

Sự kiện trong JQuery

jQuery được thiết kế riêng để phản hồi các sự kiện trong trang HTML

- 1 số sự kiện chung

Mouse Events	Keyboard Events	Form Events	Document/Window Events
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload

Cú pháp sự kiện JQuery

- Cú pháp jQuery cho các phương thức sự kiện
- Trong jQuery, hầu hết các sự kiện DOM đều có một phương thức jQuery tương đương.

Để chỉ định một sự kiện nhấp chuột cho tất cả các đoạn trên một trang, bạn có thể thực hiện điều này:

```
$("p").click();
```

Bước tiếp theo là xác định điều gì sẽ xảy ra khi sự kiện xảy ra. Bạn phải chuyển một hàm cho sự kiện:

```
$("p").click(function(){  
    // action goes here!!  
});
```

Các sự kiện thông dụng

- **`$(document).ready()`: phương thức cho phép ta thực thi một chức năng khi tài liệu được tải đầy đủ.**

```
$(document).ready(function(){
```

```
});
```

Chú ý ta nên để các sự kiện khác trong sự kiện `ready()`

Sự kiện click

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("p").click(function(){
    $(this).hide();
  });
});
</script>
</head>
<body>

<p>Click tôi sẽ biến mất.</p>
<p>Click tôi!!</p>
<p>Click tôi nữa!!!</p>

</body>
</html>
```

Click tôi sẽ biến mất.

Click tôi!!

Click tôi nữa!!!

Mỗi khi click 1 dòng sẽ ẩn đi một dòng

Click tôi sẽ biến mất.

Click tôi!!

Sự kiện dblclick(): double click vào Phần tử

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("p").dblclick(function(){
        $(this).hide();
    });
});
</script>
</head>
<body>

<p>Double Click tôi sẽ biến mất.</p>
<p>Double Click tôi!! </p>
<p>Double Click tôi nữa!!!</p>

</body>
</html>
```

Double Click tôi sẽ biến mất.

Double Click tôi!!

Double Click tôi nữa!!!

Double Click tôi sẽ biến mất.

Double Click tôi!!

Sự kiện mouseenter():

Hàm được thực thi khi con trỏ chuột vào phần tử HTML:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#p1").mouseenter(function(){
    alert("Bạn đang trở tôi!");
  });
  $(".p2").mouseenter(function(){
    alert("Bạn đang trở tôi 1 lần nữa đi!");
  });
});
</script>
</head>
<body>

<p id="p1">trở vào tôi.</p>
<p class="p2">trở vào tôi nữa đi.</p>
</body>
</html>
```

trở vào tôi.

trở vào tôi nữa đi.

Một trang được nhúng trên trang này cho biết

Bạn đang trở tôi 1 lần nữa đi!

OK

trở vào tôi.

trở vào tôi nữa đi.

Sự kiện mouseleave(): Hàm được thực thi khi con trỏ chuột rời khỏi phần tử HTML:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#p1").mouseleave(function(){
    alert("Bạn đang rời trở tôi 1!");
  });
  $(".p2").mouseleave(function(){
    alert("Bạn đang rời trở tôi 2!");
  });
});
</script>
</head>
<body>

<p id="p1">Xin chào các bạn.</p>
<p class="p2">Cảm ơn các bạn.</p>
</body>
</html>
```

Xin chào các bạn.

Cảm ơn các bạn.

Một trang được nhúng trên trang này cho biết
Bạn đang rời trở tôi 2!

OK

Xin chào các bạn.

Cảm ơn các bạn.

Sự kiện mousedown():Hàm được thực thi khi nhấn nút chuột trái, chuột giữa hoặc chuột phải trong khi chuột ở trên phần tử HTML:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#p1").mousedown(function(){
    alert("Chuột được nhấn trên tôi");
  });
});
</script>
</head>
<body>

<p id="p1">Hello all of you.</p>

</body>
</html>
```

Hello all of you.

Một trang được nhúng trên trang này cho biết

Chuột được nhấn trên tôi

OK

Hello all of you.

Sự kiện mouseup(): Hàm được thực thi khi thả nút chuột trái, chuột giữa hoặc chuột phải trong khi chuột ở trên phần tử HTML

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#p1").mouseup(function(){
    alert("Mouse up trên p1!");
  });
});
</script>
</head>
<body>

<p id="p1">Xin chào các bạn.</p>

</body>
</html>
```

Xin chào các bạn.

Một trang được nhúng trên trang này cho biết
Mouse up trên p1!

OK

Xin chào các bạn.

Sự kiện hover():Hàm đầu tiên được thực thi khi chuột vào phần tử HTML và hàm thứ hai được thực thi khi chuột rời khỏi phần tử HTML

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#p1").hover(function(){
    alert("bạn đang trở p1!");
  },
  function(){
    alert("Bye! đang đang rời p1!");
  });
});
</script>
</head>
<body>

<p id="p1">Đoạn văn bản.</p>

</body>
</html>
```

Đoạn văn bản.

Một trang được nhúng trên trang này cho biết
bạn đang trở p1!

OK

Đoạn văn bản.

Một trang được nhúng trên trang này cho biết
Bye! đang đang rời p1!

OK

Đoạn văn bản.

Sự kiện focus():Hàm được thực thi khi trường biểu mẫu có tiêu điểm

Sự kiện blur(): Hàm được thực thi khi trường biểu mẫu mất tiêu điểm:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("input").focus(function(){
    $(this).css("background-color", "yellow");
  });
  $("input").blur(function(){
    $(this).css("background-color", "green");
  });
});
</script>
</head>
<body>

Name: <input type="text" name="fullname"><br>
Email: <input type="text" name="email">

</body>
</html>
```

Name:

Email:

Name:

Email:

Sự kiện on(): Phương thức on () đính kèm một hoặc nhiều trình xử lý sự kiện cho các phần tử đã chọn.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("p").on("click", function(){
    $(this).hide();
  });
});
</script>
</head>
<body>

<p>If you click on me, I will disappear.</p>
<p>Click me away!</p>
<p>Click me too!</p>

</body>
</html>
```

If you click on me, I will disappear.

Click me away!

Click me too!

If you click on me, I will disappear.

Click me away!

Gắn nhiều sự kiện trên sự kiện on()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("p").on({
    mouseenter: function(){
      $(this).css("background-color", "lightgray");
    },
    mouseleave: function(){
      $(this).css("background-color", "lightblue");
    },
    click: function(){
      $(this).css("background-color", "yellow");
    }
  });
});
</script>
</head>
<body>

<p>Click or move the mouse pointer over this paragraph.</p>

</body>
</html>
```

Click or move the mouse pointer over this paragraph.

Click or move the mouse pointer over this paragraph.

Click or move the mouse pointer over this paragraph.

Những hiệu ứng JQuery (Jquery Effects)

jQuery hide() and show()

Cú pháp:

```
$(selector).hide(speed,callback);
```

```
$(selector).show(speed,callback);
```

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#hide").click(function(){
    $("p").hide();
  });
  $("#show").click(function(){
    $("p").show();
  });
});
</script>
</head>
<body>

<p>If you click on the "Hide" button, I will disappear.</p>

<button id="hide">Hide</button>
<button id="show">Show</button>

</body>
</html>
```

If you click on the "Hide" button, I will disappear.

Hide Show

Hide Show

Tham số tốc độ tùy chọn chỉ định tốc độ ẩn / hiển thị và có thể nhận các giá trị sau: "slow", "fast" hoặc milliseconds.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").hide(1000);
    $("b").hide("slow");
    $("i").hide("fast");
  });
});
</script>
</head>
<body>

<button>Hide</button>

<p>Đoạn văn bản.</p>
<b>Chữ đậm.</b><br>
<i>Chữ nghiêng</i>
</body>
</html>
```

Hide

Đoạn văn bản.

Chữ đậm.

Chữ nghiêng

Hide

Đoạn văn bản.

jQuery toggle(): Các phần tử được hiển thị bị ẩn và các phần tử bị ẩn được hiển thị

- Cú pháp:

```
$(selector).toggle(speed, callback);
```

Speed="slow", "fast", or milliseconds.

Ví dụ:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("p").toggle(1000);
        $("b").toggle("slow");
        $("i").toggle("fast");
    });
});
</script>
</head>
<body>

<button>Toggle between hiding and showing the paragraphs</button>

<p>Đoạn 1.</p>
<b>Chữ đậm.</b><br>
<i>Chữ nghiêng </i>

</body>
</html>
```

Toggle between hiding and showing the paragraphs

Đoạn 1.

Chữ đậm.

Chữ nghiêng

Toggle between hiding and showing the paragraphs

Đoạn 1.

jQuery Fading Methods

- Với jQuery, bạn có thể làm mờ một phần tử trong và ngoài tầm nhìn.
- jQuery có các phương thức fade sau:

- `fadeIn()`
- `fadeOut()`
- `fadeToggle()`
- `fadeTo()`

Fadein()

Cú pháp:

`$(selector).fadeIn(speed,callback);`

Speed= "slow", "fast", or milliseconds.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").fadeIn();
    $("#div2").fadeIn("slow");
    $("#div3").fadeIn(3000);
  });
});
</script>
</head>
<body>

<p>Minh họa fadeIn() với các tham số khác nhau.</p>

<button>Click</button><br><br>

<div id="div1" style="width:80px;height:80px;display:none;background-color:red;"></div><br>
<div id="div2" style="width:80px;height:80px;display:none;background-color:green;"></div><br>
<div id="div3" style="width:80px;height:80px;display:none;background-color:blue;"></div>

</body>
</html>
```

Minh họa fadeIn() với các tham số khác nhau.

Click

Minh họa fadeIn() với các tham số khác nhau.

Click



fadeOut(): Phương thức jQuery fadeOut () được sử dụng để làm mờ một phần tử có thể nhìn thấy.

- Cú pháp:

`$(selector).fadeOut(speed, callback);`

Speed= "slow", "fast", or milliseconds.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").fadeOut();
    $("#div2").fadeOut("slow");
    $("#div3").fadeOut(3000);
  });
});
</script>
</head>
<body>

<p>Minh họa fadeOut() với các tham số khác nhau.</p>

<button>Click</button><br><br>

<div id="div1" style="width:80px;height:80px;background-color:red;"></div><br>
<div id="div2" style="width:80px;height:80px;background-color:green;"></div><br>
<div id="div3" style="width:80px;height:80px;background-color:blue;"></div>

</body>
</html>
```

Minh họa fadeOut() với các tham số khác nhau.

Click



Minh họa fadeOut() với các tham số khác nhau.

Click

fadeToggle(): Phương thức jQuery fadeToggle () chuyển đổi giữa các phương thức fadeIn () và fadeOut ().

- Cú pháp:

`$(selector).fadeToggle(speed, callback);`
Speed= "slow", "fast", or milliseconds.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").fadeToggle();
    $("#div2").fadeToggle("slow");
    $("#div3").fadeToggle(3000);
  });
});
</script>
</head>
<body>

<p>Minh họa fadeToggle() với các tham số khác nhau.</p>

<button>Click to fade in/out boxes</button><br><br>

<div id="div1" style="width:80px;height:80px;background-color:red;"></div>
<br>
<div id="div2" style="width:80px;height:80px;background-color:green;"></div>
<br>
<div id="div3" style="width:80px;height:80px;background-color:blue;"></div>

</body>
</html>
```

Minh họa fadeToggle() với các tham số khác nhau.

Click to fade in/out boxes



Minh họa fadeToggle() với các tham số khác nhau.

Click to fade in/out boxes



Minh họa fadeToggle() với các tham số khác nhau.

Click to fade in/out boxes

fadeTo(): Phương thức jQuery fadeTo () cho phép mờ dần đến một độ mờ nhất định (giá trị từ 0 đến 1).

- Cú pháp:

`$(selector).fadeTo(speed, opacity, callback);`

Speed= "slow", "fast", or milliseconds.

Tham số độ mờ bắt buộc trong phương thức fadeTo () chỉ định độ mờ dần đến độ mờ nhất định (giá trị từ 0 đến 1).

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").fadeTo("slow", 0.15);
    $("#div2").fadeTo("slow", 0.4);
    $("#div3").fadeTo("slow", 0.7);
  });
});
</script>
</head>
<body>

<p>Minh họa fadeTo() với các tham số khác nhau.</p>

<button>Click</button><br><br>

<div id="div1" style="width:80px;height:80px;background-color:red;"></div><br>
<div id="div2" style="width:80px;height:80px;background-color:green;"></div><br>
<div id="div3" style="width:80px;height:80px;background-color:blue;"></div>

</body>
</html>
```

Minh họa fadeTo() với các tham số khác nhau.

Click

Minh họa fadeTo() với các tham số khác nhau.

Click



jQuery Sliding Methods

- Với jQuery, bạn có thể tạo hiệu ứng trượt trên các phần tử.
- jQuery có các phương thức slide sau:

- `slideDown()`
- `slideUp()`
- `slideToggle()`

slideDown() :Phương thức jQuery slideDown () được sử dụng để trượt xuống một phần tử.

- Cú pháp:

`$(selector).slideDown(speed, callback);`

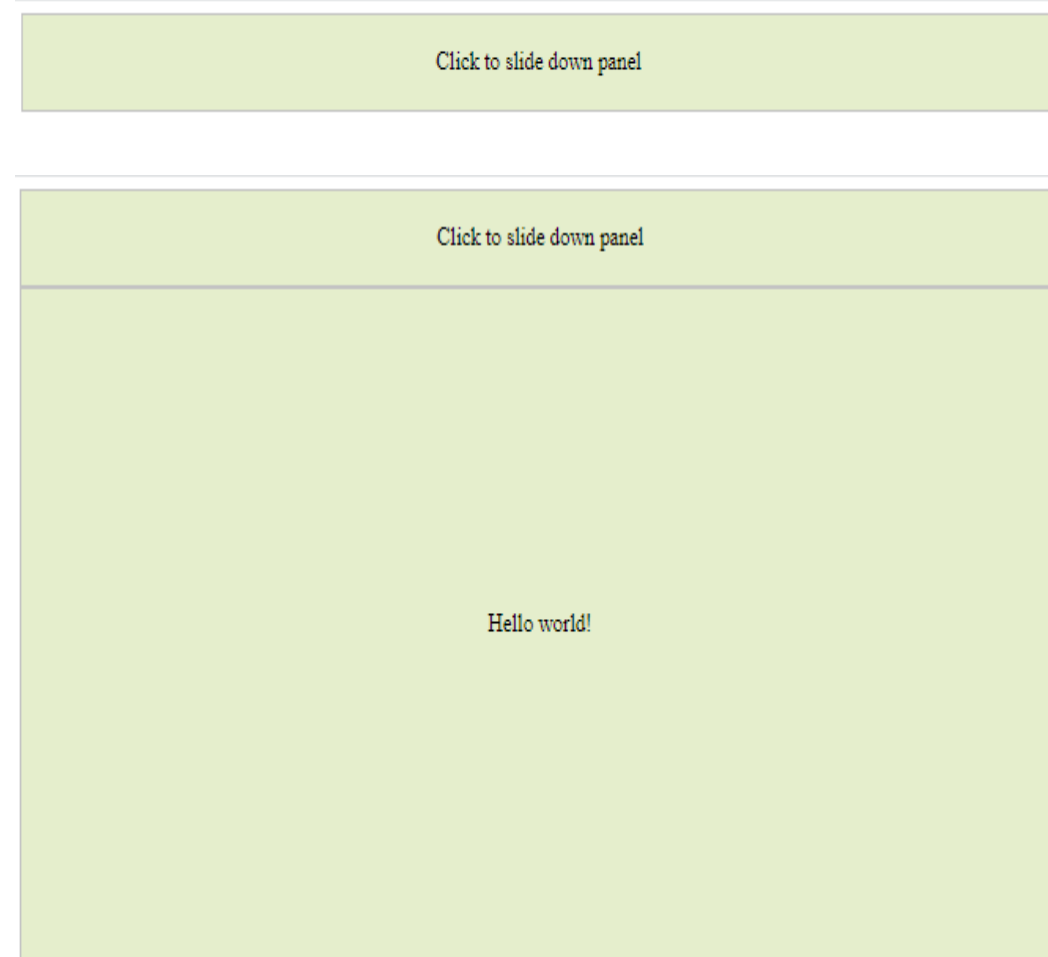
Speed= "slow", "fast", or milliseconds.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#flip").click(function(){
    $("#panel").slideDown("slow");
  });
});
</script>
<style>
#panel, #flip {
  padding: 20px;
  text-align: center;
  background-color: #e5eccc;
  border: solid 1px #c3c3c3;
}

#panel {
  padding: 200px;
  display: none;
}
</style>
</head>
<body>

<div id="flip">Click to slide down panel</div>
<div id="panel">Hello world!</div>

</body>
</html>
```



slideUp() : Phương thức jQuery slideUp () được sử dụng để trượt lên một phần tử.

- Cú pháp:
 - `$(selector).slideUp(speed,callback);`
- Speed= "slow", "fast", or milliseconds

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#flip").click(function(){
    $("#panel").slideUp("slow");
  });
});
</script>
<style>
#panel, #flip {
  padding: 5px;
  text-align: center;
  background-color: #e5e5cc;
  border: solid 1px #c3c3c3;
}

#panel {
  padding: 50px;
}
</style>
</head>
<body>

<div id="flip">Click to slide up panel</div>
<div id="panel">Hello world!</div>

</body>
</html>
```



slideToggle(): Phương thức jQuery slideToggle () chuyển đổi giữa phương thức slideDown () và slideUp ().

- Cú pháp:

`$(selector).slideToggle(speed,callback);`

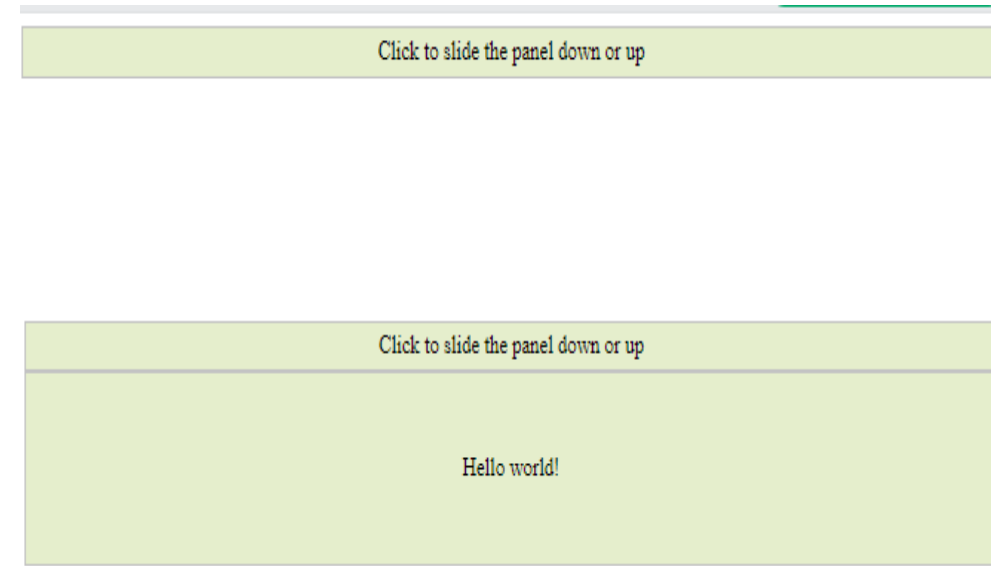
Speed= "slow", "fast", milliseconds

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("#flip").click(function(){
        $("#panel").slideToggle("slow");
    });
});
</script>
<style>
#panel, #flip {
    padding: 5px;
    text-align: center;
    background-color: #e5eccc;
    border: solid 1px #c3c3c3;
}

#panel {
    padding: 50px;
    display: none;
}
</style>
</head>
<body>

<div id="flip">Click to slide the panel down or up</div>
<div id="panel">Hello world!</div>

</body>
</html>
```



jQuery Animations - The animate() Method

Phương thức jQuery animate () được sử dụng để tạo hoạt ảnh tùy chỉnh.

- Cú pháp:

`$(selector).animate({params}, speed, callback);`

Speed="slow", "fast", or milliseconds.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("div").animate({left: '250px'});
  });
});
</script>
</head>
<body>

<button>Start Animation</button>

<p>Theo mặc định, tất cả các phần tử HTML đều có vị trí tĩnh và không thể di chuyển được. Để thao tác vị trí, trước tiên hãy nhớ đặt thuộc tính vị trí CSS của phần tử thành relative, fixed, or absolute!</p>

<div style="background:#98bf21;height:100px;width:100px;position:absolute;"></div>

</body>
</html>
```

Start Animation

Theo mặc định, tất cả các phần tử HTML đều có vị trí tĩnh và không thể di chuyển được. Để thao tác vị trí, trước tiên hãy nhớ đặt thuộc tính vị trí CSS của phần tử thành relative, fixed, or absolute!



Start Animation

Theo mặc định, tất cả các phần tử HTML đều có vị trí tĩnh và không thể di chuyển được. Để thao tác vị trí, trước tiên hãy nhớ đặt thuộc tính vị trí CSS của phần tử thành relative, fixed, or absolute!



jQuery animate(): Thao tác trên nhiều thuộc tính

- Lưu ý rằng nhiều thuộc tính có thể được làm động cùng một lúc:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("div").animate({
      left: '250px',
      opacity: '0.5',
      height: '150px',
      width: '150px'
    });
  });
});
</script>
</head>
<body>

<button>Start Animation</button>

<p>Theo mặc định, tất cả các phần tử HTML đều có vị trí tĩnh và không thể di chuyển được. Để thao
tác vị trí, trước tiên hãy nhớ đặt thuộc tính vị trí CSS của phần tử to relative, fixed, or
absolute!</p>

<div style="background:#98bf21;height:100px;width:100px;position:absolute;"></div>

</body>
</html>
```

Start Animation

Theo mặc định, tất cả các phần tử HTML đều có vị trí tĩnh và không thể di chuyển được. Để thao tác vị trí, trước tiên hãy nhớ đặt thuộc tính vị trí CSS của phần tử to relative, fixed, or absolute!



Start Animation

Theo mặc định, tất cả các phần tử HTML đều có vị trí tĩnh và không thể di chuyển được. Để thao tác vị trí, trước tiên hãy nhớ đặt thuộc tính vị trí CSS của phần tử to relative, fixed, or absolute!



Ghi chú

- Có thể thao tác TẤT CẢ các thuộc tính CSS với phương thức `animate()` không?
- Có gần như! Tuy nhiên, có một điều quan trọng cần nhớ: tất cả các tên thuộc tính phải có vỏ camel khi được sử dụng với phương thức `animate()`:
- Bạn sẽ cần viết `paddingLeft` thay vì `padding-left`, `marginRight` thay vì `margin-right`, v.v.

jQuery animate() -Sử dụng giá trị tương đối

Cũng có thể xác định giá trị tương đối (giá trị sau đó tương đối với giá trị hiện tại của phần tử).

Điều này được thực hiện bằng cách đặt + = hoặc - = trước giá trị:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("div").animate({
      left: '250px',
      height: '+=150px',
      width: '+=150px'
    });
  });
});
</script>
</head>
<body>

<button>Start Animation</button>

<p>Theo mặc định, tất cả các phần tử HTML đều có vị trí tĩnh và không thể di chuyển được. Để thao tác vị trí, trước tiên hãy nhớ đặt thuộc tính vị trí CSS của phần tử thành relative, fixed, or absolute!</p>

<div style="background:#98bf21;height:100px;width:100px;position:absolute;"></div>

</body>
</html>
```

Start Animation

Theo mặc định, tất cả các phần tử HTML đều có vị trí tĩnh và không thể di chuyển được. Để thao tác vị trí, trước tiên hãy nhớ đặt thuộc tính vị trí CSS của phần tử thành relative, fixed, or absolute!



Start Animation

Theo mặc định, tất cả các phần tử HTML đều có vị trí tĩnh và không thể di chuyển được. Để thao tác vị trí, trước tiên hãy nhớ đặt thuộc tính vị trí CSS của phần tử thành relative, fixed, or absolute!



jQuery animate () - Sử dụng các Giá trị được xác định trước

- có thể chỉ định giá trị hoạt ảnh của thuộc tính là "show", "hide", or "toggle"

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("div").animate({
      height: 'toggle'
    });
  });
});
</script>
</head>
<body>
```

<p>Nhấp vào nút nhiều lần để chuyển đổi hoạt ảnh.</p>

<button>Start Animation</button>

<p>Theo mặc định, tất cả các phần tử HTML đều có vị trí tĩnh và không thể di chuyển được. Để thao tác vị trí, trước tiên hãy nhớ đặt thuộc tính vị trí CSS của phần tử thành relative, fixed, or absolute!</p>

<div style="background:#98bf21;height:100px;width:100px;position:absolute;"></div>

</body>

</html>

Nhấp vào nút nhiều lần để chuyển đổi hoạt ảnh.

Start Animation

Theo mặc định, tất cả các phần tử HTML đều có vị trí tĩnh và không thể di chuyển được. Để thao tác vị trí, trước tiên hãy nhớ đặt thuộc tính vị trí CSS của phần tử thành relative, fixed, or absolute!



Nhấp vào nút nhiều lần để chuyển đổi hoạt ảnh.

Start Animation

Theo mặc định, tất cả các phần tử HTML đều có vị trí tĩnh và không thể di chuyển được. Để thao tác vị trí, trước tiên hãy nhớ đặt thuộc tính vị trí CSS của phần tử thành relative, fixed, or absolute!

jQuery animate () - Sử dụng Chức năng Hàng đợi

Theo mặc định, jQuery đi kèm với chức năng hàng đợi cho hoạt ảnh.

Điều này có nghĩa là nếu bạn viết nhiều lệnh gọi animate () sau nhau, jQuery sẽ tạo một hàng đợi "nội bộ" với các lệnh gọi phương thức này. Sau đó, nó chạy các cuộc gọi hoạt ảnh MỘT bởi MỘT.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        var div = $("div");
        div.animate({height: '300px', opacity: '0.4'}, "slow");
        div.animate({width: '300px', opacity: '0.8'}, "slow");
        div.animate({height: '100px', opacity: '0.4'}, "slow");
        div.animate({width: '100px', opacity: '0.8'}, "slow");
    });
});
</script>
</head>
<body>

<button>Start Animation</button>

<p>Theo mặc định, tất cả các phần tử HTML đều có vị trí tĩnh và không thể di chuyển được. Để thao tác vị trí, trước tiên hãy nhớ đặt thuộc tính vị trí CSS của phần tử thành relative, fixed, or absolute!</p>

<div style="background:#98bf21;height:100px;width:100px;position:absolute;"></div>

</body>
</html>
```

Start Animation

Theo mặc định, tất cả các phần tử HTML đều có vị trí tĩnh và không thể di chuyển được. Để thao tác vị trí, trước tiên hãy nhớ đặt thuộc tính vị trí CSS của phần tử thành relative, fixed, or absolute!



Start Animation

Theo mặc định, tất cả các phần tử HTML đều có vị trí tĩnh và không thể di chuyển được. Để thao tác vị trí, trước tiên hãy nhớ đặt thuộc tính vị trí CSS của phần tử thành relative, fixed, or absolute!



Ví dụ khác

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    var div = $("div");
    div.animate({left: '100px'}, "slow");
    div.animate({fontSize: '3em'}, "slow");
  });
});
</script>
</head>
<body>
```

```
<button>Start Animation</button>
```

```
<p>Theo mặc định, tất cả các phần tử HTML đều có vị trí tĩnh và không thể di chuyển được. Để thao tác vị trí, trước tiên hãy nhớ đặt thuộc tính vị trí CSS của phần tử to relative, fixed, or absolute!</p>
```

```
<div style="background:#98bf21;height:100px;width:200px;position:absolute;">HELLO</div>
```

```
</body>
</html>
```

Start Animation

Theo mặc định, tất cả các phần tử HTML đều có vị trí tĩnh và không thể di chuyển được. Để thao tác vị trí, trước tiên hãy nhớ đặt thuộc tính vị trí CSS của phần tử to relative, fixed, or absolute!

HELLO

Start Animation

Theo mặc định, tất cả các phần tử HTML đều có vị trí tĩnh và không thể di chuyển được. Để thao tác vị trí, trước tiên hãy nhớ đặt thuộc tính vị trí CSS của phần tử to relative, fixed, or absolute!

HELLO

jQuery stop()

Phương thức jQuery stop () được sử dụng để dừng hoạt ảnh hoặc hiệu ứng trước khi kết thúc.

Phương thức stop () hoạt động cho tất cả các hàm hiệu ứng jQuery, bao gồm cả hoạt ảnh trượt, mờ dần và tùy chỉnh.

Cú pháp: `$(selector).stop(stopAll, goToEnd);`

Tham số stopAll tùy chọn chỉ định xem hàng đợi hoạt ảnh có được xóa hay không. Mặc định là false, có nghĩa là chỉ hoạt ảnh đang hoạt động sẽ bị dừng lại, cho phép mọi hoạt ảnh được xếp hàng đợi được thực hiện sau đó.

Tham số goToEnd tùy chọn chỉ định có hoàn thành hoạt ảnh hiện tại ngay lập tức hay không. Mặc định là false.

Vì vậy, theo mặc định, phương thức stop () hủy hoạt ảnh hiện tại đang được thực hiện trên phần tử đã chọn.

Ví dụ sự kiện stop

```
<script>
$(document).ready(function(){
    $("#flip").click(function(){
        $("#panel").slideDown(500);
    });
    $("#stop").click(function(){
        $("#panel").stop();
    });
});
</script>
<style>
#panel, #flip {
    padding: 5px;
    font-size: 18px;
    text-align: center;
    background-color: #555;
    color: white;
    border: solid 1px #666;
    border-radius: 3px;
}

#panel {
    padding: 50px;
    display: none;
}
</style>
</head>
<body>

<button id="stop">Stop sliding</button>

<div id="flip">Click to slide down panel</div>
<div id="panel">Chào các bạn!</div>

</body>
```

Stop sliding

Click to slide down panel

Stop sliding

Click to slide down panel

Chào các bạn!

jQuery Callback Functions:

Một hàm gọi lại được thực thi sau khi hiệu ứng hiện tại kết thúc 100%.

Các câu lệnh JavaScript được thực thi từng dòng một. Tuy nhiên, với các hiệu ứng, dòng mã tiếp theo có thể được chạy ngay cả khi hiệu ứng chưa kết thúc. Điều này có thể tạo ra lỗi.

Để ngăn chặn điều này, bạn có thể tạo một hàm gọi lại.

Một hàm gọi lại được thực thi sau khi kết thúc hiệu ứng hiện tại.

Cú pháp: **`$(selector).hide(speed,callback);`**

Ví dụ về callback:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").hide("slow", function(){
      alert("The paragraph is now hidden");
    });
  });
});
</script>
</head>
<body>

<button>Hide</button>

<p>Đây là đoạn nội dung văn bản.</p>

</body>
</html>
```

Hide

Đây là đoạn nội dung văn bản.

Một trang được nhúng trên trang này cho biết

The paragraph is now hidden

OK

Hide

Ví dụ 2 về callback

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").hide(1000);
    alert("Đoạn bây giờ bị ẩn");
  });
});
</script>
</head>
<body>

<button>Hide</button>

<p>Đây là đoạn văn bản.</p>

</body>
</html>
```

Hide

Đây là đoạn văn bản.

Một trang được nhúng trên trang này cho biết

Đoạn bây giờ bị ẩn

OK

Hide

Đây là đoạn văn bản.

jQuery - Chaining

Để xâu chuỗi một hành động, bạn chỉ cần nối hành động đó vào hành động trước đó.

Ví dụ sau chuỗi các phương thức `css()`, `slideUp()` và `slideDown()` lại với nhau. Phần tử "p1" đầu tiên chuyển thành màu đỏ, sau đó nó trượt lên, rồi trượt xuống:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#p1").css("color", "red").slideUp(2000).slideDown(2000);
  });
});
</script>
</head>
<body>

<p id="p1">Tuyệt với!!</p>

<button>Click </button>

</body>
</html>
```

Tuyệt với!!

Click

Tuyệt với!!

Click

jQuery HTML

- Lấy nội dung và thuộc tính của HTML
- Một phần rất quan trọng của jQuery là khả năng thao tác DOM.
- jQuery đi kèm với một loạt các phương thức liên quan đến DOM giúp dễ dàng truy cập và thao tác các phần tử và thuộc tính.

DOM = Document Object Model

Lấy nội dung sử dụng các hàm - text(), html(), and val()

Có 3 phương thức jQuery đơn giản nhưng hữu ích để thao tác DOM là:

- text () - Đặt hoặc trả về nội dung văn bản của các phần tử đã chọn.
- html () - Đặt hoặc trả về nội dung của các phần tử đã chọn (bao gồm cả đánh dấu HTML)
- val () - Đặt hoặc trả về giá trị của các trường biểu mẫu

Ví dụ sau minh họa cách lấy nội dung bằng phương thức jQuery text () và html ():

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    alert("Text: " + $("#test").text());
  });
  $("#btn2").click(function(){
    alert("HTML: " + $("#test").html());
  });
});
</script>
</head>
<body>

<p id="test">Đây là đoạn <b>Đậm</b> trong văn bản.</p>

<button id="btn1">Show Văn bản</button>
<button id="btn2">Show HTML</button>

</body>
</html>
```

Đây là đoạn **Đậm** trong văn bản.

Show Văn bản

Show HTML

Một trang được nhúng trên trang này cho biết

Text: Đây là đoạn **Đậm** trong văn bản.

OK

Đây là đoạn **Đậm** trong văn bản.

Show Văn bản

Show HTML

Một trang được nhúng trên trang này cho biết

HTML: Đây là đoạn Đậm trong văn bản.

OK

Đây là đoạn **Đậm** trong văn bản.

Show Văn bản

Show HTML

Ví dụ: Sử dụng val() lấy giá trị

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        alert("Value: " + $("#test").val());
    });
});
</script>
</head>
<body>

<p>Name: <input type="text" id="test" value="Viên Thanh Nhã"></p>

<button>Show giá trị</button>

</body>
</html>
```

Name:

Show giá trị

Một trang được nhúng trên trang này cho biết

Value: Viên Thanh Nhã

OK

Name:

Show giá trị

Ví dụ: sử dụng attr() lấy giá trị thuộc tính

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        alert($("#nhagroup").attr("href"));
    });
});
</script>
</head>
<body>

<p><a href="https://www.viennhagroup.com" id="nhagroup">viennhagroup.com</a></p>

<button>Show href Value</button>

</body>
</html>
```

[viennhagroup.com](https://www.viennhagroup.com)

Show href Value

Một trang được nhúng trên trang này cho biết

<https://www.viennhagroup.com>

OK

[viennhagroup.com](https://www.viennhagroup.com)

Show href Value

jQuery - thiết lập nội dung và thuộc tính

- Đặt Nội dung - text (), html () và val ()
- Chúng tôi sẽ sử dụng ba phương pháp tương tự từ trang trước để thiết lập nội dung:
 - text () - Đặt hoặc trả về nội dung văn bản của các phần tử đã chọn
 - html () - Đặt hoặc trả về nội dung của các phần tử đã chọn (bao gồm cả đánh dấu HTML)
 - val () - Đặt hoặc trả về giá trị của các trường biểu mẫu

Ví dụ đặt nội dung text(), html() và val()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("#btn1").click(function(){
        $("#test1").text("Xin chào!");
    });
    $("#btn2").click(function(){
        $("#test2").html("<b>Xin chào bạn!</b>");
    });
    $("#btn3").click(function(){
        $("#test3").val("Chính là tôi");
    });
});
</script>
</head>
<body>

<p id="test1">Đoạn 1.</p>
<p id="test2">Đoạn 2.</p>

<p>Input field: <input type="text" id="test3" value="Viên Thanh Nhã"></p>

<button id="btn1">Set Text</button>
<button id="btn2">Set HTML</button>
<button id="btn3">Set Value</button>

</body>
</html>
```

Đoạn 1.

Đoạn 2.

Input field: Viên Thanh Nhã

Set Text

Set HTML

Set Value

Xin chào!

Xin chào bạn!

Input field: Chính là tôi

Set Text

Set HTML

Set Value

Sử dụng Callback cho hàm text(), html(), and val()

- Tất cả ba phương thức jQuery ở trên: text (), html () và val (), cũng đi kèm với một hàm gọi lại.
- Hàm callback có hai tham số: chỉ số của phần tử hiện tại trong danh sách các phần tử được chọn và giá trị ban đầu (cũ).
- Sau đó, bạn trả về chuỗi mà bạn muốn sử dụng làm giá trị mới từ hàm.

Ví dụ: sử dụng callback()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    $("#test1").text(function(i, origText){
      return "Old text: " + origText + " Văn bản mới: Chào bạn! (index: " + i + ")";
    });
  });

  $("#btn2").click(function(){
    $("#test2").html(function(i, origText){
      return "Old html: " + origText + " Văn bản mới: Chào bạn <b>yêu dấu!</b> (index: " + i + ")";
    });
  });
});
</script>
</head>
<body>

<p id="test1">Đoạn văn bản <b>đậm</b> trong đoạn.</p>
<p id="test2">Đoạn văn bản khác <b>đậm</b> trong đoạn.</p>

<button id="btn1">Show cũ/mới Text</button>
<button id="btn2">Show cũ/mới HTML</button>

</body>
</html>
```

Đoạn văn bản **đậm** trong đoạn.

Đoạn văn bản khác **đậm** trong đoạn.

Show cũ/mới Text

Show cũ/mới HTML

Old text: Đoạn văn bản đậm trong đoạn. Văn bản mới: Chào bạn! (index: 0)

Old html: Đoạn văn bản khác **đậm** trong đoạn. Văn bản mới: Chào bạn **yêu dấu!** (index: 0)

Show cũ/mới Text

Show cũ/mới HTML

Đặt thuộc tính- attr()

Phương thức jQuery attr () cũng được sử dụng để thiết lập / thay đổi các giá trị thuộc tính.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("#w3s").attr("href", "https://viennhagroup.com/administrator.html");
    });
});
</script>
</head>
<body>

<p><a href="https://www.viennhagroup.com.com" id="w3s">viennhagroup.com</a></p>

<button>Thay đổi giá trị href </button>

</body>
</html>
```

viennhagroup.com

Thay đổi giá trị href

Phương thức attr () cũng cho phép bạn thiết lập nhiều thuộc tính cùng một lúc.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#nha").attr({
      "href" : "https://www.viennhagroup.com/index.html",
      "title" : "Nha Group"
    });
  });
});
</script>
</head>
<body>

<p><a href="https://www.viennhagroup.com" title="some title" id="nha">viennhagroup.com</a></p>

<button>Thay đổi href và title của trang web</button>

</body>
</html>
```

[viennhagroup.com](https://www.viennhagroup.com)

Thay đổi href và title của trang web

Hàm Callback cho thuộc tính attr()

- Phương thức jQuery attr (), cũng đi kèm với một hàm gọi lại.
- Hàm callback có hai tham số: chỉ số của phần tử hiện tại trong danh sách các phần tử được chọn và giá trị thuộc tính ban đầu (cũ).
- Sau đó, bạn trả về chuỗi mà bạn muốn sử dụng làm giá trị thuộc tính mới từ hàm.

Ví dụ callback thay đổi link

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#nha").attr("href", function(i, origValue){
      return origValue + "/administrator.html";
    });
  });
});
</script>
</head>
<body>

<p><a href="https://www.viennhagroup.com" id="nha">viennhagroup.com</a></p>

<button>Thay đổi giá trị href</button>

</body>
</html>
```

[viennhagroup.com](https://www.viennhagroup.com)

Thay đổi giá trị href

jQuery - Add Elements(Thêm các phần tử)

- Thêm nội dung HTML mới
- Xem xét bốn phương thức jQuery được sử dụng để thêm nội dung mới:
 - `append()` - Chèn nội dung vào cuối các phần tử đã chọn
 - `prepend()` - Chèn nội dung vào đầu các phần tử đã chọn
 - `after()` - Chèn nội dung sau các phần tử đã chọn
 - `before()` - Chèn nội dung trước các phần tử đã chọn

jQuery append()

Phương thức jQuery append () chèn nội dung VÀO CUỐI của các phần tử HTML đã chọn.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    $("#p").append(" <b>Appended text</b>.");
  });

  $("#btn2").click(function(){
    $("#ol").append("<li>Appended item</li>");
  });
});
</script>
</head>
<body>

<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

<ol>
  <li>List item 1</li>
  <li>List item 2</li>
  <li>List item 3</li>
</ol>

<button id="btn1">Append text</button>
<button id="btn2">Append list items</button>

</body>
</html>
```

This is a paragraph.

This is another paragraph.

1. List item 1
2. List item 2
3. List item 3

Append text

Append list items

This is a paragraph. **Appended text.**

This is another paragraph. **Appended text.**

1. List item 1
2. List item 2
3. List item 3
4. Appended item

Append text

Append list items

jQuery prepend()

Phương thức jQuery prepend () chèn nội dung BUỐC ĐẦU của các phần tử HTML đã chọn.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    $("p").prepend("<b>Prepended text</b>. ");
  });
  $("#btn2").click(function(){
    $("ol").prepend("<li>Prepended item</li>");
  });
});
</script>
</head>
<body>

<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

<ol>
  <li>List item 1</li>
  <li>List item 2</li>
  <li>List item 3</li>
</ol>

<button id="btn1">Prepend text</button>
<button id="btn2">Prepend list item</button>

</body>
</html>
```

This is a paragraph.

This is another paragraph.

1. List item 1
2. List item 2
3. List item 3

Prepend text

Prepend list item

Prepended text. This is a paragraph.

Prepended text. This is another paragraph.

1. Prepended item
2. List item 1
3. List item 2
4. List item 3

Prepend text

Prepend list item

Thêm nhiều phần tử mới với append () và prepend ()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
function appendText() {
    var txt1 = "<p>văn bản 1.</p>";           // Create text with HTML
    var txt2 = $("<p></p>").text("văn bản 2 ."); // Create text with jQuery
    var txt3 = document.createElement("p");
    txt3.innerHTML = "văn bản 3.";           // Create text with DOM
    $("body").append(txt1, txt2, txt3);      // Append new elements
}
</script>
</head>
<body>

<p>Đây là văn bản.</p>
<button onclick="appendText()">Thêm văn bản</button>

</body>
</html>
```

Đây là văn bản.

Thêm văn bản

Đây là văn bản.

Thêm văn bản

văn bản 1.

văn bản 2 .

văn bản 3.

jQuery after() and before()

Phương thức jQuery after () chèn nội dung SAU các phần tử HTML đã chọn.

Phương thức jQuery before () chèn nội dung TRƯỚC các phần tử HTML đã chọn.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function() {
  $("#btn1").click(function() {
    $("#img").before("<b>Before</b>");
  });

  $("#btn2").click(function() {
    $("#img").after("<i>After</i>");
  });
});
</script>
</head>
<body>

<br><br>

<button id="btn1">Thêm vào trước</button>
<button id="btn2">Thêm vào sau</button>

</body>
</html>
```



Thêm vào trước

Thêm vào sau



Before

After

Thêm vào trước

Thêm vào sau

Thêm nhiều phần tử mới với after() và before()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
function afterText() {
    var txt1 = "<b>I </b>";           // Create element with HTML
    var txt2 = $("<i></i>").text("love "); // Create with jQuery
    var txt3 = document.createElement("b"); // Create with DOM
    txt3.innerHTML = "jQuery!";
    $("img").after(txt1, txt2, txt3); // Insert new elements after img
}
</script>
</head>
<body>



<p>Click the button to insert text after the image.</p>

<button onclick="afterText()">Insert after</button>

</body>
</html>
```



Click the button to insert text after the image.

Insert after



I love jQuery! I love jQuery! I love jQuery! I love jQuery! I love jQuery! I love jQuery! I love jQuery! I love jQuery! I love jQuery!

Click the button to insert text after the image.

Insert after

jQuery - Remove(): xóa các phần tử Xóa phần tử / nội dung

- Để xóa các phần tử và nội dung, chủ yếu có hai phương thức jQuery:
 - `remove ()` - Loại bỏ phần tử đã chọn (và các phần tử con của nó)
 - `empty ()` - Loại bỏ các phần tử con khỏi phần tử đã chọn

Ví dụ: remove()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("#div1").remove();
    });
});
</script>
</head>
<body>

<div id="div1" style="height:100px;width:300px;border:1px solid black;background-color:yellow;">

một số văn bản trong thẻ div.
<p>Đoạn 1 trong thẻ div.</p>
<p>Đoạn 2 trong thẻ div.</p>

</div>
<br>

<button>Xóa phần tử trong thẻ div</button>

</body>
</html>
```

một số văn bản trong thẻ div.

Đoạn 1 trong thẻ div.

Đoạn 2 trong thẻ div.

Xóa phần tử trong thẻ div

Xóa phần tử trong thẻ div

Ví dụ: empty()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("#div1").empty();
    });
});
</script>
</head>
<body>

<div id="div1" style="height:100px;width:300px;border:1px solid black;background-color:yellow;">

một số văn bản trong thẻ div.
<p>Đoạn 1 trong thẻ div.</p>
<p>Đoạn 2 trong thẻ div.</p>

</div>
<br>

<button>Xóa phần tử trong thẻ div</button>

</body>
</html>
```

một số văn bản trong thẻ div.

Đoạn 1 trong thẻ div.

Đoạn 2 trong thẻ div.

Xóa phần tử trong thẻ div

Xóa phần tử trong thẻ div

Lọc các phần tử cần loại bỏ

Phương thức jQuery remove () cũng chấp nhận một tham số, cho phép bạn lọc các phần tử cần loại bỏ.

Tham số có thể là bất kỳ cú pháp jQuery selector nào.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("p").remove(".test");
    });
});
</script>
<style>
.test {
    color: red;
    font-size: 20px;
}
</style>
</head>
<body>

<p>This is a paragraph.</p>
<p class="test">This is another paragraph.</p>
<p class="test">This is another paragraph.</p>

<button>Remove all p elements with class="test"</button>

</body>
</html>
```

This is a paragraph.

This is another paragraph.

This is another paragraph.

Remove all p elements with class="test"

This is a paragraph.

Remove all p elements with class="test"

Remove nhiều phần tử cùng lúc

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("p").remove(".test, .demo");
    });
});
</script>
<style>
.test {
    color: red;
    font-size: 20px;
}

.demo {
    color: green;
    font-size: 25px;
}
</style>
</head>
<body>

<p>This is a paragraph.</p>
<p class="test">This is p element with class="test".</p>
<p class="test">This is p element with class="test".</p>
<p class="demo">This is p element with class="demo".</p>

<button>Remove all p elements with class="test" and class="demo"</button>

</body>
</html>
```

This is a paragraph.

This is p element with class="test".

This is p element with class="test".

This is p element with class="demo".

Remove all p elements with class="test" and class="demo"

This is a paragraph.

Remove all p elements with class="test" and class="demo"

jQuery - lấy và thiết lập lớp CSS

jQuery Thao tác CSS

- jQuery có một số phương thức để thao tác CSS:
 - `addClass ()` - Thêm một hoặc nhiều lớp vào các phần tử đã chọn
 - `removeClass ()` - Xóa một hoặc nhiều lớp khỏi các phần tử đã chọn
 - `toggleClass ()` - Chuyển đổi giữa việc thêm / xóa các lớp khỏi các phần tử đã chọn
 - `css ()` - Đặt hoặc trả về thuộc tính style

jQuery addClass()

Thêm thuộc tính lớp vào các phần tử khác nhau.

```
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("h1, h2, p").addClass("blue");
    $("div").addClass("important");
  });
});
</script>
<style>
.important {
  font-weight: bold;
  font-size: xx-large;
}

.blue {
  color: blue;
}
</style>
</head>
<body>

<h1>Heading 1</h1>
<h2>Heading 2</h2>

<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

<div>This is some important text!</div><br>

<button>Add classes to elements</button>

</body>
</html>
```

Heading 1

Heading 2

This is a paragraph.

This is another paragraph.

This is some important text!

Add classes to elements

Heading 1

Heading 2

This is a paragraph.

This is another paragraph.

This is some important text!

Add classes to elements

jQuery addClass()

Thêm nhiều thuộc tính lớp vào các phần tử khác nhau.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").addClass("important blue");
  });
});
</script>
<style>
.important {
  font-weight: bold;
  font-size: xx-large;
}

.blue {
  color: blue;
}
</style>
</head>
<body>

<div id="div1">This is some text.</div>
<div id="div2">This is some text.</div>
<br>

<button>Add classes to first div element</button>

</body>
</html>
```

This is some text.

This is some text.

Add classes to first div element

This is some text.

This is some text.

Add classes to first div element

jQuery removeClass()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("h1, h2, p").removeClass("blue");
    });
});
</script>
<style>
.blue {
    color: blue;
}
</style>
</head>
<body>

<h1 class="blue">Heading 1</h1>
<h2 class="blue">Heading 2</h2>

<p class="blue">This is a paragraph.</p>
<p>This is another paragraph.</p>

<button>Remove class from elements</button>

</body>
</html>
```

Heading 1

Heading 2

This is a paragraph.

This is another paragraph.

Remove class from elements

Heading 1

Heading 2

This is a paragraph.

This is another paragraph.

Remove class from elements

jQuery toggleClass().

Phương thức này chuyển đổi giữa việc thêm / xóa các lớp khỏi các phần tử đã chọn

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("h1, h2, p").toggleClass("blue");
  });
});
</script>
<style>
.blue {
  color: blue;
}
</style>
</head>
<body>

<h1>Heading 1</h1>
<h2>Heading 2</h2>

<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

<button>Toggle class</button>

</body>
</html>
```

Heading 1

Heading 2

This is a paragraph.

This is another paragraph.

Toggle class

Heading 1

Heading 2

This is a paragraph.

This is another paragraph.

Toggle class

jQuery css()

Phương thức css () đặt hoặc trả về một hoặc nhiều thuộc tính kiểu cho các phần tử đã chọn.

Trả về thuộc tính css

Cú pháp:

css("propertyname");

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        alert("Background color = " + $("p").css("background-color"));
    });
});
</script>
</head>
<body>

<h2>This is a heading</h2>

<p style="background-color:#ff0000">This is a paragraph.</p>
<p style="background-color:#00ff00">This is a paragraph.</p>
<p style="background-color:#0000ff">This is a paragraph.</p>

<button>Return background-color of p</button>

</body>
</html>
```

This is a heading

This is a paragraph.

This is a paragraph.

This is a paragraph.

Return background-color of p

Một trang được nhúng trên trang này cho biết

Background color = rgb(255, 0, 0)

OK

This is a heading

This is a paragraph.

This is a paragraph.

This is a paragraph.

Return background-color of p

Thiết lập thuộc tính CSS

- Cú pháp:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").css("background-color", "yellow");
  });
});
</script>
</head>
<body>

<h2>This is a heading</h2>

<p style="background-color:#ff0000">This is a paragraph.</p>
<p style="background-color:#00ff00">This is a paragraph.</p>
<p style="background-color:#0000ff">This is a paragraph.</p>

<p>This is a paragraph.</p>

<button>Set background-color of p</button>

</body>
</html>
```

This is a heading

This is a paragraph.

This is a paragraph.

This is a paragraph.

This is a paragraph.

Set background-color of p

This is a heading

This is a paragraph.

This is a paragraph.

This is a paragraph.

This is a paragraph.

Set background-color of p

Thiết lập nhiều thuộc tính CSS

- Cú pháp:
 - `css({"propertyname":"value","propertyname":"value",...});`

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").css({"background-color": "yellow", "font-size": "200%"});
  });
});
</script>
</head>
<body>

<h2>This is a heading</h2>

<p style="background-color:#ff0000">This is a paragraph.</p>
<p style="background-color:#00ff00">This is a paragraph.</p>
<p style="background-color:#0000ff">This is a paragraph.</p>

<p>This is a paragraph.</p>

<button>Set multiple styles for p</button>

</body>
</html>
```

This is a heading

This is a paragraph.

This is a paragraph.

This is a paragraph.

This is a paragraph.

Set multiple styles for p

This is a heading

This is a paragraph.

This is a paragraph.

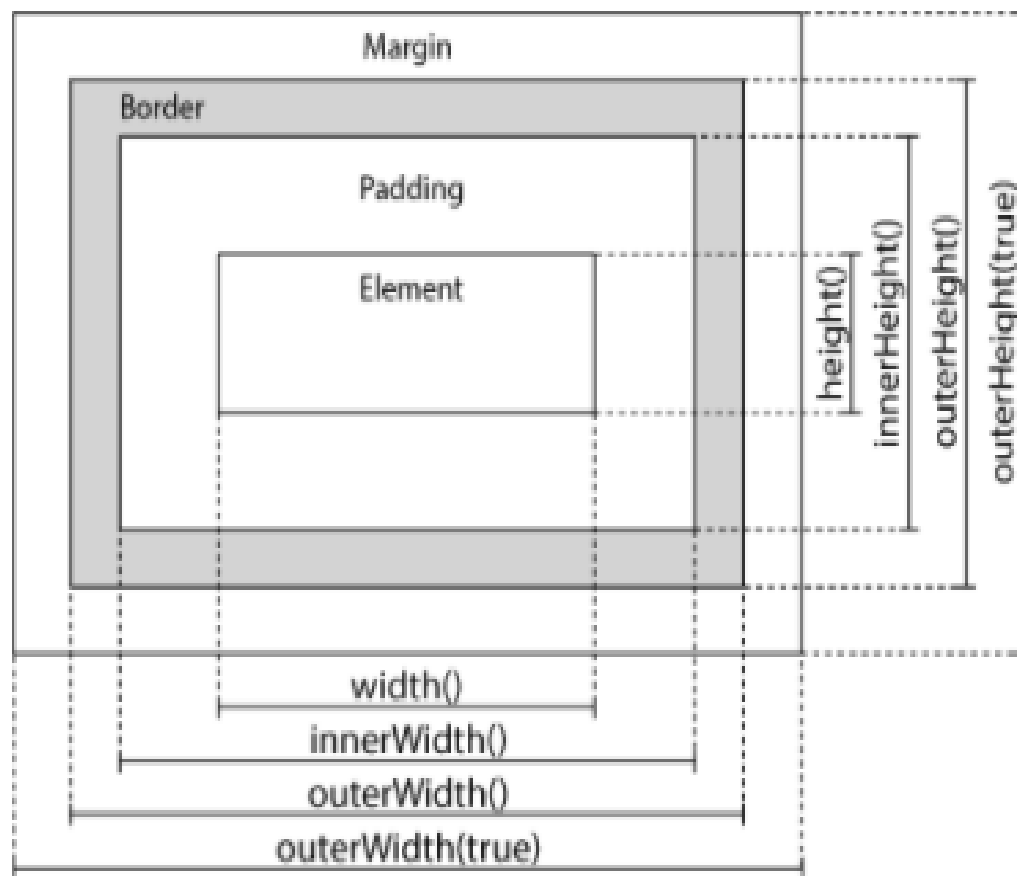
This is a paragraph.

This is a paragraph.

Set multiple styles for p

jQuery - Dimensions.

jQuery có một số phương thức quan trọng để làm việc với các kích thước:



jQuery width() and height().

Phương thức width () đặt hoặc trả về chiều rộng của một phần tử (không bao gồm phần đệm, đường viền và lề).

Phương thức height () đặt hoặc trả về chiều cao của một phần tử (không bao gồm phần đệm, đường viền và lề).

Ví dụ:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    var txt = "";
    txt += "Width of div: " + $("#div1").width() + "<br>";
    txt += "Height of div: " + $("#div1").height();
    $("#div1").html(txt);
  });
});
</script>
<style>
#div1 {
  height: 100px;
  width: 300px;
  padding: 10px;
  margin: 3px;
  border: 1px solid blue;
  background-color: lightblue;
}
</style>
</head>
<body>

<div id="div1"></div>
<br>

<button>Display dimensions of div</button>

<p>width() - returns the width of an element.</p>
<p>height() - returns the height of an element.</p>

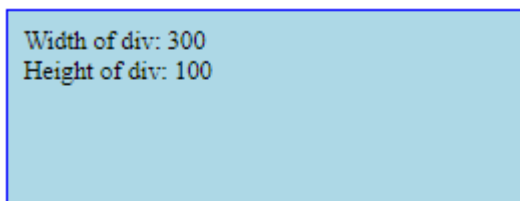
</body>
</html>
```



Display dimensions of div

width() - returns the width of an element.

height() - returns the height of an element.



Display dimensions of div

width() - returns the width of an element.

height() - returns the height of an element.

jQuery innerWidth() and innerHeight()

Phương thức innerWidth () trả về chiều rộng của một phần tử (bao gồm padding).

Phương thức innerHeight () trả về chiều cao của một phần tử (bao gồm cả phần đệm).

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        var txt = "";
        txt += "Width of div: " + $("#div1").width() + "<br>";
        txt += "Height of div: " + $("#div1").height() + "<br>";
        txt += "Inner width of div: " + $("#div1").innerWidth() + "<br>";
        txt += "Inner height of div: " + $("#div1").innerHeight();
        $("#div1").html(txt);
    });
});
</script>
</head>
<style>
#div1 {
    height: 100px;
    width: 300px;
    padding: 10px;
    margin: 3px;
    border: 1px solid blue;
    background-color: lightblue;
}
</style>
<body>

<div id="div1"></div>
<br>

<button>Display dimensions of div</button>

<p>innerWidth() - returns the width of an element (includes padding).</p>
<p>innerHeight() - returns the height of an element (includes padding).</p>

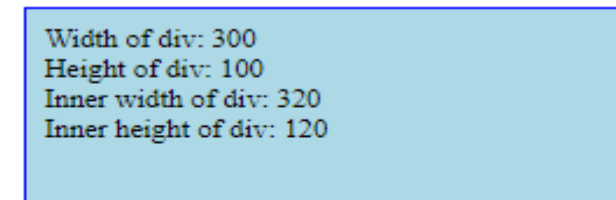
</body>
</html>
```



Display dimensions of div

innerWidth() - returns the width of an element (includes padding).

innerHeight() - returns the height of an element (includes padding).



Display dimensions of div

innerWidth() - returns the width of an element (includes padding).

innerHeight() - returns the height of an element (includes padding).

jQuery `outerWidth()` and `outerHeight()`.

Phương thức `externalWidth ()` trả về chiều rộng của một phần tử (bao gồm phần đệm và đường viền). Phương thức `externalHeight ()` trả về chiều cao của một phần tử (bao gồm phần đệm và đường viền).

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    var txt = "";
    txt += "Width of div: " + $("#div1").width() + "<br>";
    txt += "Height of div: " + $("#div1").height() + "<br>";
    txt += "Outer width of div: " + $("#div1").outerWidth() + "<br>";
    txt += "Outer height of div: " + $("#div1").outerHeight();
    $("#div1").html(txt);
  });
});
</script>
<style>
#div1 {
  height: 100px;
  width: 300px;
  padding: 10px;
  margin: 3px;
  border: 1px solid blue;
  background-color: lightblue;
}
</style>
</head>
<body>

<div id="div1"></div>
<br>

<button>Display dimensions of div</button>

<p>outerWidth() - returns the width of an element (includes padding and border).</p>
<p>outerHeight() - returns the height of an element (includes padding and border).</p>

</body>
</html>
```



Display dimensions of div

`outerWidth()` - returns the width of an element (includes padding and border).

`outerHeight()` - returns the height of an element (includes padding and border).

Width of div: 300
Height of div: 100
Outer width of div: 322.5
Outer height of div: 122.5

Display dimensions of div

`outerWidth()` - returns the width of an element (includes padding and border).

`outerHeight()` - returns the height of an element (includes padding and border).

Phương thức `externalWidth(true)` trả về chiều rộng của một phần tử (bao gồm padding, border và margin).

Phương thức `externalHeight(true)` trả về chiều cao của một phần tử (bao gồm phần đệm, đường viền và lề).

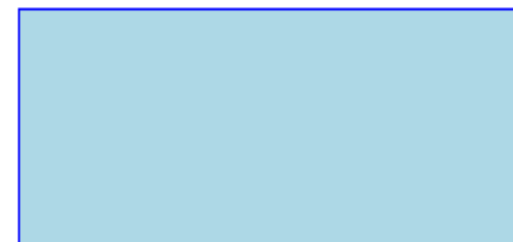
```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#button").click(function(){
    var txt = "";
    txt += "Width of div: " + $("#div1").width() + "<br>";
    txt += "Height of div: " + $("#div1").height() + "<br>";
    txt += "Outer width of div (margin included): " + $("#div1").outerWidth(true) + "<br>";
    txt += "Outer height of div (margin included): " + $("#div1").outerHeight(true);
    $("#div1").html(txt);
  });
});
</script>
<style>
#div1 {
  height: 100px;
  width: 300px;
  padding: 10px;
  margin: 3px;
  border: 1px solid blue;
  background-color: lightblue;
}
</style>
</head>
<body>

<div id="div1"></div>
<br>

<button>Display dimensions of div</button>

<p>outerWidth(true) - returns the width of an element (includes padding, border, and margin).</p>
<p>outerHeight(true) - returns the height of an element (includes padding, border, and margin).</p>

</body>
</html>
```



Display dimensions of div

`outerWidth(true)` - returns the width of an element (includes padding, border, and margin).

`outerHeight(true)` - returns the height of an element (includes padding, border, and margin).

Width of div: 300
Height of div: 100
Outer width of div (margin included): 328.5
Outer height of div (margin included): 128.5

Display dimensions of div

`outerWidth(true)` - returns the width of an element (includes padding, border, and margin).

`outerHeight(true)` - returns the height of an element (includes padding, border, and margin).

jQuery Thêm chiều rộng () và chiều cao ()

Ví dụ sau trả về chiều rộng và chiều cao của tài liệu (tài liệu HTML) và cửa sổ (chế độ xem của trình duyệt):

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        var txt = "";
        txt += "Document width/height: " + $(document).width();
        txt += "x" + $(document).height() + "\n";
        txt += "Window width/height: " + $(window).width();
        txt += "x" + $(window).height();
        alert(txt);
    });
});
</script>
</head>
<body>

<button>Display dimensions of document and window</button>

</body>
</html>
```

Display dimensions of document and window

Một trang được nhúng trên trang này cho biết

Document width/height: 839x666

Window width/height: 839x666

OK

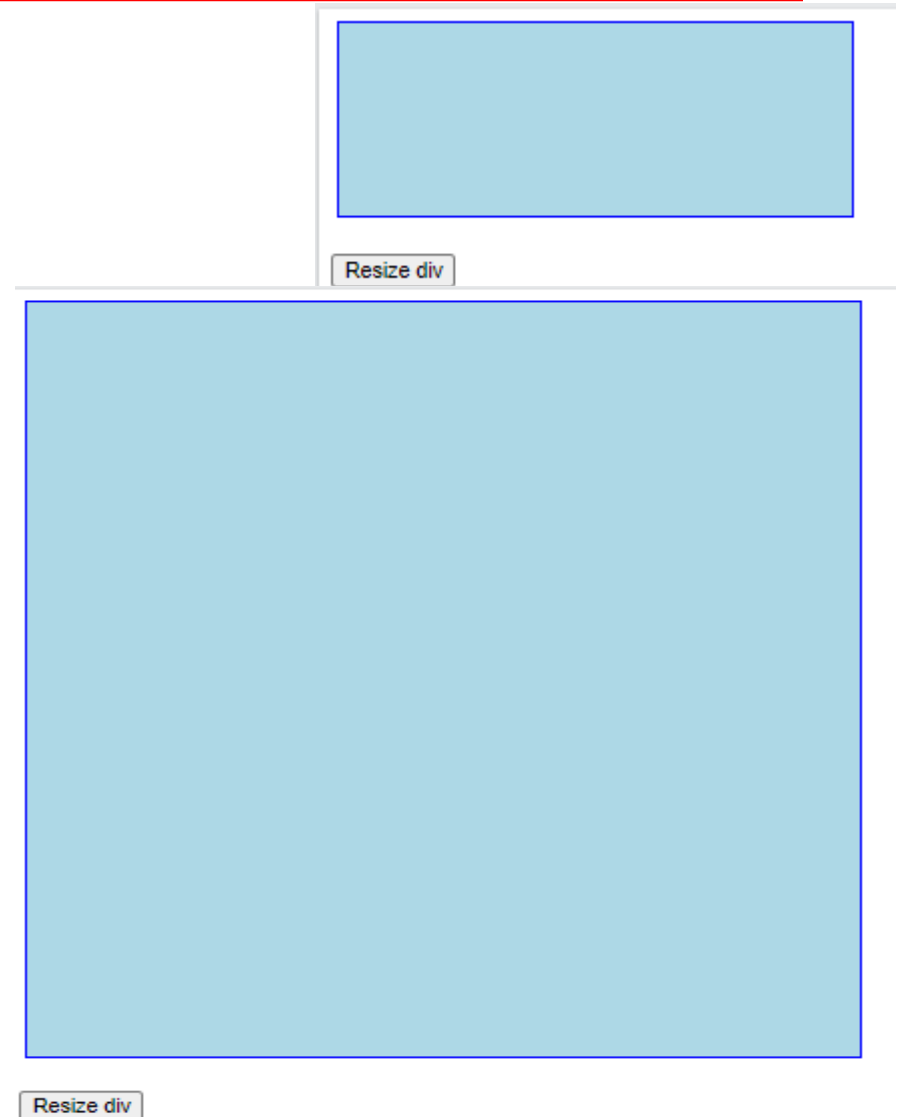
Ví dụ: đặt chiều rộng và chiều cao của phần tử `<div>` được chỉ định:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").width(500).height(500);
  });
});
</script>
<style>
#div1 {
  height: 100px;
  width: 300px;
  padding: 10px;
  margin: 3px;
  border: 1px solid blue;
  background-color: lightblue;
}
</style>
</head>
<body>

<div id="div1"></div>
<br>

<button>Resize div</button>

</body>
</html>
```



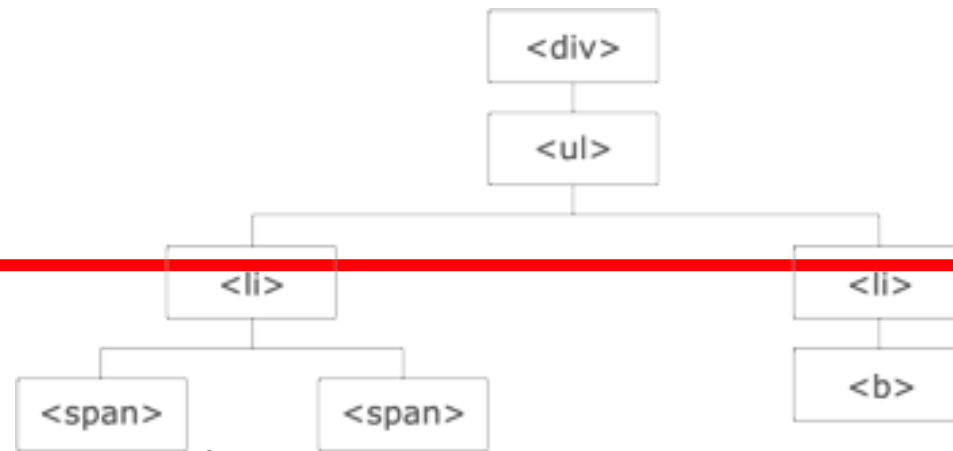
Traversing là gì?

- jQuery traversing, có nghĩa là "di chuyển qua", được sử dụng để "tìm" (hoặc chọn) các phần tử HTML dựa trên mối quan hệ của chúng với các phần tử khác.
- Bắt đầu với một lựa chọn và di chuyển qua lựa chọn đó cho đến khi bạn đạt được các yếu tố bạn mong muốn.
- Hình ảnh dưới đây minh họa một trang HTML dưới dạng cây (DOM tree).



- Với jQuery duyệt, bạn có thể dễ dàng di chuyển lên (tổ tiên), xuống (con cháu) và sang ngang (anh chị em) trong cây, bắt đầu từ phần tử đã chọn (hiện tại). Chuyển động này được gọi là đi ngang - hoặc di chuyển qua - cây DOM.

Giải thích



- Hình minh họa giải thích:
- Phần tử `<div>` là cha của `` và là tổ tiên của mọi thứ bên trong nó
- Phần tử `` là phần tử cha của cả hai phần tử `` và là phần tử con của `<div>`
- Phần tử left `` là phần tử cha của ``, phần tử con của `` và phần tử con của `<div>`
- Phần tử `` là phần tử con bên trái `` và phần tử con của `` và `<div>`
- Hai phần tử `` là anh chị em ruột (chúng có chung cha mẹ)
- Phần tử `` bên phải là phần tử cha của ``, con của `` và phần tử con của `<div>`
- Phần tử `` là phần tử con của quyền `` và phần tử con của `` và `<div>`

Ghi chú: Tổ tiên là cha mẹ, ông bà, cụ cố, vân vân. Con cháu là con, cháu, chắt, v.v. Anh chị em có chung cha mẹ.

jQuery Traversing - Ancestors (Tổ tiên)

- Ba phương thức jQuery hữu ích để duyệt qua cây DOM là:
 - `parent()`
 - `Parents()`
 - `parentUntil()`

jQuery parent():

Phương thức parent () trả về phần tử cha trực tiếp của phần tử được chọn.
Phương thức này chỉ duyệt một cấp duy nhất lên cây DOM.

```
<style>
.ancestors * {
  display: block;
  border: 2px solid lightgrey;
  color: lightgrey;
  padding: 5px;
  margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("span").parent().css({"color": "red", "border": "4px solid green"});
});
</script>
</head>
<body>

<div class="ancestors">
  <div style="width:500px;">div (great-grandparent)
    <ul>ul (grandparent)
      <li>li (direct parent)
        <span>span</span>
      </li>
    </ul>
  </div>

  <div style="width:500px;">div (grandparent)
    <p>p (direct parent)
      <span>span</span>
    </p>
  </div>
</div>
```

div (great-grandparent)

ul (grandparent)

li (direct parent)

span

div (grandparent)

p (direct parent)

span

jQuery parents()

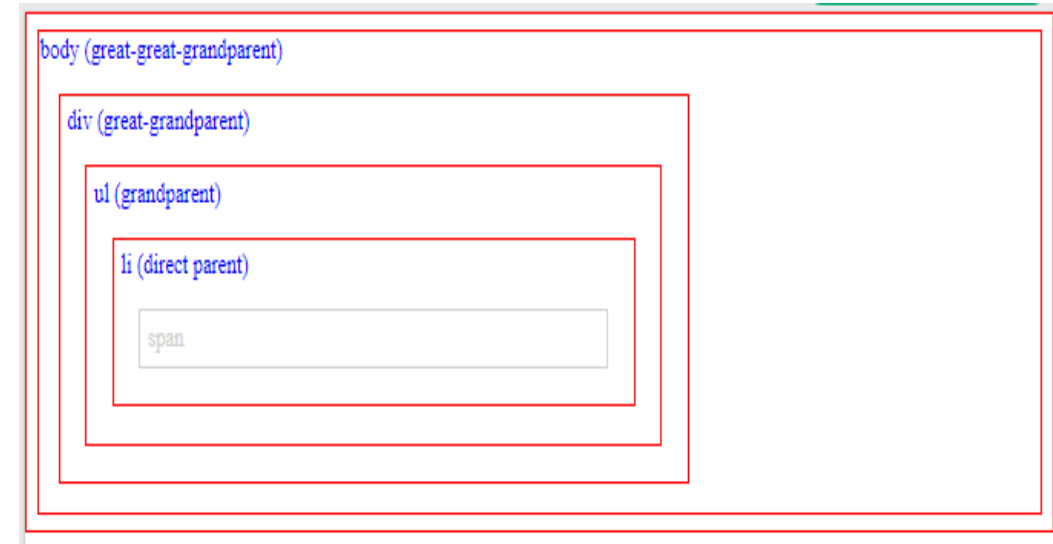
Phương thức parents() trả về tất cả các phần tử tổ tiên của phần tử đã chọn, cho đến phần tử gốc của tài liệu (<html>).

Ví dụ sau trả về tất cả tổ tiên của tất cả các phần tử :

```
<!DOCTYPE html>
<html>
<head>
<style>
.ancestors * {
  display: block;
  border: 2px solid lightgrey;
  color: lightgrey;
  padding: 5px;
  margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("span").parents().css({"color": "blue", "border": "2px solid red"});
});
</script>
</head>

<body class="ancestors">body (great-great-grandparent)
  <div style="width:500px;">div (great-grandparent)
    <ul>ul (grandparent)
      <li>li (direct parent)
        <span>span</span>
      </li>
    </ul>
  </div>
</body>

<!-- The outer red border, before the body element, is the html element (also an ancestor) -->
</html>
```



Ví dụ sau trả về tất cả tổ tiên của tất cả các phần tử `` là phần tử ``:

```
<!DOCTYPE html>
<html>
<head>
<style>
.ancestors * {
  display: block;
  border: 2px solid lightgrey;
  color: lightgrey;
  padding: 5px;
  margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("span").parents("ul").css({"color": "red", "border": "2px solid red"});
});
</script>
</head>

<body class="ancestors">body (great-great-grandparent)
  <div style="width:500px;">div (great-grandparent)
    <ul>ul (grandparent)
      <li>li (direct parent)
        <span>span</span>
      </li>
    </ul>
    <ul>ul (grandparent1)
      <li>li (direct parent1)
        <span>span</span>
      </li>
    </ul>
  </div>
</body>
</html>
```

body (great-great-grandparent)

div (great-grandparent)

ul (grandparent)

li (direct parent)

span

ul (grandparent1)

li (direct parent1)

span

jQuery parentsUntil()

Phương thức cha mẹUntil () trả về tất cả các phần tử tổ tiên giữa hai đối số đã cho.

Ví dụ sau trả về tất cả các phần tử tổ tiên giữa một phần tử và một phần tử <div>:

```
<!DOCTYPE html>
<html>
<head>
<style>
.ancestors * {
  display: block;
  border: 2px solid lightgrey;
  color: lightgrey;
  padding: 5px;
  margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("span").parentsUntil("div").css({"color": "red", "border": "2px solid red"});
});
</script>
</head>

<body class="ancestors"> body (great-great-grandparent)
  <div style="width:500px;">div (great-grandparent)
    <ul>ul (grandparent)
      <li>li (direct parent)
        <span>span</span>
      </li>
    </ul>
  </div>
</body>

</html>
```

body (great-great-grandparent)

div (great-grandparent)

ul (grandparent)

li (direct parent)

span

jQuery Traversing - Con cháu

- Hai phương thức jQuery hữu ích để duyệt qua cây DOM là:
 - `child()`
 - `find()`

jQuery children()

Phương thức children () trả về tất cả các phần tử con trực tiếp của phần tử đã chọn. Phương thức này chỉ duyệt một cấp duy nhất xuống cây DOM.

```
<!DOCTYPE html>
<html>
<head>
<style>
.descendants * {
  display: block;
  border: 2px solid lightgrey;
  color: lightgrey;
  padding: 5px;
  margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("div").children().css({"color": "red", "border": "2px solid red"});
});
</script>
</head>
<body>

<div class="descendants" style="width:500px;">div (current element)
  <p>p (child)
    <span>span (grandchild)</span>
  </p>
  <p>p (child)
    <span>span (grandchild)</span>
  </p>
</div>

</body>
</html>
```

div (current element)

p (child)

span (grandchild)

p (child)

span (grandchild)

Sử dụng một tham số tùy chọn để lọc tìm kiếm các phần tử con.
Ví dụ sau trả về tất cả các phần tử `<p>` có tên lớp là "first", là con trực tiếp của `<div>`:

```
<!DOCTYPE html>
<html>
<head>
<style>
.descendants * {
  display: block;
  border: 2px solid lightgrey;
  color: lightgrey;
  padding: 5px;
  margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("div").children("p.first").css({"color": "red", "border": "2px solid red"});
});
</script>
</head>
<body>

<div class="descendants" style="width:500px;">div (current element)
  <p class="first">p (child)
    <span>span (grandchild)</span>
  </p>
  <p class="second">p (child)
    <span>span (grandchild)</span>
  </p>
</div>

</body>
</html>
```

div (current element)

p (child)

span (grandchild)

p (child)

span (grandchild)

jQuery find()

Phương thức find () trả về các phần tử con của phần tử đã chọn, cho đến phần tử con cuối cùng.

Ví dụ sau trả về tất cả các phần tử là con của <div>:

```
<!DOCTYPE html>
<html>
<head>
<style>
.descendants * {
  display: block;
  border: 2px solid lightgrey;
  color: lightgrey;
  padding: 5px;
  margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("div").find("span").css({"color": "red", "border": "2px solid red"});
});
</script>
</head>
<body>

<div class="descendants" style="width:500px;">div (current element)
  <p>p (child)
    <span>span (grandchild)</span>
  </p>
  <p>p (child)
    <span>span (grandchild)</span>
  </p>
</div>

</body>
</html>
```

div (current element)

p (child)

span (grandchild)

p (child)

span (grandchild)

Ví dụ sau trả về tất cả các con của <div>:

```
<!DOCTYPE html>
<html>
<head>
<style>
.descendants * {
  display: block;
  border: 2px solid lightgrey;
  color: lightgrey;
  padding: 5px;
  margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("div").find("*").css({"color": "red", "border": "2px solid red"});
});
</script>
</head>
<body>

<div class="descendants" style="width:500px;">div (current element)
  <p>p (child)
    <span>span (grandchild)</span>
  </p>
  <p>p (child)
    <span>span (grandchild)</span>
  </p>
</div>

</body>
</html>
```

div (current element)

p (child)

span (grandchild)

p (child)

span (grandchild)

jQuery Traversing-Anh chị em ruột

Có nhiều phương thức jQuery hữu ích để đi ngang trong cây DOM:

- `siblings()`
 - `next()`
 - `nextAll()`
 - `nextUntil()`
 - `prev()`
 - `prevAll()`
 - `prevUntil()`
-

jQuery siblings()

Phương thức siblings () trả về tất cả các phần tử anh chị em của phần tử được chọn.

```
<!DOCTYPE html>
<html>
<head>
<style>
.siblings * {
  display: block;
  border: 2px solid lightgrey;
  color: lightgrey;
  padding: 5px;
  margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.
<script>
$(document).ready(function(){
  $("h2").siblings().css({"color": "red", "border": "2px solid red"});
});
</script>
</head>
<body class="siblings">

<div>div (parent)
  <p>p</p>
  <span>span</span>
  <h2>h2</h2>
  <h3>h3</h3>
  <p>p</p>
</div>

</body>
</html>
```



Sử dụng một tham số tùy chọn để lọc tìm kiếm anh chị em. Ví dụ sau trả về tất cả các phần tử anh em của `<h2>` là phần tử `<p>`:

```
<!DOCTYPE html>
<html>
<head>
<style>
.siblings * {
  display: block;
  border: 2px solid lightgrey;
  color: lightgrey;
  padding: 5px;
  margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("h2").siblings("p").css({"color": "red", "border": "2px solid red"});
});
</script>
</head>
<body class="siblings">

<div>div (parent)
  <p>p</p>
  <span>span</span>
  <h2>h2</h2>
  <h3>h3</h3>
  <p>p</p>
</div>

</body>
</html>
```



jQuery next()

Phương thức next () trả về phần tử anh em tiếp theo của phần tử được chọn.

Ví dụ sau trả về anh chị em tiếp theo của <h2>:

```
<!DOCTYPE html>
<html>
<head>
<style>
.siblings * {
  display: block;
  border: 2px solid lightgrey;
  color: lightgrey;
  padding: 5px;
  margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("h2").next().css({"color": "red", "border": "2px solid red"});
});
</script>
</head>
<body class="siblings">

<div>div (parent)
  <p>p</p>
  <span>span</span>
  <h2>h2</h2>
  <h3>h3</h3>
  <p>p</p>
</div>

</body>
</html>
```



jQuery nextAll()

Phương thức nextAll () trả về tất cả các phần tử anh em tiếp theo của phần tử đã chọn.

Ví dụ sau trả về tất cả các phần tử anh em tiếp theo của <h2>:

```
<!DOCTYPE html>
<html>
<head>
<style>
.siblings * {
  display: block;
  border: 2px solid lightgrey;
  color: lightgrey;
  padding: 5px;
  margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("h2").nextAll().css({"color": "red", "border": "2px solid red"});
});
</script>
</head>
<body class="siblings">

<div>div (parent)
  <p>p</p>
  <span>span</span>
  <h2>h2</h2>
  <h3>h3</h3>
  <p>p</p>
</div>

</body>
</html>
```



jQuery nextUntil()

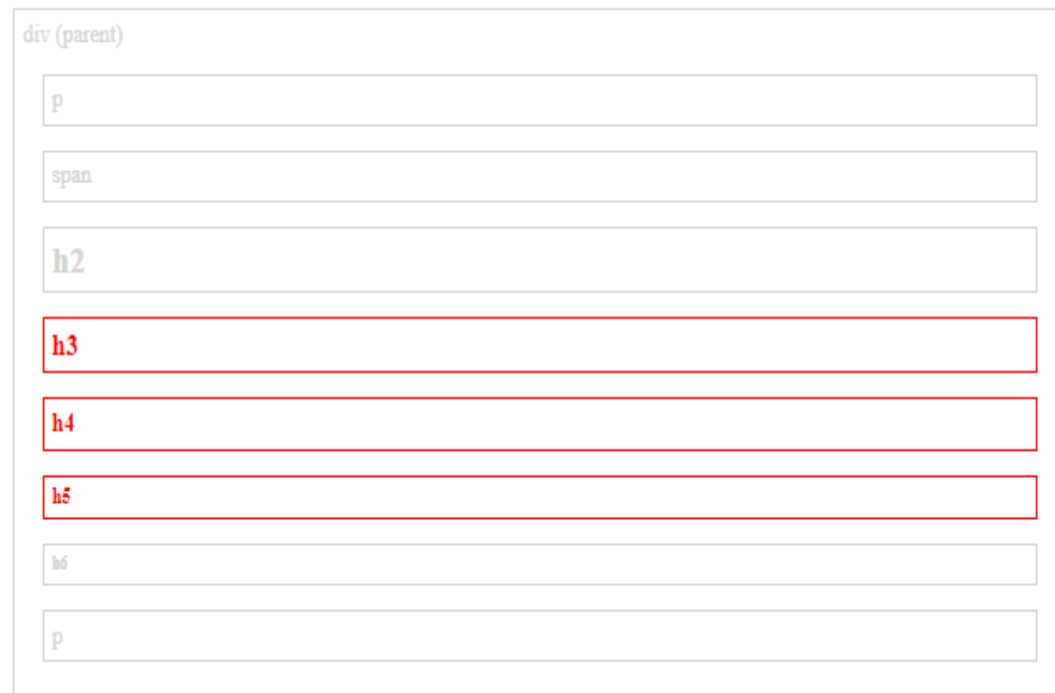
Phương thức nextUntil () trả về tất cả các phần tử anh em tiếp theo giữa hai đối số đã cho.

Ví dụ sau trả về tất cả các phần tử anh em giữa một phần tử <h2> và một phần tử <h6>:

```
<!DOCTYPE html>
<html>
<head>
<style>
.siblings * {
  display: block;
  border: 2px solid lightgrey;
  color: lightgrey;
  padding: 5px;
  margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("h2").nextUntil("h6").css({"color": "red", "border": "2px solid red"});
});
</script>
</head>
<body class="siblings">

<div>div (parent)
  <p>p</p>
  <span>span</span>
  <h2>h2</h2>
  <h3>h3</h3>
  <h4>h4</h4>
  <h5>h5</h5>
  <h6>h6</h6>
  <p>p</p>
</div>

</body>
</html>
```



jQuery prev(), prevAll() & prevUntil()

- Các phương thức prev (), prevAll () và prevUntil () hoạt động giống như các phương thức ở next(), nextAll() và nextUntil() nhưng có chức năng ngược lại: chúng trả về các phần tử anh em trước đó (đi ngược chiều dọc theo các phần tử anh em trong cây DOM)

jQuery Traversing - Filtering(lọc)

- Hàm first(), last(), eq(), filter() và not()
- Các phương pháp lọc cơ bản nhất là first (), last () và eq (), cho phép bạn chọn một phần tử cụ thể dựa trên vị trí của nó trong một nhóm phần tử.
- Các phương pháp lọc khác, như filter () và not () cho phép bạn chọn các phần tử phù hợp hoặc không khớp với một tiêu chí nhất định.

jQuery first()

Phương thức first () trả về phần tử đầu tiên của các phần tử được chỉ định.

Ví dụ sau chọn phần tử <div> đầu tiên:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("div").first().css("background-color", "yellow");
});
</script>
</head>
<body>

<h1>Welcome to My Homepage</h1>

<p>This is a paragraph.</p>

<div style="border: 1px solid black;">
  <p>A paragraph in a div.</p>
  <p>Another paragraph in a div.</p>
</div>
<br>

<div style="border: 1px solid black;">
  <p>A paragraph in another div.</p>
  <p>Another paragraph in another div.</p>
</div>
<br>

<div style="border: 1px solid black;">
  <p>A paragraph in another div.</p>
  <p>Another paragraph in another div.</p>
</div>

</body>
</html>
```

Welcome to My Homepage

This is a paragraph.

A paragraph in a div.

Another paragraph in a div.

A paragraph in another div.

Another paragraph in another div.

A paragraph in another div.

Another paragraph in another div.

jQuery last()

Phương thức last () trả về phần tử cuối cùng của các phần tử được chỉ định.

Ví dụ sau đây chọn phần tử <div> cuối cùng:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("div").last().css("background-color", "yellow");
});
</script>
</head>
<body>

<h1>Welcome to My Homepage</h1>

<p>This is a paragraph.</p>

<div style="border: 1px solid black;">
    <p>A paragraph in a div.</p>
    <p>Another paragraph in a div.</p>
</div>
<br>

<div style="border: 1px solid black;">
    <p>A paragraph in another div.</p>
    <p>Another paragraph in another div.</p>
</div>
<br>

<div style="border: 1px solid black;">
    <p>A paragraph in another div.</p>
    <p>Another paragraph in another div.</p>
</div>

</body>
</html>
```

Welcome to My Homepage

This is a paragraph.

A paragraph in a div.

Another paragraph in a div.

A paragraph in another div.

Another paragraph in another div.

A paragraph in another div.

Another paragraph in another div.

jQuery eq()

Phương thức eq () trả về một phần tử có số chỉ mục cụ thể của các phần tử được chọn. Các số chỉ mục bắt đầu từ 0, vì vậy phần tử đầu tiên sẽ có số chỉ mục 0 chứ không phải 1.

Ví dụ sau chọn phần tử <p> thứ hai (chỉ số 1):

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("p").eq(1).css("background-color", "yellow");
});
</script>
</head>
<body>

<h1>Welcome to My Homepage</h1>

<p>My name is Donald (index 0).</p>
<p>Donald Duck (index 1).</p>
<p>I live in Duckburg (index 2).</p>
<p>My best friend is Mickey (index 3).</p>

</body>
</html>
```

Welcome to My Homepage

My name is Donald (index 0).

Donald Duck (index 1).

I live in Duckburg (index 2).

My best friend is Mickey (index 3).

jQuery filter()

Phương thức filter () cho phép bạn chỉ định một tiêu chí.

Các phần tử không phù hợp với tiêu chí sẽ bị xóa khỏi lựa chọn và những phần tử phù hợp sẽ được trả lại.

Ví dụ sau trả về tất cả các phần tử <p> có tên lớp là "phần giới thiệu":

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("p").filter(".intro").css("background-color", "yellow");
});
</script>
</head>
<body>

<h1>Welcome to My Homepage</h1>

<p>My name is Donald.</p>
<p class="intro">I live in Duckburg.</p>
<p class="intro">I love Duckburg.</p>
<p>My best friend is Mickey.</p>

</body>
</html>
```

Welcome to My Homepage

My name is Donald.

I live in Duckburg.

I love Duckburg.

My best friend is Mickey.

jQuery not()

Phương thức not () trả về tất cả các phần tử không phù hợp với tiêu chí.

Lưu ý: Phương thức not () đối lập với filter ().

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("p").not(".intro").css("background-color", "yellow");
});
</script>
</head>
<body>

<h1>Welcome to My Homepage</h1>

<p>My name is Donald.</p>
<p class="intro">I live in Duckburg.</p>
<p class="intro">I love Duckburg.</p>
<p>My best friend is Mickey.</p>

</body>
</html>
```

Welcome to My Homepage

My name is Donald.

I live in Duckburg.

I love Duckburg.

My best friend is Mickey.

jQuery - AJAX

AJAX là công nghệ cho phép trao đổi dữ liệu với máy chủ và cập nhật các phần của trang web - mà không cần tải lại toàn bộ trang.

AJAX = JavaScript và XML không đồng bộ. Nói ngắn gọn; AJAX là về việc tải dữ liệu ở chế độ nền và hiển thị trên trang web mà không cần tải lại toàn bộ trang. Ví dụ về các ứng dụng sử dụng AJAX: tab Gmail, Google Maps, Youtube và Facebook.

Mối liên hệ giữa jQuery và AJAX?

jQuery cung cấp một số phương thức cho chức năng AJAX. Với các phương thức jQuery AJAX, bạn có thể yêu cầu văn bản, HTML, XML hoặc JSON từ máy chủ từ xa bằng cách sử dụng cả HTTP Get và HTTP Post - Và bạn có thể tải dữ liệu bên ngoài trực tiếp vào các phần tử HTML đã chọn của trang web của mình!

jQuery - AJAX load()

Phương thức jQuery load () là một phương thức AJAX đơn giản nhưng mạnh mẽ. Phương thức load () tải dữ liệu từ máy chủ và đưa dữ liệu trả về vào phần tử đã chọn.

Cú pháp:

```
$(selector).load(URL,data,callback);
```

Tham số URL bắt buộc chỉ định URL bạn muốn tải.

Tham số data tùy chọn chỉ định một tập hợp các cặp khóa / giá trị chuỗi truy vấn để gửi cùng với yêu cầu.

Tham số callback tùy chọn là tên của một hàm sẽ được thực thi sau khi hoàn tất phương thức load ().

Ví dụ: sử dụng tham số url lấy dữ liệu từ tập tin

```
<!DOCTYPE html>
<meta charset="UTF-8">
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function() {
  $("button").click(function() {
    $("#div1").load("data_test.txt");
  });
});
</script>
</head>
<body>

<div id="div1"><h2>Bài thơ</h2></div>

<button>Lấy nội dung từ file</button>

</body>
</html>
```

← → ↻ ⓘ localhost:9000/ex/web.html

Bài thơ

Lấy nội dung từ file

Tất Nước Đầu Đình </br>
Hôm qua tát nước đầu đình</br>
Bỏ quên cái áo trên cành hoa sen</br>
Em được thì cho anh xin</br>
Hay là em để làm tin trong nhà?</br>
Áo anh sút chỉ đường tà</br>
Vợ anh chưa có, mẹ già chưa khâu.</br>

← → ↻ ⓘ localhost:9000/ex/web.html

Tất Nước Đầu Đình
Hôm qua tát nước đầu đình
Bỏ quên cái áo trên cành hoa sen
Em được thì cho anh xin
Hay là em để làm tin trong nhà?
Áo anh sút chỉ đường tà
Vợ anh chưa có, mẹ già chưa khâu.

Lấy nội dung từ file

Ví dụ Tải nội dung của phần tử có id = "p1", bên trong tệp "demo_test.txt", vào một phần tử <div> cụ thể:

```
<!DOCTYPE html>
<meta charset="UTF-8">
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function() {
  $("button").click(function() {
    $("#div1").load("data_test.txt #p1");
  });
});
</script>
</head>
<body>

<div id="div1"><h2>Bài thơ</h2></div>

<button>Lấy nội dung từ file</button>

</body>
</html>
```

```
data_test.txt
Tát Nước Đầu Đình </br>
Hôm qua tát nước đầu đình</br>
Bỏ quên cái áo trên cành hoa sen</br>
Em được thì cho anh xin</br>
Hay là em để làm tin trong nhà?</br>
Áo anh sút chỉ đường tà</br>
Vợ anh chưa có, mẹ già chưa khâu.</br>
<div id="p1"><h2>Trích bài hát tát nước đầu đình</h2></div>
```

← → ↻ ⓘ localhost:9000/ex/web.html

Bài thơ

Lấy nội dung từ file

← → ↻ ⓘ localhost:9000/ex/web.html

Trích bài hát tát nước đầu đình

Lấy nội dung từ file

Sử dụng callback trong phương thức load()

Tham số callback() tùy chọn chỉ định một hàm gọi lại để chạy khi phương thức load () hoàn tất. Hàm gọi lại có thể có các tham số khác nhau:

responseTxt - chứa nội dung kết quả nếu cuộc gọi thành công

statusTxt - chứa trạng thái của cuộc gọi

xhr - chứa đối tượng XMLHttpRequest

Ví dụ sử dụng callback trong việc đọc file

```
<!DOCTYPE html>
<meta charset="UTF-8">
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function() {
  $("button").click(function() {
    $("#div1").load("data_test.txt", function(responseTxt, statusTxt, xhr) {
      if(statusTxt == "success")
        alert("Lấy file thành công!");
      if(statusTxt == "error")
        alert("Lỗi: " + xhr.status + ": " + xhr.statusText);
    });
  });
});
</script>
</head>
<body>

<div id="div1"><h2>Bài thơ</h2></div>

<button>Lấy nội dung từ file</button>

</body>
</html>
```

← → ↻ ⓘ localhost:9000/ex/web1.html

Bài thơ

Lấy nội dung từ file

← → ↻ ⓘ localhost:9000/ex/web1.html

Bài thơ

Lấy nội dung từ file

localhost:9000 cho biết

Lấy file thành công!

OK

← → ↻ ⓘ localhost:9000/ex/web1.html

Tát Nước Đầu Đình
Hôm qua tát nước đầu đình
Bỏ quên cái áo trên cành hoa sen
Em được thì cho anh xin
Hay là em để làm tin trong nhà?
Áo anh sứt chỉ đường tà
Vợ anh chưa có, mẹ già chưa khâu.

Trích bài hát tát nước đầu đình

Lấy nội dung từ file

jQuery - AJAX get() and post()

Yêu cầu HTTP: GET so với POST

Hai phương pháp thường được sử dụng để phản hồi yêu cầu giữa máy khách và máy chủ là: GET và POST.

GET - Yêu cầu dữ liệu từ một tài nguyên cụ thể

POST - Gửi dữ liệu được xử lý tới một tài nguyên cụ thể

Về cơ bản, GET được sử dụng để lấy (lấy) một số dữ liệu từ máy chủ.

Lưu ý:

Phương thức GET có thể trả về dữ liệu đã lưu trong bộ nhớ cache.

POST cũng có thể được sử dụng để lấy một số dữ liệu từ máy chủ.

Tuy nhiên, phương thức POST KHÔNG BAO GIỜ lưu trữ dữ liệu và thường được sử dụng để gửi dữ liệu cùng với yêu cầu.

jQuery \$.get()

Phương thức \$.get () yêu cầu dữ liệu từ máy chủ với một yêu cầu HTTP GET.

- Cú pháp:

`$.get(URL, data, callback);`

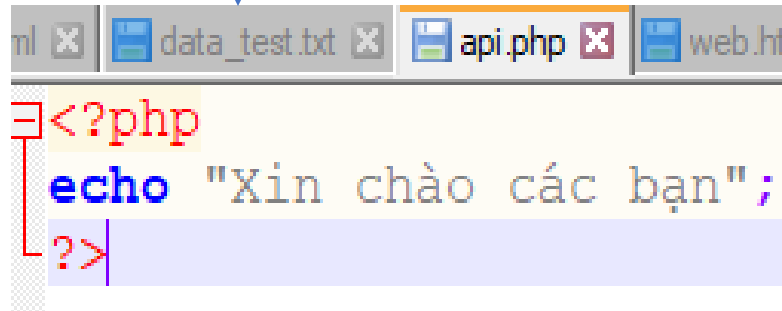
Tham số URL bắt buộc chỉ định URL bạn muốn yêu cầu.

Tham số data tùy chọn chỉ định một số dữ liệu để gửi cùng với yêu cầu.

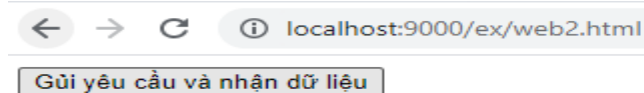
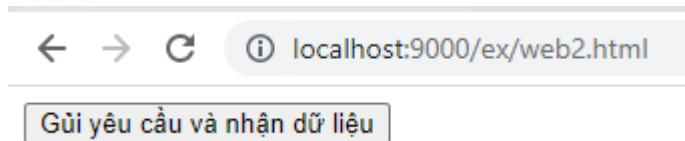
Tham số gọi lại tùy chọn là tên của một hàm sẽ được thực thi nếu yêu cầu thành công.

Ví dụ 1:

Server



```
<?php
echo "Xin chào các bạn";
?>
```



```
<!DOCTYPE html>
<meta charset="UTF-8">
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $.get("php/api.php", function(data, status){
            alert("Data: " + data + "\nStatus: " + status);
        });
    });
});
</script>
</head>
<body>

<button>Gửi yêu cầu và nhận dữ liệu</button>

</body>
```

localhost:9000 cho biết

Data: Xin chào các bạn

Status: success

OK

Ví dụ 2:

```
<!DOCTYPE html>
<meta charset="UTF-8">
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function() {
  $("button").click(function() {
    $.get("php/api.php", {
      name: "Viên Thanh Nhã",
      city: "TPHCM"
    }, function(data, status) {
      alert("Data: " + data + "\nStatus: " + status);
    });
  });
});
</script>
</head>
<body>
<button>Gửi yêu cầu và nhận dữ liệu</button>
</body>
</html>
```

```
<?php
echo "Xin chào " . $_GET["name"] . " sống tại " . $_GET["city"];
-?>
```

← → ↻ ⓘ localhost:9000/ex/web2.html

Gửi yêu cầu và nhận dữ liệu

← → ↻ ⓘ localhost:9000/ex/web2.html

Gửi yêu cầu và nhận dữ liệu

localhost:9000 cho biết

Data: Xin chào Viên Thanh Nhã sống tại TPHCM

Status: success

OK

jQuery \$.post()

Phương thức `$.post()` yêu cầu dữ liệu từ máy chủ bằng cách sử dụng yêu cầu HTTP POST.

Cú pháp:

```
$.post(URL,data,callback);
```

Tham số URL bắt buộc chỉ định URL bạn muốn yêu cầu.

Tham số data tùy chọn chỉ định một số dữ liệu để gửi cùng với yêu cầu.

Tham số callback tùy chọn là tên của một hàm sẽ được thực thi nếu yêu cầu thành công.

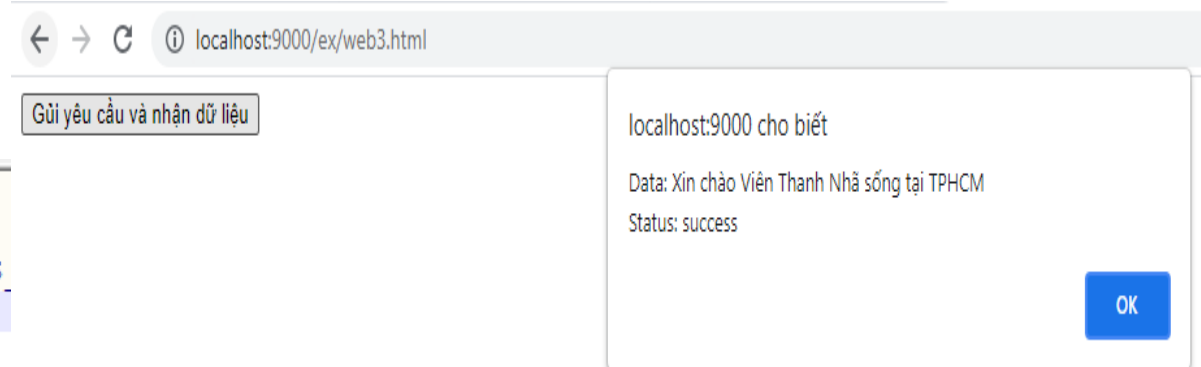
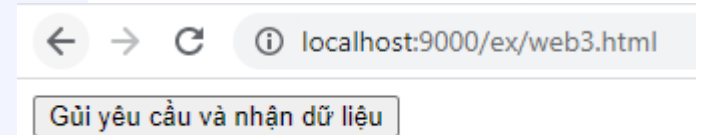
Ví dụ:

```
<!DOCTYPE html>
<meta charset="UTF-8">
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function() {
  $("button").click(function() {
    $.post("php/api.php", {
      name: "Viên Thanh Nhã",
      city: "TPHCM"
    }, function(data, status) {
      alert("Data: " + data + "\nStatus: " + status);
    });
  });
});
</script>
</head>
<body>

<button>Gửi yêu cầu và nhận dữ liệu</button>

</body>
</html>
```

```
<?php
echo "Xin chào ".$_POST["name"]." sống tại ".$_POST["city"];
?>
```



jQuery ajax()

Định nghĩa và Cách sử dụng Phương thức ajax () được sử dụng để thực hiện một yêu cầu AJAX (HTTP không đồng bộ).

Tất cả các phương thức jQuery AJAX đều sử dụng phương thức ajax (). Phương thức này chủ yếu được sử dụng cho các yêu cầu mà các phương thức khác không thể được sử dụng.

Cú pháp:

`$.ajax({name:value, name:value, ... })`

Name	Value/Description
async	A Boolean value indicating whether the request should be handled asynchronous or not. Default is true
beforeSend(<i>xhr</i>)	A function to run before the request is sent
cache	A Boolean value indicating whether the browser should cache the requested pages. Default is true
complete(<i>xhr,status</i>)	A function to run when the request is finished (after success and error functions)
contentType	The content type used when sending data to the server. Default is: "application/x-www-form-urlencoded"
context	Specifies the "this" value for all AJAX related callback functions
data	Specifies data to be sent to the server
dataFilter(<i>data,type</i>)	A function used to handle the raw response data of the XMLHttpRequest
dataType	The data type expected of the server response.
error(<i>xhr,status,error</i>)	A function to run if the request fails.
global	A Boolean value specifying whether or not to trigger global AJAX event handles for the request. Default is true

ifModified	A Boolean value specifying whether a request is only successful if the response has changed since the last request. Default is: false.
jsonp	A string overriding the callback function in a jsonp request
jsonpCallback	Specifies a name for the callback function in a jsonp request
password	Specifies a password to be used in an HTTP access authentication request.
processData	A Boolean value specifying whether or not data sent with the request should be transformed into a query string. Default is true
scriptCharset	Specifies the charset for the request
success(<i>result,status,xhr</i>)	A function to be run when the request succeeds
timeout	The local timeout (in milliseconds) for the request
traditional	A Boolean value specifying whether or not to use the traditional style of param serialization
type	Specifies the type of request. (GET or POST)
url	Specifies the URL to send the request to. Default is the current page
username	Specifies a username to be used in an HTTP access authentication request
xhr	A function used for creating the XMLHttpRequest object

Ví dụ .ajax() đọc file

```
<!DOCTYPE html>
<meta charset="UTF-8">
<html>
<head>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <script>
    $(document).ready(function() {
      $("button").click(function() {
        $.ajax({url: "data_test.txt", success: function(result) {
          $("#div1").html(result);
        }});
      });
    });
  </script>
</head>
<body>
  <div id="div1"><h2>dữ liệu</h2></div>
  <button>Gửi yêu cầu và nhận dữ liệu</button>
</body>
</html>
```

← → ↻ ⓘ localhost:9000/ex/web4.html

dữ liệu

Gửi yêu cầu và nhận dữ liệu

← → ↻ ⓘ localhost:9000/ex/web4.html

dữ liệu

Gửi yêu cầu và nhận dữ liệu

Ví dụ sử dụng tham số async

```
<!DOCTYPE html>
<meta charset="UTF-8">
<html>
<head>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
</script>
  $(document).ready(function() {
    $(document).ready(function() {
      $("button").click(function() {
        $.ajax({url: "data_test.txt", async: false, success: function(result) {
          $("div").html(result);
        }});
      });
    });
  });
</script>
</head>
<body>
  <div><h2>dữ liệu</h2></div>
  <button>Gửi yêu cầu và nhận dữ liệu</button>

</body>
</html>
```

← → ↻ ⓘ localhost:9000/ex/web5.html

dữ liệu

Gửi yêu cầu và nhận dữ liệu

← → ↻ ⓘ localhost:9000/ex/web5.html

Tát Nước Đầu Đình
Hôm qua tát nước đầu đình
Bỏ quên cái áo trên cành hoa sen
Em được thì cho anh xin
Hay là em để làm tin trong nhà?
Áo anh sứt chỉ đường tà
Vợ anh chưa có, mẹ già chưa khâu.

Trích bài hát tát nước đầu đình

Gửi yêu cầu và nhận dữ liệu

Ví dụ đọc dữ liệu từ server kiểu text

```
<!DOCTYPE html>
<meta charset="UTF-8">
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $(document).ready(function(){
    $("button").click(function(){
      $.ajax({url: "php/api.php",data:{"name":"Viên Thanh Nhã","city":"HCM"}, async: false, success: function(result){
        $("#div").html(result);
      });
    });
  });
});
</script>
</head>
<body>
<div><h2>dữ liệu</h2></div>
<button>Gửi yêu cầu và nhận dữ liệu</button>
</body>
</html>
```

← → ↻ ⓘ localhost:9000/ex/web6.html

dữ liệu

Gửi yêu cầu và nhận dữ liệu

← → ↻ ⓘ localhost:9000/ex/web6.html

Xin chào Viên Thanh Nhã sống tại HCM

Gửi yêu cầu và nhận dữ liệu

```
<?php
echo "Xin chào ".$_GET["name"]." sống tại ".$_GET["city"];
?>
```

Lưu ý: datatype='text', và kiểu type="GET" khi không chỉ định cụ thể

Ví dụ lấy dữ liệu từ server bằng JSON

```
<!DOCTYPE html>
<meta charset="UTF-8">
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function() {
  $(document).ready(function() {
    $("button").click(function() {
      $.ajax({url: "php/api.php",
        data: {"name": "Viên Thanh Nhã", "city": "HCM"},
        async: false,
        dataType: 'json',
        success: function(result) {
          $("div").html(JSON.parse(result).sdt);
        }
      });
    });
  });
});
</script>
</head>
<body>
<div><h2>dữ liệu</h2></div>
<button>Gửi yêu cầu và nhận dữ liệu</button>

</body>
</html>
```

localhost:9000/ex/web7.html

dữ liệu

Gửi yêu cầu và nhận dữ liệu

localhost:9000/ex/web7.html

0939191028

Gửi yêu cầu và nhận dữ liệu

Ta có thể viết thành 1 hàm như sau:

```
function queryDataGet(url,dataSend,callback){
```

```
    $.ajax({  
        type: 'GET',  
        url: url,  
        data: dataSend,  
        async: true,  
        dataType: 'json',  
        success: callback  
    });  
}
```

```
function queryDataPost(url,dataSend,callback){
```

```
    $.ajax({  
        type: 'POST',  
        url: url,  
        data: dataSend,  
        async: true,  
        dataType: 'json',  
        success: callback  
    });  
}
```

Ví dụ:

```
<script>
$(document).ready(function() {
    $(document).ready(function() {
        $("#button").click(function() {
            var data={"name":"Viên Thanh Nhã","city":"HCM"};

            queryDataGet("php/api.php",data,function(result) {
                $("#div").html(JSON.parse(result).sdt);
            });
        });
    });
});
function queryDataGet(url,dataSend,callback) {

    $.ajax({
        type: 'GET',
        url: url,
        data: dataSend,
        async: true,
        dataType: 'json',
        success: callback
    });
}
```

← → ↻ ⓘ localhost:9000/ex/web8.html

dữ liệu

Gửi yêu cầu và nhận dữ liệu

← → ↻ ⓘ localhost:9000/ex/web8.html

0939191028

Gửi yêu cầu và nhận dữ liệu

Ví dụ lấy dữ liệu từ database

id	name	address
1	Nhan	Hanoi
2	Ha	Vinhphuc
3	Sinh	Danang
4	Nong	Hanoi
5	Noi	Hanoi
6	Thanh Nhã	Sài Gòn

```
$(document).ready(function() {
    $(document).ready(function() {
        $("button").click(function() {
            var data={id:"1"};
            queryDataGet("php/apidata.php",data,function(result) {
                if(result.event==1){
                    var data=result.items;
                    var html='Họ tên    Địa chỉ';
                    for (item in data) {
                        var list=data[item];
                        html=list.name+"    "+list.address+"<br>";
                        $("#d").html(html);
                    }
                }else{
                    $("#d").html("Lấy dữ liệu bị lỗi");
                }
            });
        });
    });
});
```

← → ↻ ⓘ localhost:9000/ex/web9.html

dữ liệu

Gửi yêu cầu và nhận dữ liệu

← → ↻ ⓘ localhost:9000/ex/web9.html

Thanh Nhã Sài Gòn

Gửi yêu cầu và nhận dữ liệu

Ví dụ insert dữ liệu vào database Phía server

```
<?php
require_once("server.php");
$mang=array();
$id=$_GET['id'];
$name=$_GET['name'];
$address=$_GET['address'];

$rs=mysqli_query($conn,"select COUNT(*) as 'total' from student where id='".$id."' ");
$row=mysqli_fetch_array($rs);
if((int)$row['total']>0){
    $res['event'] = 2;
}else{
    $sql="INSERT INTO `student`(`id`, `name`, `address`) VALUES ('".$id."','".$name."','".$address."')";

    if (mysqli_query($conn, $sql)) {
        if(mysqli_affected_rows($conn)>0){
            $res['event'] = 1;
        }
        else{
            $res['event'] = 0;
        }
    } else {
        $res['event'] = 3;
    }
}

echo json_encode($res);
mysqli_close($conn);
```

?>

Phía client gửi dữ liệu insert

```
$(document).ready(function() {
    $(document).ready(function() {
        $("button").click(function() {
            console.log("fs");
            var data={
                id:"7",
                name:"Nguyễn Cẩm Anh",
                address:"Sóc trăng"
            };
            queryDataGet("php/apidata_insert.php",data,function(result) {
                console.log(result);
                if(result.event==1) {

                    $("#d").html("Insert Thành công");
                }
                else if(result.event==0) {
                    $("#d").html("Insert không Thành công");
                }
                else if(result.event==2) {
                    $("#d").html("Trùng ID");
                } else {
                    $("#d").html("Lỗi");
                }
            });
        });
    });
});
```

```
<body>
<div id="d"><h2>dữ liệu</h2></div>
<button>Gửi yêu cầu và nhận dữ liệu</button>

</body>
```

localhost:9000/ex/web10.html

dữ liệu

Gửi yêu cầu và nhận dữ liệu

localhost:9000/ex/web10.html

Insert Thành công

Gửi yêu cầu và nhận dữ liệu

id	name	address
1	Nhan	Hanoi
2	Ha	Vinhphuc
3	Sinh	Danang
4	Nong	Hanoi
5	Noi	Hanoi
7	Nguyễn Cẩm Anh	Sóc trăng
6	Thanh Nhã	Sài Gòn

Update dữ liệu vào database

- Server

```
<?php
require_once("server.php");
$mang=array();
$id=$_GET['id'];
$name=$_GET['name'];
$address=$_GET['address'];

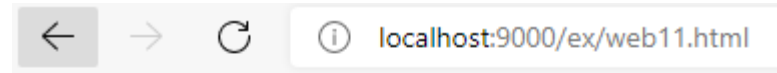
$sql="UPDATE `student` SET `name`='".$name."',`address`='".$address."' WHERE id='".$id."";
    if (mysqli_query($conn, $sql)) {
        if(mysqli_affected_rows($conn)>0){
            $res['event'] = 1;
        }else{
            $res['event'] = 0;
        }
    } else {
        $res['event'] = 2;
    }

echo json_encode($res);
mysqli_close($conn);
?>
```

Client gửi dữ liệu update

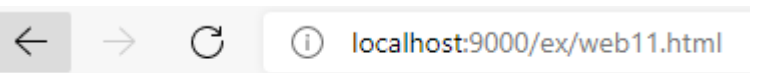
```
$(document).ready(function() {
  $(document).ready(function() {
    $("button").click(function() {
      console.log("fs");
      var data={
        id:"7",
        name:"Nguyễn Cẩm Anh",
        address:"An Giang"
      };
      queryDataGet("php/apidata_update.php",data,function(result) {
        console.log(result);
        if(result.event==1) {
          $("#d").html("Update Thành công");
        }
        else if(result.event==0) {
          $("#d").html("Update không Thành công");
        }
        else{
          $("#d").html("Lỗi");
        }
      });
    });
  });
});
```

```
<body>
<div id="d"><h2>dữ liệu</h2></div>
<button>Gửi yêu cầu và nhận dữ liệu</button>
</body>
```



dữ liệu

Gửi yêu cầu và nhận dữ liệu



Update Thành công

Gửi yêu cầu và nhận dữ liệu

id	name	address
1	Nhan	Hanoi
2	Ha	Vinhphuc
3	Sinh	Danang
4	Nong	Hanoi
5	Noi	Hanoi
7	Nguyễn Cẩm Anh	An Giang
6	Thanh Nhã	Sài Gòn

Delete dữ liệu trong database Server:

```
<?php
require_once("server.php");
$mang=array();
$id=$_GET['id'];

    $sql="delete from `student` WHERE id='".$id."'";
        if (mysqli_query($conn, $sql)) {
            if(mysqli_affected_rows($conn)>0){
                $res['event'] = 1;
            }else{
                $res['event'] = 0;
            }
        } else {
            $res['event'] = 2;
        }

    echo json_encode($res);
    mysqli_close($conn);
```

?>

Client gửi id để tiến hành delete

```
$(document).ready(function() {
    $(document).ready(function() {
        $("button").click(function() {
            console.log("fs");
            var data={
                id:"7"
            };
            queryDataGet("php/apidata_delete.php",data,function(result) {
                console.log(result);
                if(result.event==1) {
                    $("#d").html("Delete Thành công");
                }
                else if(result.event==0) {
                    $("#d").html("Delete không Thành công");
                }
                else {
                    $("#d").html("Lỗi");
                }
            });
        });
    });
});
```

← → ↻ ⓘ localhost:9000/ex/web12.html

dữ liệu

Gửi yêu cầu và nhận dữ liệu

← → ↻ ⓘ localhost:9000/ex/web12.html

Delete Thành công

Gửi yêu cầu và nhận dữ liệu

id	name	address
1	Nhan	Hanoi
2	Ha	Vinhphuc
3	Sinh	Danang
4	Nong	Hanoi
5	Noi	Hanoi
6	Thanh Nhã	Sài Gòn

Các phương thức khác jQuery AJAX

Method	Description
<code>\$.ajax()</code>	Performs an async AJAX request
<code>\$.ajaxPrefilter()</code>	Handle custom Ajax options or modify existing options before each request is sent and before they are processed by <code>\$.ajax()</code>
<code>\$.ajaxSetup()</code>	Sets the default values for future AJAX requests
<code>\$.ajaxTransport()</code>	Creates an object that handles the actual transmission of Ajax data
<code>\$.get()</code>	Loads data from a server using an AJAX HTTP GET request
<code>\$.getJSON()</code>	Loads JSON-encoded data from a server using a HTTP GET request
<code>\$.parseJSON()</code>	Deprecated in version 3.0, use <code>JSON.parse()</code> instead. Takes a well-formed JSON string and returns the resulting JavaScript value
<code>\$.getScript()</code>	Loads (and executes) a JavaScript from a server using an AJAX HTTP GET request
<code>\$.param()</code>	Creates a serialized representation of an array or object (can be used as URL query string for AJAX requests)
<code>\$.post()</code>	Loads data from a server using an AJAX HTTP POST request

<code>ajaxComplete()</code>	Specifies a function to run when the AJAX request completes
<code>ajaxError()</code>	Specifies a function to run when the AJAX request completes with an error
<code>ajaxSend()</code>	Specifies a function to run before the AJAX request is sent
<code>ajaxStart()</code>	Specifies a function to run when the first AJAX request begins
<code>ajaxStop()</code>	Specifies a function to run when all AJAX requests have completed
<code>ajaxSuccess()</code>	Specifies a function to run when an AJAX request completes successfully
<code>load()</code>	Loads data from a server and puts the returned data into the selected element
<code>serialize()</code>	Encodes a set of form elements as a string for submission
<code>serializeArray()</code>	Encodes a set of form elements as an array of names and values