JQUERY

CORE

jQuery 語法原理

一個只有函數名稱為「$」 與 class 名稱為「$」的函數庫

```
$ = (s) => {
  if ((typeof s) == "string") {
    console.log(s)
  }

  if ((typeof s) == "function") {
    console.log(s())
  }
}

$("hello")  // Print 'hello'
$(function() {
  return "hi"
})
// Print 'hi'
```

載入函數庫

完整版https://code.jquery.com/jquery-3.7.1.min.js

不含 AJAX 與特效的瘦身版

- https://code.jquery.com/jquery-3.7.1.slim.js

CDN 頁面

- https://releases.jquery.com

起手式

```
$(() =>
    $("#bn").click(() =>
        $("#test").html("Hello, World!")
    )
)
```

```
<html>
<script src="https://code.jquery.com/jquery-3.7.1.min.js"></script>
<script>
    $(function() {
        function() {
            $("#bn").click(
                function() {
                    $("#test").html("Hello, World!")
                }
            )
        }
    })
</script>
<body>
    <button id="bn">click</button>
    <div id="test"></div>
</body>
</html>
```

事件監聽

除了使用 $(物件).event() 外也可以使用 .on()，相當於 addEventListener()

取消使用 .off()

```
<script>
$(() =>
    $("#bn").on("click", () =>
        $("#test").html("Hello, World!")
    )
)
</script>
<body>
    <button id="bn">click</button>
    <div id="test"></div>
</body>
```

取得EVENT額外資訊

當事件發生時，透過傳入的event參數取得額外資訊，例如滑鼠座標、按鍵種類

點擊座標

```
$('div').click(function(event) {
    let x = event.offsetX
    let y = event.offsetY
})
```

取消事件預設行為

preventDefault()

例如超連結標籤，滑鼠 click 後預設是連到設定的網址

```
<a href="https://www.google.com">Google</a>
```

取消預設的行為改為我們自訂的行為

```
$('a').click(function(event) {
    event.preventDefault()
    alert('Hello, World!')
})
```

選取器 tag#id.class

選取 tag，例如所有 <p> 標籤

```
<script>
$(() =>
    $("p").css("color", "orange")
)
</script>
<body>
    <p>Hello</p>
    <p>World</p>
</body>
```

選取 id："#id"

選取屬性："attribute='value'"

選取tag + 屬性："input[attribute='value']"

選取class：".class"

https://api.jquery.com/category/selectors/

jQUERY物件與JS物件轉換

jQuery 轉 JS

- $('#button').get(0)
- $('#button')[0]

JS 轉 jQuery

- $(document.getElementById('button'))

取得與設定 CSS

取得

- let color = $("div").css("background-color")

設定

- $("div").css("background-color", "orange")

設定多個 CSS

```
$("div").css({
    "background-color": "brown",
    "color": "white"
})
```

取得與設定屬性

取得

- let url = $('image').attr('src')

設定

- $('image').attr('src', 'http://....')

取得value屬性

- $('input[type=text]').val()

設定value屬性

- $('input[type=text]').val('Hello, World!')

練習

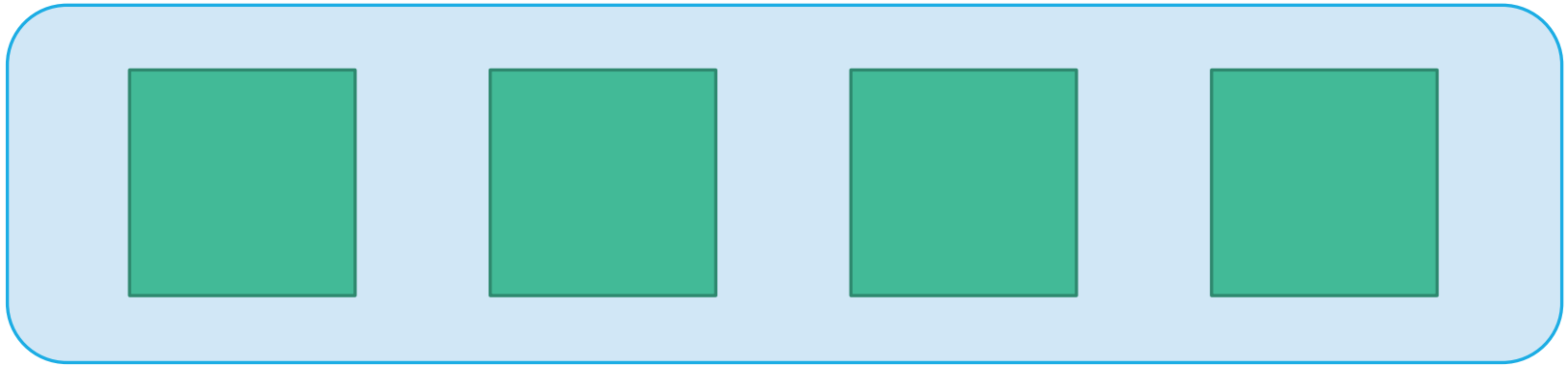
設計以下頁面

若「已閱讀」未核取時，點選超連結會讓已閱讀字串變成紅色，並且超連結無法正常運作

選取器 nth

選取器 nth-child

```
<div>
    <p>A</p>
    <p>B</p>
    <p>C</p>
    <p>D</p>
</div>
```



$('div p:first-child')

$('div p:last-child')

$('div p:nth-child(2)')

$('div p:nth-child(3)')

找父親 – parent()

```
<div>
    <p>Hello, World!</p>
</div>
```

```
$(function() {
    $('p').parent().css('background-color', 'lightgreen')
})
```

找小孩 – children()

```
<div>
    <p>1</p>
    <p>2</p>
    <p>3</p>
</div>


$(function() {
    $('div').children().css('background-color', 'lightgreen')
})
```

找鄰居 – siblings()

找 2 的鄰居

```
<div>
    <p>1</p>
    <p id="demo">2</p>
    <p>3</p>
</div>
```

```
$(function() {
    $('#demo').siblings().css('background-color', 'lightgreen')
})
```

CLASS 操作

.addClass()

.hasClass()

.removeClass()

.toggleClass()

CSS

```
.lightgreen {
    background: lightgreen;
}
```

HTML

```
<p>Hello, World!</p>
```

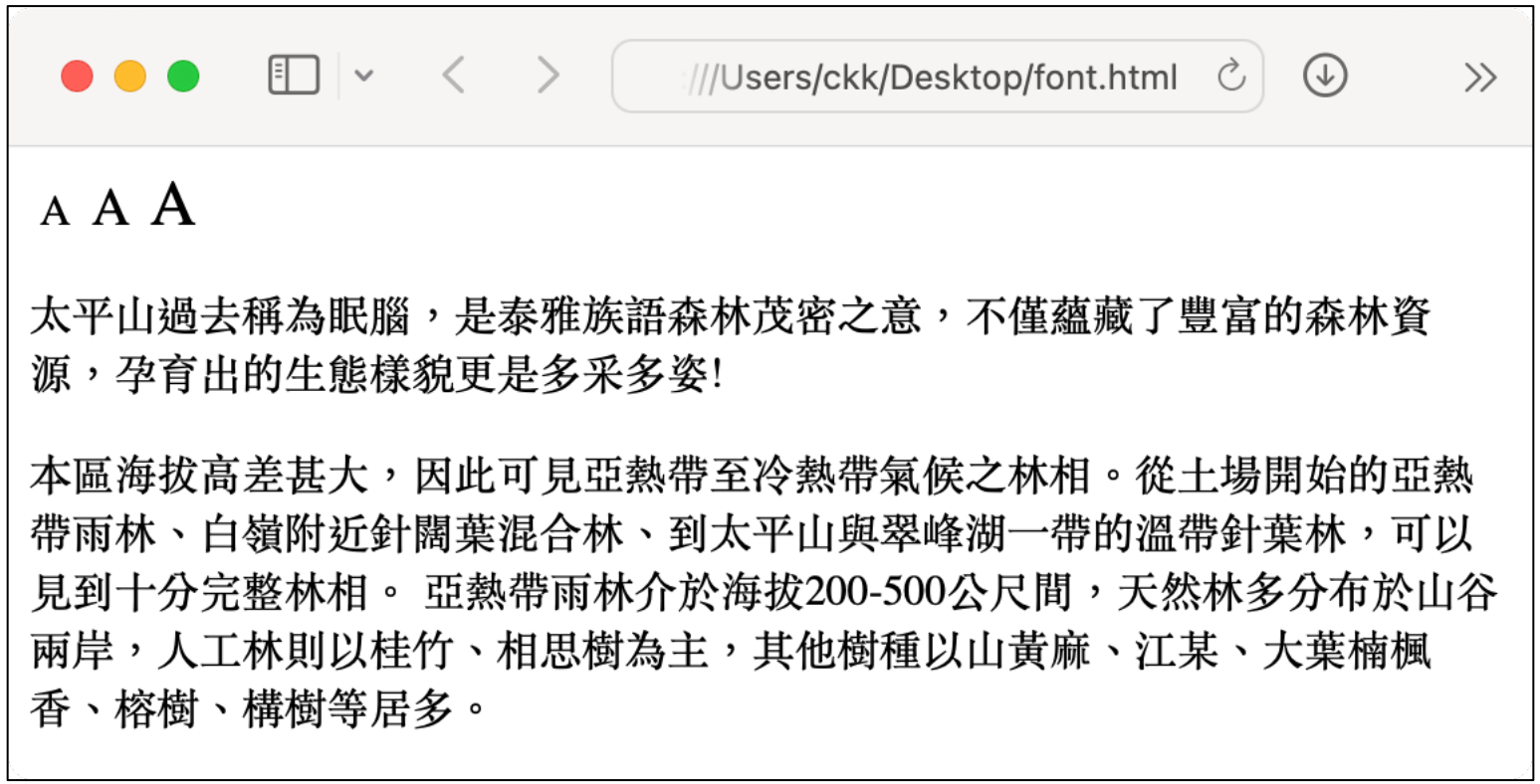
JS

```
$('p').addClass('lightgreen')

$('p').removeClass('lightgreen')
```

練習

使用 jQuery 設定文字大小，游標移到「A」圖示上需要變成手的形狀，如下圖



迴圈

jQuery 的 each 函數可以放陣列也可以放 HTML 元素

```
$(() => {
    $.each([5, 2, 7, 3], function(index, el) {
        console.log(el)
    })
})
```

```
<script>
$(() => {
    let colors = ["red", "orange", "yellow", "green"]
    $("div").each((index, item) => {
        $(item).css("background-color", colors[index])
    })
})
</script>
<body>
    <div>Red</div>
    <div>Orange</div>
    <div>Yellow</div>
    <div>Green</div>
</body>
```

隱藏與顯示

動畫效果

- hide(1000)

另外試試看 toggle()，可自動在 hide 與 show 間切換

```
<script>
$(function() {
    $("#hide").click(function() {
        $("#test").hide()
    })
    $("#show").click(function() {
        $("#test").show()
    })
})
</script>
<body>
    <button id="hide">hide</button>
    <button id="show">show</button>
    <p>
    <div id="test">Hello, World!</div>
</body>
```

ARROW FUNCTION 中的 $(this)

按鈕按下後修改按鈕上的字串

下面的程式碼語法正確，但沒有任何反應

```
<script>
$(() =>
    $("#bn").on("click", () => {
        $(this).html("Hello")
    })
)
</script>
<body>
    <button id="bn">click</button>
</body>
```

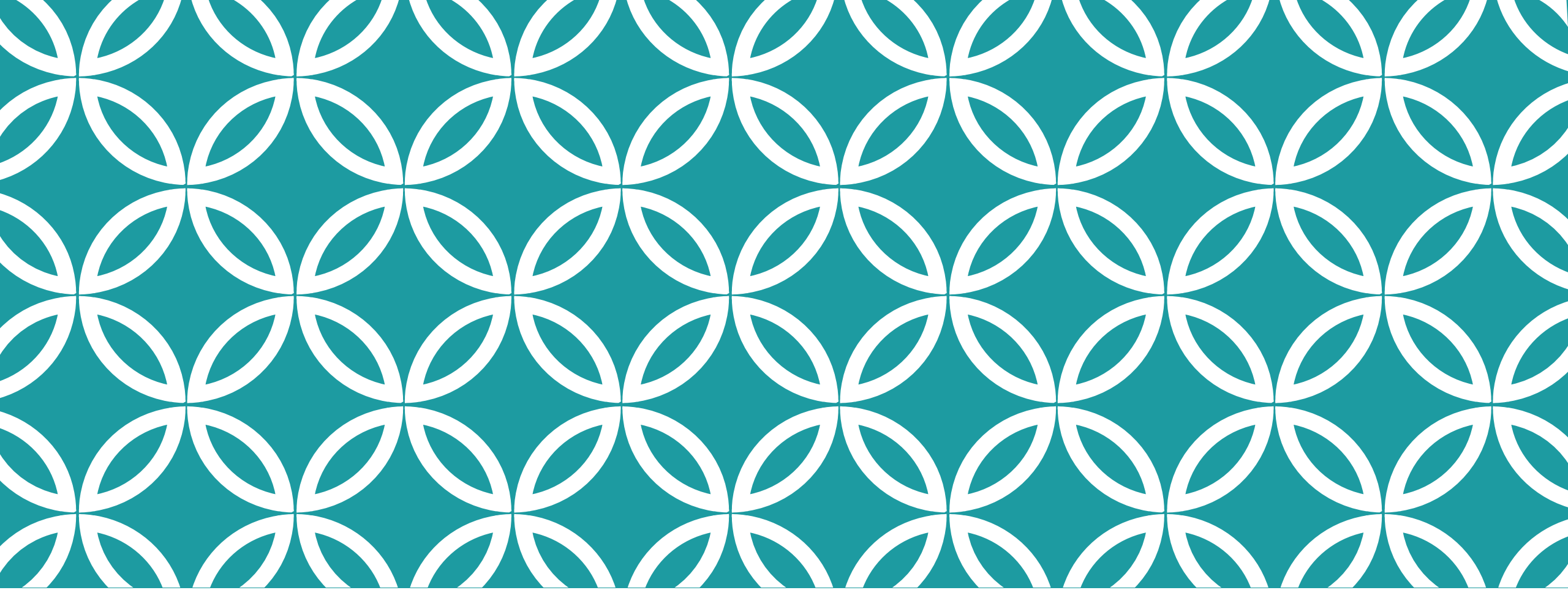
兩種處理方式

方式一：不要使用 Arrow Function

```
$("#bn").on("click", function() {
    $(this).html("Hello")
})
```

方式二：將 this 改為 event.currentTarget

```
$("#bn").on("click", (event) => {
    $(event.currentTarget).html("Hello")
})
```

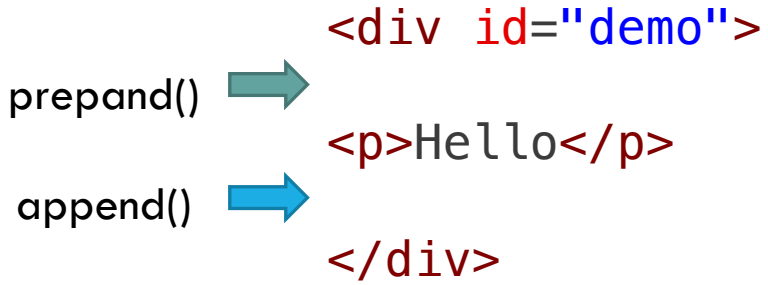


DOM操作

附加

插入文字

```
$('#demo').prepend('before hello')
$('#demo').append('after hello')
```

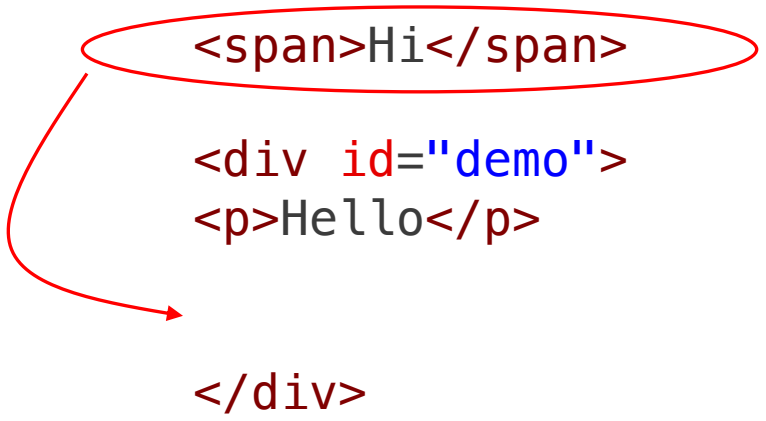


插入標籤

```
$('#demo').prepend('<h1>before hello</h1>')
$('#demo').append('<h1>after hello</h1>')
```

搬移

```
$('#demo').append($('span'))
```



插入

```
$('span').before('<br>')
```

```
<br>
<span>A</span>
<br>
<span>B</span>
```

```
$('span').after('<br>')
```

```
<span>A</span>
<br>
<span>B</span>
<br>
```

原本内容

```
<span>A</span>
<span>B</span>
```

對調

將 A、B 位置對調

```
<div>A</div>
<div>B</div>
```

解法

```
$a = $('div').first()
$b = $('div').last()
$b.after($a)
```

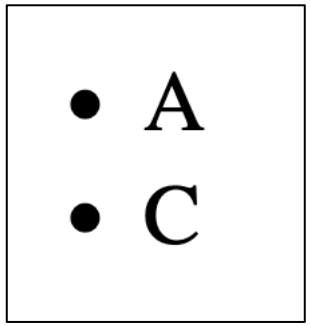
刪除

remove() 刪除標籤本身

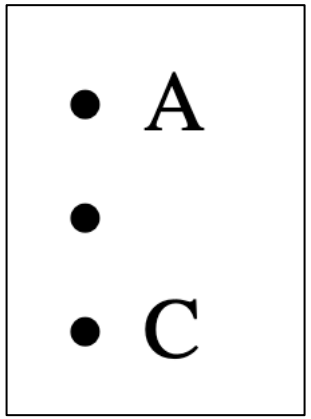
empty() 刪除標籤中的內容

```
<ul>
    <li>A</li>
    <li>B</li>
    <li>C</li>
</ul>
<button>Del</button>
```

```
$('button').click(function() {
    $('ul li:nth-child(2)').remove()
})
```




```
$('button').click(function() {
    $('ul li:nth-child(2)').empty()
})
```



練習

按一個按鈕後，把下列格子中的數字隨機打亂，請以 DOM 的方式操作

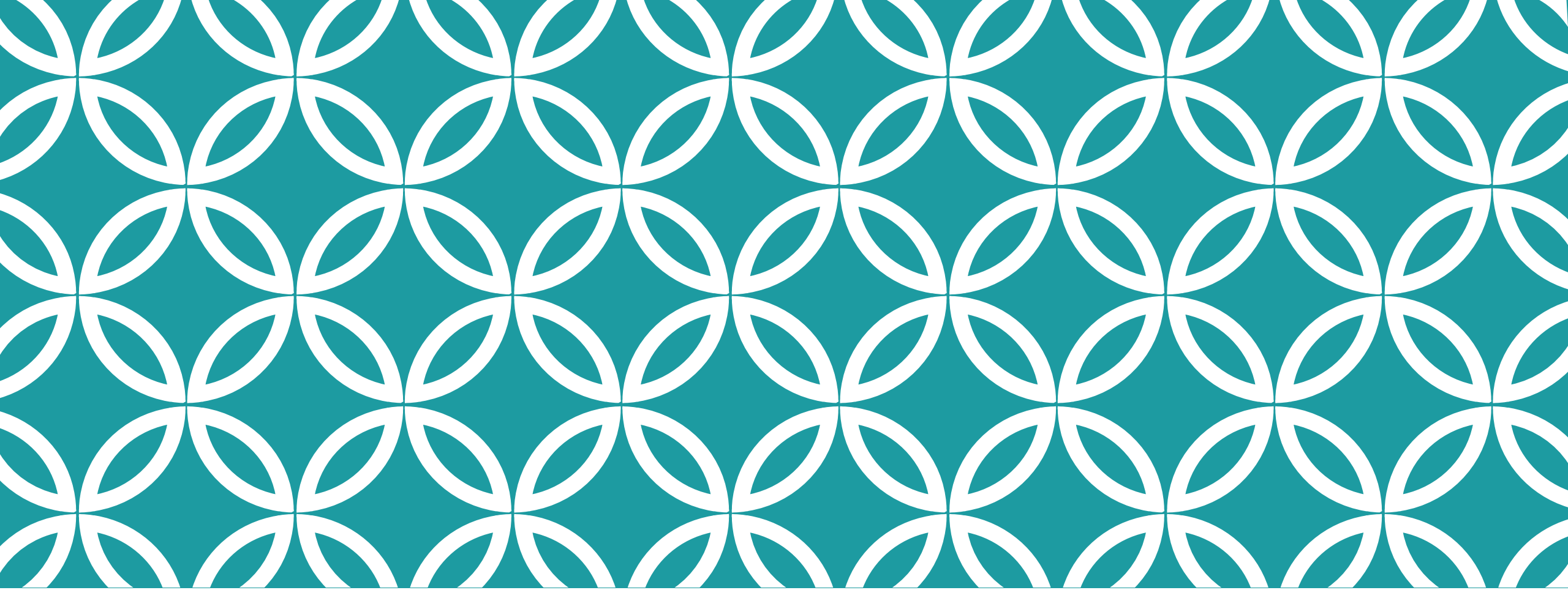
1	2	3
4	5	6
7	8	9

練習

1. 每按一次新增按鈕，在表格中新增一筆資料
2. 每一筆資料右邊具有刪除按鈕，按下後即可刪除該筆資料

```
$(function() {
    $('body').append('<table border="1"></table>')
    $('table').hide()
    $('button').click(function() {
        $('table').show()
        $('table').append(`<tr><td>${Math.random()}</td></tr>`)
    })
})
```

```
<body>
<button>Add</button>
</body>
```



AJAX

AJAX

GET

```
$.ajax({
    url: "news.txt",
    success: (result) => {
        $("#test").text(result)
    }
}) .done(() => {
    alert('done')
})
```

POST

```
$.ajax({
    type: "POST",
    url: "news.txt",
    success: (result) => {
        $("#test").text(result)
    }
}) .done(() => {
    alert('done')
})
```

省略時為 get

簡略語法

$.get(url, [data], [function() {}])

$.post(url, [data], [function() {}])

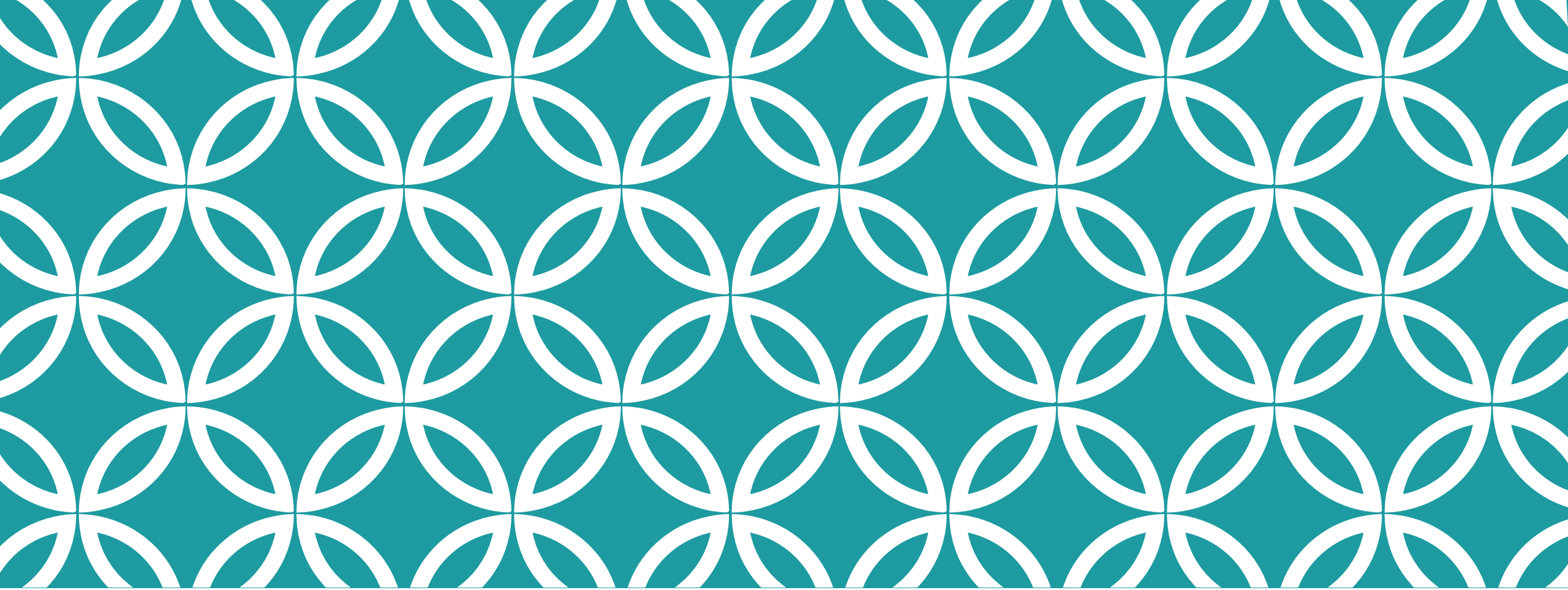
```
$.get("news.txt", (result) => {
        $("div").html(result)
    }
)
```

與表單結合

```
<form>
    Number 1: <input name="a"><br/>
    Number 2: <input name="b" ><p/>
    <input type="button" id="bn" value="click">
</form>
```

```
$.post("do.php", $("form").serialize(), function(result) {
    $("#test").text(result)
})
```

```
$("#bn").click(() => {
    $.ajax({
        type: "post",
        url: "do.php",
        data: $("form").serialize(),
        success: (result) => {
            $("#test").text(result)
        }
    })
})
```



動畫

基本動畫

動畫速度

- 預設: 400ms
- fast: 200ms
- slow: 600ms
- 可自訂，填入數字即可

```
<script>
$(function() {
    $("button").click(function() {
        $("p").animate({
            top: "+=100",
        }, "slow")
    })
})
</script>
<body>
<button>Click</button>
<p style="position:relative">Hello</p>
</body>
```

能夠有動畫的屬性

基本上值為數字的CSS屬性，例如

- width、height
- left、top、bottom、right
- opacity
- font-size

以為可以但不行的屬性，例如

- transform

STEP

每次動畫過程中都會呼叫 step 所綁定的函數

用這個技術可以透過 CSS 做到旋轉效果

```
$(function() {
    let deg = 0
    $("button").click(function() {
        $("p").animate({
            top: "+=100"
        }, {
            duration: "slow",
            step: function(now, fx) {
                $("p").css("transform", `rotate(${deg}deg)`)
                deg += 5;
            }
        })
    })
})
```