<!DOCTYPE html>

<head>

<style>

    \* {

margin: 0;

padding: 0;

    }

    div {

        width: 200px;

        height: 200px;

        border: 1px solid red;

        float: left;

        margin: 10px;

    }

    div:nth-child(2){

        border: 1px solid green;

    }

    p{

        height: 25px;

        background-color: palegreen;

        line-height: 25px;

        text-align: center;

    }

</style>

</head>

<body>

    <div id="div1">

        <p id="p1" draggable="true">拖曳内容1</p>

        <p id="p4" draggable="true">拖曳内容2</p>

        <p id="p3" draggable="true">拖曳内容3</p>

        <p id="p2" draggable="true">拖曳内容4</p>

    </div>

    <div id="div02">

    </div>

    <script>

        //拖曳开始

        document.ondragstart=function(e){

            console.log("源对象开启拖曳功能");

            //输出扑捉拖曳后事件当中的全部内容

            console.log(e);

            //找到拖曳的控件id

            console.log(e.target.id);

            //通过获取id，将此控件添加到相应的位置上

            e.dataTransfer.setData('text/html',e.target.id);

        }

        //当前拖曳到指定的位置时释放鼠标

        document.ondrop=function(e){

            console.log("源对象移动到指定位置的上方");

            console.log(e)

            //获取拖曳时鼠标的对象数据

            var id=e.dataTransfer.getData('text/html');

            //根据当前鼠标所在位置，将拖曳的数据注入到当前的标签内容

            e.target.appendChild(document.getElementById(id));

        }

        //当鼠标移动到目标控件上方

        document.ondragover=function(e){

            console.log("将拖曳的内容移动到了指定的位置");

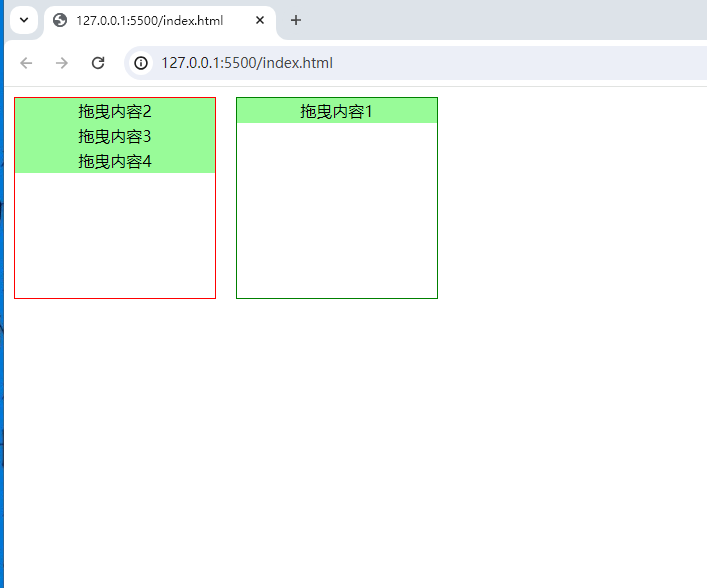
            return false;

        }

    </script>

</body>

<html>



<!DOCTYPE html>

<html>

    <head>

        <style>

            \*{

                margin: 0;

                padding: 0;

            }

            div {

                width: 200px;

                height: 200px;

                border: 1px solid red;

                float: left;

                margin: 10px;

            }

            div:nth-child(2){

                border: 1px solid green;

            }

            div:nth-child(3){

                border: 1px solid blue;

            }

            p{

                height: 25px;

                background: palegreen;

                line-height: 25px;

                text-align: center;

            }

        </style>

    </head>

    <body>

        <div id="div1">

            <p id="p1" draggable="true">拖拽内容1</p>

            <p id="p2" draggable="true">拖拽内容2</p>

            <p id="p3" draggable="true">拖拽内容3</p>

            <p id="p4" draggable="true">拖拽内容4</p>

        </div>

        <div id="div02">

        </div>

        <div id="div03"></div>

        <script>

            //拖拽开始

            document.ondragstart=function(e){

                console.log("源对象开启拖拽功能");

                //输出扑捉拖拽事件当中的全部内容

                console.log(e);

                //找到拖拽的控件id

                console.log(e.target.id)

                //通过获取id，将此控件添加到相应的位置上

                e.dataTransfer.setData('text/html',e.target.id);

            }

            //当拖拽到指定位置时释放鼠标

            document.ondrop=function(e){

                console.log("源对象移动到指定位置的上方");

                console.log(e)

                //获取拖拽时鼠标的对象数据

                var id=e.dataTransfer.getData('text/html');

                //根据当前鼠标所在的位置，将拖拽的数据注入到当前的标签内

                e.target.appendChild(document.getElementById(id));

            }

            //当鼠标移动到目标控件上方

            document.ondragover=function(e){

                console.log("将拖拽的内容移动到了指定位置")

                return false;

            }

            //拖拽结束时触发事件

            document.ondragend=function(e){

                console.log("拖拽结束后触发事件");

                console.log(e);

            }

            //从原标签进入目标标签后触发

            document.ondragenter=function(e){

                console.log("将拖拽的控件进入目标标签厚触发");

                console.log(e);

            }

            //源对象离开目标标签后触发

            document.ondragleave=function(e){

                console.log("离开标签后触发");

                console.log(e)

            }

        </script>

    </body>

</html>

