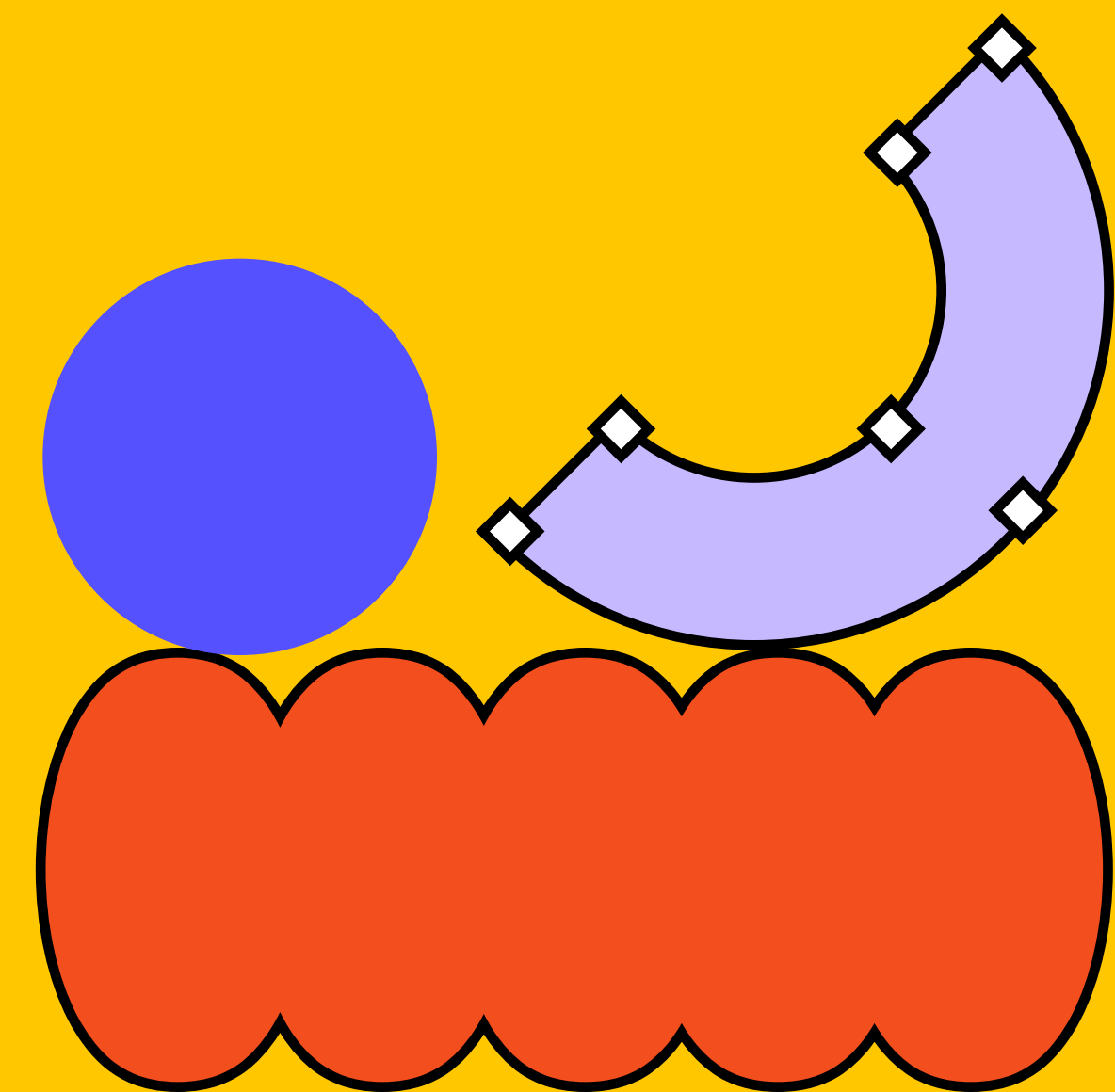
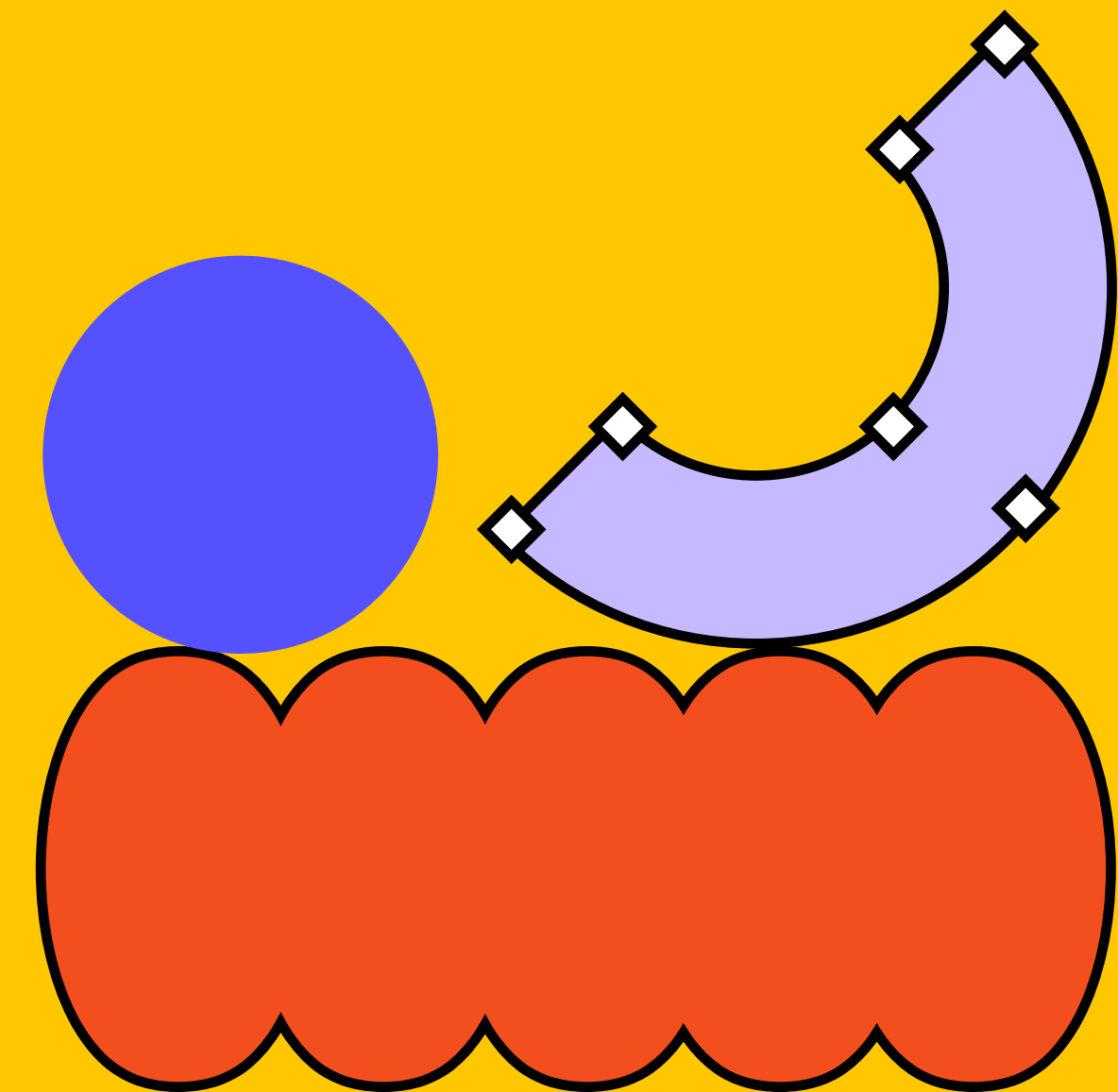


# 什麼是 jQuery



# Why jQuery?



# 簡化 JavaScript 編碼

```
// 使用原生 JavaScript 選擇元素並添加事件
document.getElementById('myButton').addEventListener('click', function() {
    alert('Button clicked!');
});
```

```
// 使用 jQuery 選擇元素並添加事件
$('#myButton').click(function() {
    alert('Button clicked!');
});
```

# 瀏覽器相容性

讓你的程式能在不同瀏覽器中正常運行。

# DOM 操作更簡單

插入、刪除、修改和遍歷 DOM 結構變得非常容易。

```
// 使用原生 JavaScript 操作 DOM  
var newElement = document.createElement('div');  
newElement.innerHTML = 'New Element';  
document.body.appendChild(newElement);
```

```
// 使用 jQuery 操作 DOM  
$('<div>New Element</div>').appendTo('body');
```

# 事件處理簡化

讓事件綁定和移除變得更加方便。

```
// 使用 jQuery 綁定和移除事件
$('#myElement').on('click', function() {
    console.log('Element clicked');
});

$('#myElement').off('click');
```

# 動畫和效果

可以輕鬆實現淡入淡出、滑動等效果。

```
// 使用 jQuery 動畫效果  
$( '#myElement' ).fadeIn();  
$( '#myElement' ).slideUp();
```

# AJAX

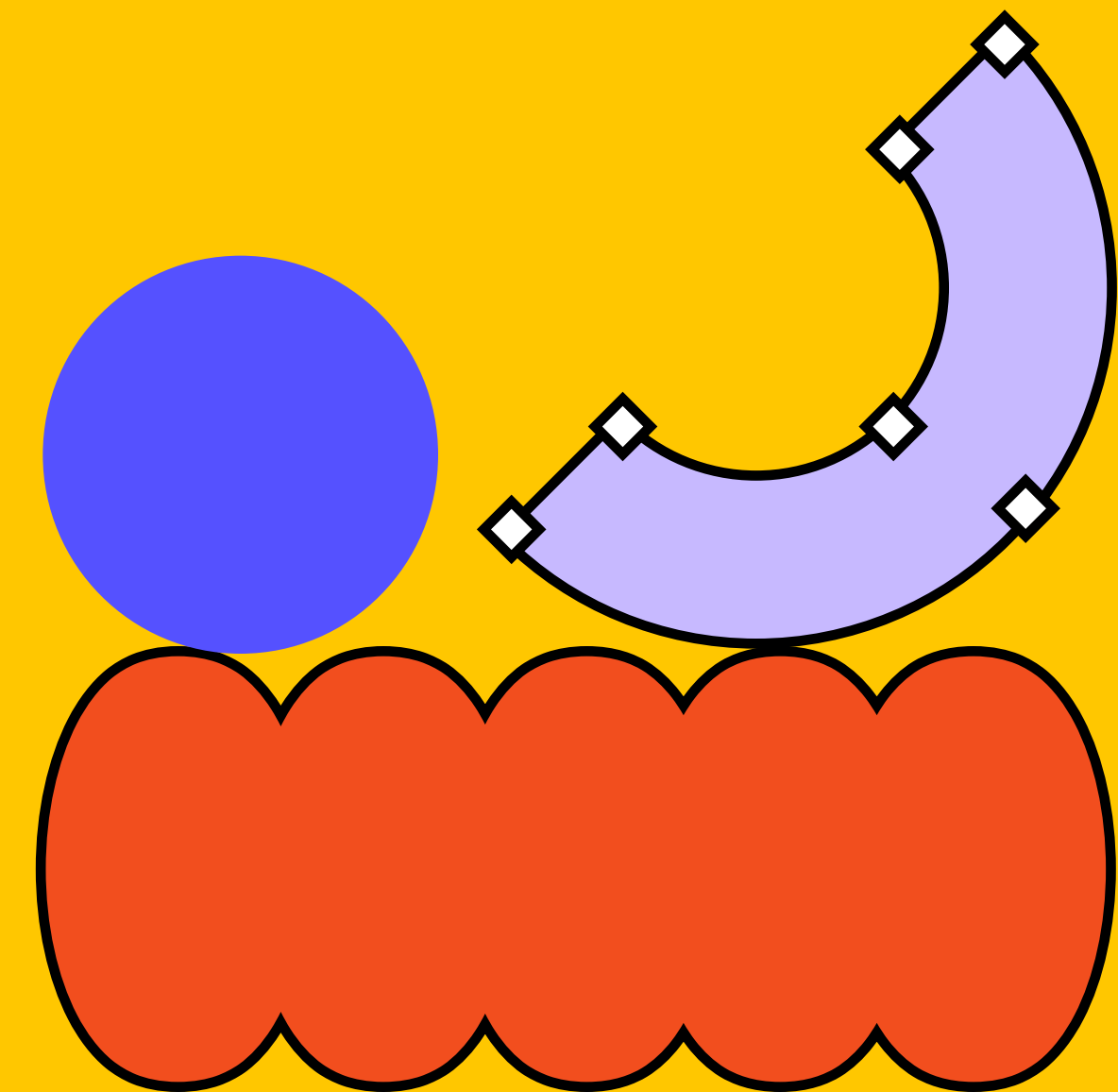
簡化與伺服器不同步通信的過程，從伺服器取得數據  
變得簡單且更有效率。



# 學習曲線平緩

語法簡單直觀，可以快速上手，並且能夠快速看到效果，增加學習興趣。

# How to get started ?



# CDN

使用 CDN 連結 jQuery 是最簡單和推薦的方式，因為它可以減少你的伺服器負載，並且用戶可能已經有緩存的 jQuery 版本。

```
$(document).ready(function() {  
    alert("jQuery is loaded and ready!");  
});
```

# 下載和連結 jQuery

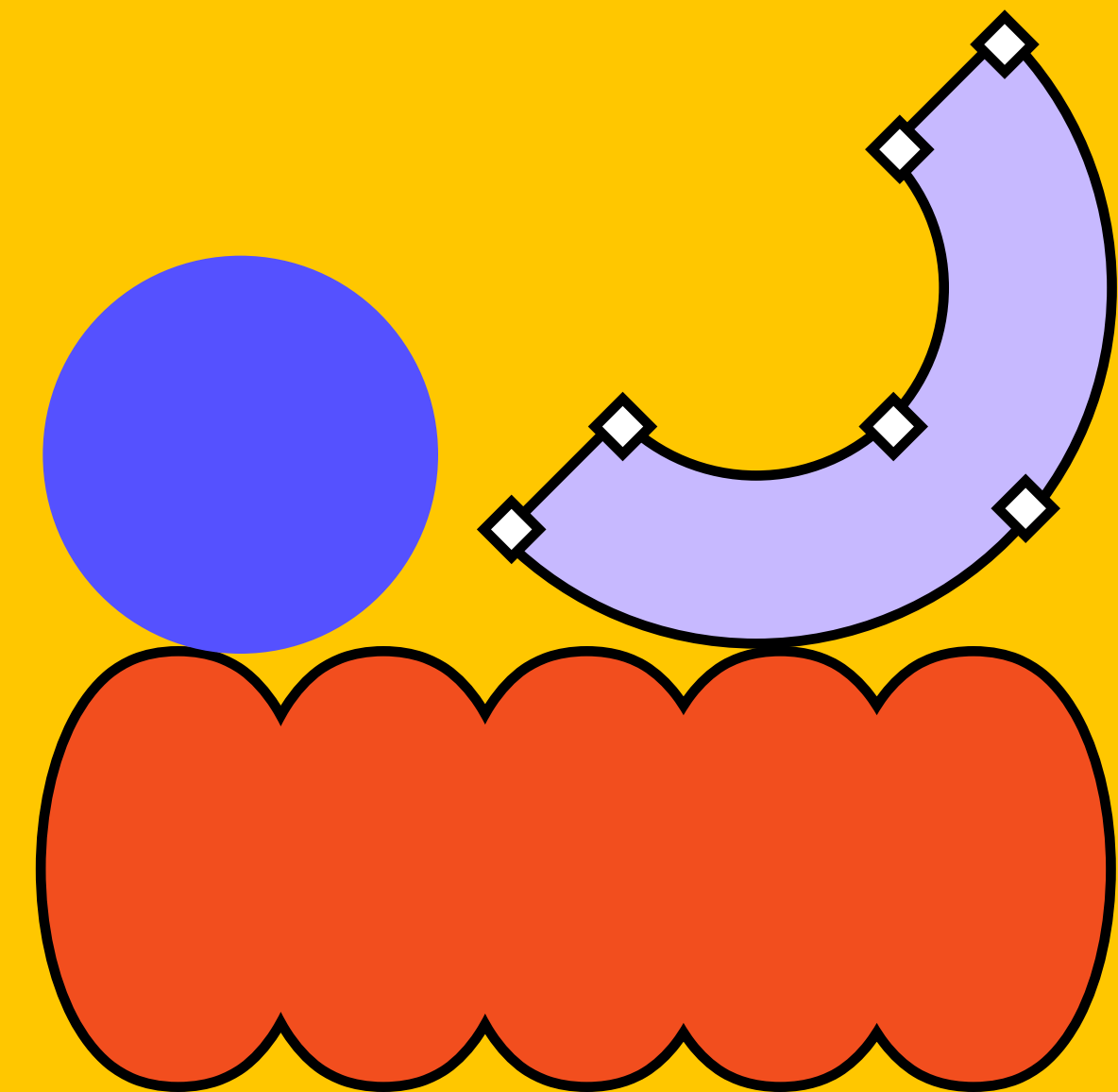
可以從 jQuery 官網下載 jQuery，並將其保存在你的專案目錄中，然後在 HTML 檔案中引用。

```
$(document).ready(function() {  
    alert("jQuery is loaded and ready!");  
});
```

# **minify**

jquery-3.7.1.min.js

# Selecting elements

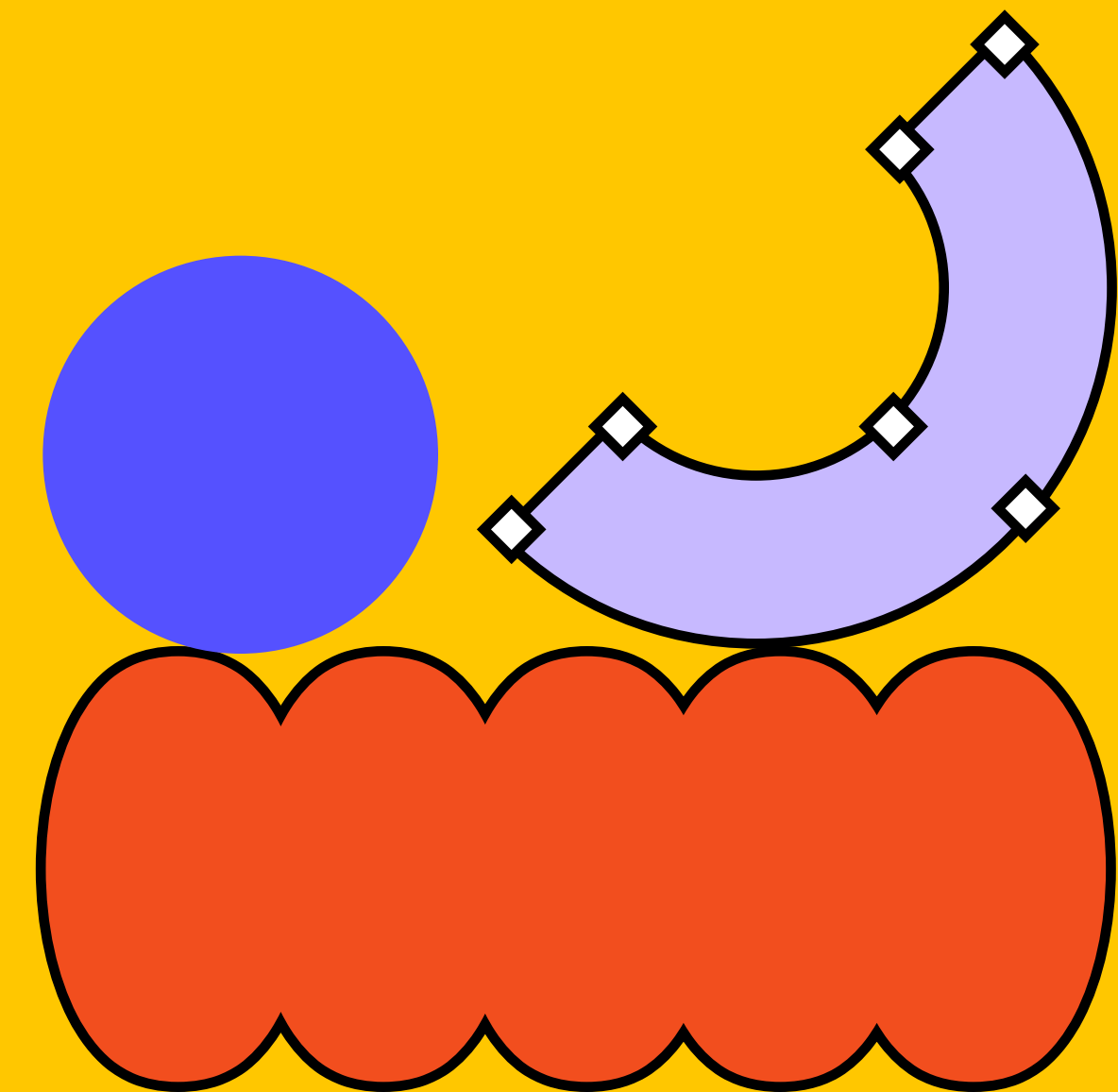


# .attr()

為每個取得元素 設置或取得 一個或多個屬性

```
| var attributeValue = $("selector").attr("attributeName");
```

# Style with jQuery





# .css()

用於 取得或設定 選擇元素集合第一個元素的 CSS 屬性值

```
$('#ul li').eq(1).css('color', 'blue');
```

```
$('#h1').css({  
  color: 'green',  
  fontSize: '40px'  
});
```

# .addClass() | .removeClass()

用於為選擇元素 新增或移除 集合中一個 類別。

```
| $("p").addClass("highlight bold");
```

# .hasClass()

用來檢查取得元素集中是否有**任何一個元素具有指定的類別**。

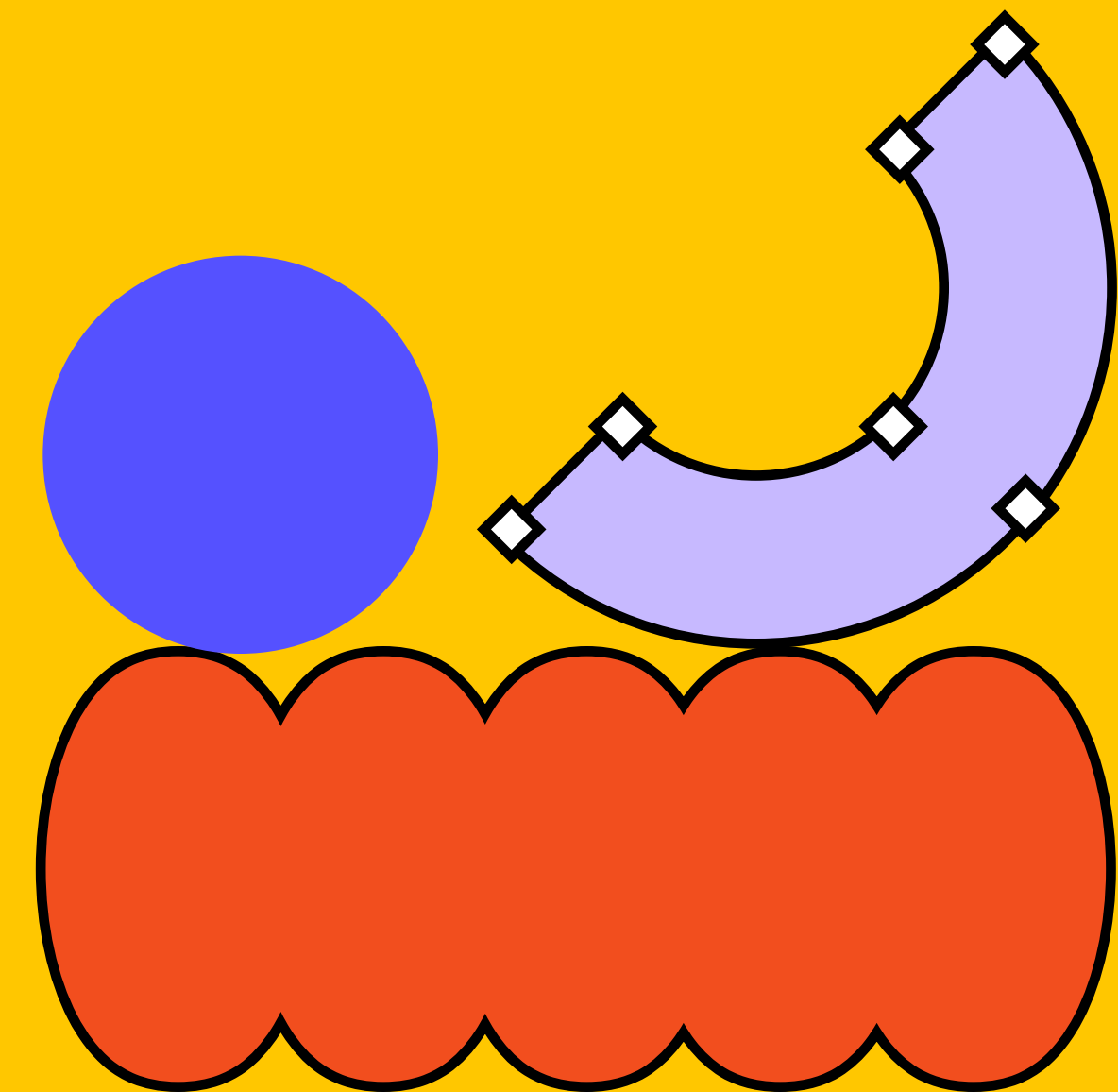
```
| $('ul li').eq(1).css('color', 'blue');
```

# **.toggleClass()**

用於切換每個取得的元素上的一個 CSS 類別。

```
| $("p").toggleClass("highlight bold");
```

# Text with jQuery



# .text()

用於 取得或設定 元素集中每個元素的內容。

```
var textContent = $("selector").text();
```

```
var text = $("p").text();  
console.log(text);
```

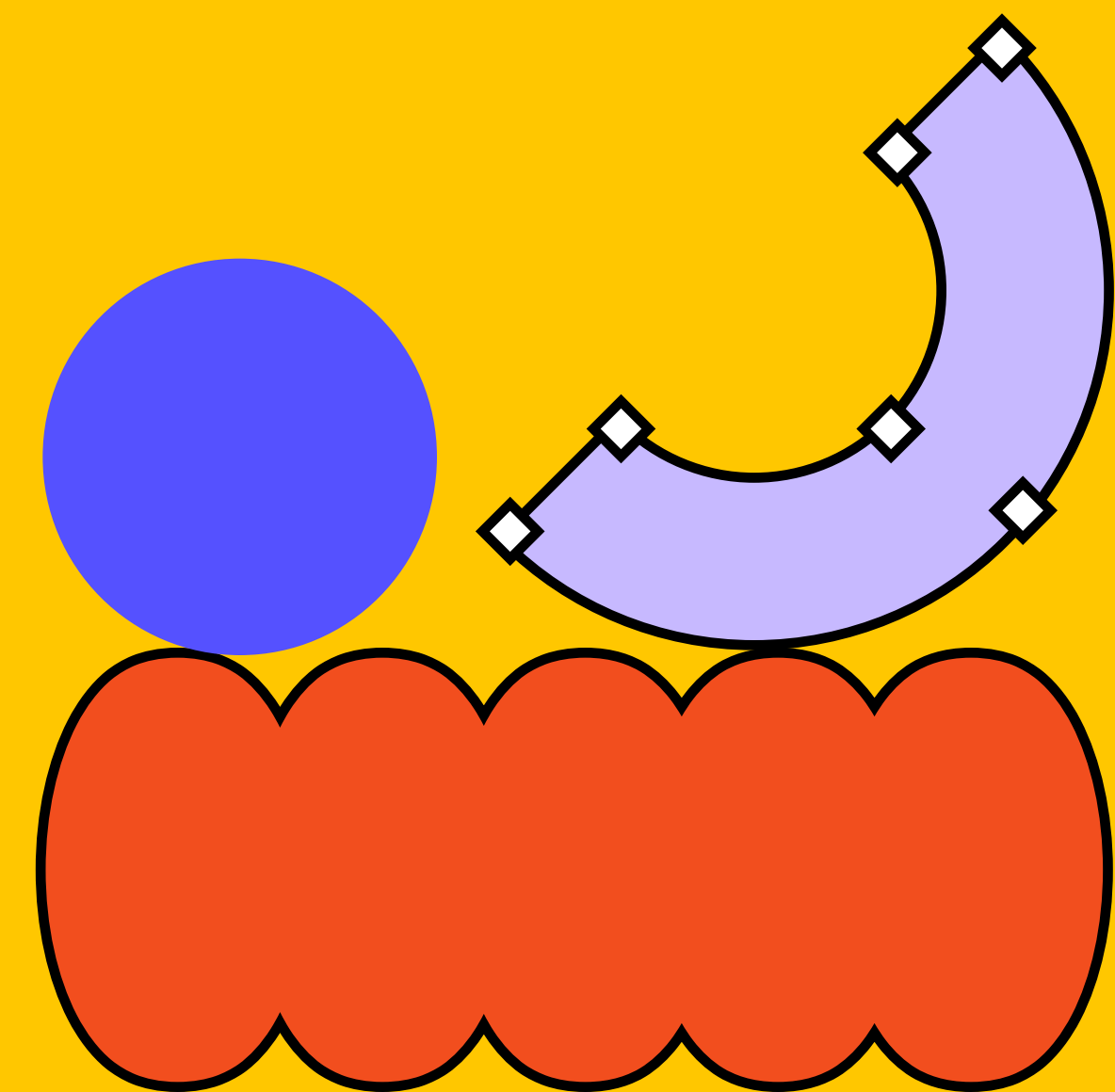
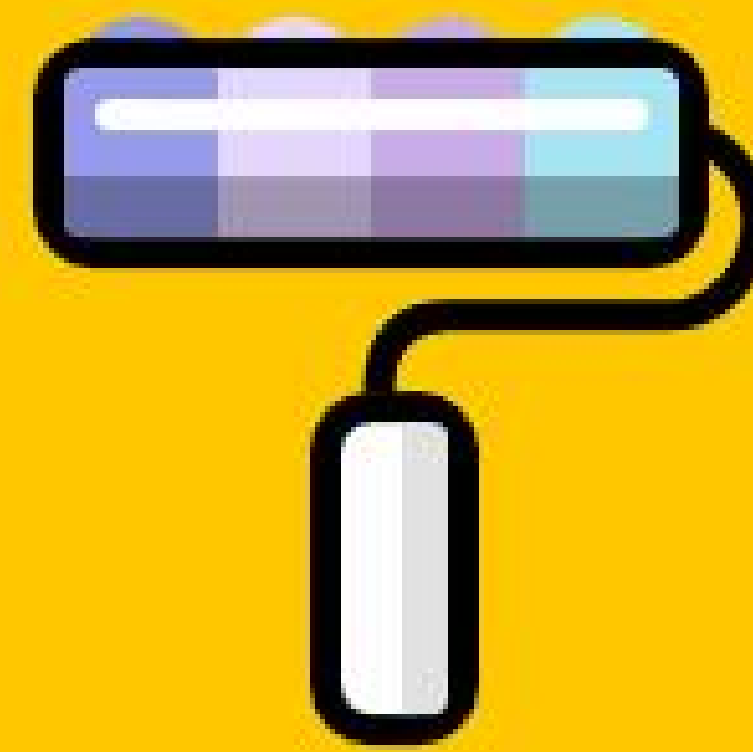
# .html()

用於 取得或設定 選擇元素集中第一個元素的 HTML 內容。

```
var htmlContent = $("selector").html();
```

```
var html = $("div").html();  
console.log(html);
```

# Operating elements





# **.eq( index )**

用於取得選擇元素集合中的一個指定索引的元素。

```
| $('ul li').eq(1).css('color', 'blue');
```

# **.first() | .last()**

用於取得選擇元素集合中的 第 | 最後 一個元素。

```
| $('ul li').first().css('color', 'red');
```

# .toArray()

將 jQuery 元素集合轉換為原生的 JavaScript 陣列。

```
var listItemsArray = $('ul li').toArray();  
  
sArray.forEach(function(item) {  
    console.log($(item).text());  
});
```

# .add()

用於將新的元素或選擇器添加到選定的元素集合中。

```
$(selector).add(selector[, context]);  
$(selector).add(elements);  
$(selector).add(html);  
$(selector).add(jQuery object);  
  
var obj = $("h1").add("p");
```

# .not()

從取得的元素集合中排除某些元素。

```
$(selector).not(selector);  
$(selector).not(elements);  
$(selector).not(function(index, element));  
$(selector).not(jQuery object);  
  
$("section").not("a").css("color", "red")
```

# .filter()

從取得的元素集合中過找出符合指定標準的元素。

```
$("#li").filter(".cl").css("background-color", "yellow");  
$(".cl").css("background-color", "red");
```

```
$("#li").filter(function(index) {  
    return index % 2 === 0;  
}).css("background-color", "yellow");
```

# .has()

用於篩選取得的元素集合，僅保留那些包含指定子元素的元素。

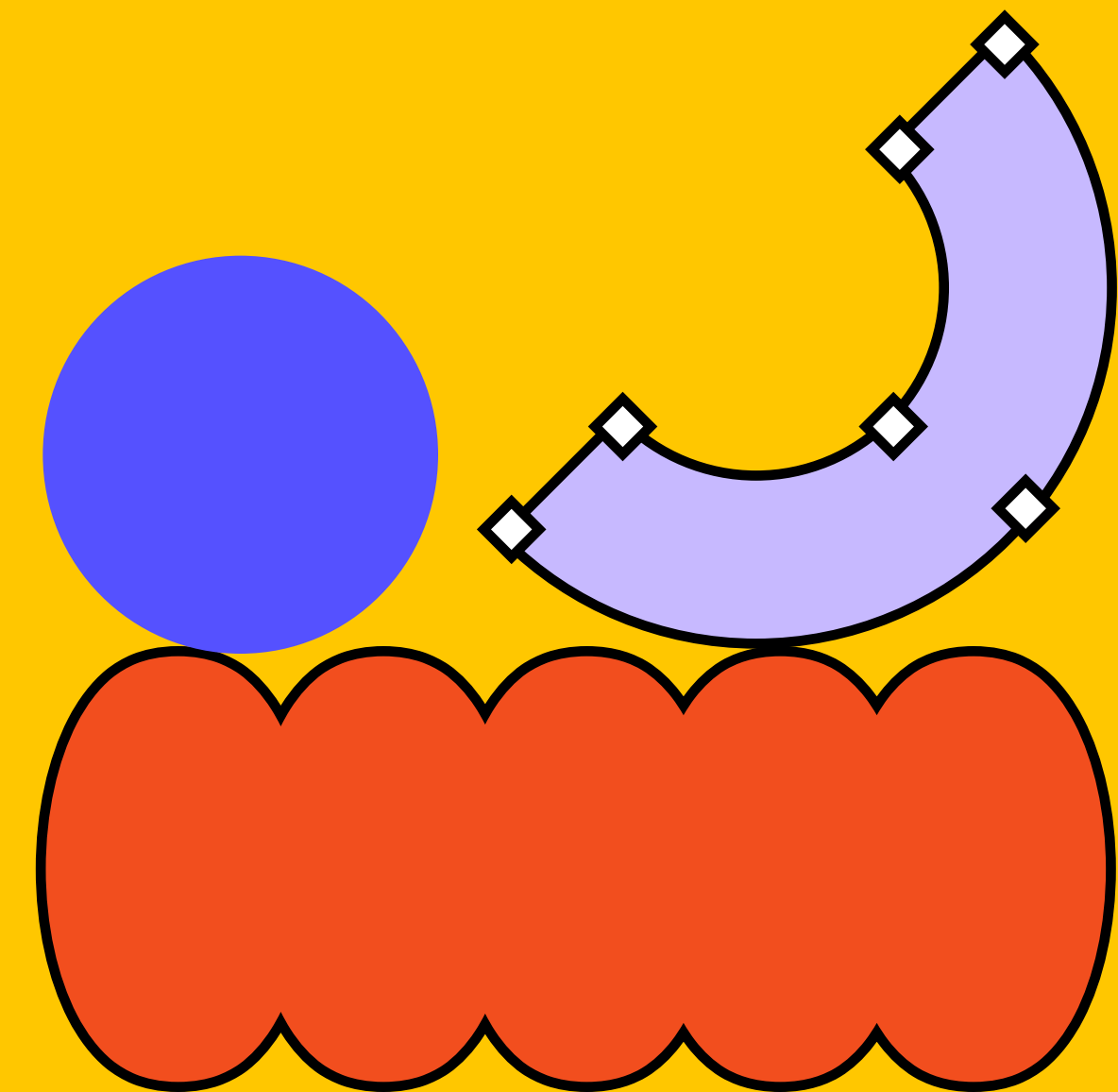
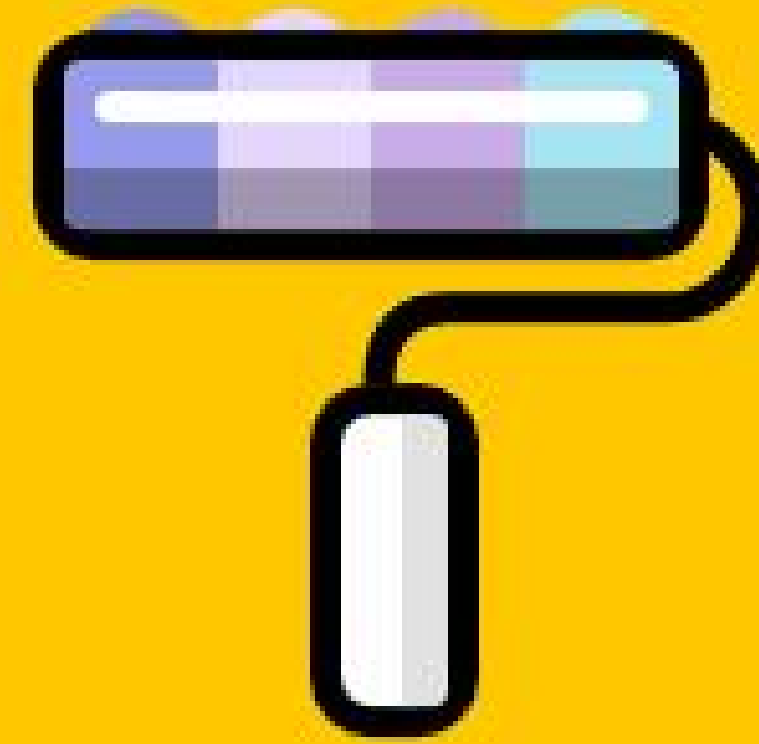
```
$(selector).has(selector);
```

```
$(selector).has(contained);
```

```
$("section").has("p").css("color","silver");
```

```
$("div").has("p").css("color","green");
```

**move & copy elements**





# .append() | .prepend()

用於將指定的內容插入到每個取得元素的 **最後 | 最前面**（作為其子元素）。

```
<ul id="list">  
  <li>item 1</li>  
</ul>
```

```
var newElement = $('<li>appends</li>');  
$('#list').append(newElement);
```

# .before() | .after()

用於在每個匹配的元素 **之前 | 之後** 插入指定的內容。

```
<ul id="list">  
  <li>item 1</li>  
</ul>
```

```
$('#list li').before('<li>段落0</li>');
```

# .appendTo() | .prependTo()

用於將指定的內容插入到每個取得元素的最後 | 最前面作為其子元素。

```
<ul id="list">  
  <li>item 1</li>  
</ul>
```

```
var newElement = $('<li>appends</li>');  
$('#list').append(newElement);
```

# .insertBefore() | .insertAfter()

用於在每個匹配的元素 **之前 | 之後** 插入指定的內容。

```
<ul id="list">  
  <li>item 1</li>  
</ul>
```

```
$('#list li').before('<li>段落0</li>');
```