

纯JS实现Web随操作系统主题切换-解决方案

— 穆贵林

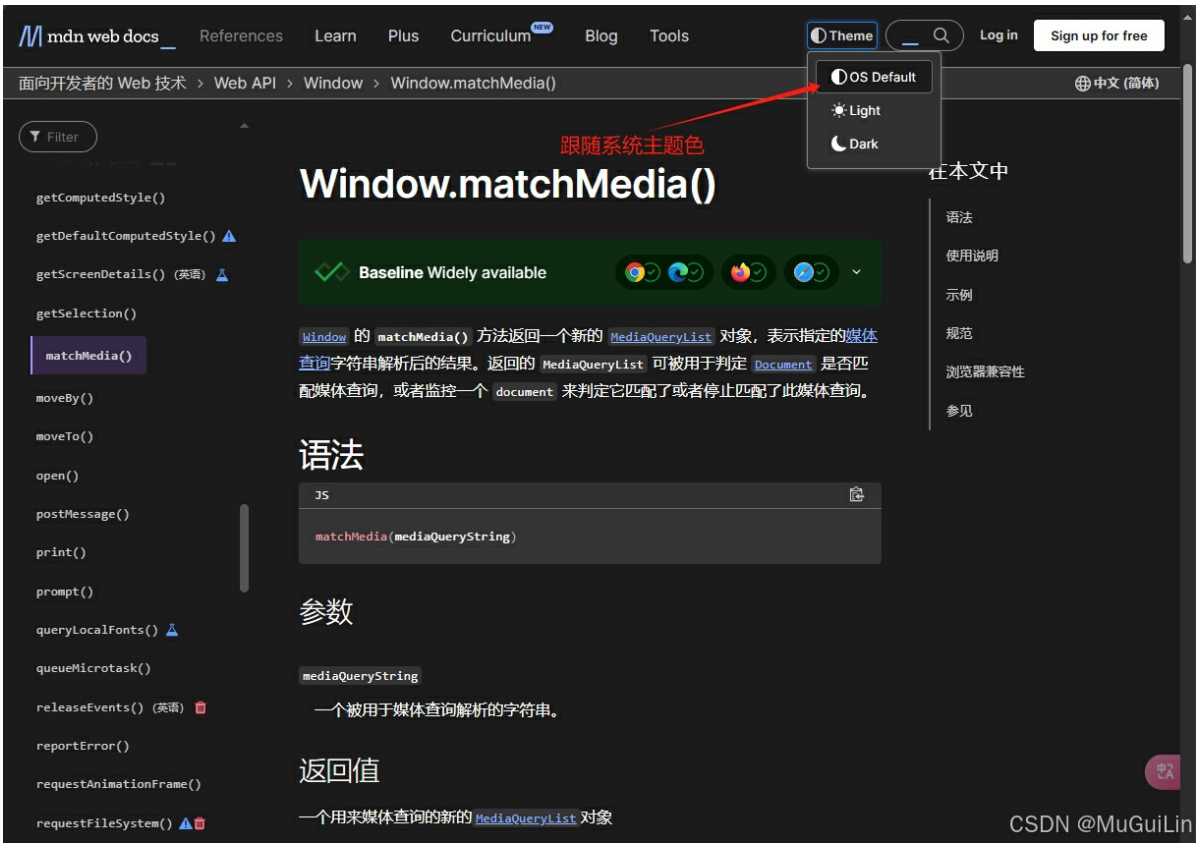
2025年03月6日

CSDN: [JS媒体查询之matchMedia API 实现跟随系统主题色切换效果](#)

写在前面

在网页设计中，跟随系统主题切换可以通过CSS和JavaScript实现。可以通过定义两套CSS变量，根据系统主题的颜色来切换变量的生效，从而实现不同主题下的页面样式变化。

例如，可以使用媒体查询API来获取系统主题的颜色，并根据匹配结果来适配页面的自定义属性。同时，可以注册媒体查询的change事件，当系统主题变化时重新调用函数，实现跟随系统的效果。



API简介

Window 的 [matchMedia\(\)](#) 方法返回一个新的 `MediaQueryList` 对象，表示指定的媒体查询字符串解析后的结果。返回的 `MediaQueryList` 可被用于判定 `Document` 是否匹配媒体查询，或者监控一个 `document` 来判定它匹配了或者停止匹配了此媒体查询。

实例效果

这里只是实现最简单的background-color背景颜色 和 color文字颜色的修改，在实际开发项目中也是一样的原理，将颜色添加到更的DOM元素（HTML标签）样式中即可。



实例代码（HTML版）

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>JS媒体查询之matchMedia API 实现跟随系统主题色切换效果</title>
  <style>
    :root {
      --text-color: #333;
      --bg-color: #fff;
    }

    html[data-theme="dark"] {
      --text-color: #fff;
      --bg-color: #333;
    }

    html[data-theme="light"] {
      --text-color: #333;
      --bg-color: #fff;
    }

    html[data-theme="red"] {
```

```

        --text-color: #fff;
        --bg-color: #ff0064;
    }

    html[data-theme="green"] {
        --text-color: #fff;
        --bg-color: #67c23a;
    }

    html[data-theme="blue"] {
        --text-color: #fff;
        --bg-color: #0091db;
    }

    body {
        background-color: var(--bg-color);
        color: var(--text-color);
    }

    h1 {
        color: var(--text-color);
        text-align: center;
    }

    h5 {
        text-align: center;
        color: red;
    }

    select {
        display: block;
        margin: 20px auto;
        font-size: 32px;
    }
</style>
</head>

<body>
    <h1>JS媒体查询之matchMedia API 实现跟随系统主题色切换效果</h1>
    <hr>

    <select name="theme" id="theme">
        <option value="light">亮色</option>
        <option value="dark">暗色</option>
        <option value="os">系统跟随</option>
        <option value="red">红色</option>
        <option value="green">绿色</option>
        <option value="blue">蓝色</option>
    </select>

    <h5>注：如果想要在系统主题色切换时，也跟着一起切换，请选中“系统跟随”选项！</h5>

    <script>
        const theme = document.getElementById('theme');

        // 获取系统主题

```

```

const metch = window.matchMedia('(prefers-color-scheme: dark)');

// 设置主题
const setTheme = (theme) => {
  document.documentElement.setAttribute('data-theme',
Object.prototype.toString.call(theme) === '[object MediaQueryListEvent]' ?
metch.matches ? 'dark' : 'light' : theme);
};

// 监听主题切换
theme.addEventListener('change', (e) => {
  if ('os' === e.target.value) {
    setTheme(metch.matches ? 'dark' : 'light');
    // 监听系统主题切换
    metch.addEventListener('change', setTheme);
  } else {
    setTheme(e.target.value);
    metch.removeEventListener('change', setTheme);
  }
});
</script>
</body>

</html>

```

实例代码 (Vue3.js + TS版)

Vue的Hooks文件 hooks/useTheme.ts

```

import { ref, watchEffect } from 'vue';

// 定义主题类型
type Theme = 'light' | 'dark' | 'os';

// 存储主题的key
const STORE_KEY = '__htheme__';

// 获取存储主题
const theme = ref<Theme>(localStorage.getItem(STORE_KEY) as Theme || 'light');

// 获取系统主题
const media = globalThis.matchMedia('(prefers-color-scheme: dark)');

// 监听系统主题变化
const listener = () => {
  document.documentElement.dataset.theme = media.matches ? 'dark' : 'light';
}

// 监听theme变量值变化
watchEffect(() => {
  // 更新存储
  localStorage.setItem(STORE_KEY, theme.value);
});

```

```

    if (theme.value === 'os') {
      // 设置系统主题
      listener();

      // 监听系统主题变化
      media.addEventListener('change', listener);

    } else {
      // 移除监听
      media.removeEventListener('change', listener);
      document.documentElement.dataset.theme = theme.value;
    }
  });

export function useTheme() {
  return {
    theme
  }
};

```

Vue模板文件 views/demo.vue

```

<script setup lang="ts">
  import { useTheme } from '@/hooks/useTheme';
  const { theme } = useTheme()
</script>

<template>
  <h1>JS媒体查询之matchMedia API 实现跟随系统主题色切换效果</h1>
  <hr>
  <select v-model="theme">
    <option value="light">亮色</option>
    <option value="dark">暗色</option>
    <option value="os">系统跟随</option>
    <option value="red">红色</option>
    <option value="green">绿色</option>
    <option value="blue">蓝色</option>
  </select>
  <h5>注：如果想要在系统主题色切换时，也跟着一起切换，请选中“系统跟随”选项！</h5>
</template>

<style>
  :root {
    --text-color: #333;
    --bg-color: #fff;
  }

  html[data-theme="dark"] {
    --text-color: #fff;
    --bg-color: #333;
  }

  html[data-theme="light"] {
    --text-color: #333;
    --bg-color: #fff;
  }
</style>

```

```
}

html[data-theme="red"] {
  --text-color: #fff;
  --bg-color: #ff0064;
}

html[data-theme="green"] {
  --text-color: #fff;
  --bg-color: #67c23a;
}

html[data-theme="blue"] {
  --text-color: #fff;
  --bg-color: #0091db;
}

body {
  background-color: var(--bg-color);
  color: var(--text-color);
}

h1 {
  color: var(--text-color);
  text-align: center;
}

h5 {
  text-align: center;
  color: red;
}

select {
  display: block;
  margin: 20px auto;
  font-size: 32px;
}
</style>
```