

HTML5 CANVAS

# THỢ ĐỒNG HỒ

## Nội dung

Làm đồng hồ với Canvas.....	3
Phần I – Tạo Canvas.....	3
HTML code:.....	3
Code ở trên viết gì?.....	3
Phần II – Dựng mặt.....	5
JavaScript:.....	5
Code ở trên viết gì?.....	5
Phần III – Gắn số.....	7
JavaScript:.....	7
Code ở trên viết gì?.....	7
Phần IV – Lắp đặt kim.....	8
JavaScript:.....	8
Code ở trên viết gì?.....	8
Phần V – Lắp pin và chạy.....	10
JavaScript:.....	10
Code ở trên viết gì?.....	10
Phần VI – Thử thách thợ đồng hồ.....	11

# Làm đồng hồ với Canvas

Trong bài hướng dẫn này chúng ta sẽ xây dựng một chiếc đồng hồ với HTML canvas.

Dịch từ:

## Phần I - Tạo Canvas

Đầu tiên chiếc đồng hồ cần có một nơi để “trung bày” (HTML container). Chúng ta hãy tạo một HTML canvas với kích thước là 300 x 300 px:

### HTML code:

```
<!DOCTYPE html>
<html>
<body>

<canvas id="canvas" width="300" height="300" style="background-
color:#333"></canvas>

<script>

    var canvas = document.getElementById("canvas");
    var ctx = canvas.getContext("2d");
    var radius = canvas.height / 2;
    ctx.translate(radius, radius);
    radius = radius * 0.90
    drawClock();

    function drawClock() {
        ctx.arc(0, 0, radius, 0, 2*Math.PI);
        ctx.fillStyle = "white";
        ctx.fill();
    }

</script>

</body>
</html>
```

### Code ở trên viết gì?

Thêm phần tử HTML <canvas> vào trang (chỗ để trưng bày đồng hồ):

```
<canvas id="canvas" width="300" height="300" style="background-
color:#333"></canvas>
```

Tạo một đối tượng canvas (var canvas) trong JavaScript từ phần tử HTML canvas vừa thêm ở trên:

```
var canvas = document.getElementById("canvas");
```

Tạo đối tượng để vẽ 2d (var ctx) từ đối tượng canvas ở trên:

```
var ctx = canvas.getContext("2d");
```

Tính toán bán kính của chiếc đồng hồ (var radius), sử dụng chiều cao của canvas:

```
var radius = canvas.height / 2;
```



Sử dụng chiều cao của canvas để tính bán kính của đồng hồ sẽ khiến đồng hồ có thể “trung bày” vừa vặn với bất kỳ kích cỡ nào của canvas.

Đặt lại vị trí (0,0) (của đối tượng vẽ - ctx) vào trung tâm của canvas:

```
ctx.translate(radius, radius);
```

Giảm bán kính của đồng hồ (xuống 90%) để đồng hồ đẹp hơn trong không gian của canvas:

```
radius = radius * 0.90;
```

Xây dựng một hàm để trung bày (vẽ) đồng hồ:

```
function drawClock() {  
    ctx.arc(0, 0, radius, 0, 2*Math.PI);  
    ctx.fillStyle = "white";  
    ctx.fill();  
}
```

## Phần II – Dựng mặt

Đồng hồ cần phải có mặt. Chúng ta sẽ viết một hàm JavaScript để xây dựng mặt cho nó (bổ sung các hàm này vào trong cặp thẻ <script> ở Phần I nhé, và nhớ bỏ hàm drawClock() đang viết dang dở đi :P):

### JavaScript:

```
function drawClock() {
    drawFace(ctx, radius);
}

function drawFace(ctx, radius) {
    var grad;

    ctx.beginPath();
    ctx.arc(0, 0, radius, 0, 2*Math.PI);
    ctx.fillStyle = 'white';
    ctx.fill();

    grad = ctx.createRadialGradient(0,0,radius*0.95, 0,0,radius*1.05);
    grad.addColorStop(0, '#0f52a0');
    grad.addColorStop(0.5, 'white');
    grad.addColorStop(1, '#0f52a0');
    ctx.strokeStyle = grad;
    ctx.lineWidth = radius*0.1;
    ctx.stroke();

    ctx.beginPath();
    ctx.arc(0, 0, radius*0.1, 0, 2*Math.PI);
    ctx.fillStyle = '#0f52a0';
    ctx.fill();
}
```

### Code ở trên viết gì?

Chúng ta viết hàm drawFace() để vẽ mặt cho chiếc đồng hồ này:

```
function drawClock() {
    drawFace(ctx, radius);
}

function drawFace(ctx, radius) {
}
```

Vẽ một hình trong màu trắng:

```
ctx.beginPath();
ctx.arc(0, 0, radius, 0, 2*Math.PI);
ctx.fillStyle = 'white';
ctx.fill();
```

Tạo một radial gradient (sử dụng 95% và 105% bán kính của đồng hồ):

```
grad = ctx.createRadialGradient(0,0,radius*0.95, 0,0,radius*1.05);
```

Tạo ba mức màu (color stop), tương ứng với ba vùng trong, giữa và ngoài cùng của các vòng tròn tạo ở trên (radial gradient):

```
grad.addColorStop(0, '#0f52a0');
grad.addColorStop(0.5, 'white');
grad.addColorStop(1, '#0f52a0');
```



Các Color stop tạo ra một hiệu ứng 3D cho chiếc đồng hồ.

Định nghĩa gradient như một kiểu (style) stroke của đối tượng vẽ ctx:

```
ctx.strokeStyle = grad;
```

Định nghĩa độ rộng đường thẳng cho đối tượng vẽ ctx (bằng 10% bán kính):

```
ctx.lineWidth = radius * 0.1;
```

Vẽ đường tròn (hàm stroke thực thi việc vẽ và hiển thị các hình tròn được vẽ với kiểu là grad):

```
ctx.stroke();
```

Vẽ tâm của đồng hồ (trục quay của các kim đồng hồ):

```
ctx.beginPath();
ctx.arc(0, 0, radius*0.1, 0, 2*Math.PI);
ctx.fillStyle = '#0f52a0';
ctx.fill();
```

## Phần III – Gắn số

Đồng hồ cần các con số chỉ giờ. Chúng ta sẽ viết một hàm JavaScript để gắn các con số cho nó (bổ sung hàm này vào trong cặp thẻ `<script>` ở Phần I nhé và thêm lời gọi hàm **drawNumbers** vào trong hàm **drawClock**):

### JavaScript:

```
function drawClock() {
    drawFace(ctx, radius);
    drawNumbers(ctx, radius);
}

function drawNumbers(ctx, radius) {
    var ang;
    var num;
    ctx.font = radius*0.15 + "px arial";
    ctx.textBaseline="middle";
    ctx.textAlign="center";
    for(num= 1; num < 13; num++){
        ang = num * Math.PI / 6;
        ctx.rotate(ang);
        ctx.translate(0, -radius*0.85);
        ctx.rotate(-ang);
        ctx.fillText(num.toString(), 0, 0);
        ctx.rotate(ang);
        ctx.translate(0, radius*0.85);
        ctx.rotate(-ang);
    }
}
```

### Code ở trên viết gì?

Thiết lập kích thước phông chữ (cho đối tượng vẽ) bằng 15% bán kính đồng hồ:

```
ctx.font = radius*0.15 + "px arial";
```

Thiết lập vị trí gắn số (nằm giữa - middle - theo chiều thẳng đứng và nằm ở trung tâm - center - theo chiều ngang đối với vị trí gắn số):

```
ctx.textBaseline="middle";
ctx.textAlign="center";
```

Tính toán vị trí gắn số (cho 12 số) bằng 85% bán kính, quay một góc  $\pi/6$  (radian) với mỗi số:

```
for(num= 1; num < 13; num++) {
    ang = num * Math.PI / 6;
    ctx.rotate(ang);
    ctx.translate(0, -radius*0.85);
    ctx.rotate(-ang);
    ctx.fillText(num.toString(), 0, 0);
    ctx.rotate(ang);
    ctx.translate(0, radius*0.85);
    ctx.rotate(-ang);
}
```

## Phần IV – Lắp đặt kim

Đồng hồ còn cần có các kim (giờ, phút, giây). Chúng ta sẽ viết một hàm JavaScript để lắp đặt kim cho nó (bổ sung hàm này vào trong cặp thẻ <script> ở Phần I và thêm lệnh gọi hàm drawTime vào hàm drawClock nhé):

### JavaScript:

```
function drawClock() {
    drawFace(ctx, radius);
    drawNumbers(ctx, radius);
    drawTime(ctx, radius);
}

function drawTime(ctx, radius){
    var now = new Date();
    var hour = now.getHours();
    var minute = now.getMinutes();
    var second = now.getSeconds();
    // Giờ
    hour=hour%12;
    hour=(hour*Math.PI/6)+(minute*Math.PI/(6*60))+(second*Math.PI/(360*60));
    drawHand(ctx, hour, radius*0.5, radius*0.07);
    // Phút
    minute=(minute*Math.PI/30)+(second*Math.PI/(30*60));
    drawHand(ctx, minute, radius*0.8, radius*0.07);
    // Giây
    second=(second*Math.PI/30);
    ctx.strokeStyle = '#f37126';
    drawHand(ctx, second, radius*0.9, radius*0.02);
}

function drawHand(ctx, pos, length, width) {
    ctx.beginPath();
    ctx.lineWidth = width;
    ctx.lineCap = "round";
    ctx.moveTo(0,0);
    ctx.rotate(pos);
    ctx.lineTo(0, -length);
    ctx.stroke();
    ctx.rotate(-pos);
}
```

### Code ở trên viết gì?

Sử dụng Date có sẵn trong JavaScript để đặt giờ, phút và giây cho đồng hồ của chúng ta:

```
var now = new Date();
var hour = now.getHours();
var minute = now.getMinutes();
var second = now.getSeconds();
```

Tính toán góc của kim giờ, và vẽ nó với một chiều dài (50% bán kính), và một chiều rộng (7% của bán kính). Nên nhớ kim quay được 1 vòng tương ứng với  $360^\circ$  (tức  $2\pi$ ), như vậy mỗi giờ tương ứng với một góc  $360^\circ/12 = 2\pi/12 = \pi/6$ , để tính góc của kim giờ còn phải căn cứ vào phút và giây tại thời điểm đó:

```
hour=hour%12;
hour=(hour*Math.PI/6)+(minute*Math.PI/(6*60))+(second*Math.PI/(360*60));
drawHand(ctx, hour, radius*0.5, radius*0.07);
```

Sử dụng cùng cách thức trên với kim phút và kim giây.



Hàm `drawHand()` không cần phải giải thích. Nó chỉ vẽ các kim (đường thẳng) với độ dài, độ rộng đã được tính toán (sẽ quay một góc tương ứng vị trí của kim trước khi vẽ kim - `ctx.rotate(pos)`).

## Phần V – Lắp pin và chạy

Để chạy chiếc đồng hồ này chúng ta cần gọi hàm drawClock() sau một khoảng thời gian:

### JavaScript:

```
var canvas = document.getElementById("canvas");
var ctx = canvas.getContext("2d");
var radius = canvas.height / 2;
ctx.translate(radius, radius);
radius = radius * 0.90
//drawClock();
setInterval(drawClock, 1000);
```

### Code ở trên viết gì?

Việc cần làm bây giờ chỉ là bạn phải gọi hàm drawClock() sau một khoảng thời gian.

Để làm việc này chúng ta sẽ thay thế dòng code:

```
drawClock();
```

Bằng dòng code:

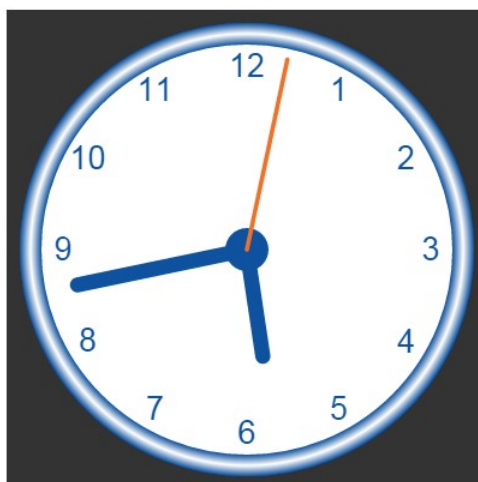
```
setInterval(drawClock, 1000);
```



Khoảng thời gian được tính bằng mili giây. Như vậy hàm drawClock() sẽ được gọi 1000 mili giây (tức 1 giây) một lần.

Code hoàn chỉnh, nếu bạn cần,

**Vậy là đồng hồ Canvas đã chạy – tích tắc !!!**

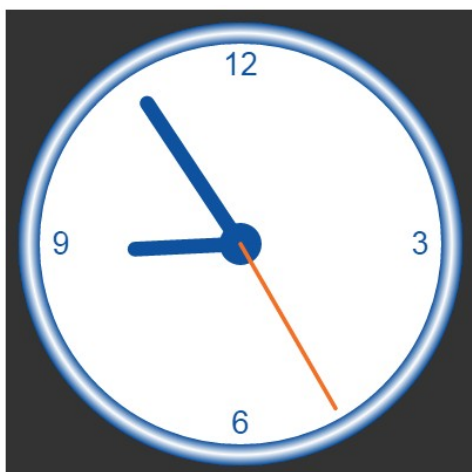


## Phần VI – Thử thách thợ đồng hồ

### 1. Lắp đặt số La Mã thay cho số Thập phân



### 2. Tháo bớt số chỉ để lại 4 số: 3, 6, 9, 12



### 3. Sơn lại đồng hồ



### 4. Sơn thêm vạch chỉ giờ



5. Thay kim giây có phần đuôi dài hơn



6. Thợ sáng tạo thêm nhé...

Chúc thợ ngày càng lành nghề :o)