

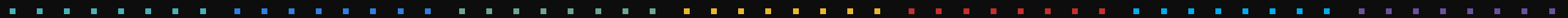
# WMNWA 2110 | WEB APPLICATIONS

## React #10

### Styling



# ΠΕΡΙΕΧΟΜΕΝΑ



## Περιεχόμενα

- **Material-UI / MUI**
- **styled-components**
- **Emotion**
- **vanilla-extract**
- **TailwindCSS**

# MATERIAL-UI

# Material-UI

## Introduction

*MUI provides a robust, customizable, and accessible library of foundational and advanced components, enabling you to build your own design system and develop React applications faster.*

# Material-UI

## Installation

```
# with npm
npm install @mui/material @emotion/react @emotion/styled

# with yarn
yarn add @mui/material @emotion/react @emotion/styled
```

# Material-UI

## Suggested fonts

*MUI was designed with the Roboto font in mind*

```
<link
  rel="stylesheet"
  href="https://fonts.googleapis.com/css?family=Roboto:300,400,500,700&display=swap"
/>
```

# Material-UI

## Icon component

*To use the font Icon component, you must first add the Material icons font*

```
<link
  rel="stylesheet"
  href="https://fonts.googleapis.com/icon?family=Material+Icons"
/>
```



# Material-UI

## SVG icons

*In order to use prebuilt SVG Material icons, such as those found in the icons demos you must first install the @mui/icons-material package.*

```
# with npm
npm install @mui/icons-material

# with yarn
yarn add @mui/icons-material
```

# STYLED-COMPONENTS

# styled-components

## Introduction

*Utilising tagged template literals (a recent addition to **JavaScript**) and the power of **CSS**, styled-components allows you to write actual **CSS** code to style your components.*

*It also removes the mapping between components and styles – using components as a low-level styling construct could not be easier!*

# styled- components

## Installation

```
# with npm  
npm install --save styled-components  
  
# with yarn  
yarn add styled-components
```

# styled-components

## Installation

If you use a package manager like yarn that supports the *"resolutions"* *package.json* field, we also highly recommend you add an entry to it as well corresponding to the major version range.

This helps avoid an entire class of problems that arise from multiple versions of **styled-components** being installed in your project.

```
{
  "resolutions": {
    "styled-components": "^5"
  }
}
```

# styled- components

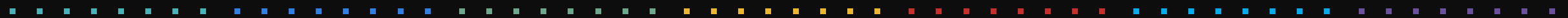
## Installation

In case you use **TypeScript**

```
# with npm
npm install --save-dev @types/styled-components

# with yarn
yarn add @types/styled-components --dev
```

# EMOTION



# Emotion

## Introduction

***Emotion** is a library designed for writing css styles with **JavaScript**.*

*It provides powerful and predictable style composition in addition to a great developer experience with features such as source maps, labels, and testing utilities.*

*Both string and object styles are supported.*



# Emotion

## Installation

### Framework Agnostic

```
# with npm  
npm i @emotion/css  
  
# with yarn  
yarn add @emotion/css
```

# Emotion

## Installation

### React

```
# with npm  
npm i @emotion/react  
  
# with yarn  
yarn add @emotion/react
```

# Emotion

## Installation

### React

The **@emotion/styled package** is for those who prefer to use the **styled.div** style **API** for creating components.

```
# with npm
npm i @emotion/styled @emotion/react

# with yarn
yarn add @emotion/styled @emotion/react
```

# VANILLA-EXTRACT

# vanilla- extract

## Introduction

Use **TypeScript** as your preprocessor.

Write type-safe, locally scoped classes, variables and themes, then generate static **CSS** files at build time.

Basically, it's "**CSS Modules-in-TypeScript**" but with scoped **CSS** Variables and heaps more.

# vanilla- extract

## Installation

```
# with npm  
npm install @vanilla-extract/css  
  
# with yarn  
yarn add @vanilla-extract/css
```

# vanilla- extract

## Installation

### Vite

```
# with npm  
npm install @vanilla-extract/css @vanilla-extract/vite-plugin
```

```
# with yarn  
yarn add @vanilla-extract/css @vanilla-extract/vite-plugin
```

```
// vite.config.js  
import { vanillaExtractPlugin } from '@vanilla-extract/vite-plugin';  
  
export default {  
  plugins: [vanillaExtractPlugin()],  
};
```

# TAILWIND CSS



# Tailwind CSS

## Introduction

A utility-first **CSS** framework packed with classes like **flex**, **pt-4**, **text-center** and **rotate-90** that can be composed to build any design, directly in your markup.

Utility classes help you work within the constraints of a system instead of littering your stylesheets with arbitrary values.

They make it easy to be consistent with color choices, spacing, typography, shadows, and everything else that makes up a well-engineered design system.

# Tailwind CSS

## Installation

```
# with npm
npm install -D tailwindcss postcss autoprefixer
npx tailwindcss init -p

# with yarn
yarn add -D tailwindcss postcss autoprefixer
yarn run tailwind init -p
```

# Tailwind CSS

## Installation

### postcss.config.js

```
module.exports = {  
  plugins: {  
    tailwindcss: {},  
    autoprefixer: {},  
  },  
}
```


# Tailwind CSS

## Installation

### tailwind.config.js

```
module.exports = {
  content: [
    "./src/**/*.{js,jsx,ts,tsx}",
  ],
  theme: {
    extend: {},
  },
  plugins: [],
}
```

# Χρήσιμα links

 Installation - MUI


<https://mui.com/getting-started/installation/>

 Getting Started - Chakra UI

<https://chakra-ui.com/docs/getting-started>

 styled-components: Basics

<https://styled-components.com/docs/basics#installati...>

 Emotion - Introduction

<https://emotion.sh/docs/introduction>

 Getting Started — vanilla-extract

<https://vanilla-extract.style/documentation/>

 Ant Design of React - Ant Design

<https://ant.design/docs/react/introduce>

 Installation: Tailwind CSS with Create React App - Tailwind CSS

<https://tailwindcss.com/docs/guides/create-react-app>

## Extra info

\* The CSS-in-React Landscape - CSS-Tricks  
<https://css-tricks.com/the-css-in-react-landscape/>

9 Ways To Implement CSS in React JS | by Dmitry Nozhenko | Medium  
<https://medium.com/@dmitrynozhenko/9-ways-to-im...>

Why I moved from Styled Components to (S)CSS modules // Puru Vijay  
<https://www.puruvj.dev/blog/move-to-css-modules-fr...>

Styled Components: To Use or Not to Use? | by Talia Marcassa | Building CrowdRiff | Medium  
<https://medium.com/building-crowdriff/styled-compo...>

DEV Using emotionJs with Vite - DEV Community  
<https://dev.to/ajitsinghkamal/using-emotionjs-with-vit...>

Getting Started — vanilla-extract  
<https://vanilla-extract.style/documentation/>

Template literals (Template strings) - JavaScript | MDN  
<https://developer.mozilla.org/en-US/docs/Web/JavaS...>

Tagged Template Literals - Wes Bos  
<https://wesbos.com/tagged-template-literals>

# THANK YOU!