字节T6外包笔试题 副本

T6

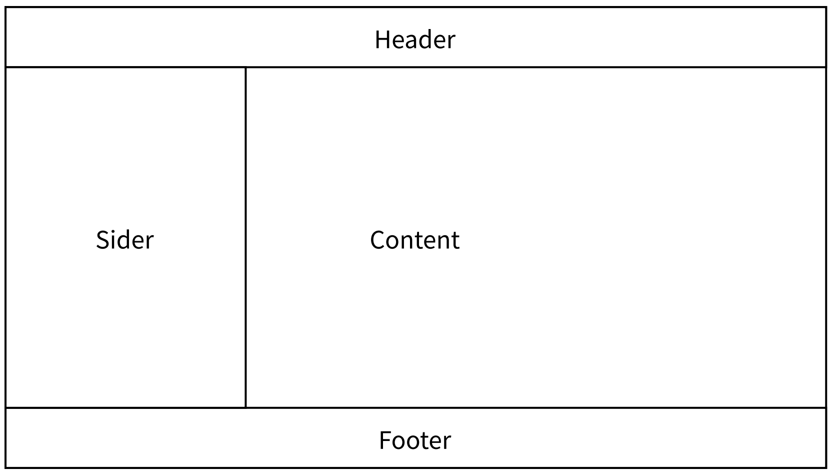
编程题目

- 设计一个函数能将字符串转换成小驼峰格式
- 假设字符串的分隔符只有" ", "-", "_"三种

```
camelCase('Foo Bar');
// => 'fooBar'

camelCase('foo-bar');
// => 'fooBar'

camelCase('FOO_BAR');
// => 'fooBar'

 function camelCase(str) {
     if (!str && !str.trim()) return "请输入正确的字符串"
     let operator = " "
     if (/\-+/g.test(str)) operator = "-"
     if (/\_+/g.test(str)) operator = "_"
     const [a, ...b] = str.split(operator)
     const firstStr= a.toLocaleLowerCase()
     const otherStr= b.map(v =>`${v.slice(0, 1).toLocaleUpperCase()}${v.slice(1).toLocaleLowerCase()}`)
     return [firstStr, ...otherStr].reduce((pre, cur) => `${pre}${cur}`, "")
}
```

组件设计

- 实现一套Layout容器组件, 包括Header,Footer,Sider,Content四种子组件



- 要求:
    - Header与Footer分别在Layout容器顶部和底部, 宽度占满全屏, 高度均为50px
    - Sider是否展示通过props可以配置, 如果展示的话宽度为50px, 高度占满剩余空间

- Content占满剩下所有空间
- 通过React或者Vue实现: 用户可以在Header/Content/Footer中自由填入内容

```
// 实现组件功能及样式
const Header = () => {}
const Footer = () => {}
const Content = () => {}
const Layout = () => {}

// 使用场景
const App = () => (
    <Layout hasSider={true}>
        <Header>我是头</Header>
        <Content>我是身子</Content>
        <Footer>我是尾</Footer>
    </Layout>
)
```

```
/**--------Layout--------------------------------------------------------------
*/
import {
    useMemo,
    useCallback,
    MapHTMLAttributes,
    ReactElement,
    CSSProperties
 } from 'react'

export interface LayoutProps extends MapHTMLAttributes<ReactElement> {
    hassider?: boolean
}
type LayoutChildrenName = 'Header' | 'Content' | 'Footer'
export const Layout = (props: LayoutProps): JSX.Element => {
    const { hassider } = props
    const layOutStyle: CSSProperties = {
        height: '100vh',
        display: 'flex',
        flexDirection: 'column'
    }
    const siderStyle: CSSProperties = {
        flexBasis: '50px',
        flexGrow: 0,
        backgroundColor: 'green'
    }

    const renderDom = useCallback(
        (type: LayoutChildrenName) => {
            if (!props.children) return 'Empty'
            if (Array.isArray(props.children)) return props.children.find(v =>
v.type.name === type) || null
            if (typeof props.children === 'object' && (props.children as any).t
ype.name === type) return props.children
            return null
        },
        [props.children]
    )
    const renderSider = useMemo(() => hassider ? (
                <div className="sider" style={siderStyle}> sider</div>
          ) : null,[hassider, siderStyle])
    return (
        <div style={layOutStyle}>
            {renderDom('Header')}
            <div style={{ display: 'flex', flexGrow: 1 }}>
                {renderSider}
                {renderDom('Content')}
            </div>
            {renderDom('Footer')}
        </div>
    )
}

/**--------Header-----------------------------------------------------------
*/
import { MapHTMLAttributes, ReactElement,CSSProperties  } from 'react'

export interface HeaderPorps extends MapHTMLAttributes<ReactElement> {}
export const Header = (props: HeaderPorps): JSX.Element => {
    const style: CSSProperties = {
            display: 'flex',
            height: '50px',
            backgroundColor: 'skyblue'
    }
    return <div style={style}>{props.children}</div>
}

/**--------Content----------------------------------------------------------
*/
import { MapHTMLAttributes, ReactElement,CSSProperties  } from 'react'

export interface ContentPorps extends MapHTMLAttributes<ReactElement> {}
export const Content = (props: ContentPorps): JSX.Element => {
    const style: CSSProperties = {
        display: 'flex',
        flexGrow: 1,
        backgroundColor: '#ccc',
    }
    return <div style={style}>{props.children}</div>
}

/**--------Footer----------------------------------------------------------
*/
import { MapHTMLAttributes, ReactElement, CSSProperties } from 'react'

export interface FooterPorps extends MapHTMLAttributes<ReactElement> {}
export const Footer = (props: FooterPorps): JSX.Element => {
    const style: CSSProperties = {
        display: 'flex',
        height: '50px',
        backgroundColor: 'skyblue',
    }
    return <div style={style}>{props.children}</div>
}

```