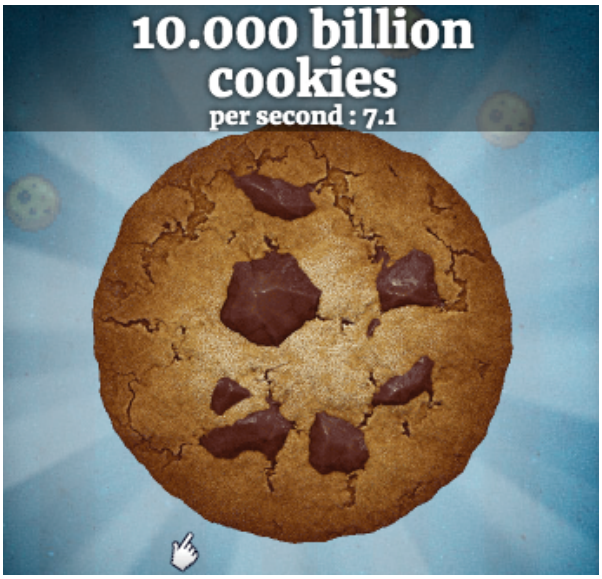
Cookie Clicker

Introduction

This tutorial will teach you the basics of creating an example Cookie Clicker application.

https://orteil.dashnet.org/cookieclicker/

Online app	Your app
	 6

Steps

Step 1 - Create your project folder

Use your Finder or File Explorer to navigate to your workspace. Create a new folder called **cookie**.

CS Tip: files and folders should be 8 characters or less with no spaces or special characters

Step 2 - Create your project files

Create 3 new files in your new project “cookie” folder, using your HTML editor (i.e., Brackets) :

1. cookie.html
2. cookie.css
3. cookie.js

Step 3 - Download your cookie image

Download https://i.postimg.cc/9z5C1S3h/cookie.png to your new “cookie” project folder.

Step 4 - Create your canvas with a background image

Use your HTML editor to create the following code in your cookie.html. Remember to save and view your code in a browser.

```
<html>
      <canvas id="mycanvas" style="background-image:url('cookie.png');"
width="266" height="264">
      </canvas>
</html>
```

CS Tip: style tags inside another HTML are called internal CSS

Step 5 - Link and source your CSS and Javascript to your HTML

```
<html>
      <link rel="stylesheet" type ="text/css" href="cookie.css" />
      <canvas id="mycanvas" style="background-image:url('cookie.png');"
width="266" height="264"></canvas>
      <script src="cookie.js"></script>
</html>
```

Step 6 - Add score paragraph tag that will contain updated user score

```
<html>
      <link rel="stylesheet" type ="text/css" href="cookie.css" />
      <canvas id="mycanvas" style="background-image:url('cookie.png');"
width="266" height="264"></canvas>
      <p id="score"></p>
      <script src="cookie.js"></script>
</html>
```

Step 7 - Add CSS to create shaking animation for Cookie

Use your editor and example https://www.w3schools.com/howto/howto_css_shake_image.asp from W3Schools, to add a shaking animation to your cookie image, in your cookie.css.

CS Tip: Remember that no <style> tag is required for your external CSS in your cookie.css file.

```
#mycanvas:hover {
    animation: shake 0.5s;
    animation-iteration-count: infinite;
}

@keyframes shake {
    0% { transform: translate(1px, 1px) rotate(0deg); }
    10% { transform: translate(-1px, -2px) rotate(-1deg); }
    20% { transform: translate(-3px, 0px) rotate(1deg); }
    30% { transform: translate(3px, 2px) rotate(0deg); }
    40% { transform: translate(1px, -1px) rotate(1deg); }
    50% { transform: translate(-1px, 2px) rotate(-1deg); }
    60% { transform: translate(-3px, 1px) rotate(0deg); }
    70% { transform: translate(3px, 1px) rotate(-1deg); }
    80% { transform: translate(-1px, -1px) rotate(1deg); }
    90% { transform: translate(1px, 2px) rotate(0deg); }
    100% { transform: translate(1px, -2px) rotate(-1deg); }
}
```

Step 8 - Find your canvas tag in your JavaScript file

Use your editor to edit your cookie.js file. Add this code to gain control of canvas and set up the 2d drawing context to access the drawing tools of <canvas>.

```
// find and gain control of your canvas tag in Javascript
var c = document.getElementById("mycanvas");
var ctx = c.getContext('2d');
```

Step 9 - Find and populate your score paragraph tag in your Javascript file

Create a cookie counter variable and set it to 0, find the paragraph tag and set the paragraph to the value stored in the cookies variable.

```
var c = document.getElementById("mycanvas");
var ctx = c.getContext('2d');

var cookies = 0;
var s = document.getElementById("score");
s.innerHTML = cookies
```

Step 10 - Add an event listener to the canvas in your JavaScript file

Event listeners can be set on on any HTML element, and listen for user events (such as “click” or “mouseover”). In the case of the event listener code below, there is a click event added to the canvas tag that calls a function called ***draw*** (next step!).

W3Schools site: https://www.w3schools.com/jsref/met_element_addeventlistener.asp

```
var c = document.getElementById("mycanvas");
var ctx = c.getContext('2d');

var cookies = 0;
var s = document.getElementById("score");
s.innerHTML = cookies

c.addEventListener("click",draw, true);
```

Step 11 - Create a draw function in your JavaScript file

The draw function will be called each time your cookie is clicked on. In this step, you will create the structure of the function and add code to update the cookie count with each click.

```
var c = document.getElementById("mycanvas");
var ctx = c.getContext('2d');

var cookies = 0;
var s = document.getElementById("score");
s.innerHTML = cookies

c.addEventListener("click",draw, true);

function draw() {
    cookies = cookies + 1;
    s.innerHTML = cookies;
}
```

Step 12 - Add +1 text to each time cookie is clicked in JavaScript file

The 2d drawing context provides access to an array of drawing tools.

You can see them all on W3Schools: https://www.w3schools.com/tags/ref_canvas.asp. The code below uses the font, fillStyle and fillText functionality to draw +1 text at the location of the click.

Also notice that the **event** parameter was added to the function definition to provide access to the event details, such as x and y click locations on the canvas.

```
var c = document.getElementById("mycanvas");
var ctx = c.getContext('2d');

var cookies = 0;
var s = document.getElementById("score");
s.innerHTML = cookies

c.addEventListener("click",draw, true);

function draw(event) {
    cookies = cookies + 1;
    s.innerHTML = cookies;

    ctx.font = "normal 30px Arial";
    ctx.fillStyle = "white";
    ctx.fillText("+1",event.clientX,event.clientY);
}
```

Step 13 - Create an interval function around the fillText() in your JavaScript

To make the added +1 text slowly *fade out*, you must create an interval function.

Algorithm explanation:

1. when user clicks on the cookie, the alpha variable is set to 1.0 (full opacity).
2. Interval function is created which will continue to run every 25 ms until cleared
    a. <canvas> tag opacity (globalAlpha) is set to alpha variable value
    b. … draw +1 text at alpha level
    c. Reduce alpha variable value by .05
    d. If alpha is 0 then clear the +1 text and clear the interval

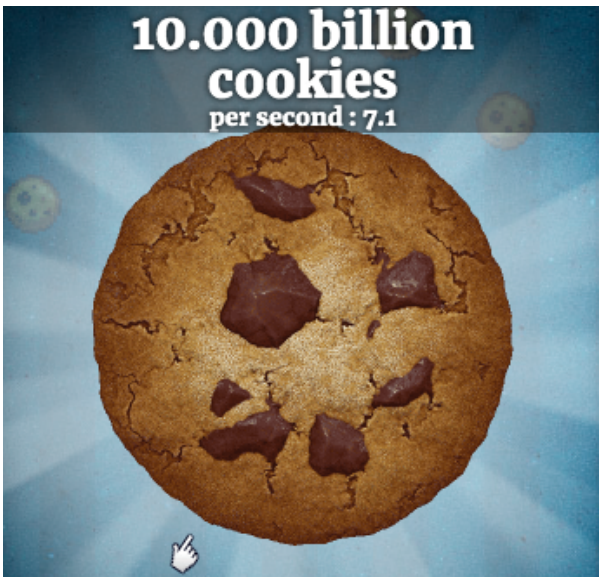
```
var c = document.getElementById("mycanvas");
var ctx = c.getContext('2d');

var cookies = 0;
var s = document.getElementById("score");
s.innerHTML = cookies

c.addEventListener("click",draw, true);

function draw(event) {
    cookies = cookies + 1;
    s.innerHTML = cookies;

    var alpha = 1.0,   // full opacity
    interval = setInterval(function () {
        ctx.globalAlpha = alpha;
        ctx.font = "normal 30px Arial";
        ctx.fillStyle = "white";
        ctx.fillText("+1",event.clientX,event.clientY);
        alpha = alpha - 0.05; // decrease opacity (fade out)
        if (alpha <= 0) {
            ctx.clearRect(0,0,300,300);
            clearInterval(interval);
        }
    }, 25);
}
```

Online app	Your app
	 6

Enrichment #1 - x2 cursor

Score:

24

Congratulations! x2 cursor

Description

If you have gotten all the basic cookie clicker code to work, why not go further to add an achievement that grants a privilege for a user.

In this lesson you will learn to use the following JavaScript techniques:

1. Variables
2. Complex if statement
3. Use functions with parameters
4. Build an HTML table, table using string concatenation
5. Find an HTML tag in script using document.getElementById()
6. Update the value of an HTML tag using innerHTML

Step 1 - Replace hard coded increment value (+1) with variable

This first step is an easy one. Simply create an increment variable and make its value equal to 1. Secondly, replace the hard coded increment value (+1) with the new variable. Lastly, update the fillText() line so increment is not hard coded to +1, but will adapt to the variable.

Test your code, it should work exactly as before.

```
var c = document.getElementById("mycanvas");
var ctx = c.getContext('2d');

var increment = 1;
var cookies = 0;
var s = document.getElementById("score");
s.innerHTML = cookies
```

```
c.addEventListener("click",draw, true);

function draw(event) {
    cookies = cookies + increment;
    s.innerHTML = cookies;

    var alpha = 1.0,   // full opacity
    interval = setInterval(function () {
        ctx.globalAlpha = alpha;
        ctx.font = "normal 30px Arial";
        ctx.fillStyle = "white";
        ctx.fillText("+" + increment,event.clientX,event.clientY);
        alpha = alpha - 0.05; // decrease opacity (fade out)
        if (alpha <= 0) {
            ctx.clearRect(0,0,300,300);
            clearInterval(interval);
        }
    }, 25);
}
```

Step 2 - Check for achievement award condition

If statements can be used to check conditions and perform code accordingly. In this step, check to see if number of clicks is greater than or equal to 20 AND clicks are also less than 30 AND the value of the increment value is also equal to 1.

If these conditions are all true then set the increment to 2 (clicks will go up by 2 now) and call a function called award that accepts two argument values, the increment value and a colour.

```
...

function draw(event) {
    cookies = cookies + increment;
    s.innerHTML = cookies;

    if (cookies >= 20 && cookies < 30 && increment == 1) {
       increment = 2;
       award(increment,"green");
    }

    var alpha = 1.0,   // full opacity
    interval = setInterval(function () {

    ...
```

Step 3 - Create award function that accepts parameters

String concatenation is a technique where you can combine existing strings with variable values to create new string. For example:

```
var name = “joe”;
alert(“my name is “ + name);
```

In this step you will construct a string variable using concatenation containing a table with a <td> with a specified style and increment value. The resultant string will be an HTML table that can be inserted into the inner HTML of a specific HTML tag (with an id of “score”).

```
    ...

    var alpha = 1.0,   // full opacity
    interval = setInterval(function () {
        ctx.globalAlpha = alpha;
        ctx.font = "normal 30px Arial";
        ctx.fillStyle = "white";
        ctx.fillText("+" + increment,event.clientX,event.clientY);
        alpha = alpha - 0.05; // decrease opacity (fade out)
        if (alpha <= 0) {
            ctx.clearRect(0,0,300,300);
            clearInterval(interval);
        }
    }, 25);
}

function award(increment,colour) {
    mytab = "<table><tr>";
    mytab = mytab + "<td style='background-color:" + colour +
";padding:25px;'>";
    mytab = mytab + "Congratulations! x" + increment + " cursor";
    mytab = mytab + "</td>";
    mytab = mytab + "</tr></table>";

    mydiv = document.getElementById("store");
    mydiv.innerHTML = mytab;
}
```

Step 4 - Create a tag to store your score

The final step is to add an HTML tag with an id of “score” to store the updated score for the game.

```
<html>
   <link rel="stylesheet" type="text/css" href="cookie.css"/>
   <canvas id="mycanvas" style="background-image: url('cookie.png')" width="266"
height="264"></canvas>
   <h1>Score:</h1>
   <p style="font-size: 50" id="score"></p>
   <div id="store"></div>
   <script src="cookie.js"></script>
</html>
```

Congratulations!