**Java Script Code to Calculate Speed / Distance of Falling Object**

By Stephen Bucaro

Bovenkant formulier

|  |  |  |
| --- | --- | --- |
|  | **Gravity Calculator** |  |
|  | Feet Meters |  |
| Time (sec): |  |  |
| Distance: |  | feet |
| Velocity: |  | feet/sec |
|  | Calculate |  |

Onderkant formulier

The Greek philosopher Aristotle taught that everything was made of four elements: air, fire, earth. and water. He taught that lighter objects fall more slowly than heavier ones because they contain more of the lighter elements, air and fire. Heavier objects fall faster because they contain more of the heavier elements, earth and water. Aristotle also taught that an object will stop moving when the force acting on it is removed.

The Italian scientist Galileo proved that, in the absence of friction, all bodies accelerate toward the earth with the same rate of acceleration, regardless of their weight. He also proved that, in the absence of friction, bodies would keep moving forever.

I like to brag about Italian scientists because I’m Italian. In fact, according to me, almost all discoveries ever made where made by Italians. You throw in Leonardo da Vinci, Luigi Galvani, Guglielmo Marconi, and Enrico Fermi, and I think I'm pretty much correct.

But in reality Aristotle lived between 384 BC and 322 BC, long before the Renaissance. when science was done by thinking about things, not actually doing experiments. So for his time, Aristotle was a genius. While Galileo lived between 1564 and 1642, right in the heart of the Renaissance. The Renaissance, by the way, started in Italy.

The acceleration due to gravity is 32.173 feet per second per second. Using this value, and the number of seconds an object has been falling, you can calculate how far it has fallen or how fast the object is moving. In this article, you'll learn how to use Java Script to create a *Gravity Calculator* that that does this and much more.

Note: The value of 32.173 ft/s2 is accurate only at sea level at the equator. Gravity decreases with greater distance from the Earth's center. And since the earth is not a perfect sphere, it also varies with latitude.

I have programmed many Java Script calculators and discovered that, the process of coding any calculator can be broken down into a step-by-step procedure. Shown below are the steps to build a calculator.

1. Acquire the formula  
2. Determine the inputs and outputs  
3. Design the html form  
4. Write Java Script to process the form  
5. Test and Final Touches

**Step 1: Gravity Formulas**

The formula for the acceleration due to gravity is:

g = 32.173 ft/s2 (feet per second squared)

or

g = 9.8 m/s2 (meters per second squared)

That formula rearranged to calculate distance traveled in time *t* is:

d = gt2/2

That formula rearranged to calculate time elapsed to fall distance *d* is:

t = sqrt(2d/g)

You can also calculate the instantaneous velocity of a falling object after elapsed time *t*:

vi = gt

Or the instantaneous velocity of a falling object that has traveled distance *d*:

vi = sqrt(2gd)

Bovenkant formulier



Onderkant formulier

**Java Script Code to Calculate Speed / Distance of Falling Object**

**Step 2. The Inputs and Outputs**

Since the acceleration due to gravity is a constant, you need only one other parameter to calculate any other given parameter. In other words, if the user enters elapsed time in the calculator, we can calculate distance traveled and instantaneous velocity at the end of that elapsed time.

Similarly, if the user enters distance traveled in the calculator, we can calculate elapsed time and instantaneous velocity. In fact, if the user enters instantaneous velocity, we can calculate elapsed time and distance traveled.

**Step 3. The html Form**

The code for the html form for the Gravity Calculator is shown below. Paste this code into the *body* section of your web page.

<form action="javascript:return false;">

Time (sec): <input id="t" type="text" onchange="Gravity('t')"></input><br />

Distance: <input id="d" type="text" onchange="Gravity('d')"></input><br />

Velocity: <input id="v" type="text" onchange="Gravity('v')"></input><br />

<button>Calculate</button>

</form>

Note the form tag's *action* attribute is set to *javascript:return false*. This is to prevent the form from being submitted to the server. The form consists of three text boxes, each of which on their *onchange* event executes a function named *Gravity*. Also note the form's *Calculate* button, which does nothing. This button is here strictly for Internet Explorer users because the *onchange* event doesn't work properly in Internet Explorer.

**Step 4. Java Script to Process the Form**

Shown below is the Java Script code for the Gravity Calculator. Paste this code into the *head* section of your web page.

<script type="text/javascript">

function Gravity(box)

{

if(box == "t")

{

var seconds = document.getElementById(box).value;

var feet = (32.173 \* seconds \* seconds)/2;

var miles = feet/5280;

document.getElementById("d").value = feet;

var vi = 32.173 \* seconds;

document.getElementById("v").value = vi;

}

if(box == "d")

{

var feet = document.getElementById(box).value;

var seconds = Math.sqrt( ( (2 \* feet) / 32.173) );

document.getElementById("t").value = seconds;

var vi = 32.173 \* seconds;

document.getElementById("v").value = vi;

}

if(box == "v")

{

var velocity = document.getElementById(box).value;

var seconds = velocity / 32.173;

document.getElementById("t").value = seconds;

var feet = (32.173 \* seconds \* seconds)/2;

document.getElementById("d").value = feet;

}

}

</script>

When someone enters a number in one of the form's text boxes and presses the keyboard [Enter] key, that text box's *onchange* event executes the *Gravity* function, passing its *id* to the function. The code uses *if* statements to determine which box generated the event, then calculates the other parameters.

For example, if the user enters a distance, the *if(box == "d")* statement calculates the elapsed time *seconds*, and the instantaneous velocity *vi*. It uses the document's *getElementById* method to place the results in the proper text box.

As I mentioned earlier, because the *onchange* event doesn't work properly in Internet Explorer, IE users will need to click on the [Calculate] button. Actually they can click anywhere outside the text box because IE's onchange event works only when the text box loses the focus. Remember, the [Calculate] button actually does nothing.

|  |  |
| --- | --- |
| Bovenkant formulier    Onderkant formulier | Bovenkant formulier    Onderkant formulier |

**Java Script Code to Calculate Speed / Distance of Falling Object**

**5. Final Touches**

Final touches involve testing for proper operation of the calculator, and good appearance of the form. When using the calculator, you may have noticed that the results return numbers with a large number of digits to the right of the decimal point. I've written a function called *roundTo4* that I use with all my calculators to round the results to a maximum of 4 digits to the right of the decimal point. Shown below is the *roundTo4* function and how it's implemented in the *getElementById* statements.

<script type="text/javascript">

function roundTo4(raw)

{

strFloat = raw.toString();

var newFloat;

if(strFloat.indexOf(".") > 0 && strFloat.length - strFloat.indexOf(".") > 5)

{

strFloat = strFloat.substring(0,strFloat.indexOf(".") + 6);

var numFloat = parseFloat(strFloat);

var rndFloat = Math.round(numFloat \* 10000);

var temp1Float = rndFloat / 10000;

var temp2Float = temp1Float.toString();

newFloat = temp2Float.substring(0,temp2Float.indexOf(".") + 5);

}

else

{

newFloat = strFloat;

}

return newFloat;

}

function Gravity(box)

{

if(box == "t")

{

// calculate distance

var seconds = document.getElementById(box).value;

var feet = (32.173 \* seconds \* seconds)/2;

document.getElementById("d").value = roundTo4(feet);

// calculate velocity

var vi = 32.173 \* seconds;

document.getElementById("v").value = roundTo4(vi);

}

if(box == "d")

{

// calculate time

var feet = document.getElementById(box).value;

var seconds = Math.sqrt( ( (2 \* feet) / 32.173) );

document.getElementById("t").value = roundTo4(seconds);

// calculate velocity

var vi = 32.173 \* seconds;

document.getElementById("v").value = roundTo4(vi);

}

if(box == "v")

{

// calculate time

var velocity = document.getElementById(box).value;

var seconds = velocity / 32.173;

document.getElementById("t").value = roundTo4(seconds);

// calculate distance

var feet = (32.173 \* seconds \* seconds)/2;

document.getElementById("d").value = roundTo4(feet);

}

}

</script>

**Adding Metric System Calculation**

Our Gravity Calculator works only with English system units. However, most of the world uses the metric system. Therefore, one of our final touches might include making the calculator work with metric units as well as English units. This is actually quite easy, basically you just repeat the same code, replacing 32.173 with 9.8, and labeling the output as *meters* instead of *feet*. Then you put a radio button in the form to select *English* or *Metric*. Required changes to the form html code are shown below.

<form action="javascript:return false;">

<input id="system" type="radio" name="system" value="feet" onclick="clearForm()" checked />Feet

<input type="radio" name="system" value="meters" onclick="clearForm()" />Meters<br />

Time (sec):<input id="t" style="text-align:right" type="text" onchange="Gravity('t')"></input><br />

Distance: <input id="d" style="text-align:right" type="text" onchange="Gravity('d')"></input><label id="ld">feet</label><br />

Velocity: <input id="v" style="text-align:right" type="text" onchange="Gravity('v')"></input><label id="lv">feet/sec</label><br />

<button>Calculate</button><br />

</form>

Note the addition of the radio buttons. The *onclick* event of each radio button executes a function named *clearForm*. Also note the htnl *label* elements added next to the Distance and Velocity text boxes. When the user sets the Meters radio button, the *clearForm* function will change the text in the label elements from *feet* and *feet/sec* to *meters* and *meters/sec*. The code for the *clearForm* function is shown below.

function clearForm()

{

document.getElementById("t").value = "";

document.getElementById("d").value = "";

document.getElementById("v").value = "";

if(document.getElementById("system").checked)

{

document.getElementById("ld").innerHTML = "feet";

document.getElementById("lv").innerHTML = "feet/sec";

}

else

{

document.getElementById("ld").innerHTML = "meters";

document.getElementById("lv").innerHTML = "meters/sec";

}

}

|  |  |
| --- | --- |
| Bovenkant formulier    Onderkant formulier | Bovenkant formulier    Onderkant formulier |

# Java Script Code to Calculate Speed / Distance of Falling Object

Shown below is the entire code for the Gravity Calculator, including the *clearForm* function, the *roundTo4* function, the repeat of the main block of code to calculate in *meters*, and the updated html form.

<html>

<head>

<script type="text/javascript">

function clearForm()

{

document.getElementById("t").value = "";

document.getElementById("d").value = "";

document.getElementById("v").value = "";

if(document.getElementById("system").checked)

{

document.getElementById("ld").innerHTML = "feet";

document.getElementById("lv").innerHTML = "feet/sec";

}

else

{

document.getElementById("ld").innerHTML = "meters";

document.getElementById("lv").innerHTML = "meters/sec";

}

}

function roundTo4(raw)

{

strFloat = raw.toString();

var newFloat;

if(strFloat.indexOf(".") > 0 && strFloat.length - strFloat.indexOf(".") > 5)

{

strFloat = strFloat.substring(0,strFloat.indexOf(".") + 6);

var numFloat = parseFloat(strFloat);

var rndFloat = Math.round(numFloat \* 10000);

var temp1Float = rndFloat / 10000;

var temp2Float = temp1Float.toString();

newFloat = temp2Float.substring(0,temp2Float.indexOf(".") + 5);

}

else

{

newFloat = strFloat;

}

return newFloat;

}

function Gravity(box)

{

if(document.getElementById("system").checked)

{

if(box == "t")

{

// calculate distance

var seconds = document.getElementById(box).value;

var feet = (32.173 \* seconds \* seconds)/2;

document.getElementById("d").value = roundTo4(feet);

// calculate velocity

var vi = 32.173 \* seconds;

document.getElementById("v").value = roundTo4(vi);

}

if(box == "d")

{

// calculate time

var feet = document.getElementById(box).value;

var seconds = Math.sqrt( ( (2 \* feet) / 32) );

document.getElementById("t").value = roundTo4(seconds);

// calculate velocity

var vi = 32.173 \* seconds;

document.getElementById("v").value = roundTo4(vi);

}

if(box == "v")

{

// calculate time

var velocity = document.getElementById(box).value;

var seconds = velocity / 32;

document.getElementById("t").value = roundTo4(seconds);

// calculate distance

var feet = (32.173 \* seconds \* seconds)/2;

document.getElementById("d").value = roundTo4(feet);

}

}

else

{

if(box == "t")

{

// calculate distance

var seconds = document.getElementById(box).value;

var meters = (9.8 \* seconds \* seconds)/2;

var miles = meters/5280;

document.getElementById("d").value = roundTo4(meters);

// calculate velocity

var vi = 9.8 \* seconds;

document.getElementById("v").value = roundTo4(vi);

}

if(box == "d")

{

// calculate time

var feet = document.getElementById(box).value;

var seconds = Math.sqrt( ( (2 \* feet) / 9.8) );

document.getElementById("t").value = roundTo4(seconds);

// calculate velocity

var vi = 9.8 \* seconds;

document.getElementById("v").value = roundTo4(vi);

}

if(box == "v")

{

// calculate time

var velocity = document.getElementById(box).value;

var seconds = velocity / 9.8;

document.getElementById("t").value = roundTo4(seconds);

// calculate distance

var feet = (9.8 \* seconds \* seconds)/2;

document.getElementById("d").value = roundTo4(feet);

}

}

}

</script>

</head>

<body>

<form action="javascript:return false;">

<input id="system" type="radio" name="system" value="feet" onclick="clearForm()" checked />Feet

<input type="radio" name="system" value="meters" onclick="clearForm()" />Meters<br />

Time (sec):<input id="t" style="text-align:right" type="text" onchange="Gravity('t')"></input><br />

Distance: <input id="d" style="text-align:right" type="text" onchange="Gravity('d')"></input><label id="ld">feet</label><br />

Velocity: <input id="v" style="text-align:right" type="text" onchange="Gravity('v')"></input><label id="lv">feet/sec</label><br />

<button>Calculate</button><br />

</form>

</body>

</html>

Note in the *Gravity* function how a *if/else* statement that uses *document.getElementById("system").checked* to select the proper block of code based upon the users setting of the *feet/meters* radio buttons.

You can spend more time with final touches than any other part of calculator design. For example as a final touch, I added *style* tags to the text boxes to align the text to the right side of the text boxes. You're free to continue adding final touches as you desire. For example a title over the form might be nice. but that's it for this article.

In this article you learned how to use Java Script to perform gravity calculations, how to use the *onchange* event to execute functions, how to use radio buttons to select between English and metric units, and much more.