

Laporan Teknologi Web

Fibonacci Sequence



By :

Putu Ananda Adi Savitri

2215101057

Fakultas Teknik dan Kejuruan

Jurusan Teknik Informatika

Prodi Ilmu Komputer

Universitas Pendidikan Ganesha

Singaraja

2023

CHAPTER I

INTRODUCTION

A. Basic Theory

In mathematics, the Fibonacci sequence is a sequence in which each number is the sum of the two preceding ones. Numbers that are part of the Fibonacci sequence are known as Fibonacci numbers, commonly denoted F_n . The sequence commonly starts from 0 and 1, although some authors start the sequence from 1 and 1 or sometimes (as did Fibonacci) from 1 and 2. Starting from 0 and 1, the first few values in the sequence are:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144.

The Fibonacci numbers were first described in Indian mathematics, as early as 200 BC in work by Pingala on enumerating possible patterns of Sanskrit poetry formed from syllables of two lengths. They are named after the Italian mathematician Leonardo of Pisa, also known as Fibonacci, who introduced the sequence to Western European mathematics in his 1202 book *Liber Abaci*.

In this report, the objectives to be achieved include:

- a. Understand about basic web.
- b. Understand about Fibonacci.

CHAPTER II

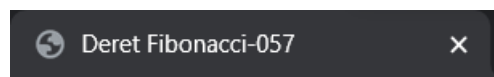
MAIN CONTENT

HTML (Hypertext Markup Language), CSS (Cascading Style Sheets) and JavaScript are the basic technologies used in web development to create and design websites with interactive features, including Each type serves a different purpose. HTML is the backbone of web content and is used to structure the content of a website. It identifies elements and their relationships on a web page. CSS is used to control the presentation and style of HTML elements. It determines how web content is displayed, including layout, fonts, colors, and spacing. JavaScript is a programming language used to add interactivity and behavior to web pages. It allows you to create dynamic content, respond to user actions, and manipulate your website's Document Object Model (DOM). Here In the context of a Fibonacci web calculator, HTML provides the structure and layout for the user interface, allowing users to input values and see the results, CSS can be applied to make the calculator visually appealing and user-friendly, and JavaScript code defines the generateFibonacci() function, which calculates and displays the Fibonacci sequence based on the user's input.

a) *HTML*

This HTML code represents the structure of a web page that includes input fields and a button for generating a Fibonacci sequence. I will breakdown each section:

1. `<!DOCTYPE html>`: This declaration specifies the document type and version of HTML being used, which is HTML5 in this case.
2. `<html>`: This is the root element of the HTML document, encapsulating all the content.
3. `<head>`: The head section of the HTML document contains metadata about the document, such as character encoding, viewport settings, and the page title. Here the thing that contain in the `<head>` :
 - i. `<meta charset="UTF-8">`: Specifies the character encoding for the document as UTF-8, which is a widely used character encoding for handling various characters and symbols.
 - ii. `<meta name="viewport" content="width=device-width, initial-scale=1">`: This meta tag is commonly used for responsive web design. It sets the viewport to the width of the device and sets the initial zoom level to 1.
 - iii. `<title>Deret Fibonacci-057</title>`: This sets the title of the web page, which appears in the browser's title bar or tab. Here is “Deret Fibonacci-057”



- iv. `<link rel="stylesheet" href="style.css">`: This links an external CSS stylesheet named "style.css" to the HTML document. This stylesheet likely contains styles such as layout, fonts, colours, etc that will be applied to the page's elements.
- 4. `<body>`: The body section contains the visible content of the web page.
 - i. `<div class="container">`: This is a container div that wraps all the page content. It's often used for layout and styling purposes.
 - ii. `<div class="fibonanci">`: Inside the container, there's a div with the class "fibonanci," which appears to be a section or block of content related to Fibonacci sequences.
 - iii. `<h1>Fibonacci Sequence Generator</h1>`: This is a heading displaying "Fibonacci Sequence Generator."

Fibonacci Sequence Generator

- iv. `<label for="input">Input Number:</label>`: This label is associated with an input element using the `for` attribute. It tells the user what the input field is for.
- v. `<div class="row">`: This div with the class "row" appears to be used for layout purposes.
- vi. `<input type="number" id="input" placeholder="Input desire number">`: This is an input field of type "number" with the id "input." Users can enter a number here. The "placeholder" attribute provides a hint to the user.
- vii. `<button onclick="generateFibonacci()">Generate</button>`: This is a button element with the label "Generate." When clicked, it triggers the JavaScript function `generateFibonacci()` (which should be defined in the linked "script.js" file).

Input Number:

Generate

- viii. `<label for="answer">The Answer:</label>`: Similar to the first label, this one is associated with an element with the id "answer."
- ix. `<div id="output"></div>`: This empty div with the id "output" is likely where the generated Fibonacci sequence will be displayed. The content will be inserted into this div dynamically via JavaScript.

Input Number:

5

Generate

The Answer:

Fibonacci Sequence with 5 number:

0, 1, 1, 2, 3

5. `<script src="script.js"></script>`: This line includes an external JavaScript file named "script.js." This JavaScript file likely contains the code responsible for generating the Fibonacci sequence and updating the "output" div with the result.

b) CSS

This CSS code defines the styles and layout for various elements of a web page. I'll break down each section right here.

1. `body`: These styles are applied to the entire body of the web page.
 - i. ``font-family: 'Poppins', sans-serif;``: It specifies the font family for the entire page. It first tries to use the "Poppins" font and, if not available, falls back to a generic sans-serif font.
 - ii. ``box-sizing: border-box;``: This ensures that when you set the width or height of an element, it includes padding and borders in the calculation, making it easier to control the layout.
 - iii. ``margin: 0;``: It removes any default margin around the body.
 - iv. ``padding: 0;``: It removes any default padding around the body.
2. ``.container``: These styles are applied to an element with the class "container."
 - i. ``width: 100%;``: It makes the container span the entire width of its parent element.
 - ii. ``min-height: 100vh;``: It sets the minimum height of the container to 100% of the viewport height, ensuring that it at least covers the entire viewport.
 - iii. ``background: linear-gradient(145deg, #257715, #08145f);``: It sets a linear gradient background for the container. The gradient goes from color #257715 to color #08145f at a 145-degree angle.
 - iv. ``padding: 10px;``: It adds 10 pixels of padding to the container.
3. ``.fibonanci``: These styles are applied to an element with the class "fibonanci."

- i. ``max-width: 400px;``: It sets the maximum width of the "fibonanci" element to 400 pixels.
 - ii. ``margin: 0 auto;``: It centers the "fibonanci" element horizontally within its parent.
 - iii. ``background-color: #fff;``: It sets the background color of the "fibonanci" element to white.
 - iv. ``border: 2px solid #6b6b6b;``: It adds a 2-pixel solid border with a color of #6b6b6b.
 - v. ``border-radius: 10px;``: It rounds the corners of the "fibonanci" element with a 10-pixel border radius.
 - vi. ``padding: 30px;``: It adds 30 pixels of padding inside the "fibonanci" element.
 - vii. ``text-align: center;``: It centers the text content within the "fibonanci" element.
- 4. ``h1``: These styles are applied to all ``h1`` elements on the page.
 - i. ``font-size: 24px;``: It sets the font size of ``h1`` elements to 24 pixels.
 - ii. ``color: #004c9c;``: It sets the text color to #004c9c, which is a shade of blue.
- 5. ``label``: These styles are applied to all ``label`` elements on the page.
 - i. ``display: block;``: It makes ``label`` elements block-level elements, so they start on a new line and have some spacing around them.
 - ii. ``font-weight: bold;``: It makes the text in ``label`` elements bold.
 - iii. ``margin: 30px auto 20px;``: It adds margin to the top, auto (horizontally centered), and 20 pixels to the bottom of ``label`` elements.
- 6. ``input-container`` and ``row``: These styles are applied to elements with the classes "input-container" and "row," respectively. However, it appears that there are no elements with these classes in the provided HTML code.
- 7. ``input``: These styles are applied to all ``input`` elements on the page.
 - i. Various styles are applied, such as removing borders, outlines, and setting a transparent background.
- 8. ``button``: These styles are applied to all ``button`` elements on the page.
 - i. Various styles are applied, including padding, background color, text color, border, border-radius, cursor style, and a hover effect to change the background color on hover.
- 9. ``#output``: These styles are applied to an element with the ID "output."
 - i. ``max-width: 400px;``: It sets the maximum width of the "output" element to 400 pixels.
 - ii. ``margin: auto;``: It centers the "output" element horizontally within its parent.
 - iii. ``background: #d9d9d9;``: It sets the background color of the "output" element to a light gray.

- iv. ``border-radius: 20px;``: It rounds the corners of the "output" element with a 20-pixel border radius.
- v. ``padding: 20px;``: It adds 20 pixels of padding inside the "output" element.
- vi. ``text-align: center;``: It centers the text content within the "output" element.
- vii. ``font-weight: 15px;``: It sets the font weight to 15 pixels (which is not a valid value; it should typically be a "bold" or numeric value like 400 or 700).

c) *JavaScript*

This JavaScript code defines a function called ``generateFibonacci()`` that generates a Fibonacci sequence based on user input and displays the result on a web page. Let's break down the code step by step:

1. ``const inputElement = document.getElementById("input");`` and ``const outputElement = document.getElementById("output");``: These lines retrieve references to HTML elements with the IDs "input" and "output" using the ``getElementById`` method. The ``inputElement`` will represent the input field where the user enters a number, and the ``outputElement`` will represent the area where the generated Fibonacci sequence will be displayed.
2. ``const n = parseInt(inputElement.value);``: This line extracts the numeric value entered by the user in the input field and stores it in the variable ``n``. It uses ``parseInt`` to convert the input value from a string to an integer.
3. ``if (!isNaN(n) && n > 0) {``: This conditional statement checks if ``n`` is a valid positive number. It does the following checks:
 - i. ``!isNaN(n)``: This checks if ``n`` is a valid number by using the ``isNaN`` (is not a number) function. If ``n`` is not a valid number, this condition evaluates to ``false``.
 - ii. ``n > 0``: This checks if ``n`` is greater than zero, ensuring that the number is positive.
4. If both conditions in the ``if`` statement are met (i.e., ``n`` is a valid positive number), the code inside the ``if`` block executes:
 - i. ``let fibonacci = [0, 1];``: This initializes an array called ``fibonacci`` with the first two Fibonacci numbers (0 and 1). These are the starting values for generating the sequence.
 - ii. The ``for`` loop: This loop iterates from ``i = 2`` to ``i < n``, where ``n`` is the user-provided input.
 - iii. ``fibonacci.push(fibonacci[i - 1] + fibonacci[i - 2]);``: In each iteration, it calculates the next Fibonacci number by adding the two previous numbers in the sequence ($F(i) = F(i-1) + F(i-2)$). The calculated Fibonacci number is pushed to the ``fibonacci`` array.
 - iv. ``outputElement.innerHTML = `...```: Once the Fibonacci sequence is generated, it is displayed in the ``outputElement`` using the ``innerHTML`` property. This

replaces the content inside the `outputElement` with HTML that includes the Fibonacci sequence with the input value.

5. If the `if` condition is not met (i.e., `n` is not a valid positive number), the code inside the `else` block executes:

- i. `outputElement.innerHTML = "<p>Please input the valid number
(positive number)</p>";`: This sets the `innerHTML` of `outputElement` to a message asking the user to input a valid positive number. In here, there is put `` is for bolding the message that will shown. The `
` code is for put the next message into the new line.

CHAPTER III

CLOSING

A. Conclusion

In summary, this HTML code defines a web page with input fields for entering a number, a button for generating a Fibonacci sequence, and an area to display the result. The page's appearance and functionality are likely further defined and controlled by the linked CSS and JavaScript files ("style.css" and "script.js"). The CSS code defines the styling for various elements of a web page, including fonts, colors, backgrounds, spacing, and alignment. These styles are applied to create a visually appealing and responsive user interface. The JavaScript code defines a function that generates a Fibonacci sequence based on user input and updates a designated HTML element with the result. It performs input validation to ensure that the user enters a positive number and provides a message if the input is invalid.