|  |  |  |
| --- | --- | --- |
|  | <html lang="en"> | |
|  | <head> | |
|  | <meta charset="UTF-8"> | |
|  | <title>JS + CSS Clock</title> | |
|  | </head> | |
|  | <body> | |
|  |  | |
|  |  | |
|  | <div class="clock"> | |
|  | <div class="clock-face"> | |
|  | <div class="hand hour-hand"></div> | |
|  | <div class="hand min-hand"></div> | |
|  | <div class="hand second-hand"></div> | |
|  | </div> | |
|  | </div> | |
|  |  | |
|  |  | |
|  | <style> | |
|  | html { | |
|  | background:#018DED url(http://unsplash.it/1500/1000?image=881&blur=50); | |
|  | **background-size:cover;** | |
|  | font-family:'helvetica neue'; | |
|  | text-align: center; | |
|  | font-size: 10px; | |
|  | } | |
|  |  | |
|  | body { | |
|  | margin: 0; | |
|  | font-size: 2**rem**; | |
|  | display:flex; | |
|  | flex:1; | |
|  | **min-height: 100vh;** | |
|  | align-items: center; | |
|  | } | |
|  |  | |
|  | .clock { | |
|  | width: 30rem; | |
|  | height: 30rem; | |
|  | border:20px solid white; | |
|  | border-radius:50%; | |
|  | margin:50px auto; | |
|  | position: relative; | |
|  | padding:2rem; | |
|  | box-shadow: | |
|  | 0 0 0 4px rgba(0,0,0,0.1), | |
|  | inset 0 0 0 3px #EFEFEF, | |
|  | inset 0 0 10px black, | |
|  | 0 0 10px rgba(0,0,0,0.2); | |
|  | } | |
|  |  | |
|  | .clock-face { | |
|  | position: relative; | |
|  | width: 100%; | |
|  | height: 100%; | |
|  | **transform: translateY(-3px);** /\* account for the height of the clock hands \*/ | |
|  | } | |
|  |  | |
|  | .hand { | |
|  | width:50%; | |
|  | height:6px; | |
|  | background:black; | |
|  | position: absolute; | |
|  | top:50%; |
|  | **transform-origin: 100%;** |
|  | transform: rotate(90deg); |
|  | transition: all 0.05s; |
|  | transition-timing-function: cubic-bezier(0.1, 2.7, 0.58, 1); |
|  | } |
|  | </style> |
|  |  |
|  | <script> |
|  | const secondHand = document.querySelector('.second-hand'); |
|  | const minsHand = document.querySelector('.min-hand'); |
|  | const hourHand = document.querySelector('.hour-hand'); |
|  |  |
|  | function setDate() { |
|  | const now = **new Date();** |
|  |  |
|  | const seconds = now.getSeconds(); |
|  | **const secondsDegrees = ((seconds / 60) \* 360) + 90;** |
|  | secondHand.style.transform = `rotate(${secondsDegrees}deg)`; |
|  |  |
|  | const mins = now.getMinutes(); |
|  | **const minsDegrees = ((mins / 60) \* 360) + ((seconds/60)\*6) + 90;** |
|  | minsHand.style.transform = `rotate(${minsDegrees}deg)`; |
|  |  |
|  | const hour = now.getHours(); |
|  | **const hourDegrees = ((hour / 12) \* 360) + ((mins/60)\*30) + 90;** |
|  | hourHand.style.transform = `rotate(${hourDegrees}deg)`; |
|  | } |
|  |  |
|  | **setInterval(setDate, 1000);** |
|  |  |
|  | setDate(); |
|  |  |
|  | </script> |
|  | </body> |
|  | </html> |

**css**

一、

**background-size:cover;**

|  |  |
| --- | --- |
| cover | 把背景图像扩展至足够大，以使背景图像完全覆盖背景区域。  背景图像的某些部分也许无法显示在背景定位区域中。 |

二、

**min-height: 100vh;**

## 定义和用法

min-height 属性设置元素的最小高度。

### 说明

该属性值会对元素的高度设置一个最低限制。因此，元素可以比指定值高，但不能比其矮。不允许指定负值。

|  |  |
| --- | --- |
| vh | Relative to 1% of the height of the viewport\* |

vh, vw: 相對於瀏覽器展示網頁區域的大小(不是整個瀏覽器的大小，沒包含瀏覽器的工具列，只有展示網頁的區域)

例如:100vh和100vw時，意思就是我的這個div要是整個螢幕的可視範圍，所以你會看到，瀏覽器全部都是黃色的，而且很重要的是，這個區塊會隨著瀏覽器的縮放而改變

三、**rem**

|  |  |
| --- | --- |
| rem | Relative to font-size of the root element |

rem 也是相對的文字尺寸，和 em 使用方法接近，不同的是他僅相對於 root 層級的文字大小(網頁中的 html)。

以這個範例來說，此網站的文字大小是 16px，rem的尺寸就是以 16px 為基準，故下方的 1rem 會與 16px 相同尺寸。所以無論外層是否有其他文字大小，皆不會影響 rem 的尺寸，唯一會影響 rem 比例的只有 **html** 的文字大小(可以打開 *開發者工具*試試看喔)。

四、**transform: translateY(-3px);**

transform: transform 属性向元素应用 2D 或 3D 转换。该属性允许我们对元素进行旋转、缩放、移动或倾斜。

|  |  |
| --- | --- |
| translateY(*y*) | 定义转换，只是用 Y 轴的值。 |

The **translateY()** CSS function moves the element vertically on the plane.

<http://www.w3school.com.cn/tiy/c.asp?f=css_transform_translatey>

五、**transform-origin: 100%;**

Set a rotated element's base placement(设置旋转元素的基点位置)

The transform-origin property allows you to change the position of transformed elements.

2D transformations can change the x- and y-axis of an element. 3D transformations can also change the z-axis of an element.

transform-origin 属性允许您改变被转换元素的位置。

2D 转换元素能够改变元素 x 和 y 轴。3D 转换元素还能改变其 Z 轴。

## 语法

transform-origin: *x-axis* *y-axis* *z-axis*;

<http://www.w3school.com.cn/example/css3/demo_css3_transform-origin.html>

## 六、CSS3 Transitions

CSS3 transitions allows you to change property values smoothly (from one value to another), over a given duration.

**js**

1. const

在 ES6 之前，JavaScript 並沒有常數這種東西。大多數時候，人們會按照命名慣例，將一個**全部以英文字母大寫表示的變數當作一個常數**。小心留意不將它們放在左值。

ES6 增加了 const 這個定義。凡是用 const 定義的符號，其繫結的內容僅能在定義時設定初值，之後不允許再改變。這就是常數了。試圖改變 const 常數的敘述，都是語法錯誤。除此之外，const 的語法限制和 let 相同，不允許重複宣告、不允許宣告前使用。

1. **new Date();**

Using new Date(), creates a new date object with the **current date and time**

Tue Jul 04 2017 13:32:06 GMT+0800 (台北標準時間)

|  |
| --- |
| const now = **new Date();** |
|  |
| const seconds = now.getSeconds(); |

The getSeconds() method returns the seconds (from 0 to 59) of the specified date and time.

const mins = now.getMinutes();

Return the minutes, according to local time

The getMinutes() method returns the minutes (from 0 to 59) of the specified date and time.

1. const hour = now.getHours();

Return the hour, according to local time

The getHours() method returns the hour (from 0 to 23) of the specified date and time.

1. **setInterval(setDate, 1000);**

The setInterval() method calls a function or evaluates an expression at specified intervals (in milliseconds).

The setInterval() method will continue calling the function until [clearInterval()](https://www.w3schools.com/jsref/met_win_clearinterval.asp) is called, or the window is closed.

The ID value returned by setInterval() is used as the parameter for the clearInterval() method.

**Tip:** 1000 ms = 1 second.

**Tip:** To execute a function only once, after a specified number of milliseconds, use the [setTimeout()](https://www.w3schools.com/jsref/met_win_settimeout.asp) method.

1. **const minsDegrees = ((mins / 60) \* 360) + ((seconds/60)\*6) + 90;**

6是指6degrees,長針一分鐘走六度(360度/60分鐘=6度)

1. **const hourDegrees = ((hour / 12) \* 360) + ((mins/60)\*30) + 90;**

30是指30degrees,短針1小時走30度(360度/12小時=30度)

1. secondHand.style.transform = **`rotate(${secondsDegrees}deg)`**;

# [JavaScript ES6 中的模版字符串（template literals）](https://pjchender.blogspot.tw/2017/01/javascript-es6-template-literalstagged.html)

### 在模版字符串中嵌入變數 另外，在模版字符串中，我們還可以透過 ${...} 這樣的方式，嵌入變量或任何的表達式：

let myName = "PJCHENder",

numA = 4,

numB = 7;

let content = `Hello, my name is ${myName}, my lucky number is ${2\*(numA + numB)}`;

console.log(content); // "Hello, my name is PJCHENder, my lucky number is 22"

我們可以看到透過 ${...}，裡面我們不只可以放入變數，還可以放入表達式（例如${3+4}）。