WELCOME TO

TAILWIND CSS

Table of Contents

- What is Tailwind CSS
- Colors
- Example for Colors
- Layout
- Examples for Layout
- Spacing & sizing
- Effects
- Effects with images - examples 1,2
- Effects with animations - example

WHAT IS TAILWIND CSS

Tailwind CSS is a tool that helps in styling your website /web app in a easy way. It provides a set of ready-made classes that you can add to utilize when working on your projects. These classes have specific styling rules, like setting colors or margins and by using these classes, you can quickly change the look and feel of your project without writing a lot of custom CSS code. It's like having a set of shortcuts that make styling faster and more convenient

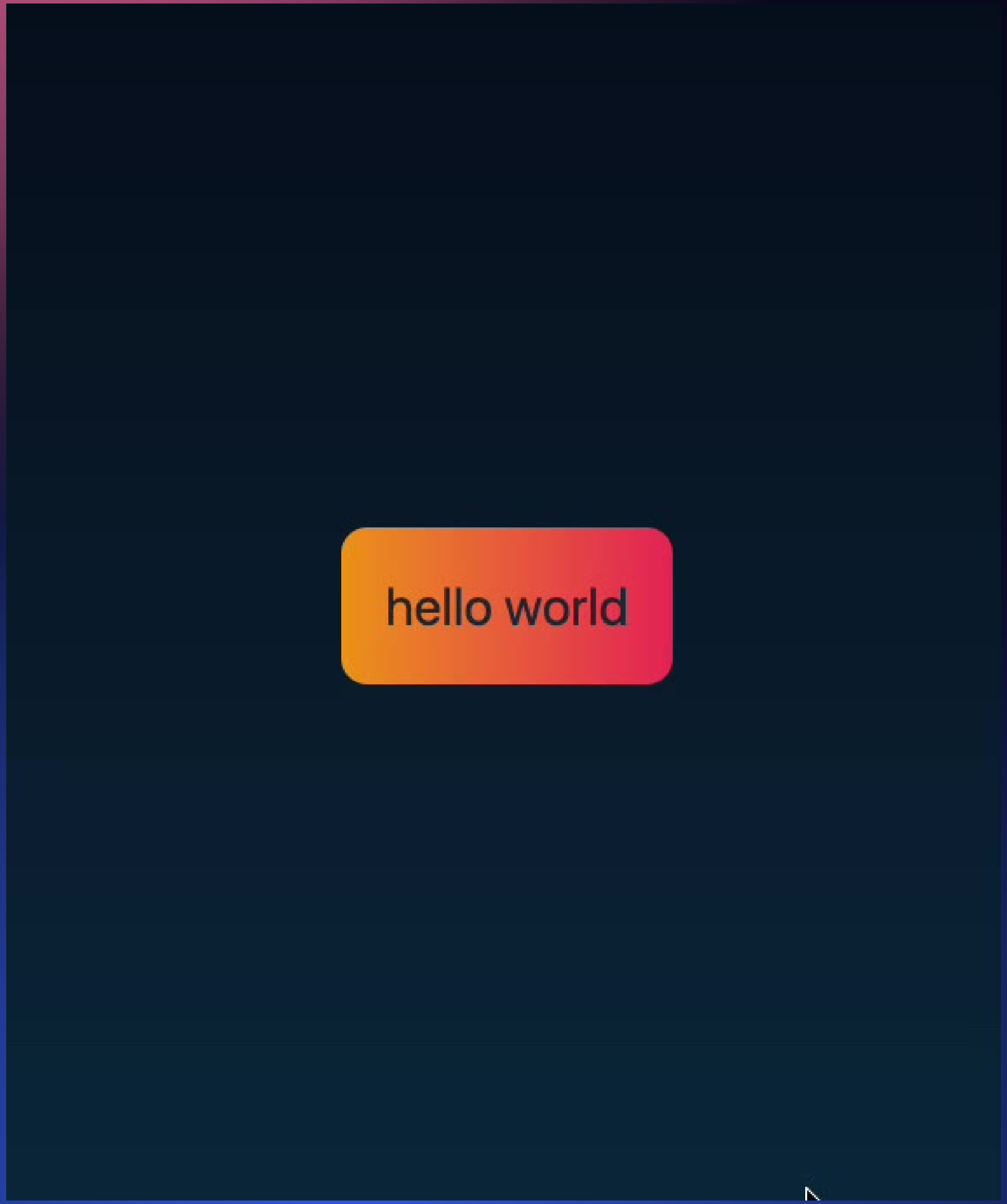
COLORS

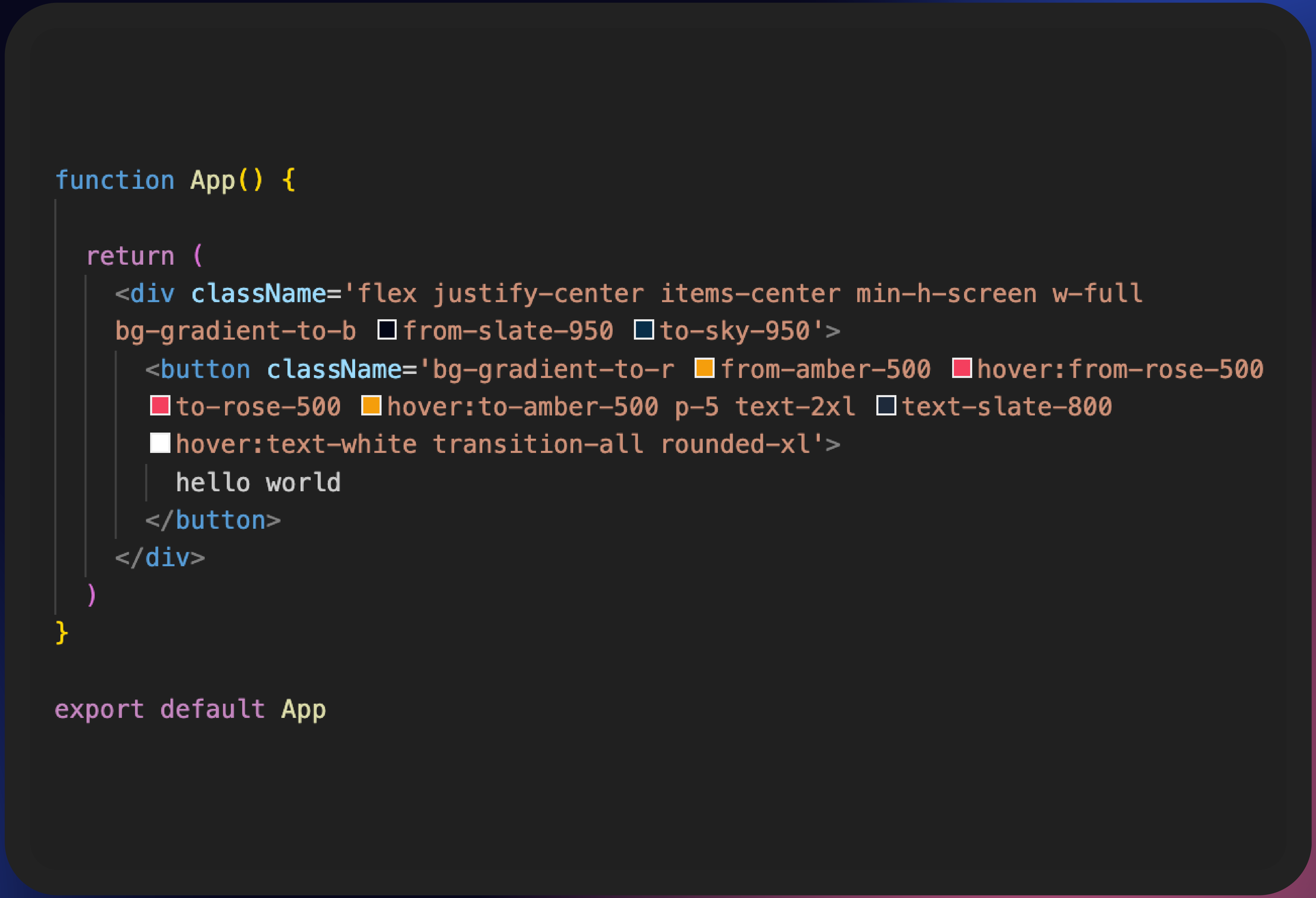
Inherit, current, transparent, black, white, slate, gray, zinc, neutral, stone, red, orange, amber, yellow, lime, green, emerald, teal, cyan, sky, blue, indigo, violet, purple, fuchsia, pink, rose

These color names (except the first 5) are followed by a scale from 50 to 950 in increments of 100 (first & last one by 50), indicating different shades of each color. For example, gray-50, gray-100, gray-200, gray-300, and so on up to 950. Additionally, Tailwind CSS enables adding custom colors in configuration file

bg-slate-600	bg-rose-500	bg-indigo-400	bg-amber-500	bg-sky-500
text-slate-600	text-rose-500	text-indigo-400	text-amber-500	text-sky-500

EXAMPLE 1





```
function App() {

  return (
    <div className='flex justify-center items-center min-h-screen w-full
    bg-gradient-to-b from-slate-950 to-sky-950'>
      <button className='bg-gradient-to-r from-amber-500 hover:from-rose-500
      to-rose-500 hover:to-amber-500 p-5 text-2xl text-slate-800
      hover:text-white transition-all rounded-xl'>
        hello world
      </button>
    </div>
  )
}

export default App
```

LAYOUT

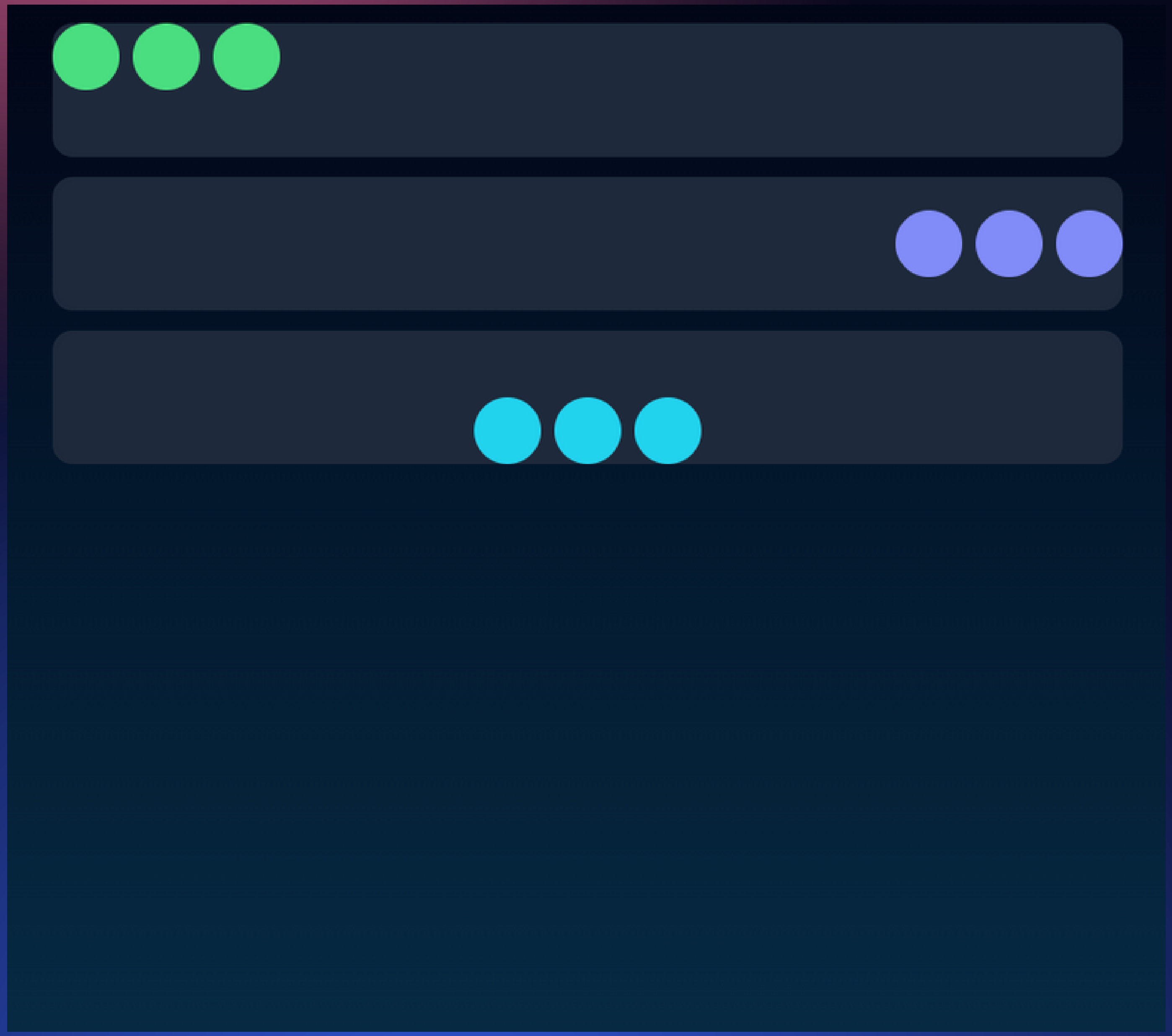
Flex

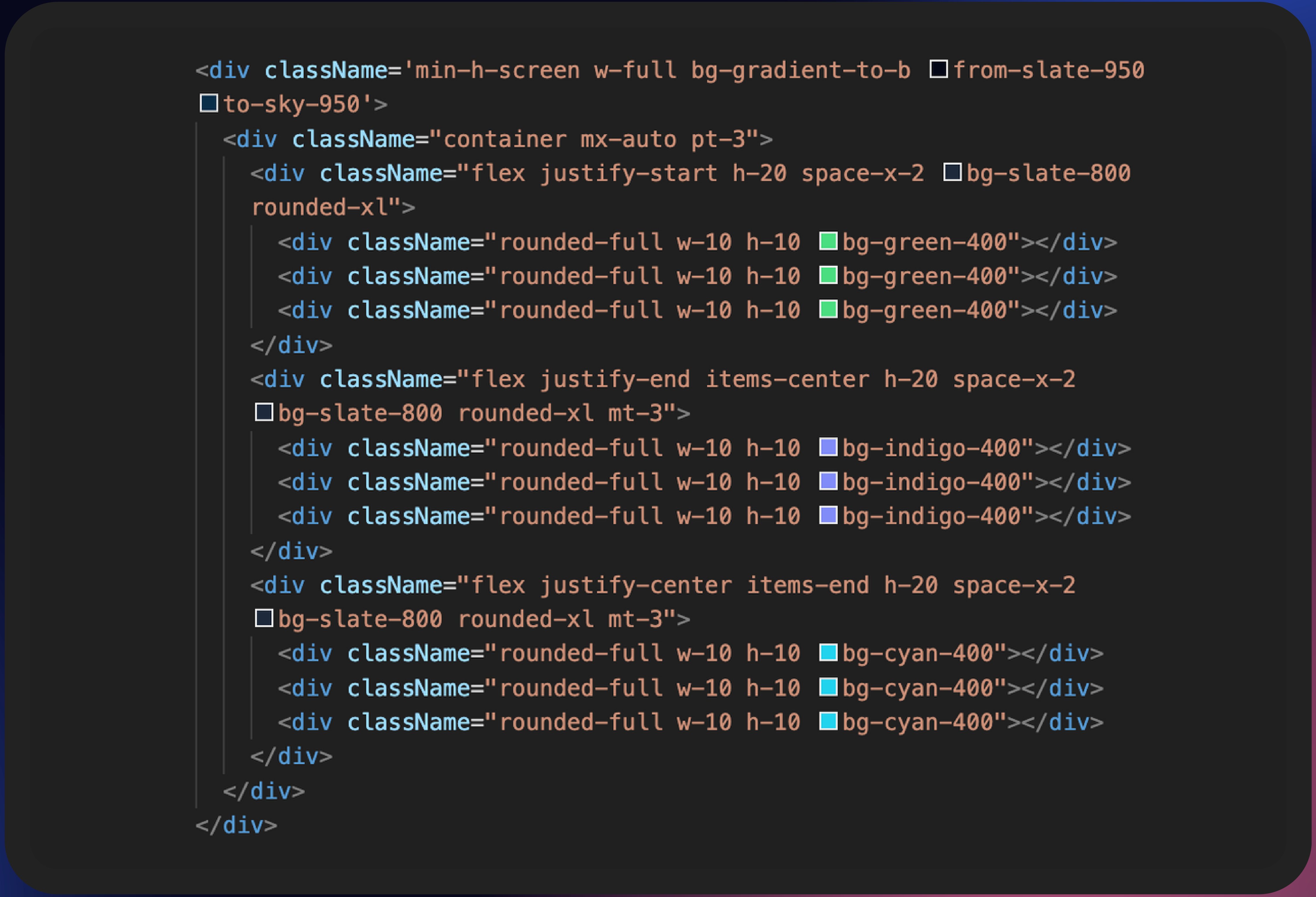
Grid

Great for arranging items in a linear fashion and handling flexible and responsive layouts

Ideal for creating more structured, grid-based designs with precise control over rows and columns

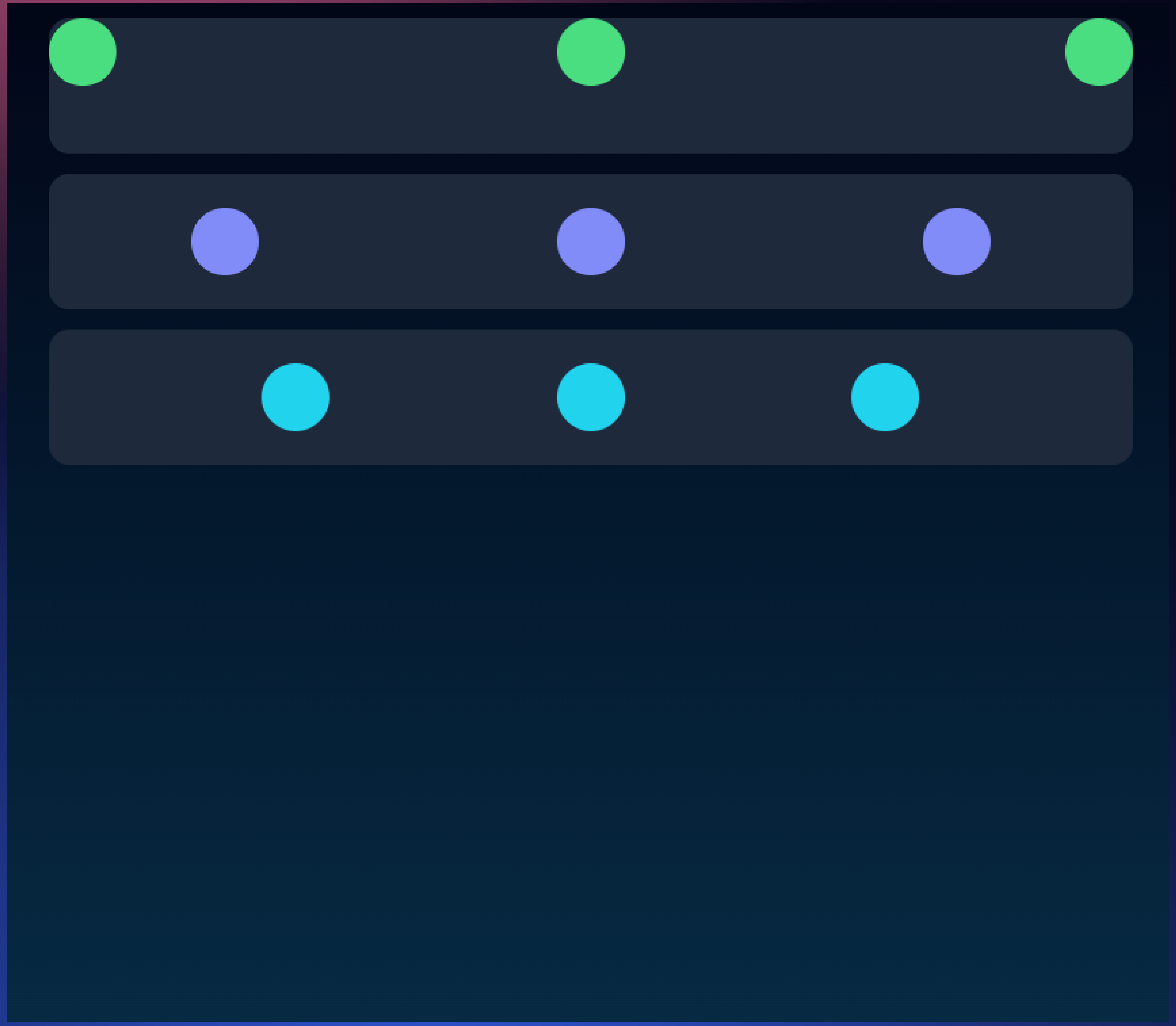
EXAMPLE 1.1 - FLEX

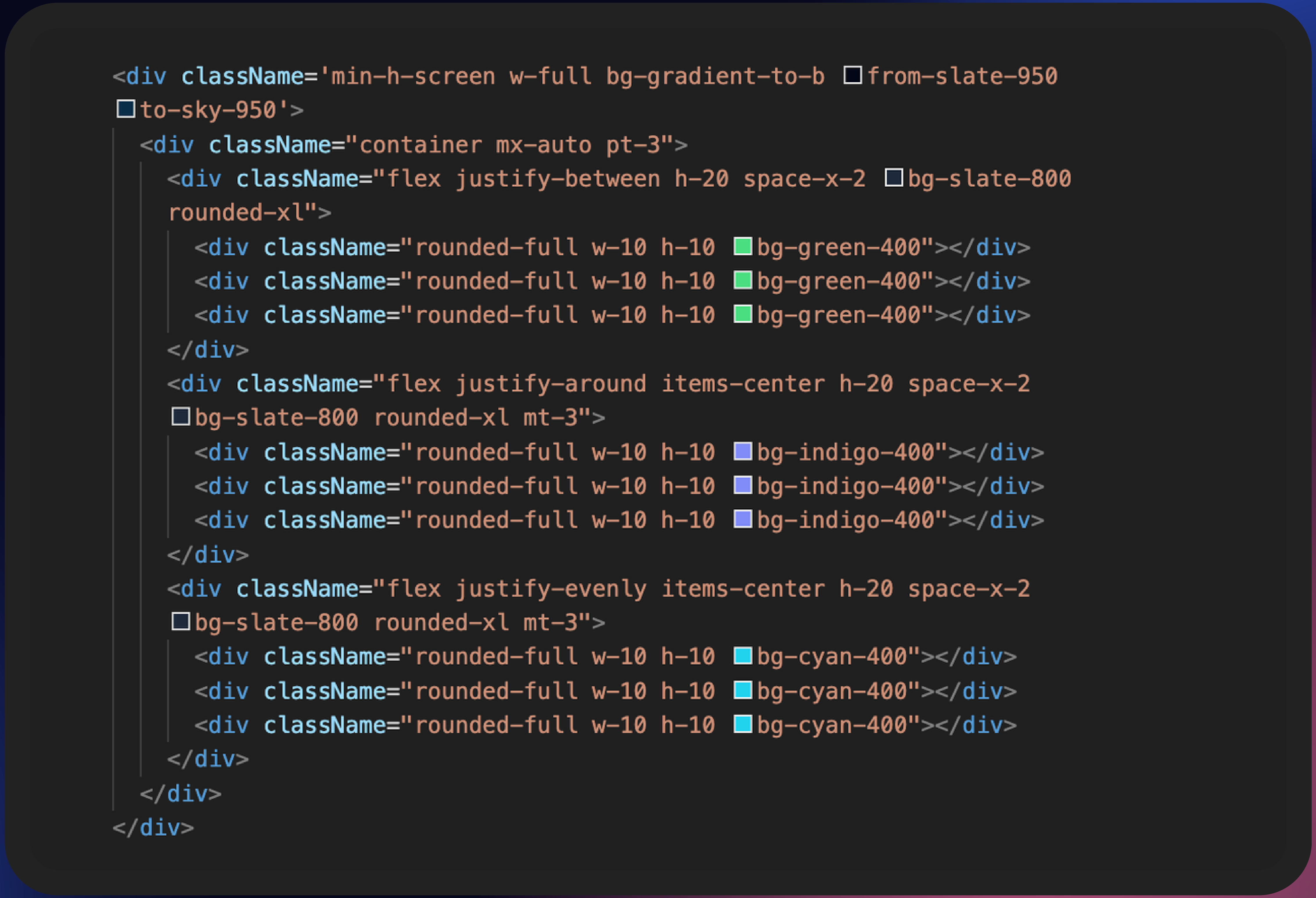




```
<div className='min-h-screen w-full bg-gradient-to-b □from-slate-950
□to-sky-950'>
  <div className="container mx-auto pt-3">
    <div className="flex justify-start h-20 space-x-2 □bg-slate-800
    rounded-xl">
      <div className="rounded-full w-10 h-10 🟩bg-green-400"></div>
      <div className="rounded-full w-10 h-10 🟩bg-green-400"></div>
      <div className="rounded-full w-10 h-10 🟩bg-green-400"></div>
    </div>
    <div className="flex justify-end items-center h-20 space-x-2
    □bg-slate-800 rounded-xl mt-3">
      <div className="rounded-full w-10 h-10 🟪bg-indigo-400"></div>
      <div className="rounded-full w-10 h-10 🟪bg-indigo-400"></div>
      <div className="rounded-full w-10 h-10 🟪bg-indigo-400"></div>
    </div>
    <div className="flex justify-center items-end h-20 space-x-2
    □bg-slate-800 rounded-xl mt-3">
      <div className="rounded-full w-10 h-10 🟦bg-cyan-400"></div>
      <div className="rounded-full w-10 h-10 🟦bg-cyan-400"></div>
      <div className="rounded-full w-10 h-10 🟦bg-cyan-400"></div>
    </div>
  </div>
</div>
```

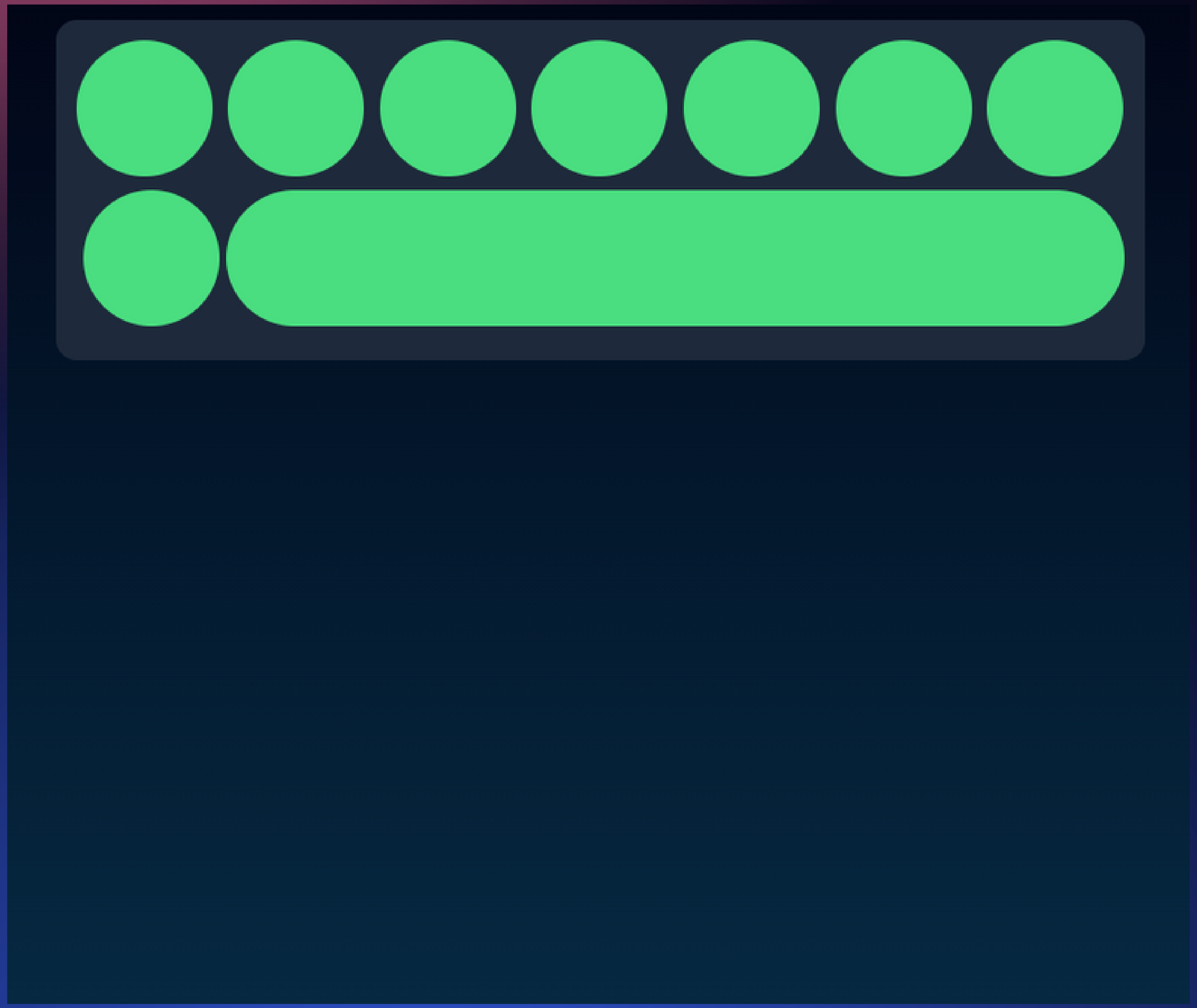
EXAMPLE 1.2 - FLEX

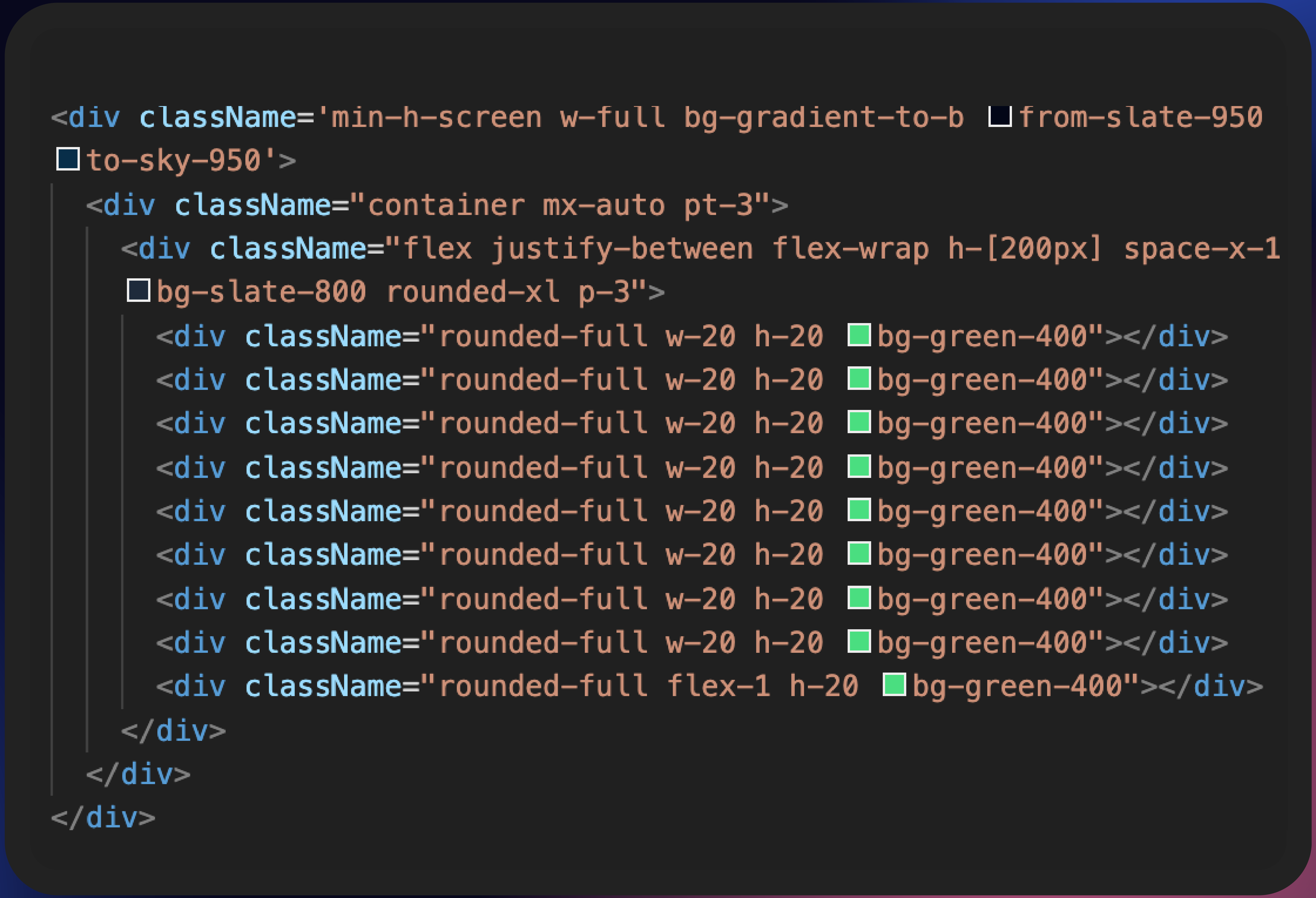




```
<div className='min-h-screen w-full bg-gradient-to-b □from-slate-950
□to-sky-950'>
  <div className="container mx-auto pt-3">
    <div className="flex justify-between h-20 space-x-2 □bg-slate-800
    rounded-xl">
      <div className="rounded-full w-10 h-10 🟩bg-green-400"></div>
      <div className="rounded-full w-10 h-10 🟩bg-green-400"></div>
      <div className="rounded-full w-10 h-10 🟩bg-green-400"></div>
    </div>
    <div className="flex justify-around items-center h-20 space-x-2
    □bg-slate-800 rounded-xl mt-3">
      <div className="rounded-full w-10 h-10 🟪bg-indigo-400"></div>
      <div className="rounded-full w-10 h-10 🟪bg-indigo-400"></div>
      <div className="rounded-full w-10 h-10 🟪bg-indigo-400"></div>
    </div>
    <div className="flex justify-evenly items-center h-20 space-x-2
    □bg-slate-800 rounded-xl mt-3">
      <div className="rounded-full w-10 h-10 🟦bg-cyan-400"></div>
      <div className="rounded-full w-10 h-10 🟦bg-cyan-400"></div>
      <div className="rounded-full w-10 h-10 🟦bg-cyan-400"></div>
    </div>
  </div>
</div>
```

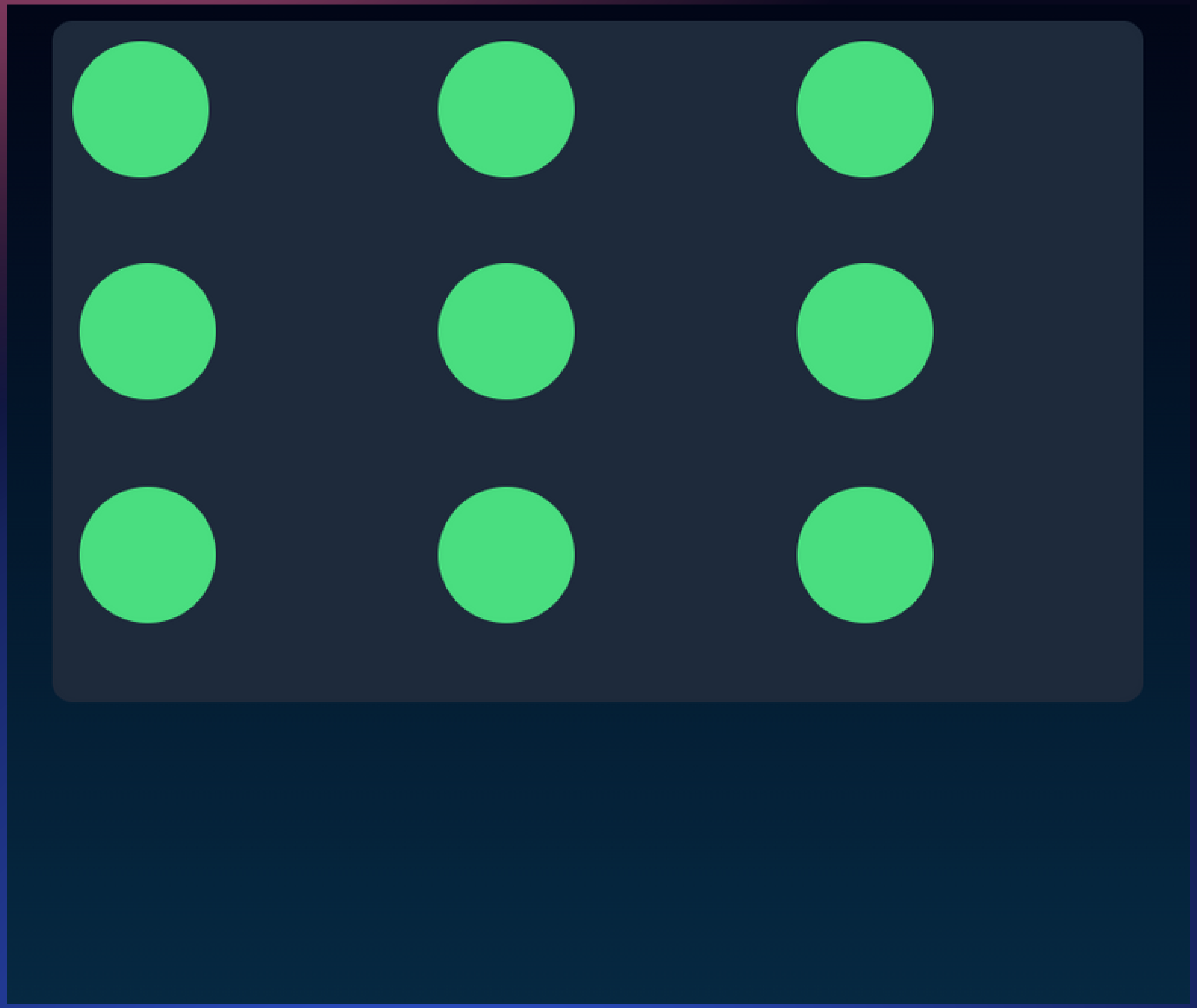
EXAMPLE 1.3 - FLEX

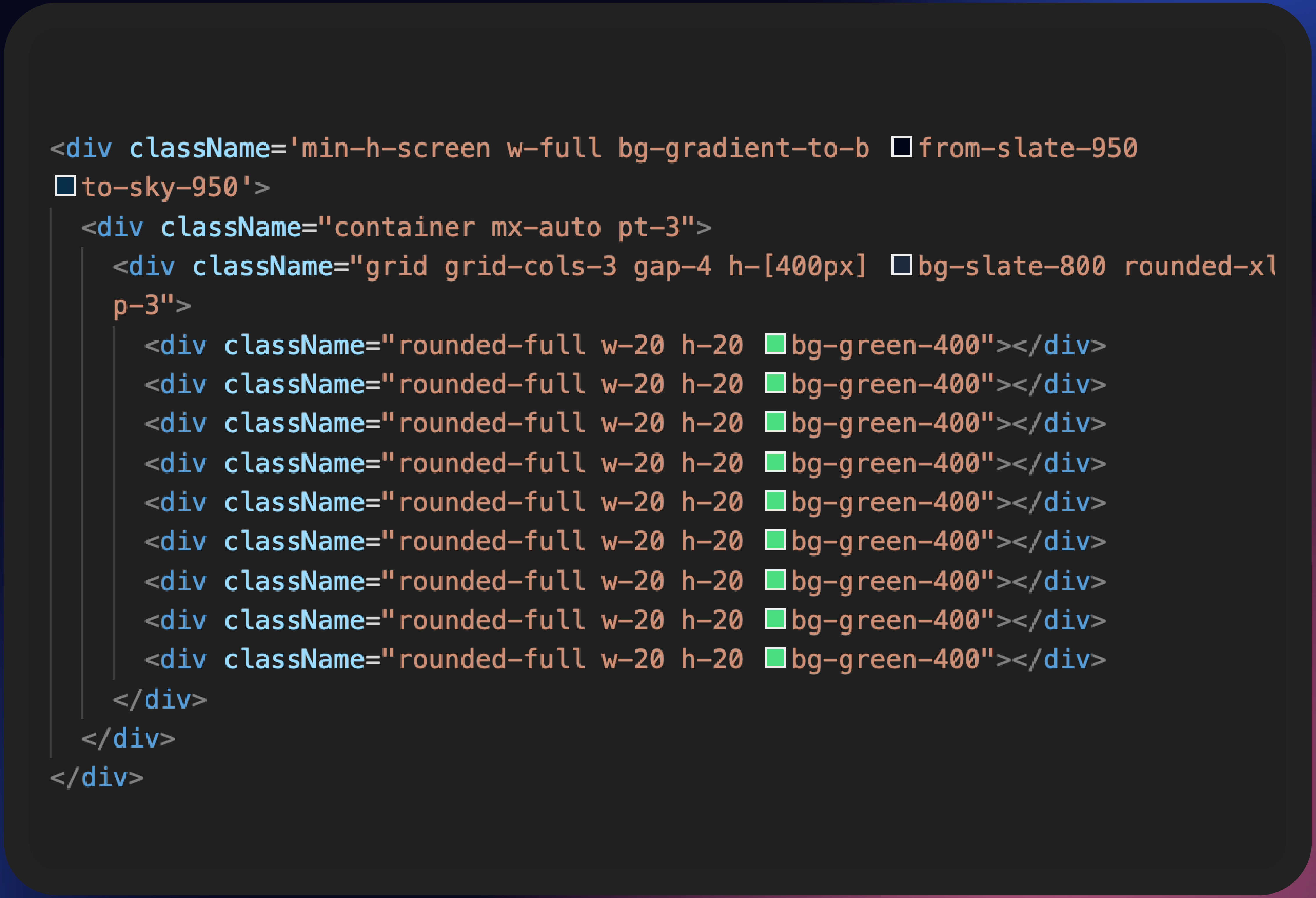




```
<div className='min-h-screen w-full bg-gradient-to-b □from-slate-950
□to-sky-950'>
  <div className="container mx-auto pt-3">
    <div className="flex justify-between flex-wrap h-[200px] space-x-1
    □bg-slate-800 rounded-xl p-3">
      <div className="rounded-full w-20 h-20 🟩bg-green-400"></div>
      <div className="rounded-full w-20 h-20 🟩bg-green-400"></div>
      <div className="rounded-full w-20 h-20 🟩bg-green-400"></div>
      <div className="rounded-full w-20 h-20 🟩bg-green-400"></div>
      <div className="rounded-full w-20 h-20 🟩bg-green-400"></div>
      <div className="rounded-full w-20 h-20 🟩bg-green-400"></div>
      <div className="rounded-full w-20 h-20 🟩bg-green-400"></div>
      <div className="rounded-full w-20 h-20 🟩bg-green-400"></div>
      <div className="rounded-full flex-1 h-20 🟩bg-green-400"></div>
    </div>
  </div>
</div>
```

EXAMPLE 2 - GRID





```
<div className='min-h-screen w-full bg-gradient-to-b □from-slate-950
□to-sky-950'>
  <div className="container mx-auto pt-3">
    <div className="grid grid-cols-3 gap-4 h-[400px] □bg-slate-800 rounded-xl
    p-3">
      <div className="rounded-full w-20 h-20 🟩bg-green-400"></div>
      <div className="rounded-full w-20 h-20 🟩bg-green-400"></div>
      <div className="rounded-full w-20 h-20 🟩bg-green-400"></div>
      <div className="rounded-full w-20 h-20 🟩bg-green-400"></div>
      <div className="rounded-full w-20 h-20 🟩bg-green-400"></div>
      <div className="rounded-full w-20 h-20 🟩bg-green-400"></div>
      <div className="rounded-full w-20 h-20 🟩bg-green-400"></div>
      <div className="rounded-full w-20 h-20 🟩bg-green-400"></div>
      <div className="rounded-full w-20 h-20 🟩bg-green-400"></div>
    </div>
  </div>
</div>
```

SPACING & SIZING

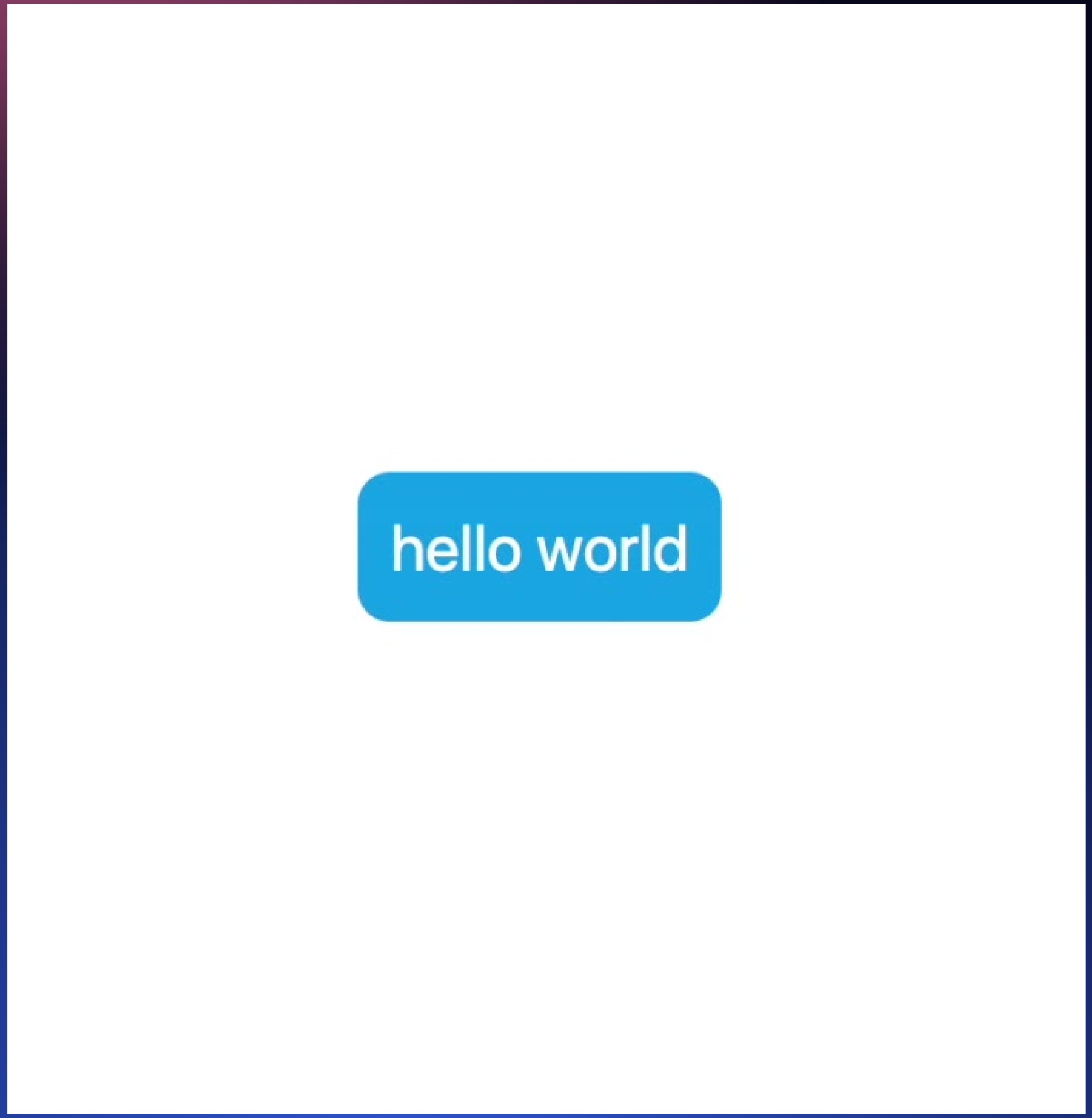
spacing

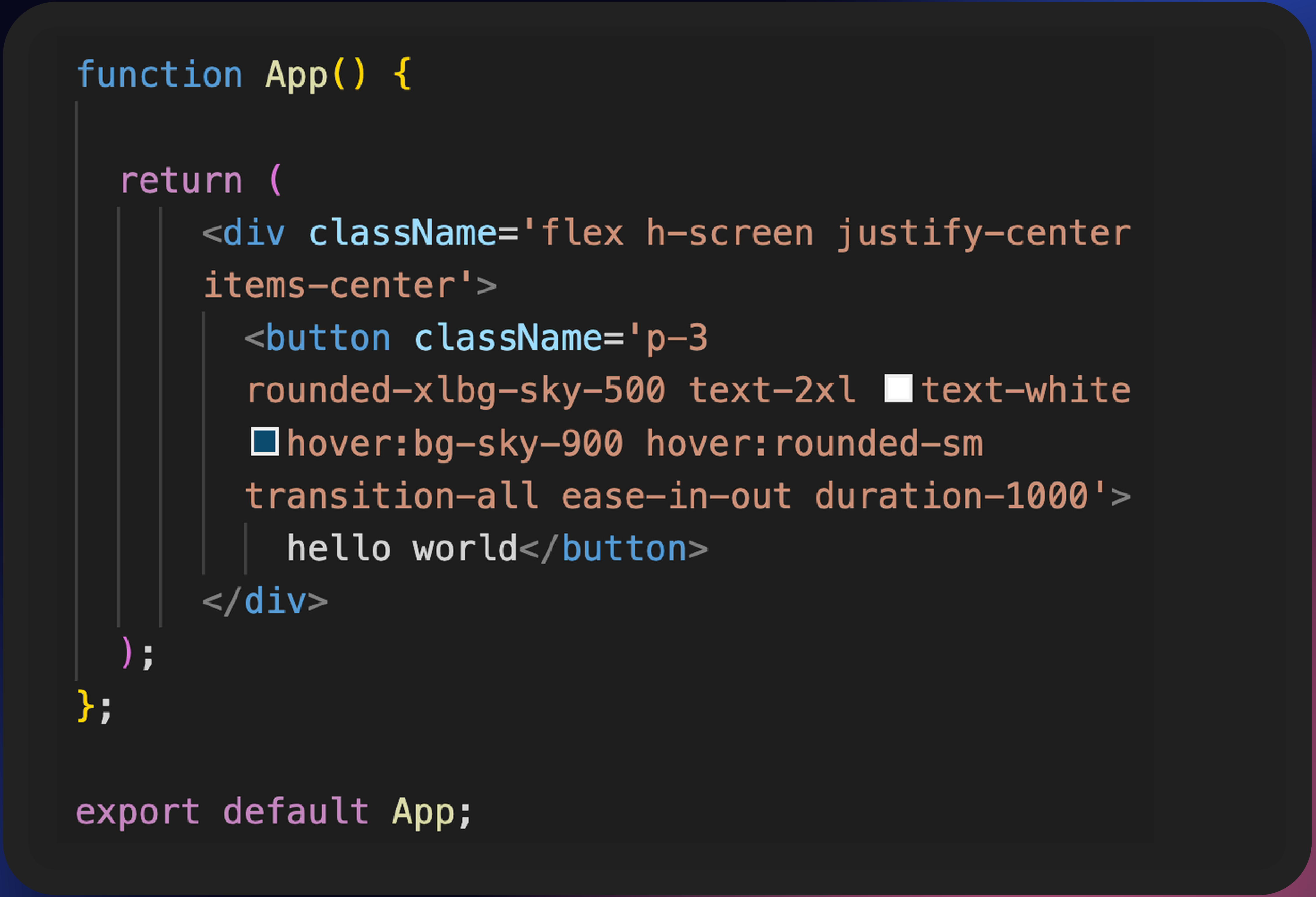
- Classes like m-{size}, p-{size}, mx-{size}, my-{size} are used to specify margin and padding sizes.
- The {size} can be defined using a scale ranging from 0 to 96, representing multiples of 0.25rem.

sizing

- Classes like w-{size}, h-{size}, min-w-{size}, max-w-{size} can be used to specify element dimensions.
- The {size} can be defined using fixed sizes, percentages, or using the responsive sizing approach.

EFFECTS



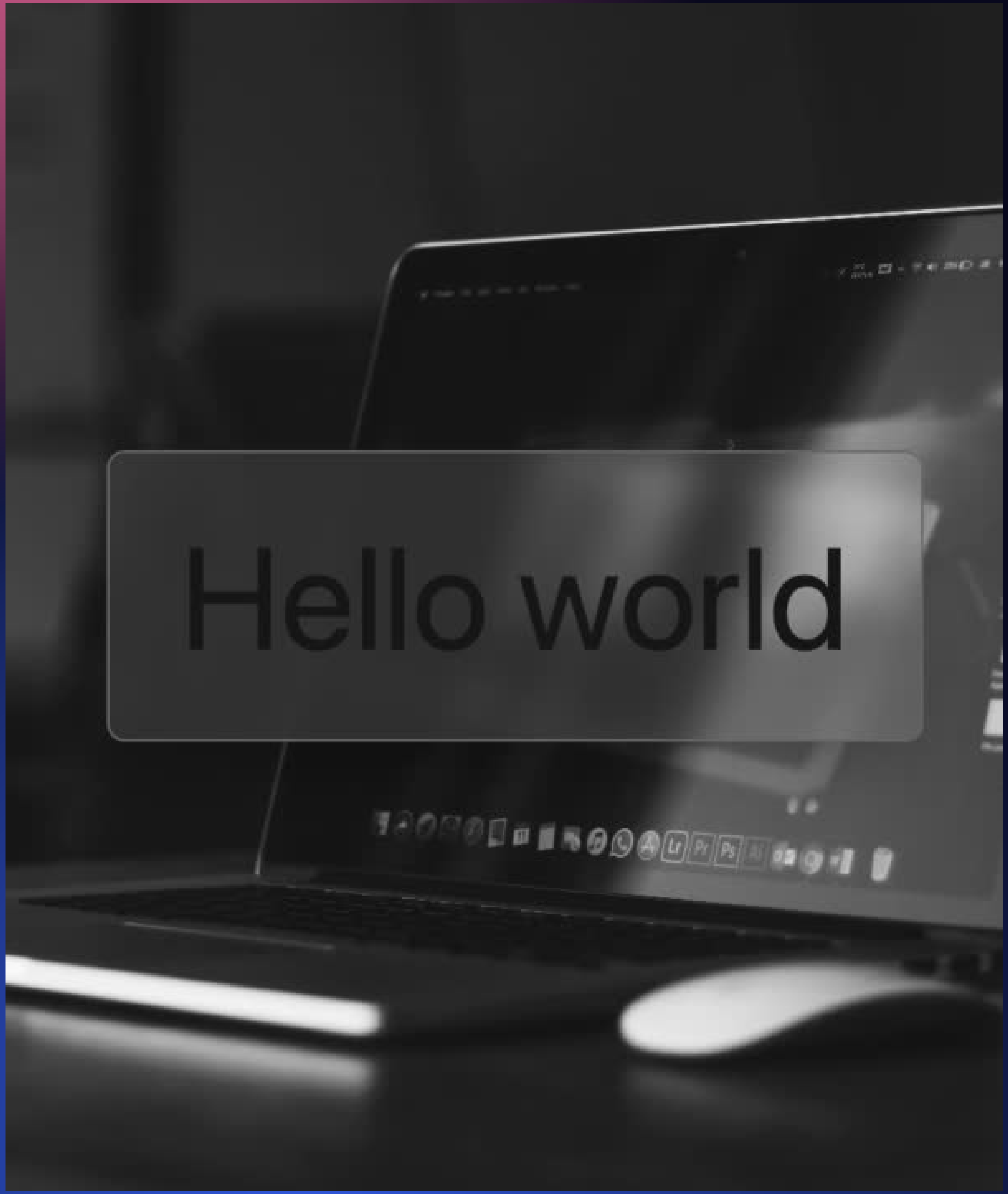


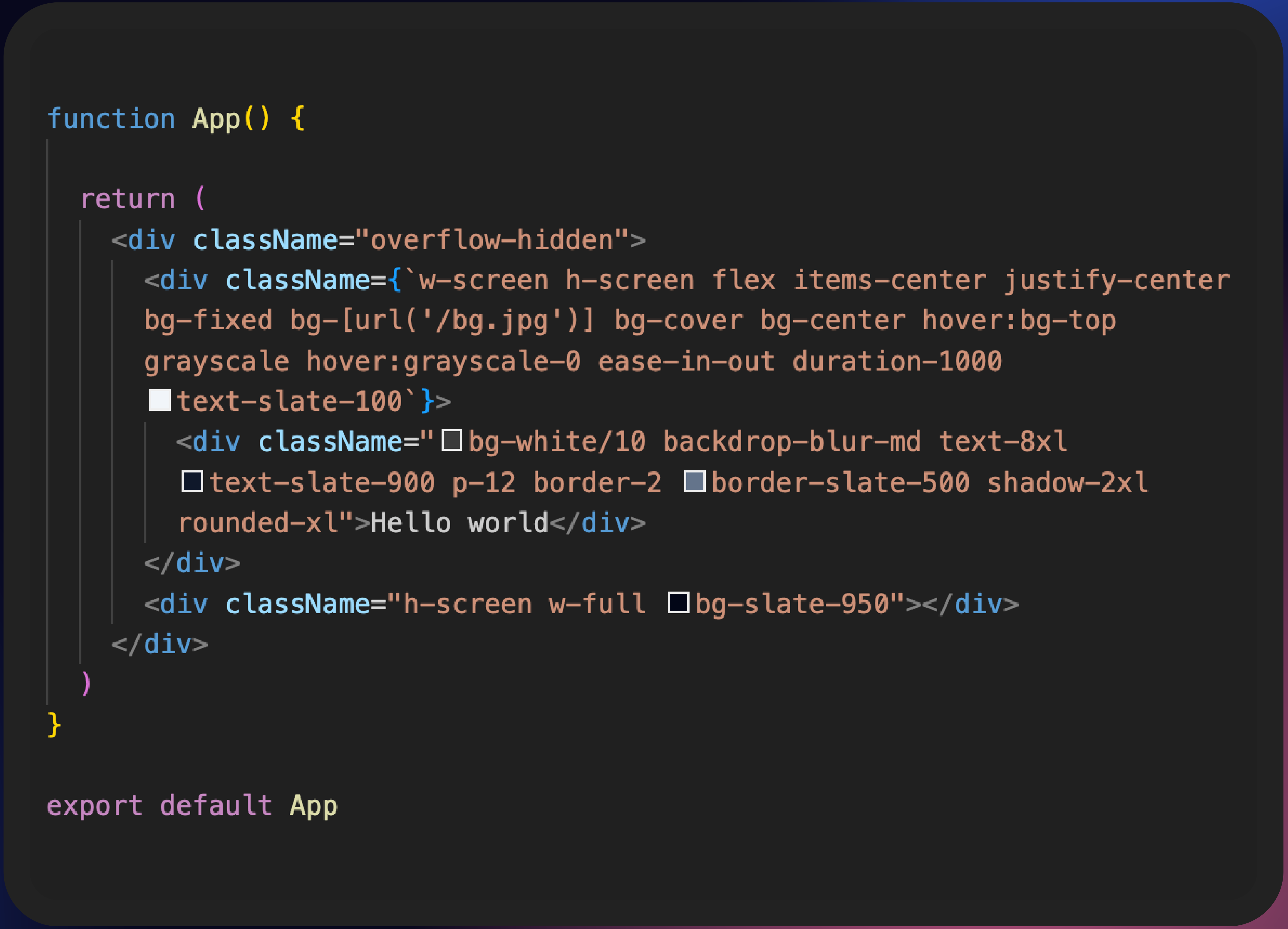
```
function App() {

  return (
    <div className='flex h-screen justify-center
    items-center'>
      <button className='p-3
      rounded-xlbg-sky-500 text-2xl □text-white
      ■hover:bg-sky-900 hover:rounded-sm
      transition-all ease-in-out duration-1000'>
        hello world</button>
    </div>
  );
};

export default App;
```

EFFECTS WITH IMAGES - 1



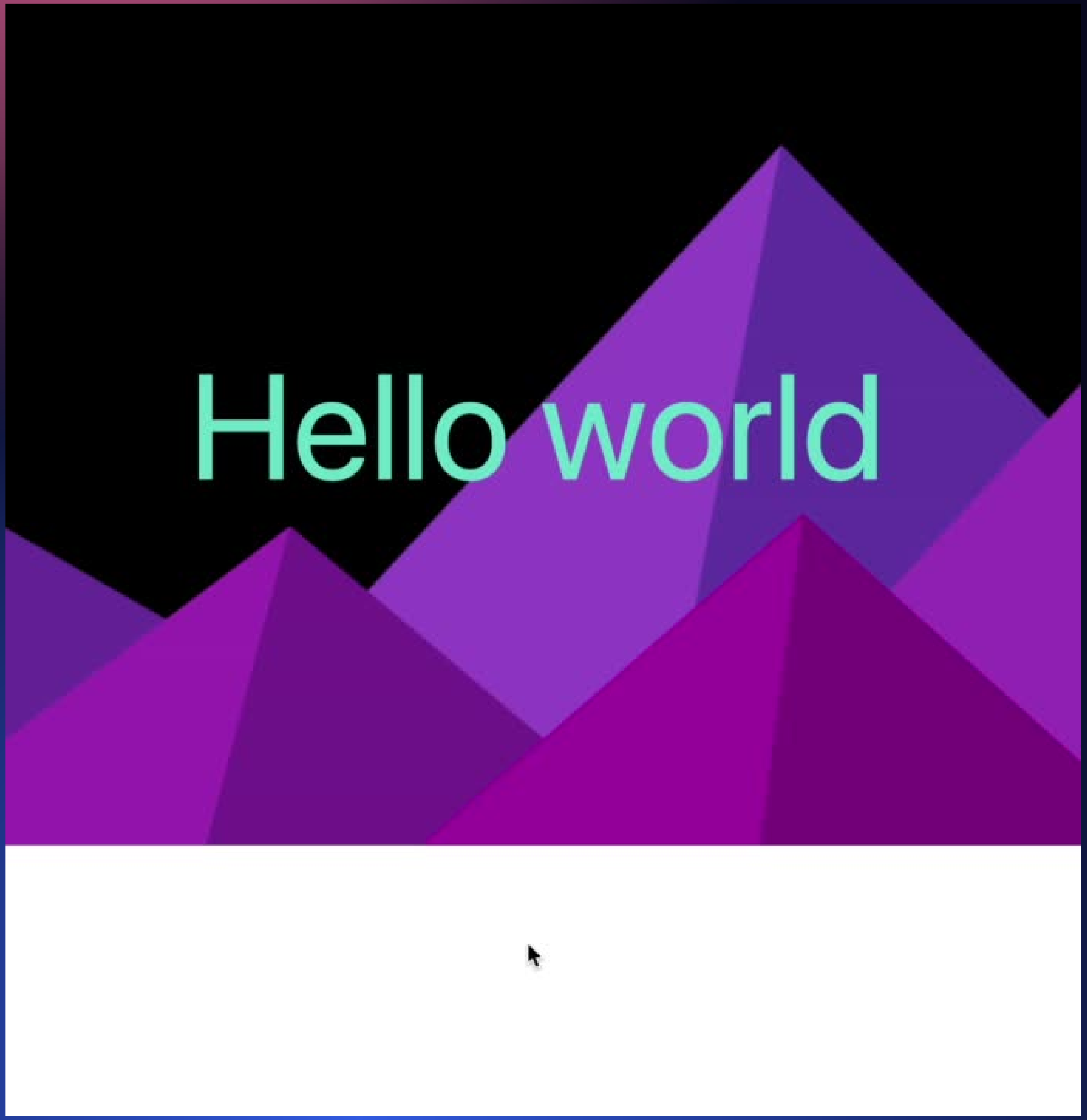


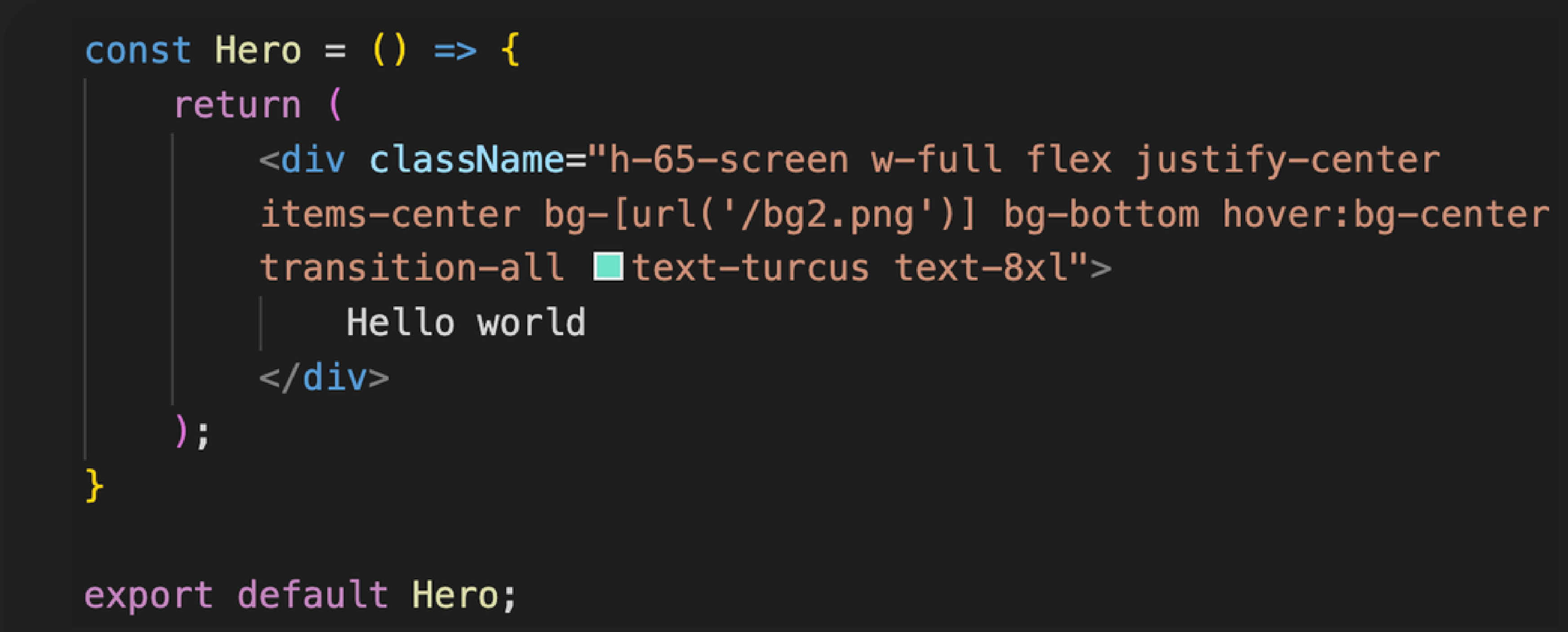
```
function App() {

  return (
    <div className="overflow-hidden">
      <div className={`w-screen h-screen flex items-center justify-center
      bg-fixed bg-[url('/bg.jpg')] bg-cover bg-center hover:bg-top
      grayscale hover:grayscale-0 ease-in-out duration-1000
      text-slate-100`}>
        <div className="bg-white/10 backdrop-blur-md text-8xl
        text-slate-900 p-12 border-2 border-slate-500 shadow-2xl
        rounded-xl">Hello world</div>
      </div>
      <div className="h-screen w-full bg-slate-950"></div>
    </div>
  )
}

export default App
```

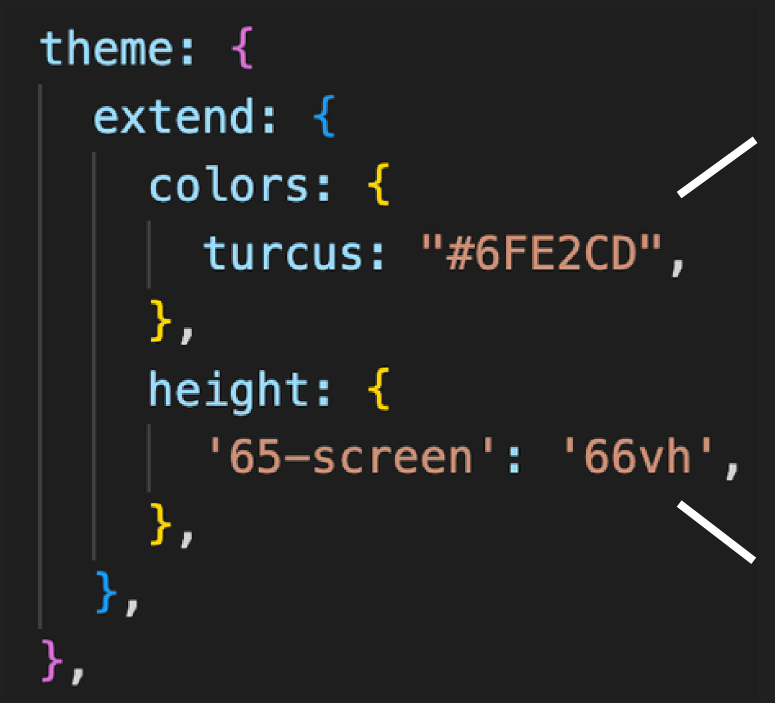
EFFECTS WITH IMAGES - 2





```jsx
const Hero = () => {
    return (
        <div className="h-65-screen w-full flex justify-center
        items-center bg-[url('/bg2.png')] bg-bottom hover:bg-center
        transition-all text-turcus text-8xl">
            Hello world
        </div>
    );
}

export default Hero;
```



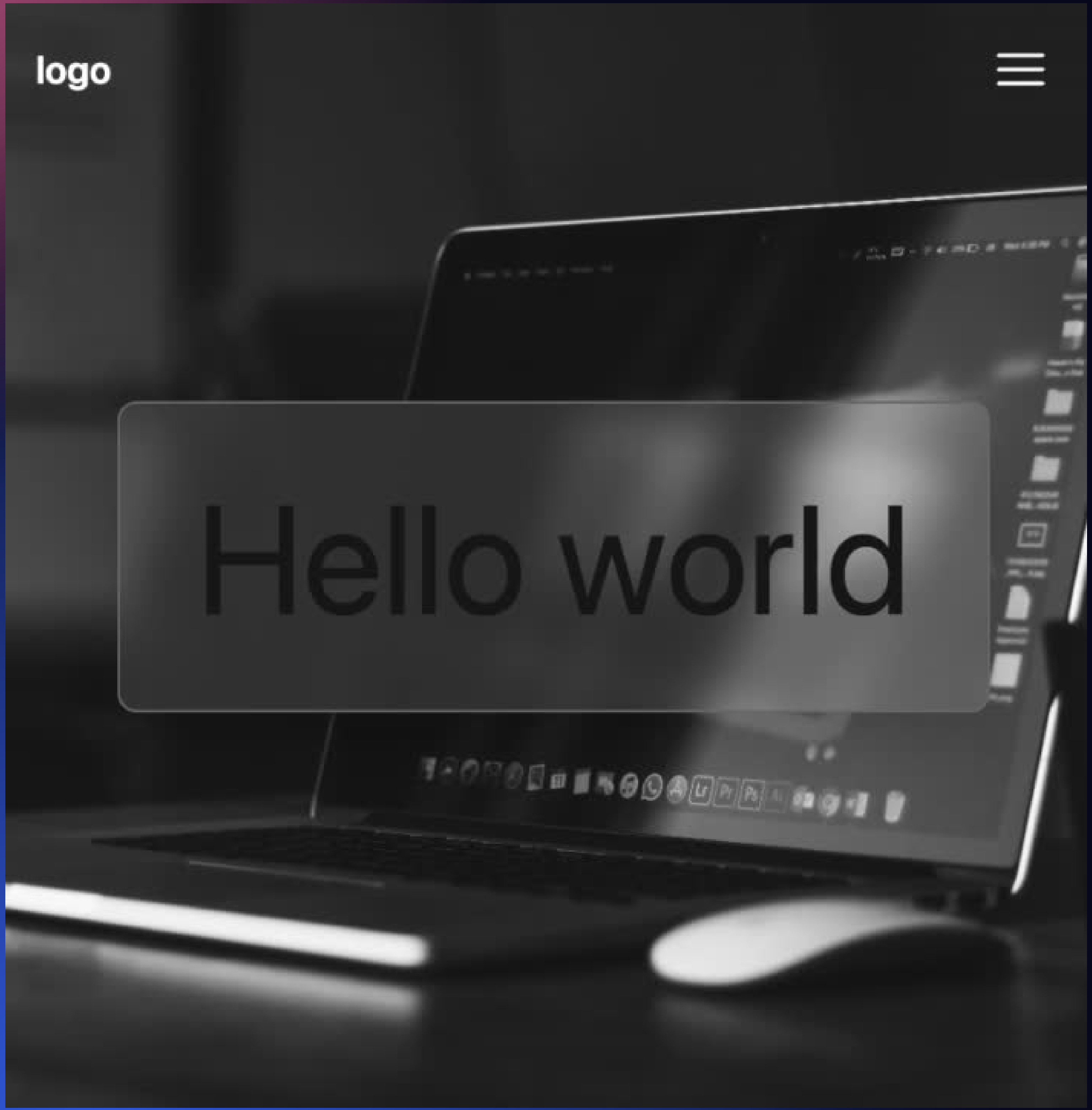
```js
theme: {
  extend: {
    colors: {
      turcus: "#6FE2CD",
    },
    height: {
      '65-screen': '66vh',
    },
  },
},
```

custom color

tailwind.config.js

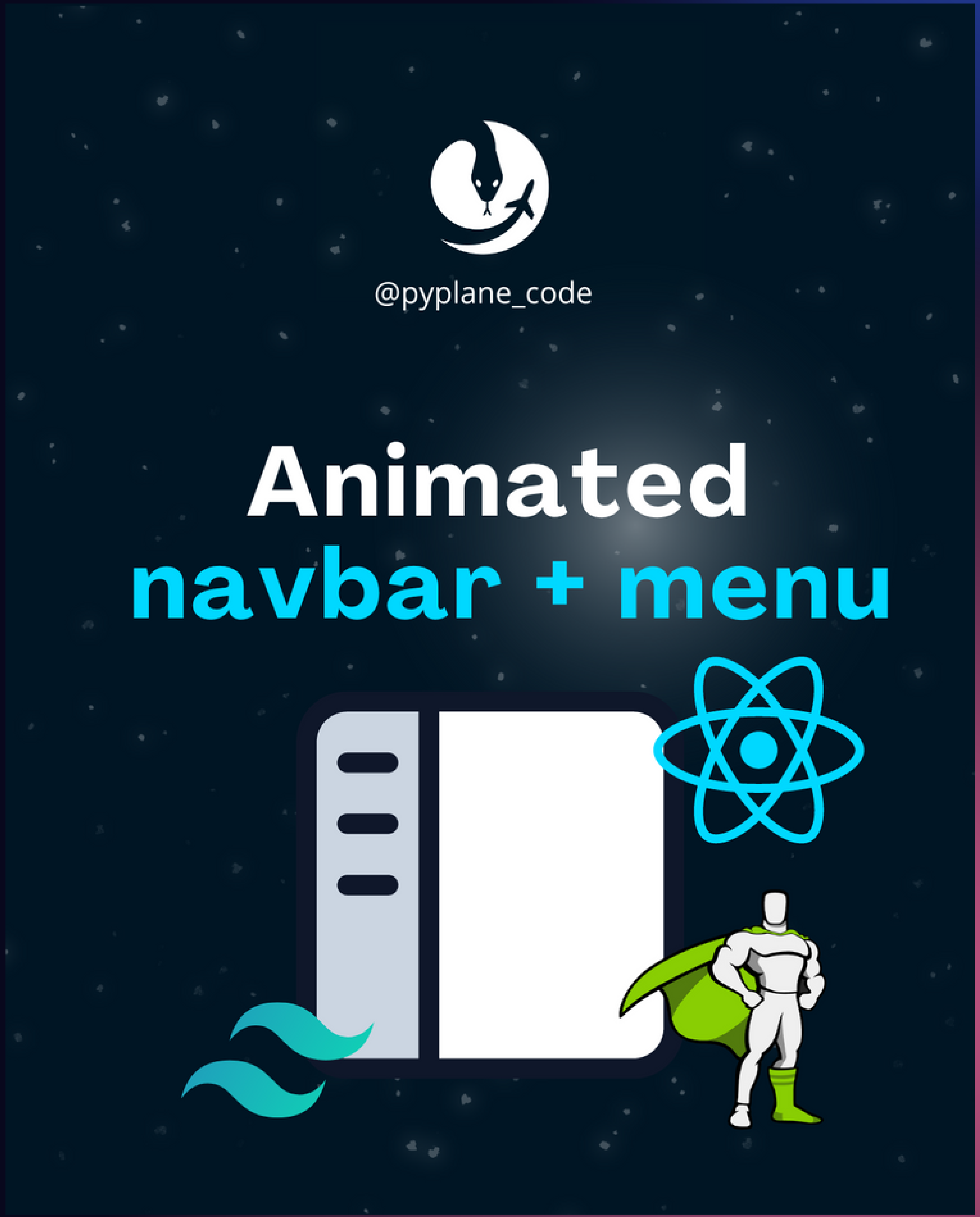
custom height

EFFECTS WITH ANIMATIONS



logo

Hello world



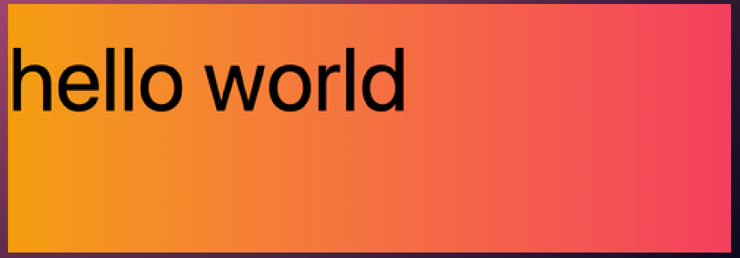
RESPONSIVNESS

Tailwind CSS follows a mobile-first approach, where styles for smaller screens are defined first and progressively enhanced for larger screens

- Tailwind CSS provides a set of pre-defined breakpoints for different screen sizes.
- You can use responsive classes like sm, md, lg, xl, 2xl to apply styles at specific breakpoints.
- For example, sm:text-lg applies a larger font size starting from the small screen breakpoint

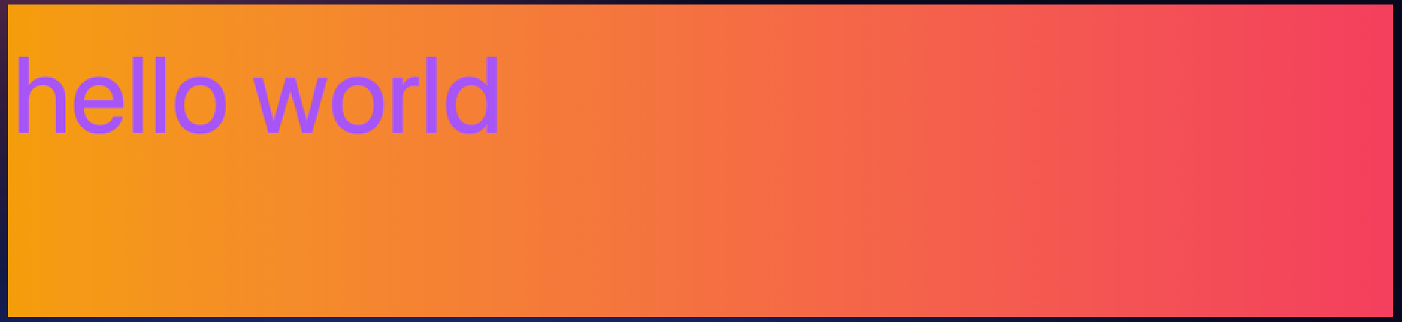
sm	**md**	**lg**	**xl**	**2xl**	
640	768	1024	1280	1536	**min-width**

RESPONSIVNESS - EXAMPLE



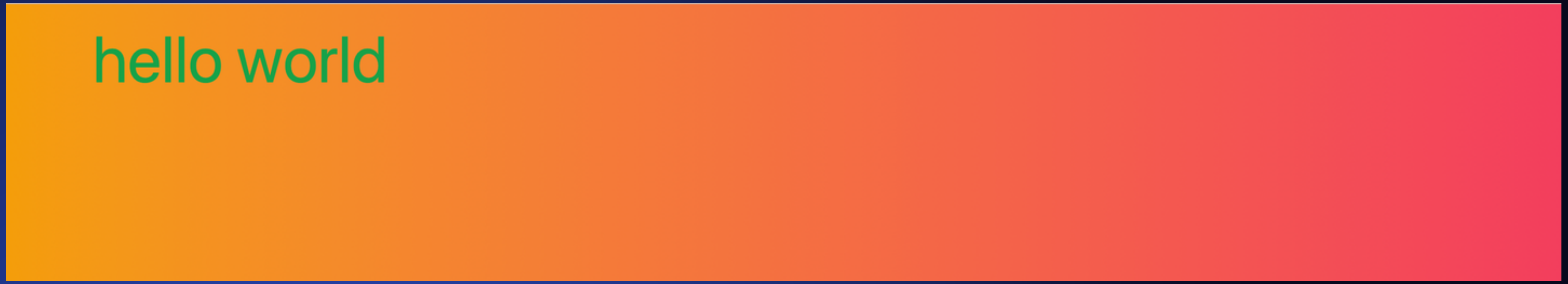
hello world

black up to sm (640px)



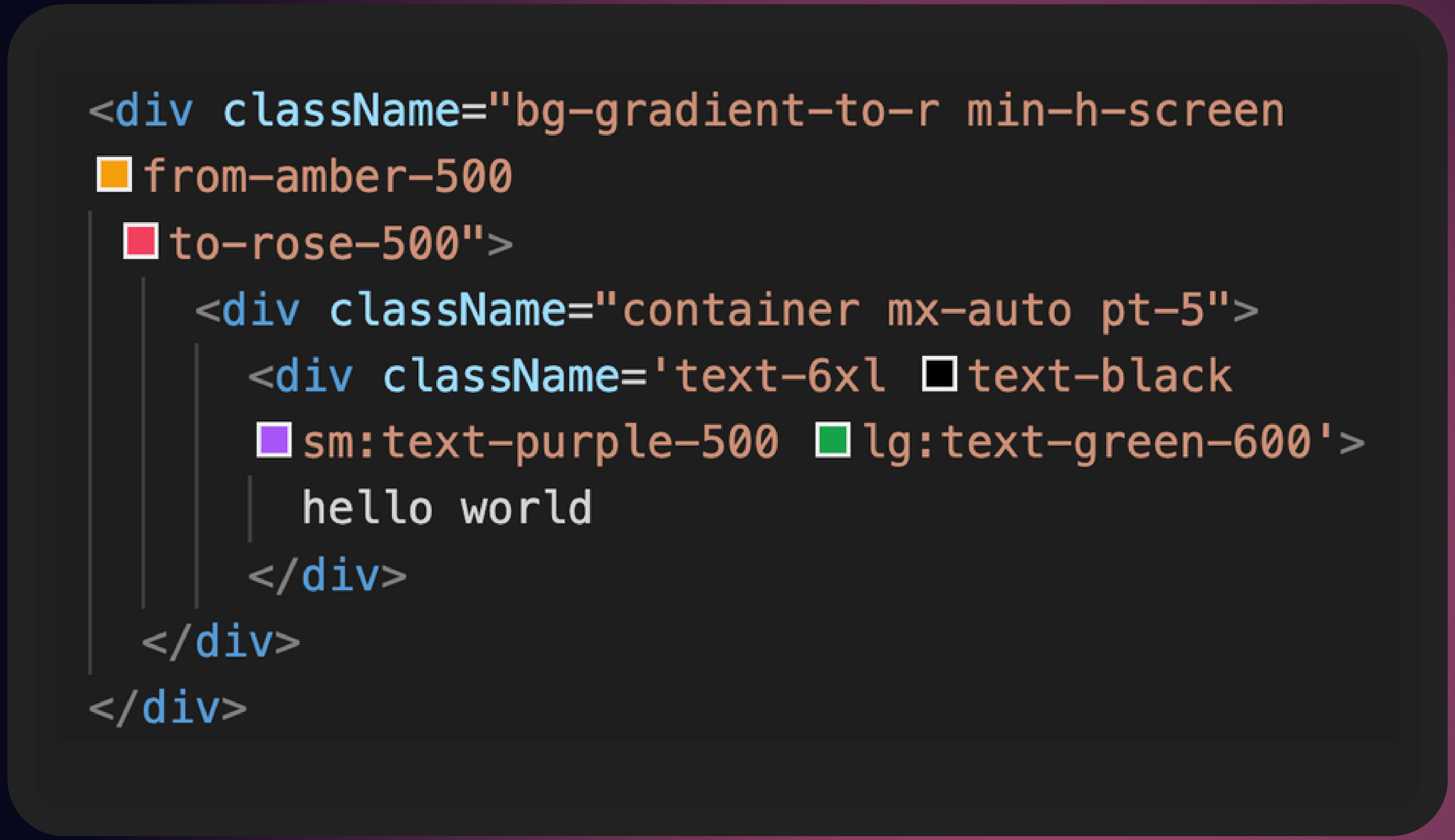
hello world

purple from sm to lg (1024px)



hello world

from lg green



```
<div className="bg-gradient-to-r min-h-screen
🟧from-amber-500
 🟥to-rose-500">
  <div className="container mx-auto pt-5">
   <div className='text-6xl ■text-black
   🟪sm:text-purple-500 🟩lg:text-green-600'>
    hello world
   </div>
  </div>
</div>
```

THANK YOU