**更多套用CSS的方法**

以下為套用在所有h1標籤

    h1{

    color: brown;

    background-color: aqua;

}

.class-1{

    color: red;

    background-color: antiquewhite;

}

以上為套用在class

以下為套用在id

<h2 id="id-1">小黑你好阿</h2>

#id-1{

    color: blue;

    background-color: beige;

同時套用在id與class #id-1, .class-1

#id-1, .class-1{

    color: blue;

    background-color: aqua;

}

同時套用在id與h1

#id-1, h1

整個網頁的背景顏色

\*{

    background-color: red;

}

套用在ul列表內的li

ul li{

    background-color: gray;

}

套用在ul裡面的a標籤

ul a{

    background-color: gray;

}

套用在type屬性上,用一個中括號 [ ]

[type]{

    background-color: gray;

}

<input type="text">

        <br/>

        <input type="password">

以上這input背景顏色會變成灰色

套用在href上,用一個中括號 [ ]

[href]{

    background-color: gray;

}

當滑鼠滑到h1標籤時,背景呈灰色, 滑開就會不見

h1:hover {

    background-color: gray;

}

當滑鼠滑到p標籤時,背景呈灰色, 滑開就會不見

p:hover {

    background-color: gray;

}

當滑鼠滑到id-1標籤時,背景呈灰色, 滑開就會不見

#id-1:hover {

    background-color: gray;

}

當滑鼠滑到有type這屬性時,背景呈灰色, 滑開就會不見

[type]:hover {

    background-color: gray;

}

更多css標籤使用,google w3c css selector CSS選擇器

flex一種顯示的方式.預設是橫的.由左到右呈現

 <!-- flex -->

        <div class="container">

            <div class="box-1>">區塊1</div>

            <div class="box-2>">區塊2</div>

            <div class="box-3>">區塊3</div>

寫出三個區塊

.container div{

    border:1px solid rgb(202, 201, 201);

    padding: 10px;

    width: 100px;

}

3個區塊各占比例1份 order:2; order:1; order:3;寫上順序改為213呈現

.box-1{

    flex:1;

    order:2;

}

.box-2{

    flex:1;

    order:1;

}

.box-3{

    flex:1;

    order:3;

}

區塊空間不夠自動換行 flex-wrap:wrap;

.container{

    display:flex;

    flex-wrap:wrap;

flex-direction: column;

主軸改由上而下 flex-direction: column;

主軸改由下而上

flex-direction: column-reverse;

flex-direction: row;

以上是主軸改由左而右

主軸由右而左

flex-direction: row-reverse;

把區塊放到主軸的中間

justify-content: center;

把區塊放到主軸的未端(右邊)

justify-content: flex-end;

把區塊放到主軸的開始

justify-content: flex-start;

空白處平均分配在區塊的左邊右邊

justify-content: space-around;

把區塊1放在主軸開始.區塊3放在主軸結束.區塊2放在1.3的中間

justify-content: space-between;

平均分配

justify-content: space-evenly;

次軸置中

align-items: center;

次軸結束

align-items: flex-end;

以下會在正中間

justify-content: center;

    align-items: center;

主軸有換行,要用align-content: center;來置中

align-content: center;

**animation(動畫)**

<!DOCTYPE HTML>

<HTML>

    <HEAD>

        <meta charset="UTF-8">

        <title>jiaru</title>

        <link rel="stylesheet" href="style.css" />

    </HEAD>

    <body>

        <h1>css教學</h1>

        <div class="box box-animation"></div>

    </body>

</HTML>

@keyframes 先定義一個動畫 然後給動畫一個名字 change-color

.box-animation這邊要設定

Animation-name:輸入上方的名字

animation-duration: 2s; <<持續時間2秒,最後會轉回原本的灰色

animation-iteration-count: 2 定義這個動畫跑兩次

animation-iteration-count:infinite;

以上表示動畫會跑無限次

animation-delay: 2s;

以上是網頁載入後會延遲2秒,才開始跑動畫

前面的from跟to也可改成在持續時間內的0~100%呈現

@keyframes change-color {

    0% {background-color: red;}

    20% {background-color: orange;}

    40% {background-color: yellow;}

    60% {background-color: green;}

    80% {background-color: blue;}

    100% {background-color: purple;}

就會變成紅橙黃綠籃紫呈現

設定一個動畫由左上到右下

@keyframes change-color {

    0% {background-color: red;top: 10px;left: 10px;}

    20% {background-color: orange;top: 20px;left: 20px;}

    40% {background-color: yellow;top:30px;left: 30px;}

    60% {background-color: green;top: 40px;left: 40px;}

    80% {background-color: blue;top:50px;left: 50px;}

    100% {background-color: purple;top:60px;left: 60px;}

}

.box {

    width: 200px;

    height: 200px;

    background-color: grey;

    position: relative;

}

.box-animation {

    animation-name: change-color;

    animation-duration: 3s;

    animation-iteration-count:1;

若box-animation:hover { } 加上一個hover表示滑鼠滑過去後動畫才開始跑