# Orifice Meter Experiment using JS

## Project Documentation and Test Cases for *Orifice Meter Experiment Using JavaScript*

*Prepared By B. N. V. VISWESWAR ABHAY*

# Documentation

*Prepared By B. N. V. VISWESWAR ABHAY*

Orifice Meter Experiment using JS

# 1.Introduction:

An orifice meter or orifice plate is a device used for measuring the rate of flow of a fluid through a pipe. The objective of this simulation is to determine the coefficient of discharge (Cd) using the orifice meter.
This simulation is written in JavaScript.

# 2.Languages Used:

1) HTML
2) CSS
3) JAVA SCRIPT

*Prepared By B. N. V. VISWESWAR ABHAY*

# 3.Working Description:

## Functionality of all the functions and scripts used in the program.

### 3.1 layout.js :

**Functions:**

1.fill_text

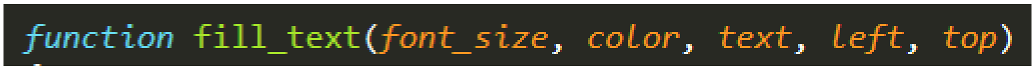
2.triangle

3.draw_line

4.lines

5.line_filling

### fill_text function



```
function fill_text(font_size, color, text, left, top)
```

**Description-**

Fills the text in the canvas.

**Parameters**

font_size: font size of the text

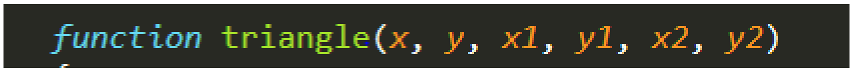
color: color of the text

text: text content

left: distance of text from left border

top: distance of the text form the top border

## triangle function



```
function triangle(x, y, x1, y1, x2, y2)
```

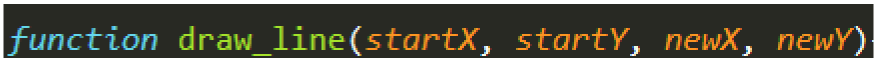
## Description-

Draws a triangle at desired location

## Parameters

x: initial x coordinate of the triangle
y: initial y coordinate of the triangle
x1: second x coordinate of the triangle
y1: second y coordinate of the triangle
x2: third x coordinate of the triangle
y2: third y coordinate of the triangle

# Draw_line function



```
function draw_line(startX, startY, newX, newY)
```

## Description-

Draws a line between desired coordinates.

## Parameters

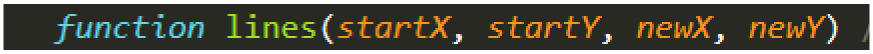
startX: initial x coordinate of the line

startY: initial y coordinate of the line

newX: final x coordinate of the line

newY: final y coordinate of the line

# lines function:



```
function lines(startX, startY, newX, newY)
```

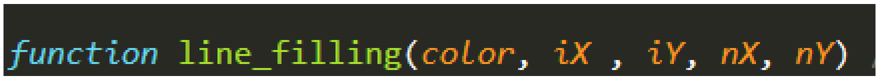
## Description-

Draws a  black line between  the desired coordinates.

## Parameters

startX: initial x coordinate of the line
startY: initial y coordinate of the line
newX: final x coordinate of the line
newY: final y coordinate of the line

# line_filling function:



```
function line_filling(color, iX , iY, nX, nY)
```

## Description-

Draws a line with desired width , height and colour at a desired position.

## Parameters

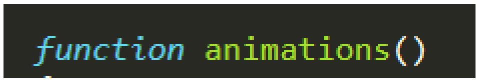
color: color of the rectangle
iX: initial point of the rectangle
iY: initial point of the rectangle
nX: width of the rectangle from iX
nY: height of the rectangle from iY

# 3.2 Animation.js:

### Functions:

1. animations

## animation function



```
function animations()
```

**Description-**

Performs the animation as per the required simulation (filling the tube with water).

**Parameters-**

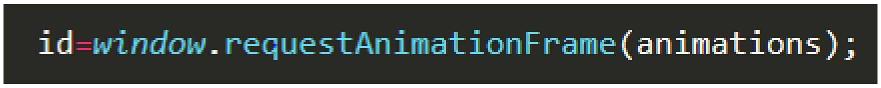
No parameters required.

### Animations frames used:

1. **window.requestAnimationFrame()**

2. **window.cancleAnimationFrame()**

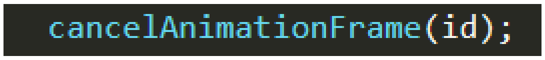
## window.requestAnimationFrame()



```
id=window.requestAnimationFrame(animations);
```

The `window.requestAnimationFrame()` method tells the browser to perform an animation and requests that the browser to call **animations** function to update an animation before the next repaint. The method takes a callback as an argument to be invoked before the repaint.

## window.cancleAnimationFrame()



```
cancelAnimationFrame(id);
```

The `window.cancelAnimationFrame()` method cancels an animation frame request previously scheduled through a call to `window.requestAnimationFrame()`.

# 3.3. calculator.js

## Functions:

1. Results

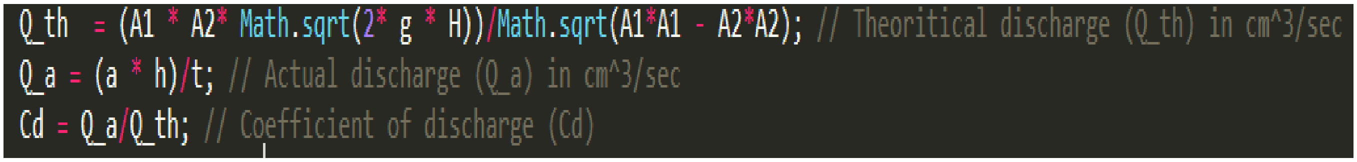
## Results function

```
function Results()
```

## Description-

This function performs all the necessary calculations required to get *Theoretical discharge (Q_th)*, *Actual discharge (Q_a)* and *coefficient of discharge (Cd).*

## Formulas Used:

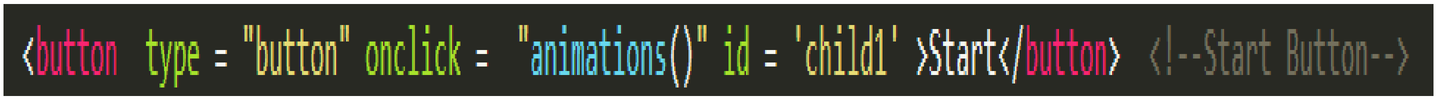


```
Q_th  = (A1 * A2* Math.sqrt(2* g * H))/Math.sqrt(A1*A1 - A2*A2); // Theoritical discharge (Q_th) in cm^3/sec
Q_a = (a * h)/t; // Actual discharge (Q_a) in cm^3/sec
Cd = Q_a/Q_th; // Coefficient of discharge (Cd)
```

# 4.Buttons:

1. Start
2. Re-set
3. Calculate

### Start Button:

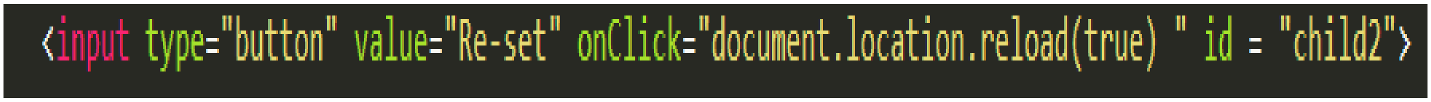


```
<button  type = "button" onclick =  "animations()" id = 'child1' >Start</button>  <!--Start Button-->
```

### Description-

On clicking on this button animations() functions starts executing.

### Re-set Button:

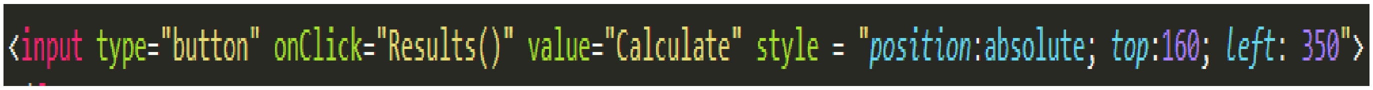


```
<input type="button" value="Re-set" onClick="document.location.reload(true) " id = "child2">
```

### Description-

On clicking this button the simulation goes to the starting stage and page gets refreshed.

## Calculate Button:

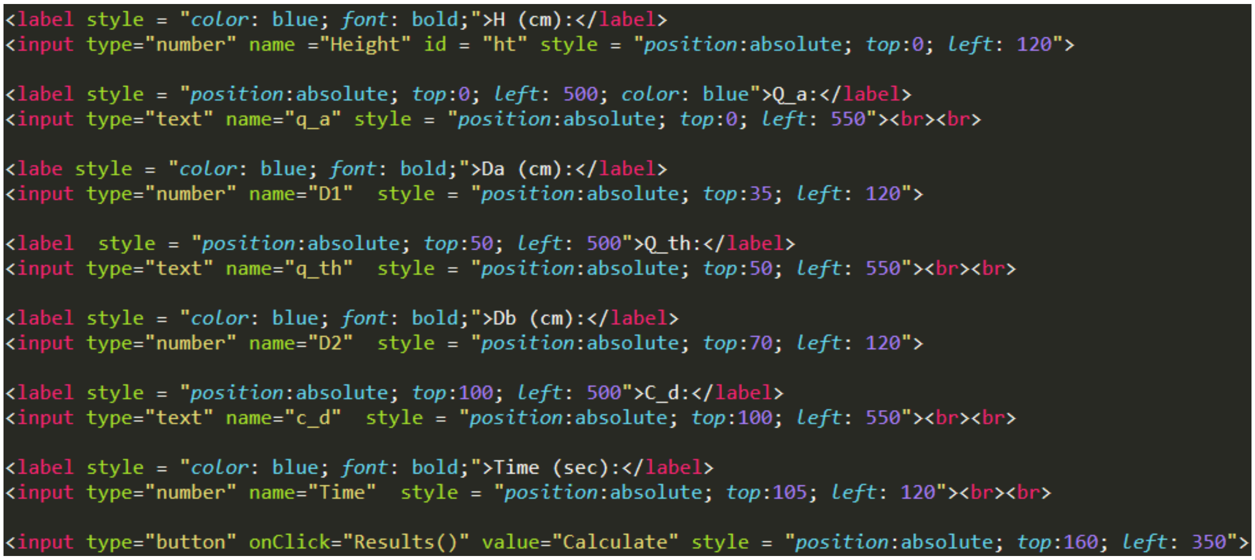


```
<input type="button" onClick="Results()" value="Calculate" style = "position:absolute; top:160; left: 350">
```

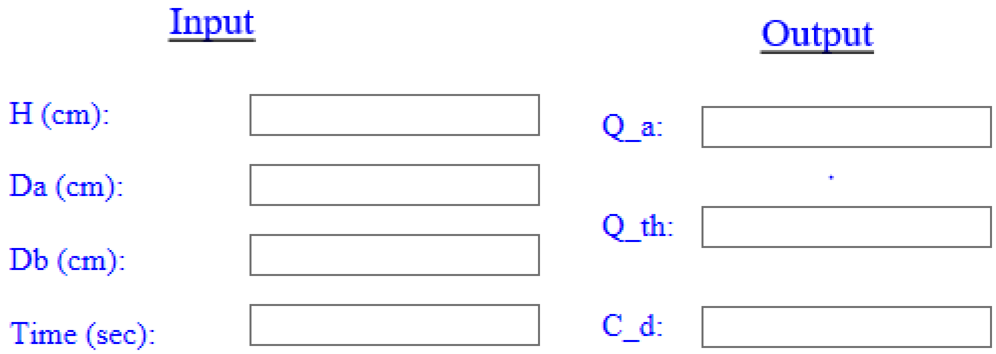
## Description-

On clicking the calculate button Results() function will be invoked and calculations will be performed and displayed on the output form.

# 5.Input and Output:



```
<label style = "color: blue; font: bold;">H (cm):</label>
<input type="number" name ="Height" id = "ht" style = "position:absolute; top:0; left: 120">

<label style = "position:absolute; top:0; left: 500; color: blue">Q_a:</label>
<input type="text" name="q_a" style = "position:absolute; top:0; left: 550"><br><br>

<labe style = "color: blue; font: bold;">Da (cm):</label>
<input type="number" name="D1"  style = "position:absolute; top:35; left: 120">

<label  style = "position:absolute; top:50; left: 500">Q_th:</label>
<input type="text" name="q_th"  style = "position:absolute; top:50; left: 550"><br><br>

<label style = "color: blue; font: bold;">Db (cm):</label>
<input type="number" name="D2"  style = "position:absolute; top:70; left: 120">

<label style = "position:absolute; top:100; left: 500">C_d:</label>
<input type="text" name="c_d"  style = "position:absolute; top:100; left: 550"><br><br>

<label style = "color: blue; font: bold;">Time (sec):</label>
<input type="number" name="Time"  style = "position:absolute; top:105; left: 120"><br><br>

<input type="button" onClick="Results()" value="Calculate" style = "position:absolute; top:160; left: 350">
```



# 6.Test Cases:

## Input/Output case 1:

**Input:**

**H = 40 cm**
**Da = 90 cm**
**Db = 30 cm**
**Time = 8  sec**

**Output:**

| **Expected Output** | **Actual Output** |
|---|---|
| **Q_a = 1.8** | **Q_a = 1.8000** |
| **Q_th = 199255.0805** | **Q_th = 199255.0805** |
| **C_d = 0.0000090336** | **C_d = 0.0000090336** |

# Input/Output case 2:

**Input:**

**H = 20 cm**<br>
**Da = 10 cm**<br>
**Db = 5 cm**<br>
**Time = 2  sec**

**Output:**

| **Expected Output** | **Actual Output** |
|---|---|
| **Q_a = 3.6** | **Q_a = 3.6000** |
| **Q_th = 4017.0640** | **Q_th = 4017.0640** |
| **C_d = 0.0008961769** | **C_d = 0.0008961769** |