

**HELLO**

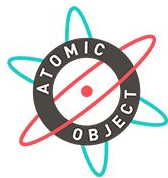
my name is

*TypeScript*

# Intro to TypeScript

TypeScript Bootcamp

# THANKS TO ALL OUR SPONSORS!

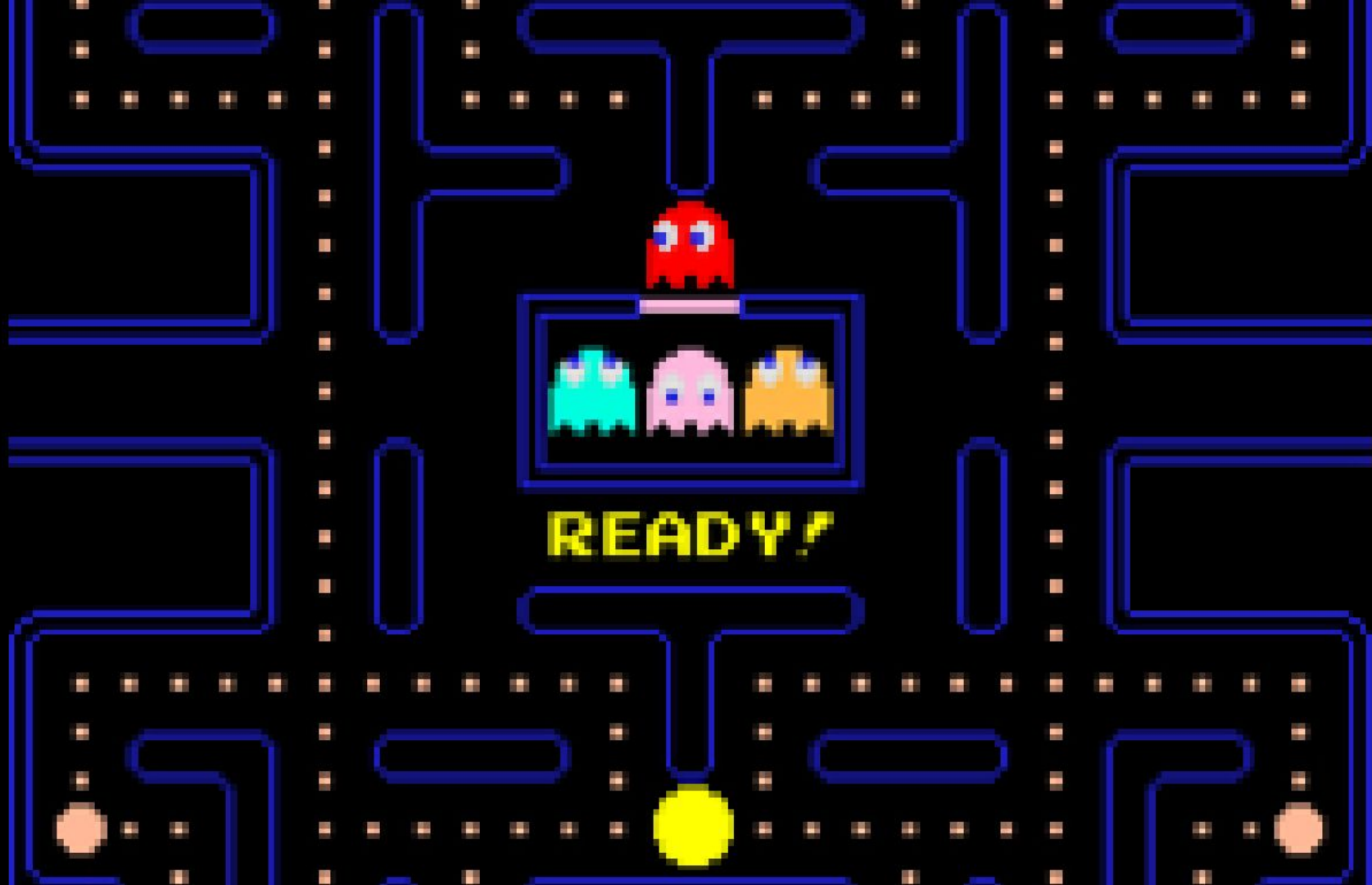


# Jody “Gus” Gustafson

- Hagerty.com
- WorldTreeSoftware.com
- PowerChord.app
- Git: @jodymgustafson
- TypeScript experience
  - 3 year writing AWS Lambdas

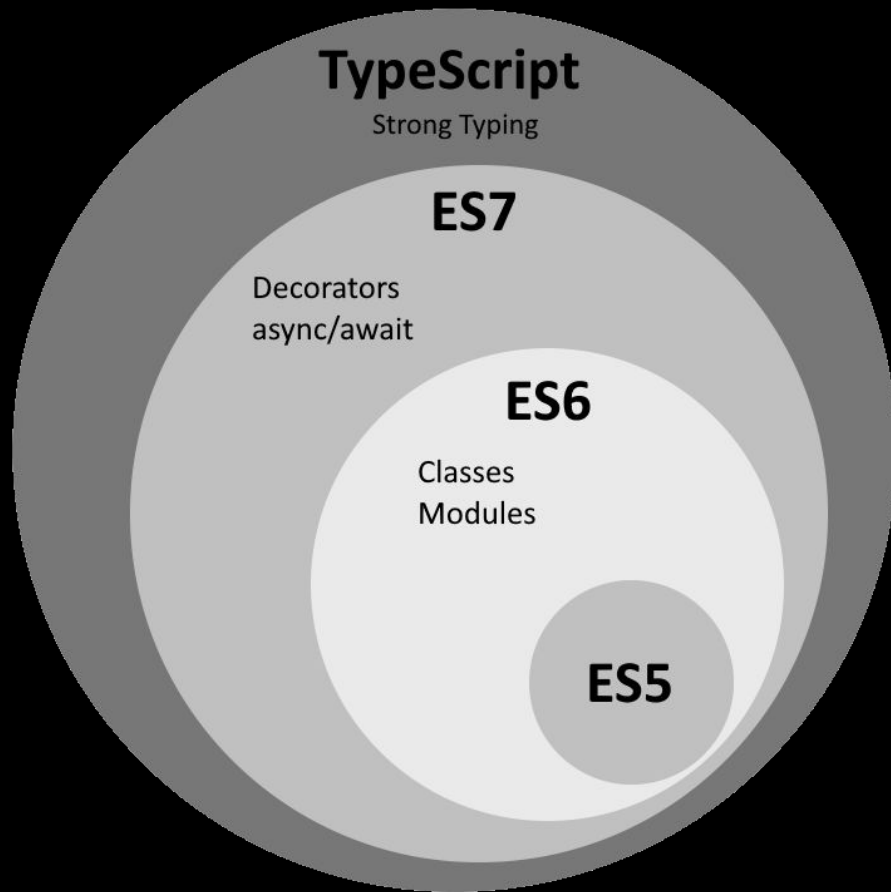
# Agenda

- Theory
- Code
- Questions



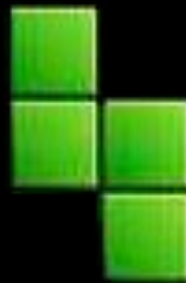
# What is TypeScript?

TypeScript is a typed superset of JavaScript that compiles to plain JavaScript



# TypeScript Features

- TypeScript is JavaScript
- Static typing
- JavaScript transpiler
  - Use cutting edge features now
- Full OOP & generics support



# Why TypeScript?

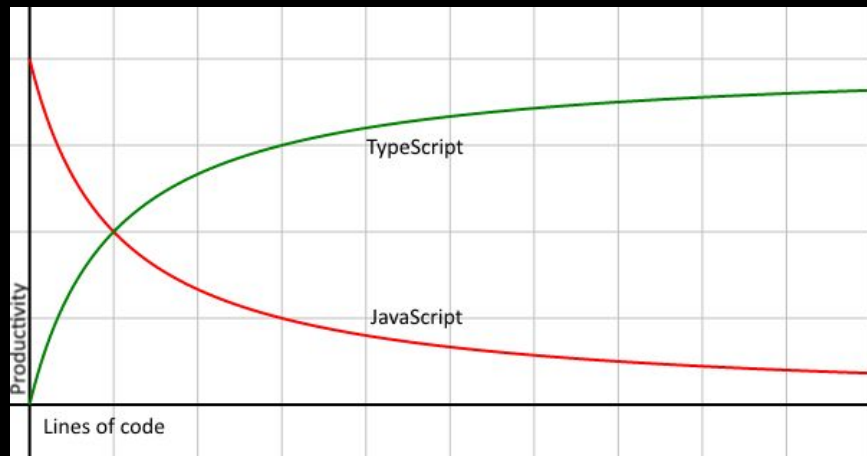
- Types enhance code quality and understandability
  - Compile time error checking
  - Easier to develop and maintain large apps
- Enhanced IDE tooling
  - Autocomplete and intellisense
  - Code navigation
  - Refactoring support
- Compiles to plain JavaScript
  - Will run in any browser or Node.js
  - Works with your favorite JS libraries





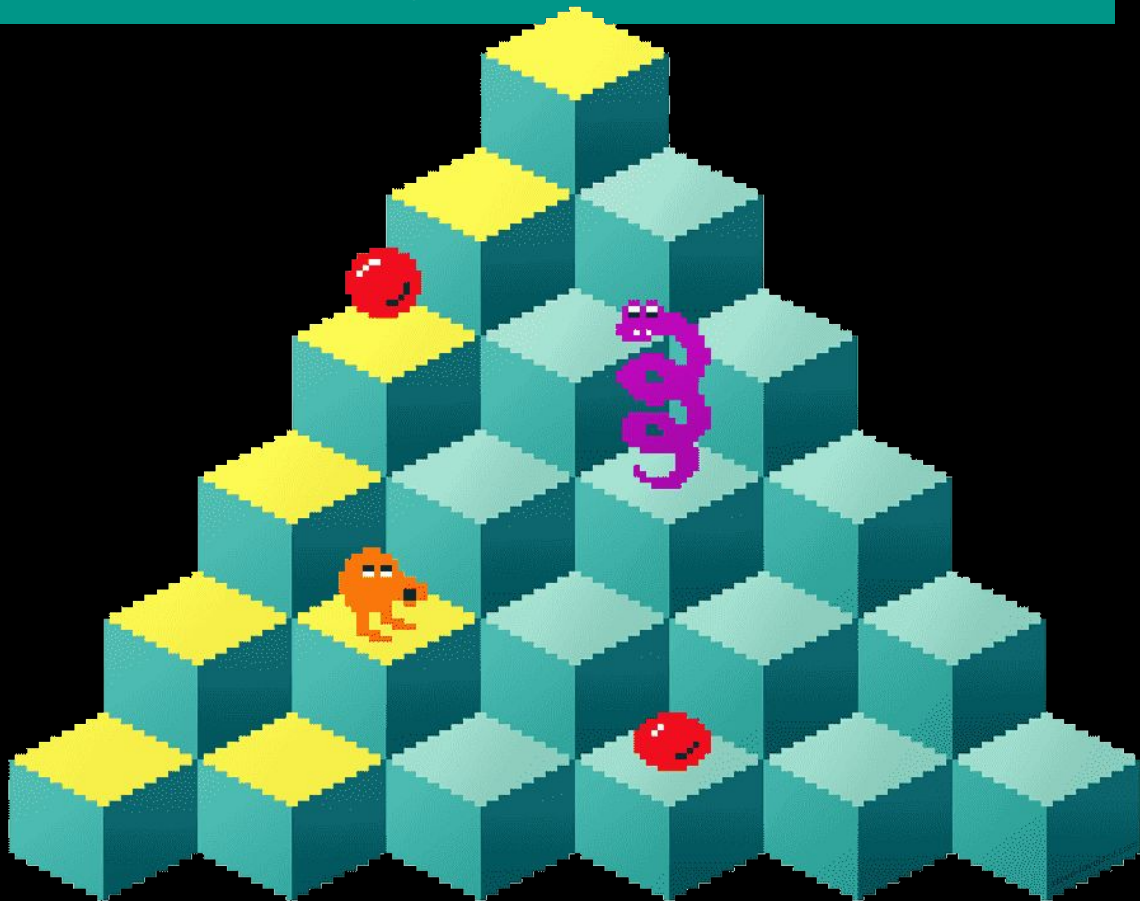
# When to use TypeScript?

- Apps vs. Scripts
- In team environments
- Rule of thumb:
  - When you can't hold it all in your head



# TypeScript FUNdamentals - Basic Types

- ❖ Boolean
- ❖ Number
- ❖ String
- ❖ Array
- ❖ Any
- ❖ Void
- ❖ Enum



# TypeScript FUNDamentals - Structural Typing

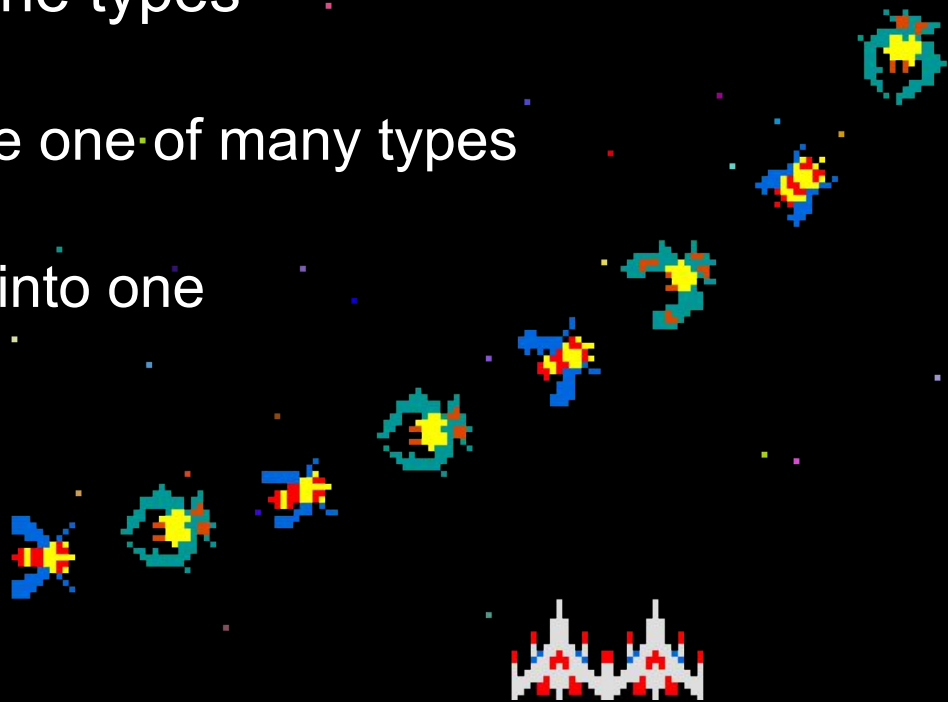
- Type compatibility is based on structural subtyping
- Type x is compatible with y if it has at least the same members as y

```
point2d = {
    x: 1,
    y: 2
};

point3d = {
    x: 1,
    y: 2,
    z: 3
};
```

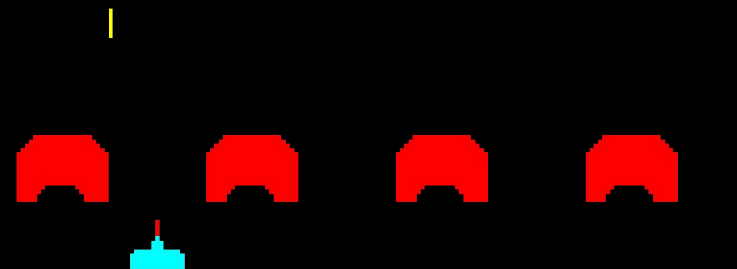
# TypeScript FUNdamentals - Type Aliases

- a.k.a User defined types
- Use `type` keyword to define types
- Union Types
  - Defines a type that can be one of many types
- Intersection Types
  - Combines multiple types into one



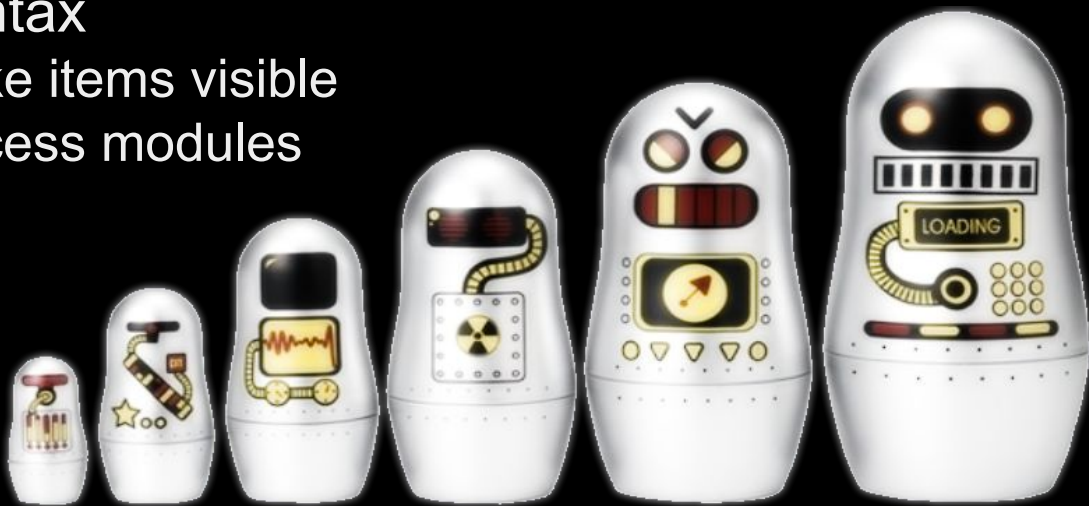
# TypeScript FUNdamentals - OOP

- Classical object-oriented support
- Constructors
- Super (base class)
- Static properties
- Access modifiers
  - public, protected, private
- Interfaces
- Abstract classes



# TypeScript FUNdamentals - Modules

- A self contained collection of related code
- Defined in a single file
- Are reusable
- Keeps items out of global namespace
- Uses ES6 module syntax
  - Use `export` to make items visible
  - Use `import` to access modules



# TypeScript FUNdamentals - Functions

- Rest parameters
- Optional parameters
- Default parameter values
- Function overloading



②



②



A

# TypeScript FUNdamentals - Generics

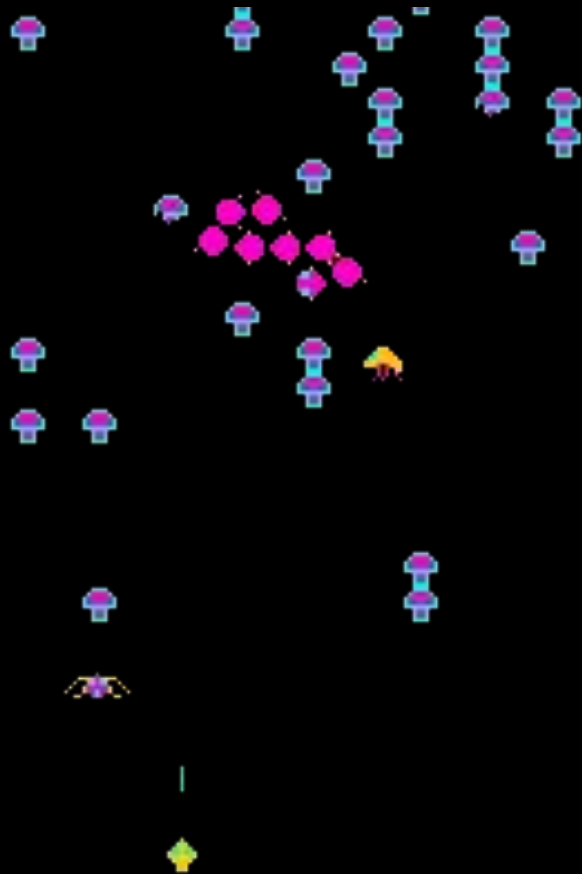
- Build components that work over many types
- Use angle brackets to declare
  - Similar to Java and C#
- Use with functions, types, interfaces, classes
- Type constraints supported





# TypeScript FUNdamentals - Type Declarations

- Defines info about JS code
- Provides type information and docs
- Makes JS libraries easier to use
- DefinitelyTyped
  - The repository for high quality type definitions
  - Install via npm `@types/*`



# TypeScript FUNdamentals - Compiler Options

- tsconfig.json
- Target JavaScript version
- Module system to use
  - CommonJS, AMD, etc.
- Generate typedef and map files
- Type checking strictness



GAME OVER

INSERT COIN

# TypeScript Resources

<https://github.com/jodymgustafson/IntroToTS>

<https://www.typescriptlang.org>

<https://www.w3schools.com/typescript>

<https://www.typescripttutorial.net/>

