CSS: CONCEPTS, ARCHITECTURE AND OUTLOOK

Vukasin Golijanin

H1615742

BIS SEMINAR JUNE 2024

TABLE OF CONTENT

01 Basic Concepts

02 Architecture

03 Outlook

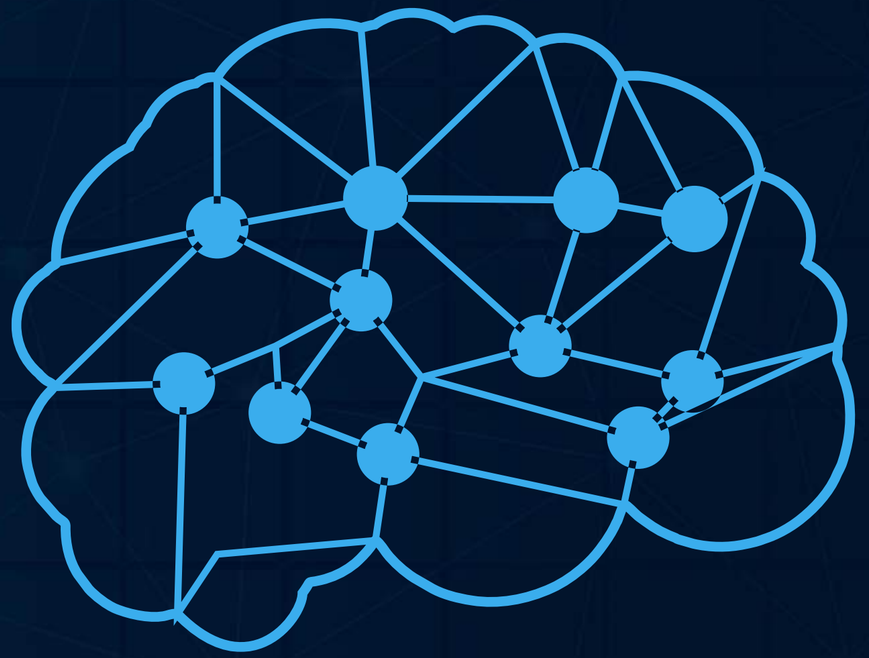
INTRODUCTION

- Initiated by Hakon Wium Lie at CERN in 1994; CSS addressed the need for styling HTML texts without altering their structure
- Formally released in December 1996 and quickly adopted by major browsers (Netscape, Internet Explorer)
- Has advanced to options like Google Fonts, Typekit, and animations, significantly improving user experiences and website customization.
- As of today, CSS is used by approximately 97% of all websites to visually enhance their interfaces.

1

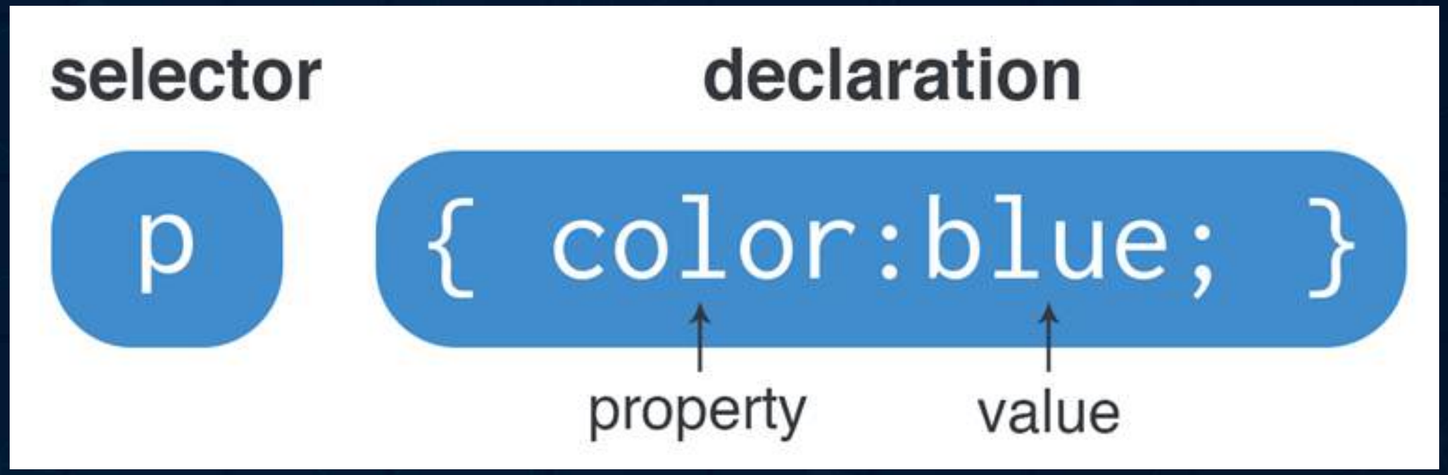
BASIC CONCEPTS

- CSS (Cascading Style Sheets) is designed to enhance the visual appeal of web pages
- Applies styles to HTML documents, defining fonts, colors, spacing, and more

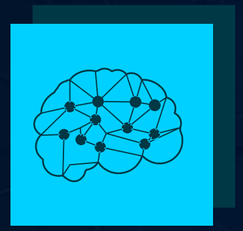


SYNTAX

- CSS syntax consists of rules and guidelines for styling HTML documents
- Includes a selector and a declaration block, which contains one or more declarations.
- Each declaration is made up of a property and its value, separated by a colon.

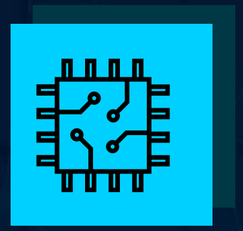


SELECTORS



SIMPLE

- Element Selector
- Class Selector
- ID Selector



PSEUDO-ELEMENT/ CLASS

- Pseudo-Classes
- Pseudo-Elements

COMBINATOR

- Descendant Combinator
- Child Combinator
- Sibling Combinators

SELECTORS

class selector.html > ...

```html
<html lang="en">
  <head>
    <link rel="stylesheet" type="text/css" href="css/BIS.css" />
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
  </head>
  <body>
    <h1 class="BIS">Class selector</h1>
  </body>
</html>
```

css > # BIS.css > h1.BIS

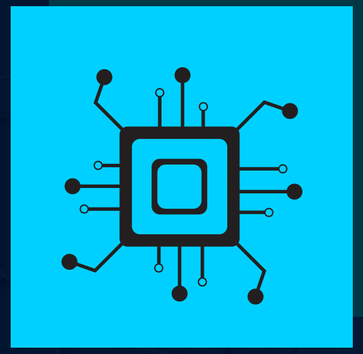
```css
h1.BIS {
font-family: Times New Roman;
 color: blue;}
```

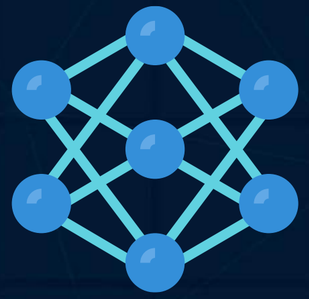
Class selector

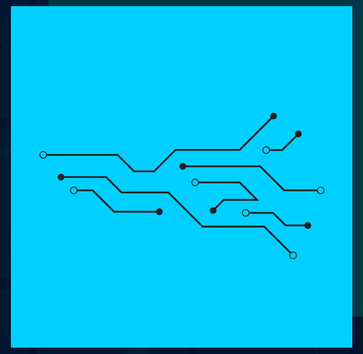
SELECTORS

Selector	Example	Example description
element element	div p	Selects all <p> elements inside elements
element>element	div > p	Selects all <p> elements where the parent is a element
element+element	div + p	Selects the first <p> element that are placed immediately after elements
element1~element2	p ~ ul	Selects every element that are preceded by a <p> element

PROPERTIES AND VALUES







PROPERTIES

- Define the styles applied to specified selectors, appearing before the values in the CSS ruleset and separated by a colon
- Common property types include list properties, font properties, border properties, and text properties
- These are essential for styling web pages and can be applied to various selectors

VALUES

- CSS values come in different types, each serving a specific purpose (example: Text values are used for properties like color and text-align)
- Relative and Absolute Units
- URLs and Colors

INCLUDING CSS IN HTML

01

Inline CSS

involves adding CSS styles directly within HTML elements using the "style" attribute

Inline CSS.html > ...

```html
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
  </head>
  <body>
    <h1 style="font-family: Times New Roman; color: red">BIS Seminararbeit</h1>
  </body>
</html>
```

02

Internal CSS

defined within the <style> tag inside the <head> section of an HTML document

Internal CSS.html > ...

```html
<html lang="en">
  <head>
    <style>
      h1 {
        font-family: Times New Roman;
        color: red;
      }
    </style>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
  </head>
  <body>
    <h1>BIS Seminararbeit</h1>
  </body>
</html>
```

03

External CSS

involving a separate CSS file linked to the HTML file using a <link> tag in the <head> section

external CSS.html > ...

```html
<html lang="en">
  <head>
    <link rel="stylesheet" type="text/css" href="css/BIS.css" />
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
  </head>
  <body>
    <h1>BIS Seminararbeit</h1>
  </body>
</html>
```

CSS SPECIFICITY

SPECIFICITY

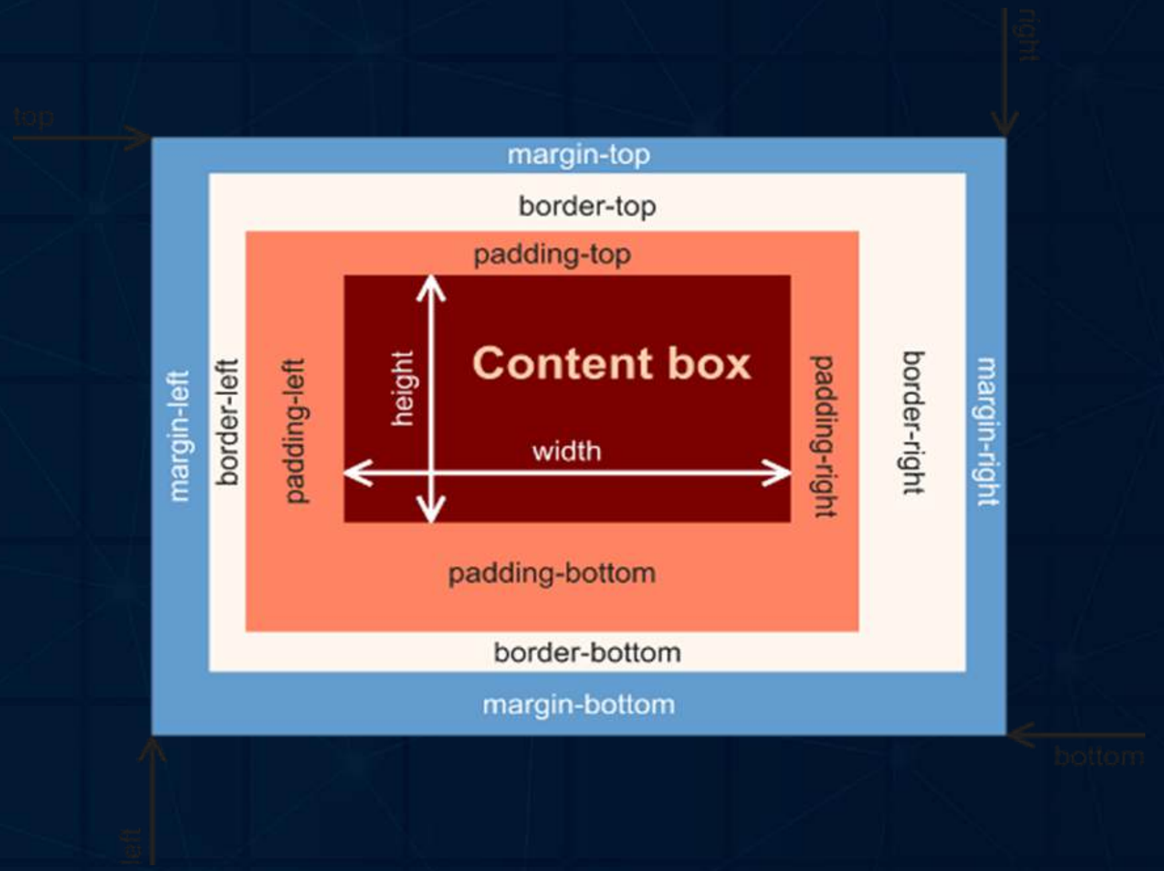
The total specificity value determines which rule is applied when multiple rules target the same element

!IMPORTANT RULE

It overrides any previously calculated specificity values, giving it the highest rank in terms of specificity

BOX MODEL

- Everything on webpages is determined by boxes, often not noticeable due to uniform background colors.
- The box model consists of the content, padding, border, and margin



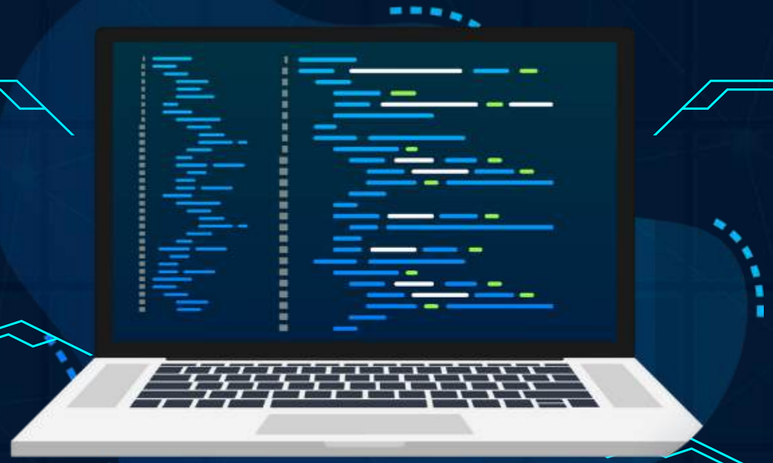

```
css > # BIS.css > ...
  1   #BIS {
  2   font-family: Times New Roman;
  3   color: red;
  4   width: 240px;
  5   height: 100px;
  6   background-color: green;
  7   padding-bottom: 1px;
  8   border: solid 5px;
  9   border-color: blue;
 10   border-left: 10px dotted yellow;
 11   padding-left: 25px;
 12   }
 13
```

CSS SPECIFICITY

Selector	Specificity Value	Calculation
p	1	1
p.test	11	1 + 10
p#demo	101	1 + 100
<p style="color: pink;">	1000	1000
#demo	100	100
.test	10	10
p.test1.test2	21	1 + 10 + 10
#navbar p#demo	201	100 + 1 + 100
*	0	0 (the universal selector is ignored)

2

ARCHITECTURE



READABILITY

Code should be simple and clear, with uniform formatting, easy-to-understand names, and comments when needed

SPECIFICITY

Selectors should be only as specific as necessary to avoid overly specific or complicated selectors that negatively impact maintainability

MODULARITY

Styles should be reusable across different parts of the application to reduce repetition and boost efficiency

SEPARATION OF CONCERNS

CSS files should be divided into logical sections, each focusing on a specific aspect of styling

ARCHITECTURE METHODOLOGIES

OBJECT-ORIENTED CSS (OOCSS)

BLOCK, ELEMENT, MODIFIER (BEM)

SCALABLE AND MODULAR ARCHITECTURE FOR CSS (SMACSS)

ATOMIC CSS (UTILITY-FIRST CSS)

ARCHITECTURE METHODOLOGIES

OBJECT-ORIENTED CSS (OOCSS)

BLOCK, ELEMENT, MODIFIER (BEM)

```
# OOCSS.css > ...
/* Object Part*/

.Object {
    font-size: 2em;
    font-weight: bold;
    text-align: center;
    width: 250px;
    height: 100px;

}

/* Skin Part*/

.Skin {
    background-color: black;
    color: aqua;
    border: solid 5px;
    border-color: aqua;
}
```

BIS Seminararbeit

```
<div class="title">
  <h1 class="title__text title__text_highlight">BIS Seminararbeit</h1>
</div>
```

```
1   /* Block */
2   .title {
3       padding: 10px;
4       background-color: black;
5       border: 1px solid;
6       border-color: aqua;
7   }
8
9   /* Element */
10  .title__text {
11      font-size: 2em;
12      font-weight: bold;
13      text-align: center;
14      margin: 0;
15  }
16
17  /* Modifier */
18  .title__text_highlight {
19      background-color: black;
20      color: aqua;
21      border: solid 3px aqua;
22  }
```

BIS Seminararbeit

ARCHITECTURE METHODOLOGIES

SCALABLE AND MODULAR ARCHITECTURE FOR CSS (SMACSS)

```html
<body class="theme-light">
  <header class="header">
    <h1 class="header__title">BIS Seminararbeit</h1>
  </header>
</body>
```

```css
/* Base Rule */
html, body {
    margin: 0;
    padding: 0;
    font-family: 'Times New Roman';
}

h1 {
    margin: 0;
    padding: 0;
}

/* Layout Rule */
.header {
    background-color: black;
    padding: 20px;
    border-bottom: 2px solid red;
    text-align: center;
}

/* Module Rule */
.header__title {
    font-size: 4em;
    font-weight: bold;
}

/* Theme Rule */
.theme-light {
    color: lightblue;
}
```

BIS Seminararbeit

ATOMIC CSS (UTILITY-FIRST CSS)

```html
<body class="font-times">
  <header class="bg-black p-20 text-center border-bottom-red">
    <h1 class="m-0 font-bold text-lightblue">BIS Seminararbeit</h1>
  </header>
```

```css
/* Atomic CSS Classes */
.m-0 { margin: 0; }
.p-20 { padding: 20px; }
.bg-black { background-color: yellow; }
.text-center { text-align: center; }
.font-bold { font-weight: bold; }
.text-lightblue { color: green; }
.border-bottom-red { border-bottom: 2px solid red; }
.font-times { font-family: 'Times New Roman'; }
```

BIS Seminararbeit

OOCSS

- Separates structure from visual content (object from skin)
- No specific naming conventions, making it intuitive to understand and use
- Lack of naming conventions can hinder shared understanding among developers
- Different perceptions of properties (e.g., border) can lead to inconsistencies

BEM

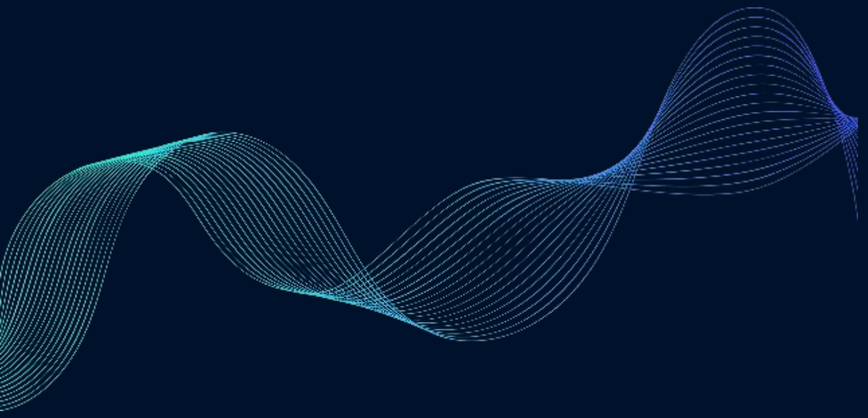
- Focuses on modularity and reusable code with a strict naming convention
- Naming conventions define relationships between element for more organized and structured files
- Promotes collaboration among multiple developers
- Potential for over-nesting and increased complexity in large codebases
- Strict naming conventions can be challenging for new developers to learn

SMACSS

- Aims to logically divide CSS code for scalability and maintainability
- Divides code into five categories for easier management and updates
- Larger CSS files become more structured
- No specific naming conventions despite appearing rigid
- Adhering to five categories can be time-consuming and tends to be mis categorized
- Miscommunication among developers can lead to categorization issues

ATOMIC CSS

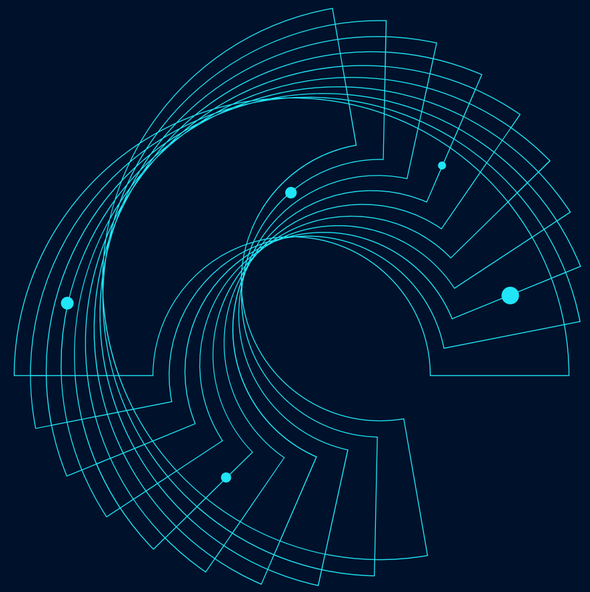
- Involves creating reusable single-property classes
- Simplifies code updates by naming properties as classes in HTML
- Promotes quick code writing and reduces redundancy with pre processors
- Multiple properties attached to classes can make the code complex and hard to read
- Numerous single-property classes can bloat the CSS file
- Naming conventions introduce another learning curve

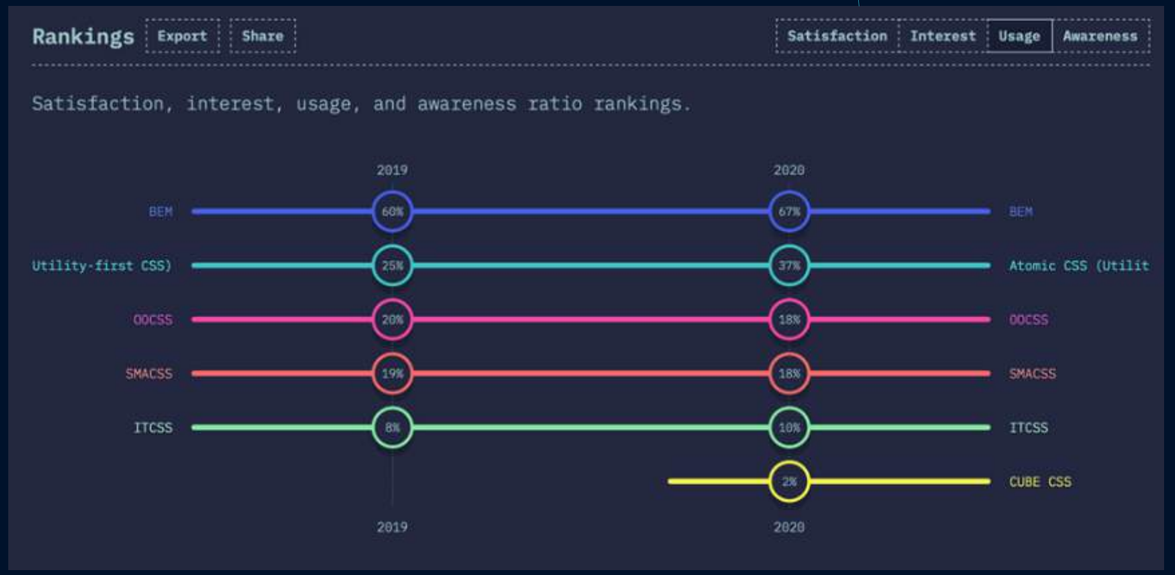


SATISFACTION

AWARENESS

INTEREST





USAGE

3

OUTLOOK

- CSS will continue to evolve and adapt to the changes of web development and become more versatile
- Methodologies like BEM, OOCSS, SMACSS, and Atomic CSS have laid the foundation for scalable and reusable CSS
- CSS will play an important role in enabling new forms of interaction and animation, especially with the rise of VR
- The integration of AI into web development could change how CSS is written and managed

“AI is here and it’s being used at scale. 92% of U.S.-based developers are already using AI coding tools both in and outside of work.“

(STAFF, 2023)

“Developers also see big benefits to AI. 70% say AI coding tools will offer them an advantage at work and cite better code quality, completion time, and resolving incidents as some of the top anticipated benefits.“

(STAFF, 2023)

Thank you for your attention!

REFERENCES

- A brief history of CSS until 2016. (o. J.). Abgerufen 18. März 2024, von https://www.w3.org/Style/CSS20/history.html
- Advance CSS - 101.school. (o. J.). Abgerufen 3. Mai 2024, von https://101.school/courses/advance-css/modules/9-css-architecture-and-methodologies/units/1-introduction-to-bem-and-smacss
- Base Rules—Scalable and Modular Architecture for CSS. (o. J.). Abgerufen 31. Mai 2024, von https://smacss.com/book/type-base/
- Categorizing CSS Rules—Scalable and Modular Architecture for CSS. (o. J.). Abgerufen 31. Mai 2024, von https://smacss.com/book/categorizing/
- Combinators—Learn web development | MDN. (2024, Jänner 1). https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/Selectors/Combinators
- contributors, V. S., Vladimir Starkov and. (o. J.). BEM — Block Element Modifier. Abgerufen 29. Mai 2024, von http://getbem.com/
- Cruz, J. C. Y. (o. J.). CSS Methodologies | Juan Carlos Yovera Cruz. Abgerufen 16. Mai 2024, von https://jconsoftwares.com/blog/24
- CSS Combinators. (o. J.). Abgerufen 2. April 2024, von https://www.w3schools.com/css/css_combinators.asp
- CSS Introduction. (2018, April 10). GeeksforGeeks. https://www.geeksforgeeks.org/css-introduction/
- CSS Pseudo-elements. (o. J.). Abgerufen 2. Juni 2024, von https://www.w3schools.com/css/css_pseudo_elements.asp
- CSS Selectors. (o. J.). Abgerufen 1. April 2024, von https://www.w3schools.com/css/css_selectors.asp
- CSS Selectors, Properties and Values. (o. J.). CodeCoda.Com. Abgerufen 31. Mai 2024, von https://codecoda.com/en/blog/entry/css-selectors-properties-and-values?limit=all
- CSS Specificity. (o. J.). Abgerufen 1. April 2024, von https://www.w3schools.com/css/css_specificity.asp
- CSS Syntax. (2019, Februar 13). GeeksforGeeks. https://www.geeksforgeeks.org/css-syntax/
- CSS values and units - CSS: Cascading Style Sheets | MDN. (2024, März 21). https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Values_and_Units
- Difference Between CSS, CSS2 And CSS3 | What is CSS? (o. J.). Hackr.Io. Abgerufen 18. März 2024, von https://hackr.io/blog/difference-between-css-css2-and-css3
- History / Methodology / BEM. (o. J.). Abgerufen 29. Mai 2024, von https://en.bem.info/methodology/history/
- How to Link CSS to HTML – Stylesheet File Linking. (2022, Juni 14). freeCodeCamp.Org. https://www.freecodecamp.org/news/how-to-link-css-to-html/
- HTML a rel Attribute. (o. J.). Abgerufen 29. März 2024, von https://www.w3schools.com/tags/att_a_rel.asp
- HTML and CSS – Inline Style, External Stylesheet, CSS Code Examples. (2021, August 12). freeCodeCamp.Org. https://www.freecodecamp.org/news/html-and-css-inline-style-external-stylesheet-css-code-examples/
- HTML Standard. (o. J.). Abgerufen 29. März 2024, von https://html.spec.whatwg.org/#the-link-element
- HTML Styles CSS. (o. J.). Abgerufen 28. März 2024, von https://www.w3schools.com/html/html_css.asp

REFERENCES

- Introduction—Scalable and Modular Architecture for CSS. (o. J.). Abgerufen 31. Mai 2024, von https://smacss.com/book/
- Layout Rules—Scalable and Modular Architecture for CSS. (o. J.). Abgerufen 31. Mai 2024, von https://smacss.com/book/type-layout/
- Let's Define Exactly What Atomic CSS is. (2017, April 10). CSS-Tricks. https://css-tricks.com/lets-define-exactly-atomic-css/
- Mastering Object-Oriented CSS (OOCSS). (2023, Juni 22). DEV Community. https://dev.to/ladunjexa/mastering-object-oriented-css-oocss-3b5h
- Module Rules—Scalable and Modular Architecture for CSS. (o. J.). Abgerufen 31. Mai 2024, von https://smacss.com/book/type-module/
- Most commonly used tags in HTML. (2018, August 23). GeeksforGeeks. https://www.geeksforgeeks.org/most-commonly-used-tags-in-html/
- olawanletjoel. (2022, Juni 8). Inline Style in HTML – CSS Inline Styles. freeCodeCamp.Org. https://www.freecodecamp.org/news/inline-style-in-html/
- Pseudo-classes and pseudo-elements—Learn web development | MDN. (2024, Jänner 1). https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/Selectors/Pseudo-classes_and_pseudo-elements
- Quick start / Methodology / BEM. (o. J.). Abgerufen 29. Mai 2024, von https://en.bem.info/methodology/quick-start/
- Simuratli. (2020, Oktober 16). What is OOCSS(Object Oriented CSS)? Simuratli. https://medium.com/simuratli/what-is-oocss-object-oriented-css-c8667fd65dfe
- Staff, I. S., GitHub. (2023, Juni 13). Survey reveals AI's impact on the developer experience. The GitHub Blog. https://github.blog/2023-06-13-survey-reveals-ais-impact-on-the-developer-experience/
- State Rules—Scalable and Modular Architecture for CSS. (o. J.). Abgerufen 31. Mai 2024, von https://smacss.com/book/type-state/
- Team, E. A. (2023, Dezember 21). Will Generative AI Replace or Empower HTML & CSS Developers? Exadel. https://exadel.com/news/gen-ai-and-html-css-developers/
- The box model—Learn web development | MDN. (o. J.). Abgerufen 30. April 2024, von https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/The_box_model#using_display_inline-block
- The Making of Atomic CSS: An Interview With Thierry Koblentz. (2022, Februar 3). CSS-Tricks. https://css-tricks.com/thierry-koblentz-atomic-css/
- The State of CSS 2020: CSS Methodologies. (o. J.). Abgerufen 2. Juni 2024, von https://2020.stateofcss.com/en-US/technologies/methodologies/
- Theme Rules—Scalable and Modular Architecture for CSS. (o. J.). Abgerufen 31. Mai 2024, von https://smacss.com/book/type-theme/
- Usage Statistics of CSS for Websites, March 2024. (o. J.). Abgerufen 18. März 2024, von https://w3techs.com/technologies/details/ce-css