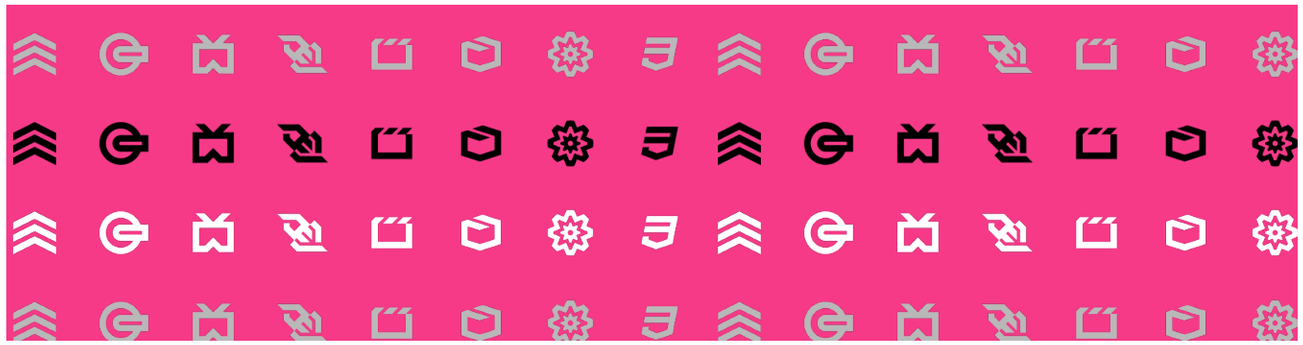
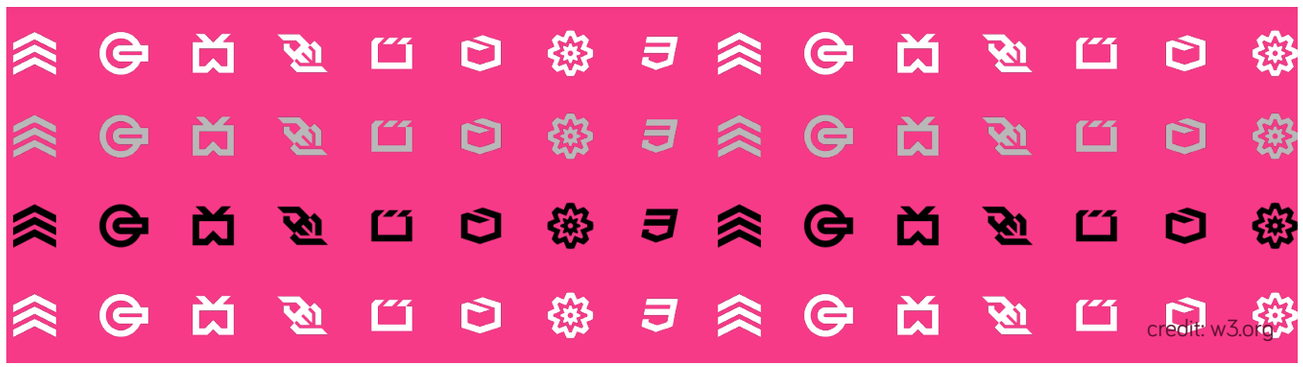
Let's make HTML5 & JavaScript Games!

Before we begin making the game, we're going to go over what each building block will contribute to your game.



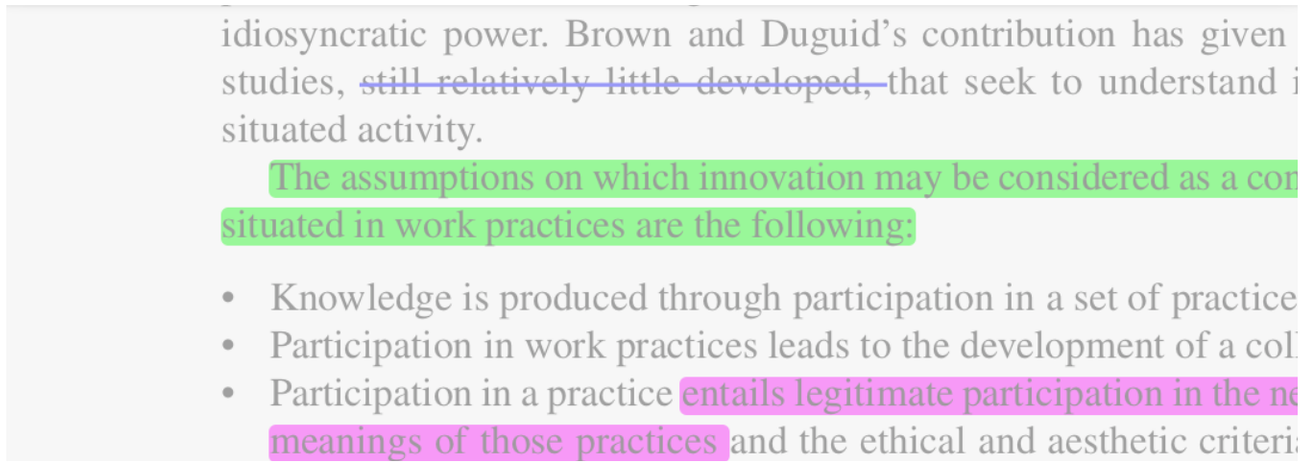
What is HTML?



Let’s start with HTML. What is it, and why are there 5 of them?

HTML is a markup language

- Markup languages are code-based annotation systems
- HTML is the backbone of every website


idiosyncratic power. Brown and Duguid's contribution has given
studies, still relatively little developed, that seek to understand i
situated activity.

The assumptions on which innovation may be considered as a cor
situated in work practices are the following:

- Knowledge is produced through participation in a set of practice
- Participation in work practices leads to the development of a col
- Participation in a practice entails legitimate participation in the ne
meanings of those practices and the ethical and aesthetic criteri

You mark-up real documents with a pen to separate and highlight content.. In markup languages, you use code!

HTML is made of elements

HTML defines the structure and layout of a website by using different elements.
Elements are composed of tags and their descriptors, also known as attributes.
Every element has a different purpose, similar to the way that each button in Microsoft Word does something different.
Tags are like sides of a box that help to organize content by type. They tell the browser how the content will be used.

HTML5 is new & dynamic

- HTML5 was designed for all of today's internet-capable devices
- Includes new tags for more types of content
- Check out diveintohtml5.info

The previous revision of HTML, HTML4, was released in 1997, when almost no one used a cell phone, dial-up was rampant, and no one thought video game consoles or TVs would be able to connect to the internet.

HTML5 is the 5th revision of HTML -- it has been created to keep up with the demand for more dynamic content on the web. Because it's new, it isn't supported by older versions of browsers (like IE8)

It includes new elements, including the one we're going to use to make games today!

Let’s make a simple HTML5 page

This is where we’ll start writing code. Yay!

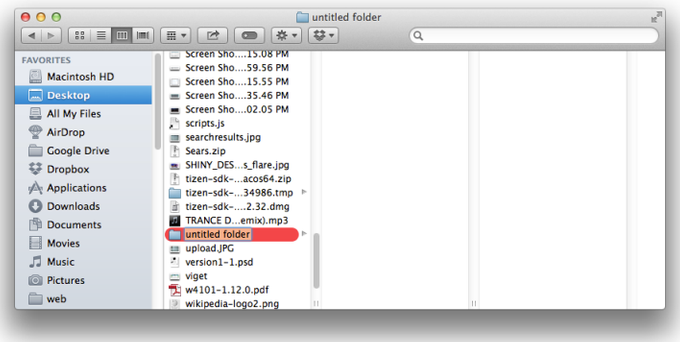
How do I begin?

- To make an HTML website, you only need a text editor and a browser
- You can use Notepad or TextEdit — there just won't be color-coding
- You can edit your files on any operating system

You only need 2 programs to code in HTML, and they both come pre-installed! However, you should probably download a better one like Sublime 2.

Now open your text editor of choice.

Create a folder to work in

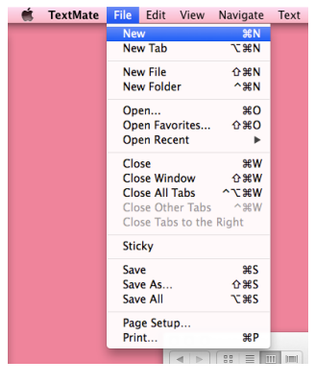


Name your folder CLF-html5-game

You can create a new folder on your desktop (or anywhere on your computer that you'd like to store this file)

Name it CLF-html5-game.

Create a new file



Save it as index.html in your new folder

Go into your text editor and create a new file. Save it in your folder as index.html.

Index.html is a naming convention — on the internet it's the first file the browser looks for when loading a website

Save as often as possible; occasionally computers are dumb. I'm scarred from Photoshop, so I save my files very often!

And now...the fun stuff!

We’re going to actually start typing. Prepare yourselves.

Components of an HTML5 page

- <!DOCTYPE html>: tells the browser it is looking at an HTML5 page
- <html>: begins the HTML code
  - <head>: the area where meta information will be located
    - <meta charset="utf-8">: the character encoding you want to use
    - <title>: the website title
  - <body>: the part of the page where we will be working!

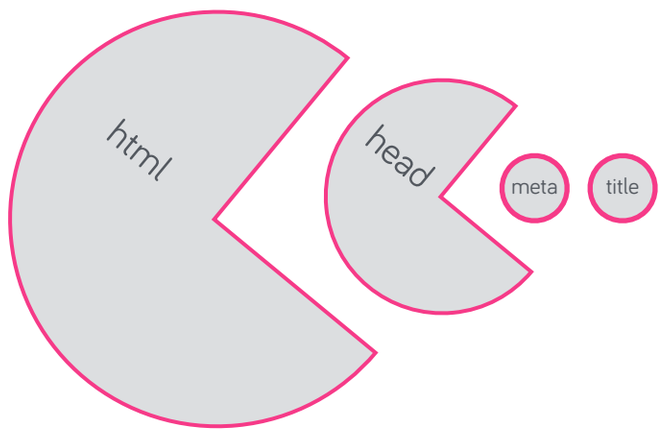
To begin, you have to tell the browser what kind of page it is looking at. There are several doctypes; some are for older versions of HTML and some are for XML or SVG.

Next, we tell the browser it's time to read HTML! I may use the word "parse" sometimes when referring to the browser reading code.

The HTML tag must be closed. The head is where we put all the information(data) that will be seen by search engines.

Remember!

- Close tags in the opposite order they were opened



Except for <!DOCTYPE>, <meta>, <img>, and <input> tags

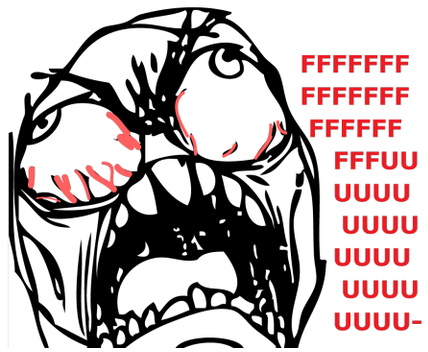
Your code will be messy otherwise.

Also indent your code so it's easier to read.
A shortcut to indent quickly is command +] or command + [

Comment your code

5 minutes after you write code
without comments

When you come back to it
in 3 weeks



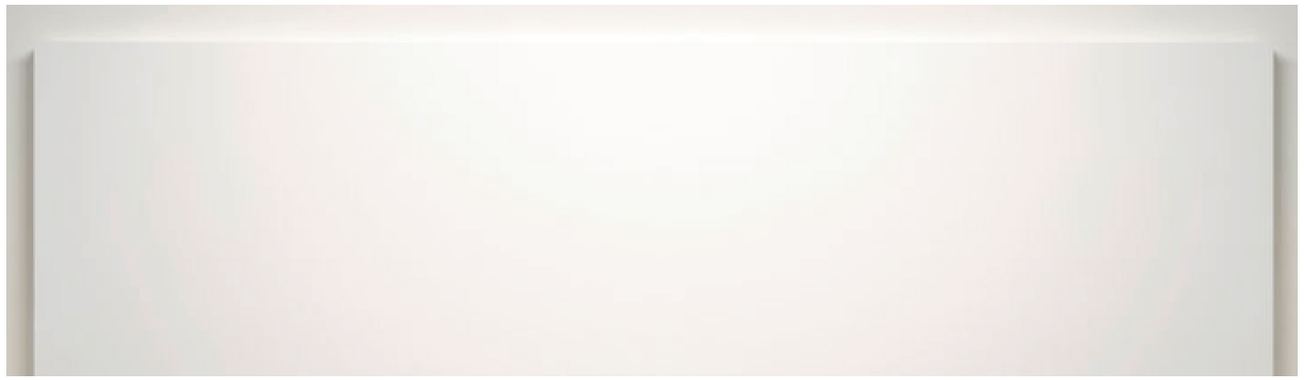
You should always take notes in your code so you remember what it does.

Commenting code is easy

- Preface your comment with < , a bang and two dashes (<!--)
- End it with two dashes and > (-->)
- Most text editors have shortcuts (like + /)

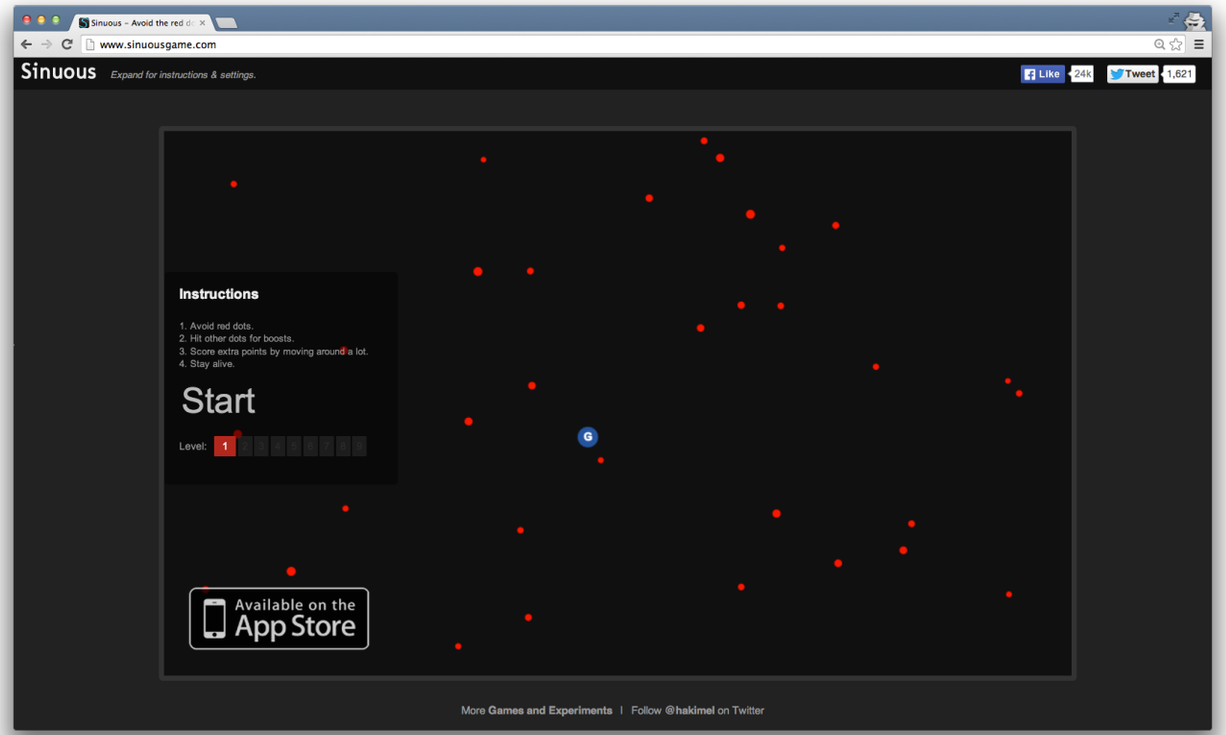
If you open index.html in your browser, you should see “This isn’t a comment!”, but not “This is a comment”.

Does anyone have questions so far?
Is everyone set up with a basic web page?



The canvas element

The canvas element is exactly what it sounds like — a dynamic artboard for both interactive animations and games.



It's HTML5's answer to Flash.

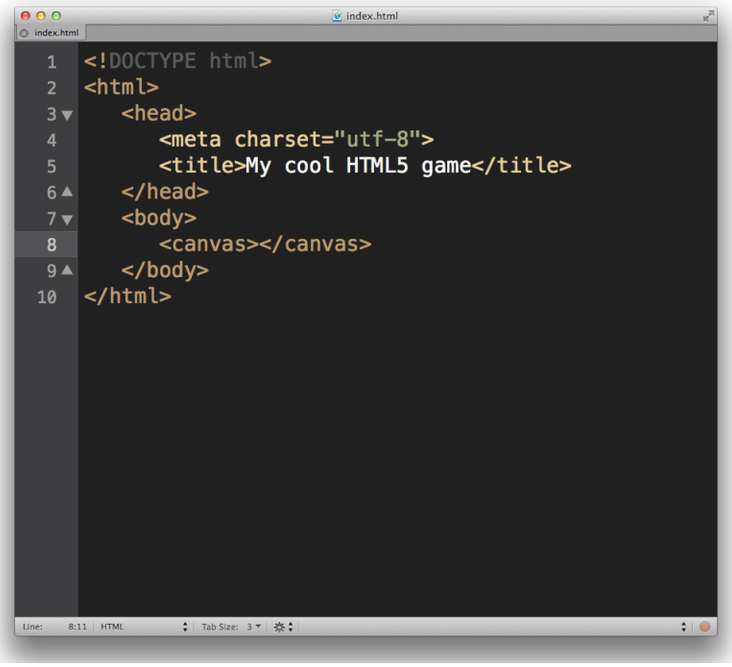
credit: Photon Storm

Once iPhones and Android dropped support for Flash, the canvas element began to look more appealing as a way to spread content across web-enabled devices.

credit: Wikimedia, stockrockandroll.com

Unlike flash, canvas works using only HTML and JavaScript. This means it doesn't require Adobe Flash maker, which also makes it a very affordable alternative.

Create a <canvas> element



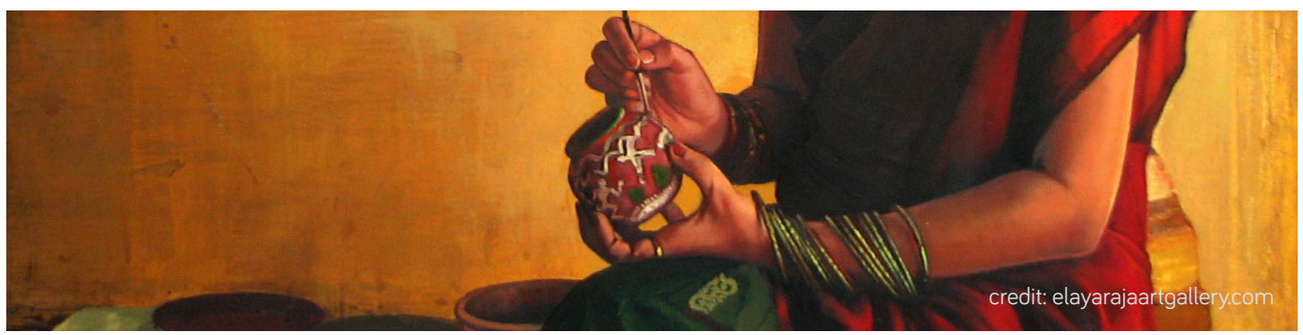
To begin, open and close your canvas element.

Older browsers (like IE8 and even IE9) won't be able to support canvas, so provide fallback text so that people aren't confused.

Make your <canvas> official

Bring your canvas to the DMV and give it an ID. This way, we can point to it later in JavaScript.

JavaScript



We’re about to go in deep, so please let me know if anything is unclear.

Make a javascript file
Include the javascript file in the HTML below the canvas — must be after the canvas so JS can find it

Find the canvas and tell it it will work in 2D. Show commenting and console logging.

Is everyone ready?

Set up the game ingredients

- Before your game can start, you need to have all the ingredients ready.
- Let’s go through these steps together.

Create the game loop. Without a game loop, our canvas will not become an interactive environment.

In order to simulate the appearance of smooth and continuous gameplay, we want to update the game and redraw the screen just faster than the human mind and eye can perceive (60 times in one second)

Set interval.

Variables

- Variables are useful for storing data that may change throughout the course of your app (e.g. your player's health)
- To create a variable, you have to tell JavaScript:
  - The name you're going to refer to it by
  - The value (information) that the variable holds

By variables, I am referring to numbers that will change, such as the location of the character and enemy on the screen.

Create an FPS variable for our game loop.

Functions

- Function: a named section of a program that does a specific task
  - Wraps up code in an easy-to-reference way
  - Parameter: additional information you can give the function to change the output

JavaScript has functions, much like a body does. Some functions are included in JavaScript, like math() and random(), while others must be defined before being used.

- Really all a function is is a named section of a program that does a specific task
- Makes it easier to execute the code you want to execute
- Functions can also take parameters, which is additional info you can give the function to change the output

Function structure

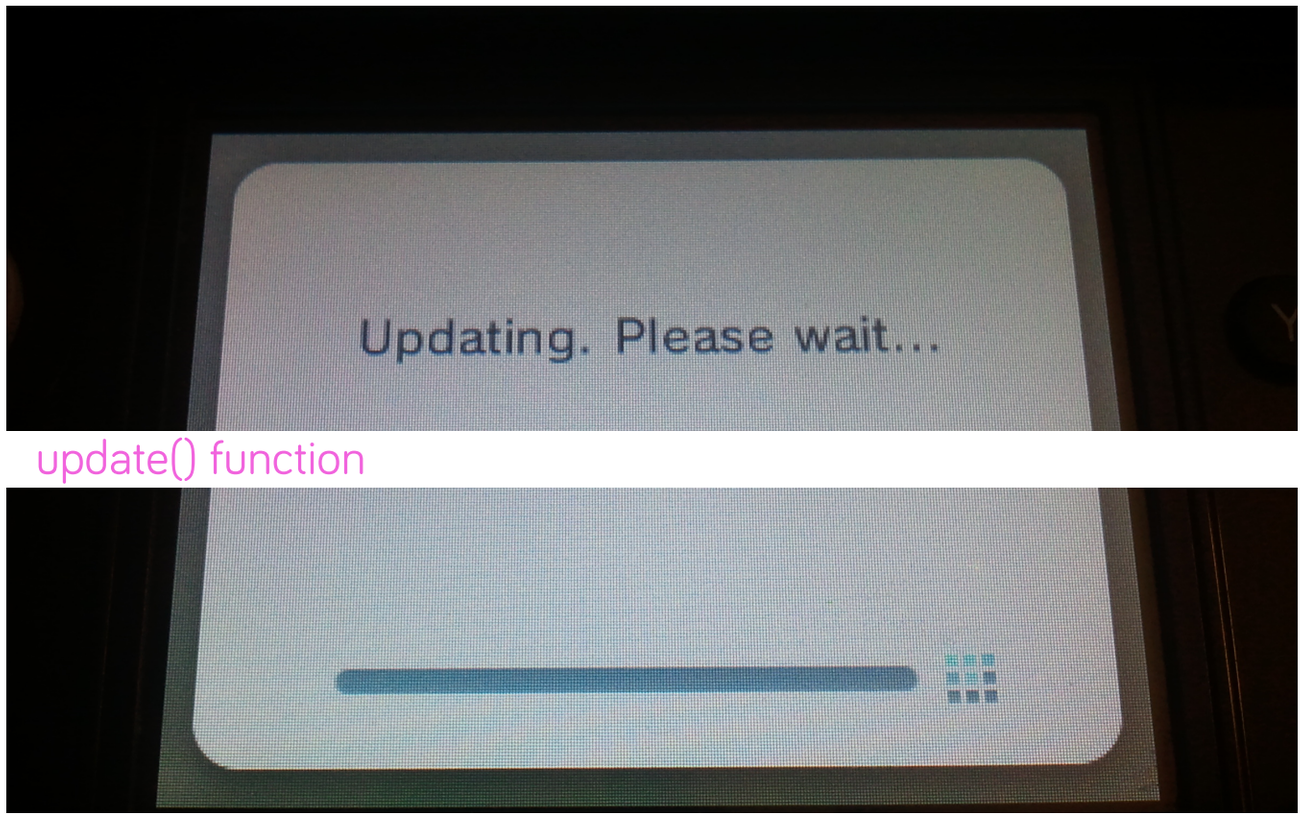
```
var dance = function (danceType) {};
function dance (danceType) {};
```

- Either of these is syntactically correct.
- Name of the function
- Parentheses: Hold any modifiers (also known as arguments)
- Brackets: What to do in the function
- Semicolon: end of line, move onto the next thing

- We will use the first syntax
- Name of function
- Parentheses
- Semicolon: end of line, p. much

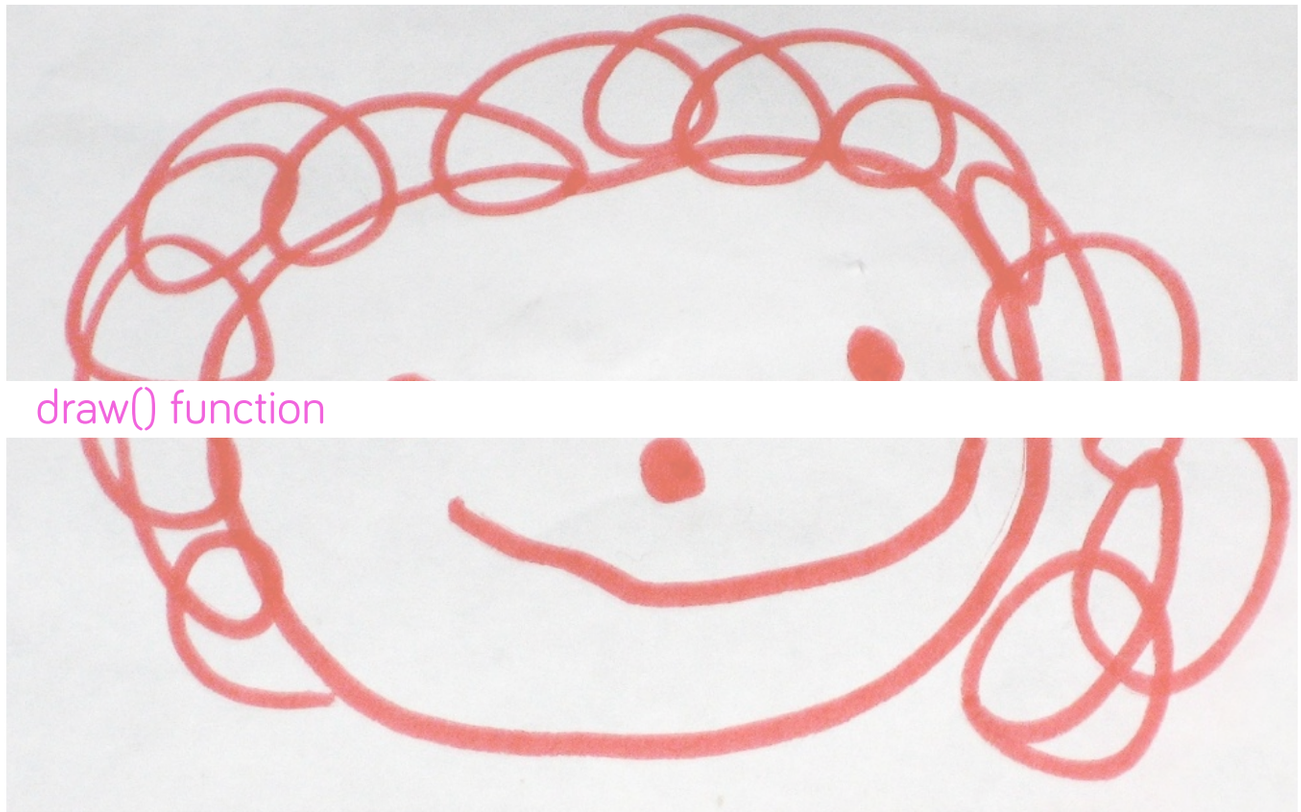
Our game's functions

- setInterval(): runs update and draw 60 times per 1000 milliseconds
- update(): runs once per frame, before draw()
  - Can be used to update player location, health, score, etc.
- draw(): runs once per frame, after update()
  - Where you should put all your drawing, e.g. player.draw();



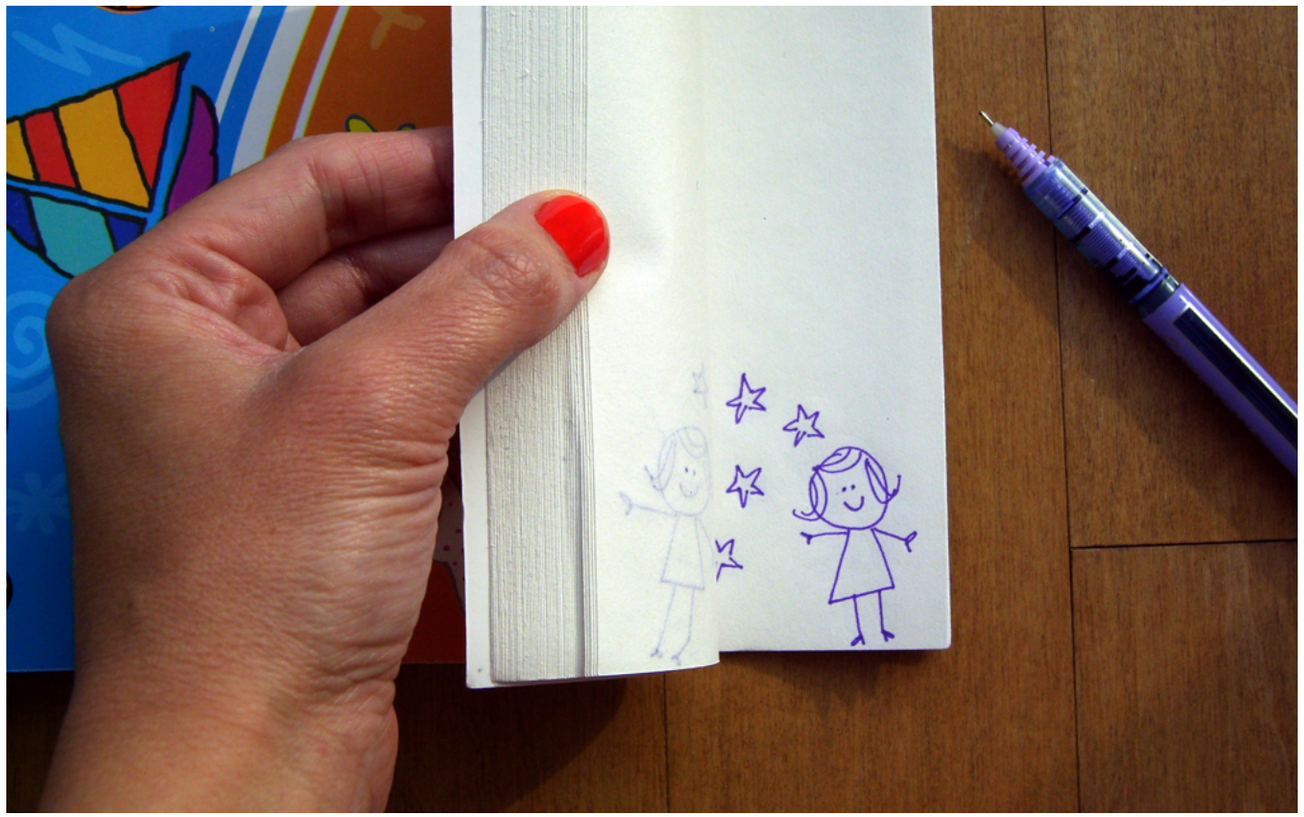
update() function

Variables we create will be changed in the update function.

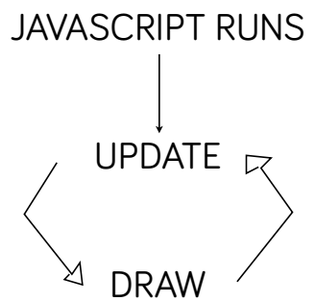


- draw()
- Once the update function is finished, the canvas will be redrawn.

- Will draw later shapes over earlier shapes, and use earlier colors for all subsequent shapes (unless you change the color/turn it off)



With an FPS of 60, the canvas will be redrawn 60 times in one second. It works similarly to a flipbook!

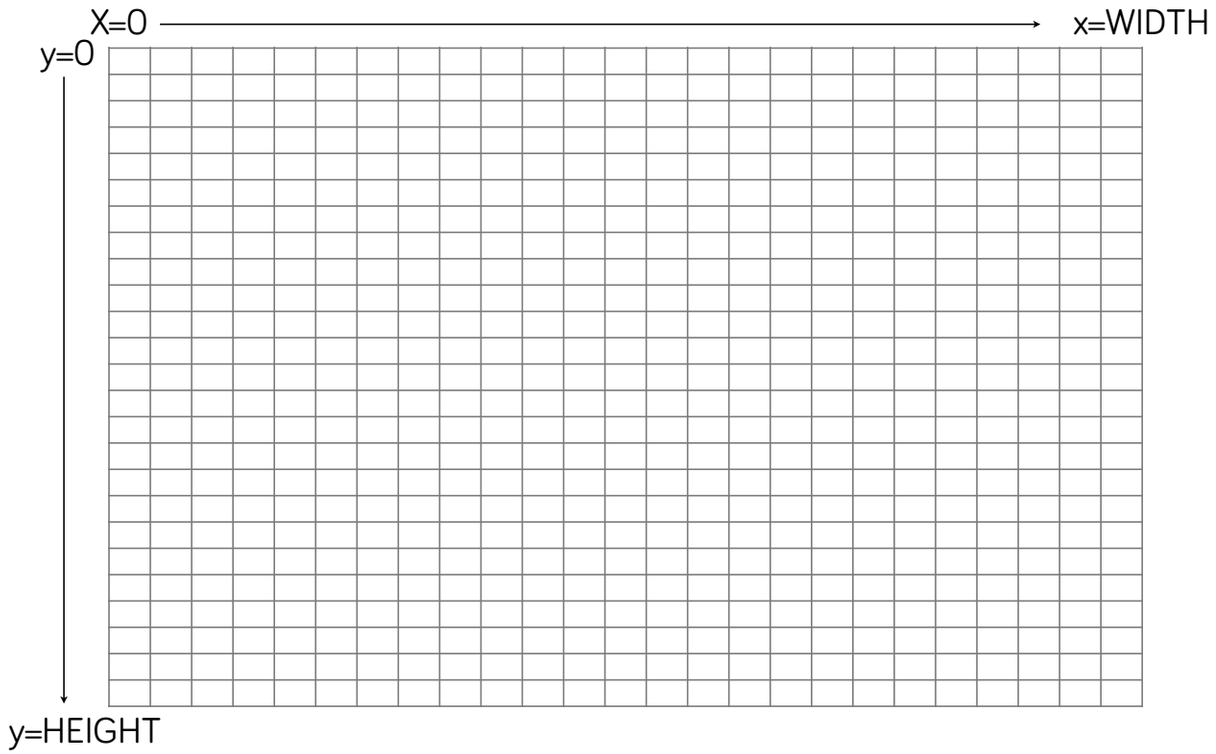


60 times per second

FLOW:
- Javascript begins
- initial variable values are defined
- update executes
- draw executes
- update and draw alternate until the game is finished

How does positioning work?

In order to draw something on screen, you have to tell the computer exactly where to put it.



Positioning
- X: gets bigger as it goes to the right
- Y: gets bigger as it goes down

Let’s draw our player at 3,2.

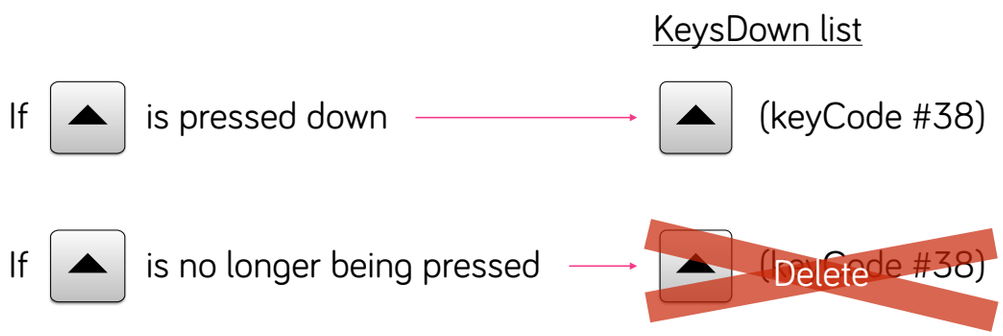
How can I incorporate interactivity?

- Keyboard inputs to move your character

Create a variable object for keys that are currently being pressed. This will allow us to move at diagonals because you can press more than one key.

Next, add an event listener, which will check for

Keyboard interactivity

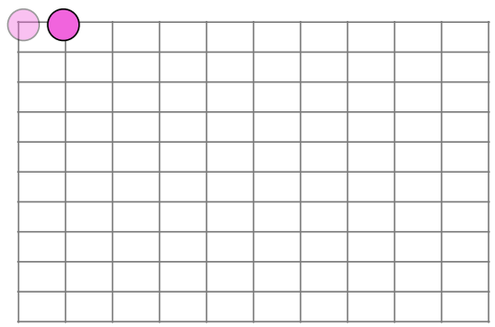


- Keyboard inputs to move your character

Create a variable object for keys that are currently being pressed. This will allow us to move at diagonals because you can press more than one key.

Next, add an event listener, which will check for

Movement

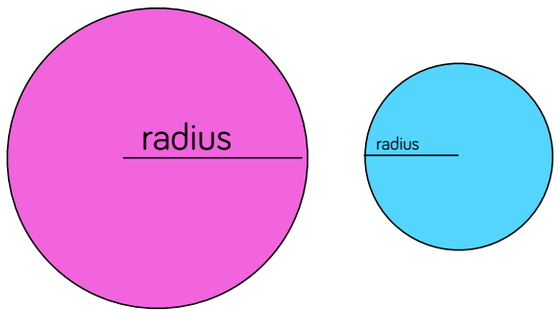


New position = old position + speed

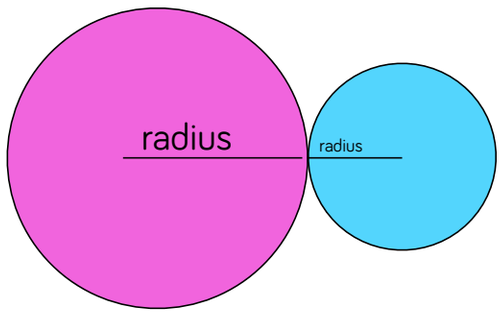
Move our player using the keyboard!

How can I test for collisions?

Have these circles collided yet?



How about now?



If player’s X or Y minus the radius equals the location of any edge of the canvas, it must stop moving.

If the player touches the enemy, the enemy must be eaten and the score must increase.

Questions? How is everyone doing

Game: eat the squares!
Randomize the location of all enemies every time one is eaten
Aim to eat all the squares
Keep score
If you eat all of them, you win and the game ends.