1}^N 2N/(1 + N) \text{ or } \prod_{i=1}^N 2N/(1 + N)$$

, where  $N$  is integer number and last element in this case is  $2N/(1 + N)$ .

So, if we are able to enter these two parameters in our app we will be able to calculate the result.

- Once you setup your virtual environment with Flash and Python, set up your folder structure inside your project and at the end create your files. (NOTE: If you want more details how to set up your working environment with Flask and Python feel free to ask from us at [support@centriumtech.com](mailto:support@centriumtech.com). If you learn with us we will provide you licenced Python IDE for free).

`index.html` file defines our web page that integrate `<style> ...something... </style>` HTML code part which contain designs shape of the web page (this also can be part of separate `style.css` file as well). In HTML code there is also `Form` part, where we use web form in which user enters data (in our case we enter *Nth element* as string and we enter number  $N$  as integer number. When we enter some data in the form we are redirected to page `BaseURL/send` (in our example that is `http://127.0.0.1:5000/send`). Using Jinja2 templating we return from our Python script named `example.py` (into the `index.html` file) value of the variable `result` with command: `{{ result }}`.

`index.html`

```
<!DOCTYPE html>
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>WebFFormCalculators</title>
    <link
      rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/skeleton/2.0.4/skeleton.min.css"
    />
    <style>
      .alert {
        background: #D8D8D8;
        padding: 1rem;
        border-radius: 5px;
        color: #000033;
        margin: 1rem;
      }
    </style>
  </head>
  <body>
```





## 4. Web Apps with Python and Flask - Web Forms

```
<div class="alert">
<div class="container">
<h5>Summation Calculator/Product of a sequence Calculator</h5>
</div>

<!-- Form -->
<div class="container">
<form action="/send" method="POST">
<label for="Number One">Enter your Nth element. As an example: 2*N+1:</label>
<input
  type="text"
  placeholder="Nth element"
  name="Nth_element"
/>

<label for="Number Two">Enter value of N:</label>
<input
  type="text"
  placeholder="Value of N"
  name="N"
/>

<label for="Operation">Operation</label>
<select class="u-full-width" name="operation">
<option value="sum">Summation</option>
<option value="product">Product of a sequence</option>
</select>

<input type="submit" value="Calculate" id="calc_btn" />
<br />

  {{ result }}

</div>
</form>
</div>
</body>
</html>
```

Our second file *example.py* is Python script. In this script: *product\_N(Nth\_element, N)* and *product\_N(Nth\_element, N)* are recursive functions that are used to do our calculations. We call these two functions into the *send()* function. Using *render\_template('index.html', result=result)* we are returning into *index.html* calculated value of the variable named *result*. *Nth\_element=request.form['Nth\_element']* takes entered value in the web form from our *index.html* HTML code (similarly also for *N = int(request.form['N'])* and *operation = request.form['operation']*).

example.py

```
from flask import Flask, render_template, request

app = Flask(__name__)

def sum_N(Nth_element, N):
    z=0
    if N > 0:
```





## 4. Web Apps with Python and Flask - Web Forms

```
z=eval(Nth_element)+sum_N(Nth_element, N-1)
return z

def product_N(Nth_element, N):
    z=1
    if N > 0:
        z=eval(Nth_element)*product_N(Nth_element, N-1)
    return z

@app.route('/')
def main():
    return render_template('index.html')

@app.route('/send', methods=['POST'])
def send():
    if request.method == 'POST':
        Nth_element= request.form['Nth_element']
        N = int(request.form['N'])
        operation = request.form['operation']

        if operation == 'sum':
            result = float(sum_N(Nth_element, N))
            return render_template('index.html', result=result)

        elif operation == 'product':
            result = float(product_N(Nth_element, N))
            return render_template('index.html', result=result)

        else:
            return render_template('index.html')

if __name__ == "__main__":
    app.run()
```





## 4. Web Apps with Python and Flask - Web Forms

### How our web app will look like?

- If we enter `http://127.0.0.1:5000/` resulting web app is:

Summation Calculator/Product of a sequence Calculator

Enter your Nth element. As an example:  $2 \cdot N + 1$ :

Enter value of N:

Operation

CALCULATE

Then we enter our data in the form (for example: last element is  $2N/(1+N)$  and value of  $N$  is 4), and choose combo option *Summation*:

Summation Calculator/Product of a sequence Calculator

Enter your Nth element. As an example:  $2 \cdot N + 1$ :

Enter value of N:

Operation

CALCULATE

After we press CALCULATE button we will got our result:

Summation Calculator/Product of a sequence Calculator

Enter your Nth element. As an example:  $2 \cdot N + 1$ :

Enter value of N:

Operation

CALCULATE

5.433333333333334





## 4. Web Apps with Python and Flask - Web Forms

- Similarly if we hit `http://127.0.0.1:5000/` we enter our data in the form (for example: last element is  $2N/(1+N)$  and value of  $N$  is 4), and choose combo option *Product of a sequence*:

Summation Calculator/Product of a sequence Calculator

Enter your Nth element. As an example:  $2 \cdot N + 1$ :

Enter value of N:

Operation

CALCULATE

After we press CALCULATE button we will got our result:

Summation Calculator/Product of a sequence Calculator

Enter your Nth element. As an example:  $2 \cdot N + 1$ :

Enter value of N:

Operation

CALCULATE

3.2

