# Let's make HTML5 & JavaScript Games!

credit: w3.org

# What is HTML?

credit: w3.org

# 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
situated activity.

The assumptions on which innovation may be considered as a co
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 n
  meanings of those practices and the ethical and aesthetic criteri

# HTML is made of elements

opening tag

attribute

closing tag

<section id="box">HTML is very easy to use!</section>

element

# HTML5 is new & dynamic

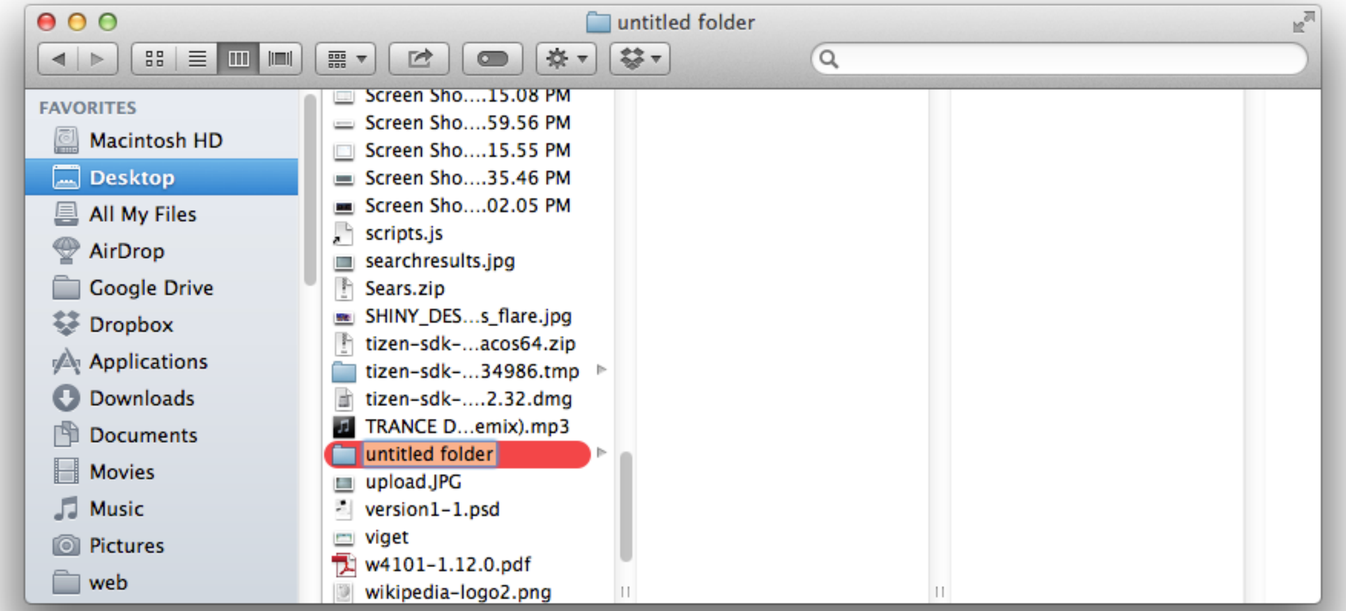
- HTML5 was designed for all of today's internet-capable devices
- Includes new tags for more types of content
- Check out diveintohtml5.info

# Let’s make a simple HTML5 page

# How do I begin?

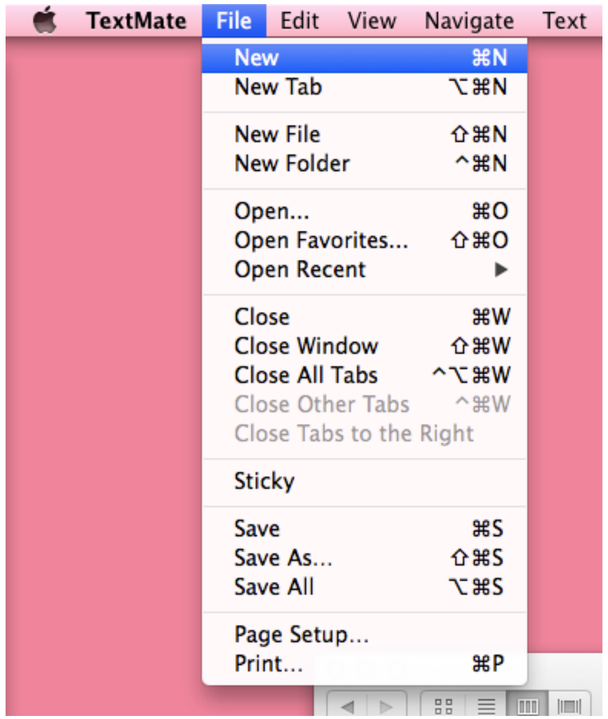
- To make an HTML website, you only need a text editor and a browser
- You can use Notepad or TextEdit — there just won't be color-coding
- You can edit your files on any operating system

# Create a folder to work in




Name your folder CLF-html5-game

# Create a new file




Save it as index.html in your new folder

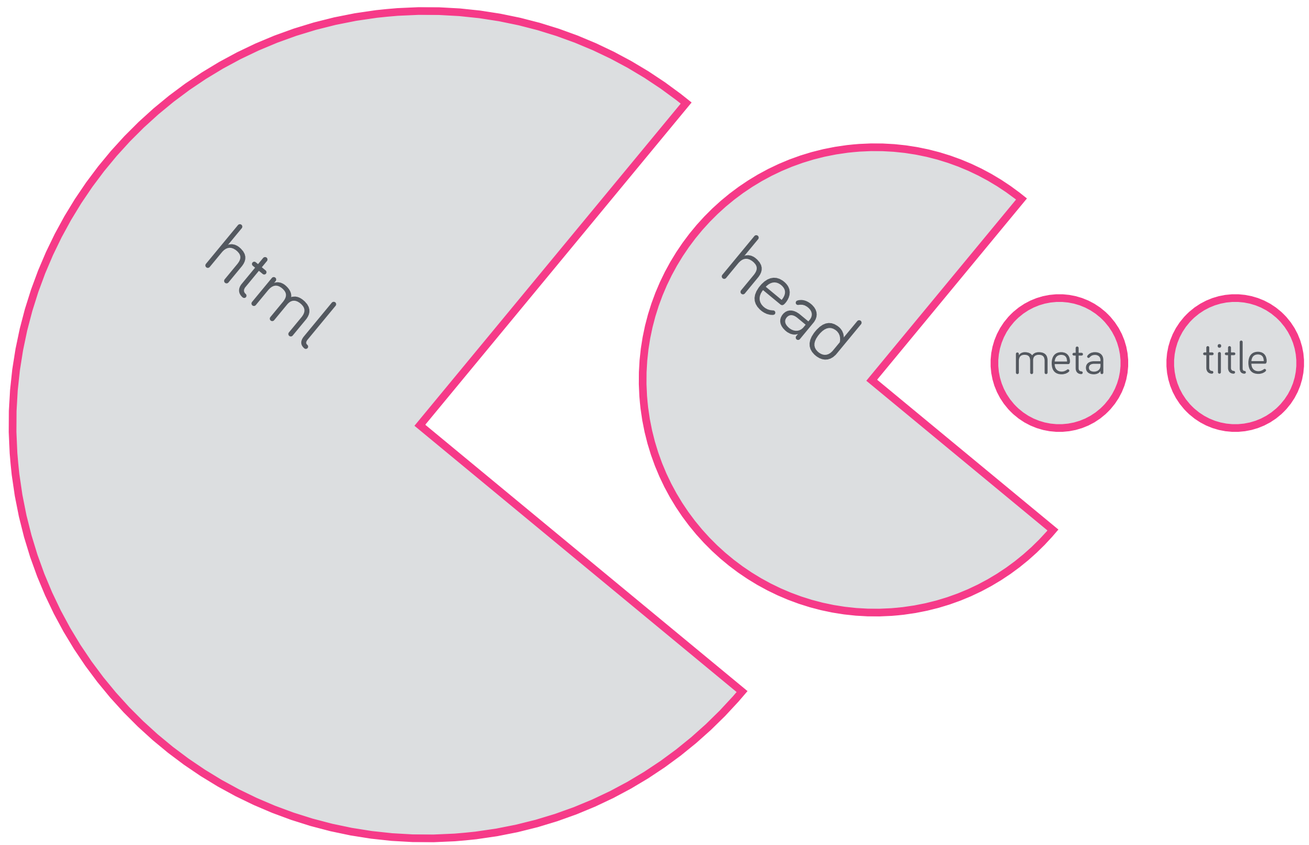
# And now…the fun stuff!

# 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!

# Remember!

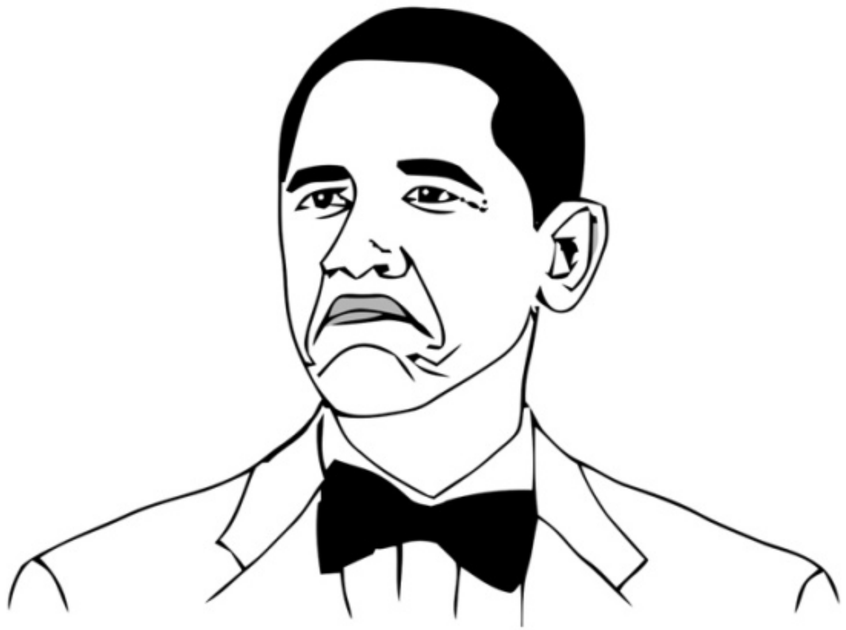
- Close tags in the opposite order they were opened




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

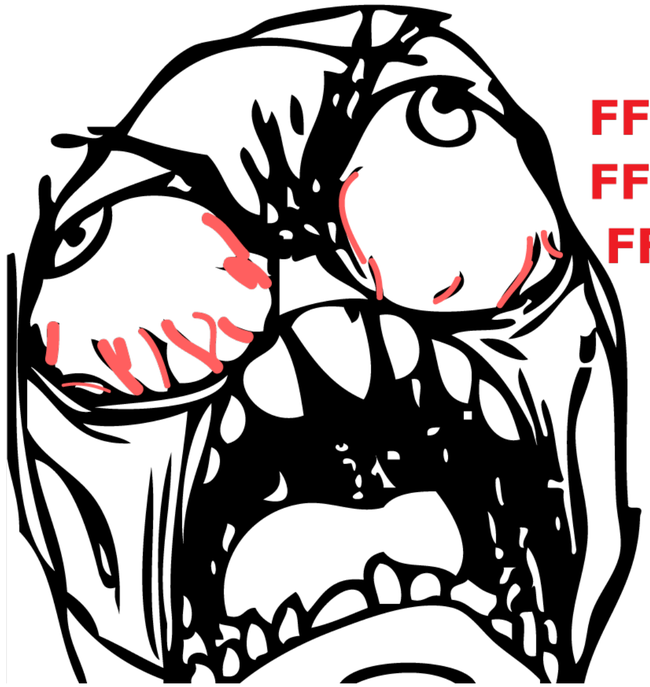
# Comment your code

5 minutes after you write code
without comments



NOT BAD

When you come back to it
in 3 weeks



# 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  + /)

# The canvas element

credit: thesilverpen.com

Sinuous – Avoid the red do

www.sinuousgame.com

# Sinuous

*Expand for instructions & settings.*

Like 24k Tweet 1,621

## Instructions

1. Avoid red dots.
2. Hit other dots for boosts.
3. Score extra points by moving around a lot.
4. Stay alive.

## Start

Level: 1 2 3 4 5 6 7 8 9

Available on the App Store

More **Games and Experiments** | Follow **@hakimel** on Twitter



credit: Photon Storm

credit: Wikimedia, stockrockandroll.com

# Create a <canvas> element

index.html

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title>My cool HTML5 game</title>
    </head>
    <body>
        <canvas></canvas>
    </body>
</html>
```

Line: 8:11 | HTML | Tab Size: 3

# Provide fallback text for older browsers

index.html

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>My cool HTML5 game</title>
6      </head>
7      <body>
8          <canvas>Canvas not supported</canvas>
9      </body>
10 </html>
```

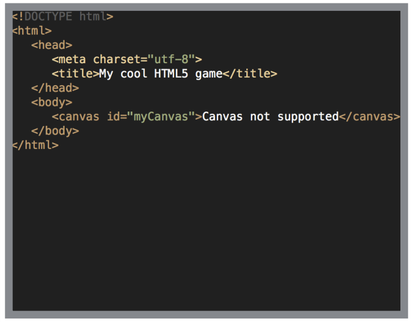
Line: 9:9 | HTML | Tab Size: 3

# Make your <canvas> official

index.html

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>My cool HTML5 game</title>
6      </head>
7      <body>
8          <canvas id="myCanvas">Canvas not supported</canvas>
9      </body>
10 </html>
```

Line: 9:9 | HTML | Tab Size: 3 | ID: myCanvas




## myCanvas

**Element:** canvas
**Born:** 5 minutes ago
**Address:** html > body

html id

# JavaScript

# Set up the game ingredients

# 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

# 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

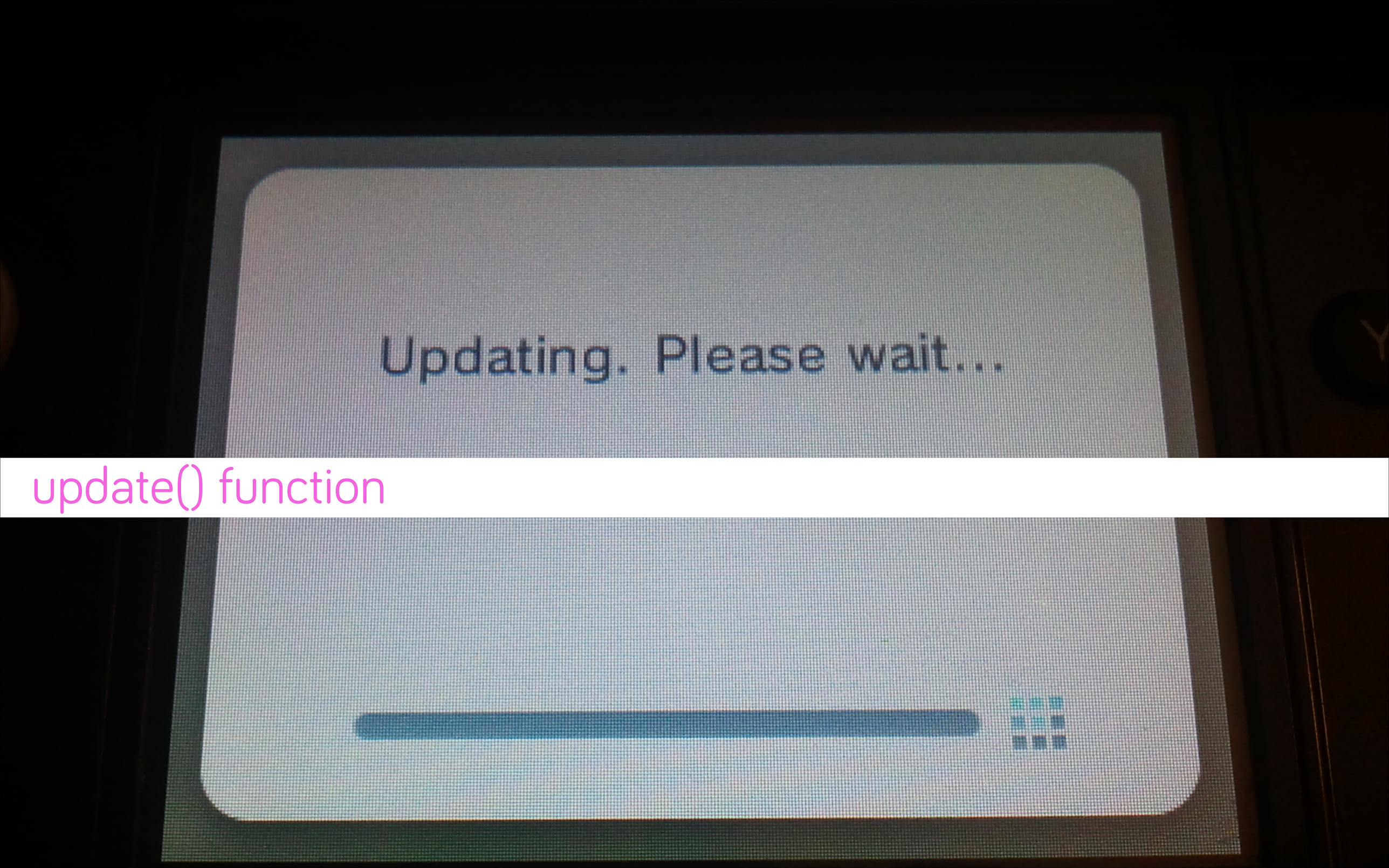
# Function structure

```
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

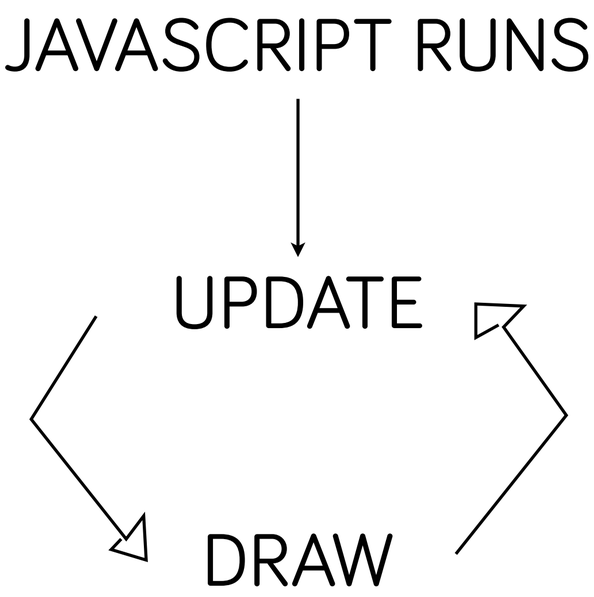
# Our game's functions

- 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();



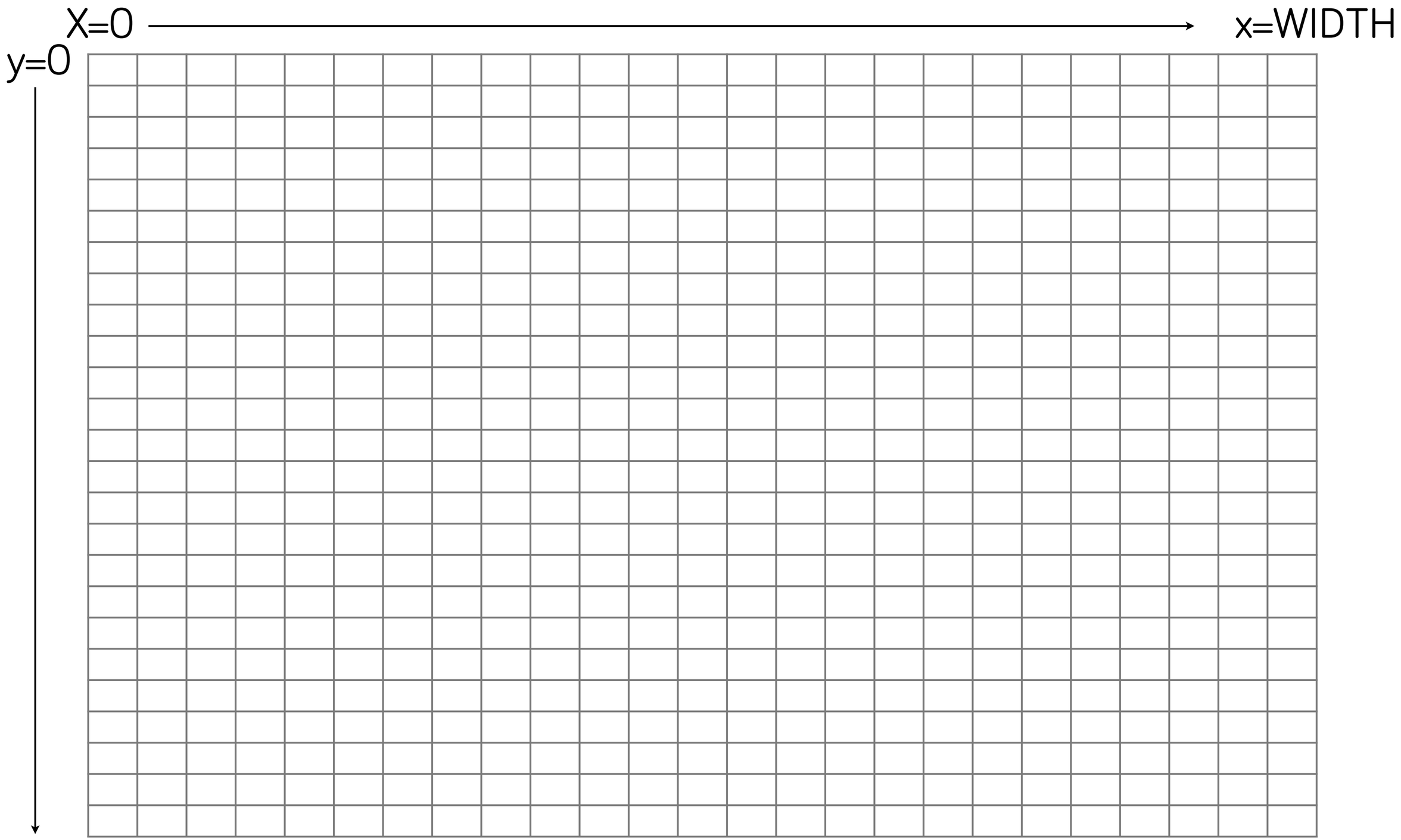

# update() function

# draw() function




60 times per second

# How does positioning work?

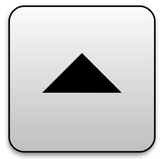

X=0
x=WIDTH
y=0
y=HEIGHT


X=0
x=WIDTH
y=0
(3,2)
y=HEIGHT

# How can I incorporate interactivity?

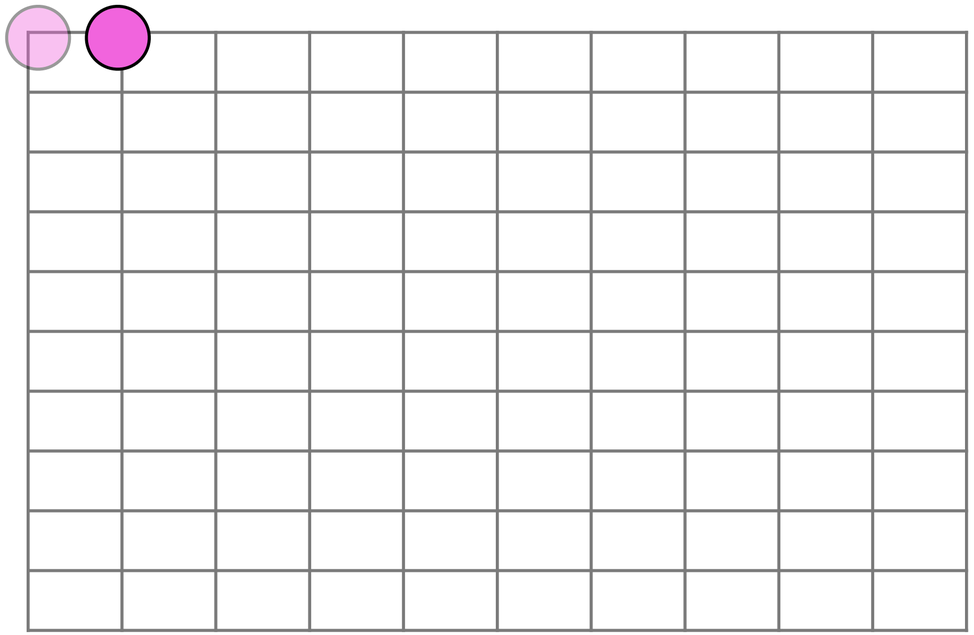
# Keyboard interactivity

KeysDown list

If  is pressed down →  (keyCode #38)

If  is no longer being pressed →  (keyCode #38) Delete

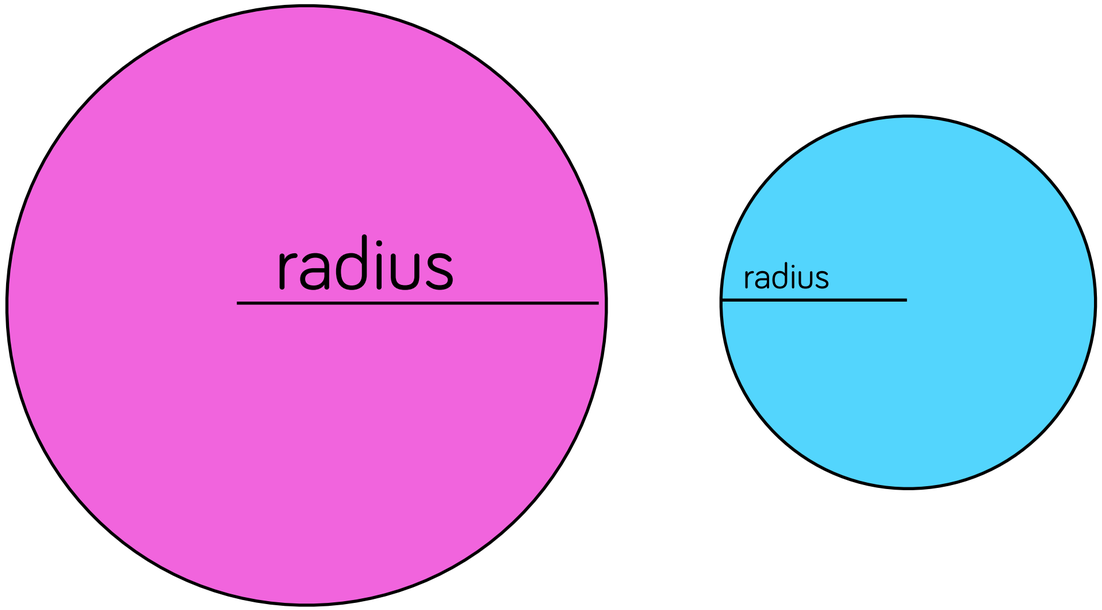
# Movement



New position = old position + speed

# How can I test for collisions?

Have these circles collided yet?



How about now?