OOP JavaScript

* Object- Instance of a class
  + Property- Holds value within an object
  + Method- Defines logic within an object
    - Behavior- Multiple methods within an object
  + Factory/Constructor Functions
    - Factory Function creates an object based on contained properties
      * Returns object
        + Uses “Return” 🡪 function *createfactory* (property) {*return* {property, method}}
    - Constructor Function constructs a new object via demanding a new instance of the object
      * Uses “this”
        + Refers to global object by default
        + ALWAYS use “new” operator or the function will assign props/methods to the global “this”

Will throw error

* + - * Uses “new” operator 🡪 const *another* = new Circle(1);
      * Constructors are built into JavaScript
    - Can use either. Use what you prefer.
  + Every object has a constructor property. It references the function used to create that object.
    - new
      * String(); ‘’, “”, ``
      * Boolean(); True, False
      * Number(); 1,2,3
  + Primitives- Copied by their value
  + Objects- Copied by their reference
  + Dot vs Bracket notation
    - circle.location = { x:1 };
* Coding Notes
  + Syntax
    - When using a .Identifier, ALWAYS PUT AFTER SQUARE BRACKETS
      * //console.log(JSON.stringify(animalFax[1].Name)) ALWAYS PUT THE .Identifier AFTER THE SQUARE BRACKETS