ES6:
ECMAScript 6

GIRLSCRIPT EDUCATION OUTREACH PROGRAM

TOPICS TO COVER:

1. What is es6 ?
2. let , const
3. Template strings
4. Fat-Arrow functions
5. Destructuring objects
6. Rest Parameter
7. Spread operator
8. JS classes
9. use strict
10. What is Babel ?

1. What is ES6 ?

- ✓ *European Computer Manufacturers Association* (ECMAScript) or (ES) :

a **standard for scripting languages** like JavaScript, ActionScript and Jscript.

- ✓ ES6 = ECMAScript 6 = ECMAScript 2015 , was released in 2015.
- ✓ ES6 allows you to write the code in such a way that makes your code more modern and readable.

Image source

What's there in ES6?

- JavaScript let
- JavaScript const
- Template Strings
- Arrow Functions
- Destructuring objects
- Spread operator
- JavaScript Classes
- Exponentiation (**) (EcmaScript 2016)
- .. and more

2. let , const

1. Var

Variables declared with **var** , are either function-wide or global. Var has no block scope.

They are accessible globally or throughout the function.

2. Let

Variables declared with **let** , are **block scoped**.

They are accessible only within the block they are declared in.

```
var x = 10;
// Here x is 10
{
  let x = 2;
  // Here x is 2
}
// Here x is 10
```

3. Const

Const is similar to Let but the value once assigned cannot be changed.

Value stays **constant**. Any attempt to change value will throw an error.

3. Template Strings

This allows you to easily implement variables with a very simple syntax ${ } and embed expressions.

```
1  let name  = "Sarah";
2  const greeting = `Hello my name is ${name}`;
3
4  console.log(greeting);
```

Image source

4. Arrow Functions

Arrow functions allows a short syntax for writing function expressions.

You don't need the **function** keyword, the **return** keyword, and the **curly brackets**.

```
// ES5
var x = function(x, y) {
   return x * y;
}

// ES6
const x = (x, y) => x * y;
```

5. Destructuring objects

With **Destructuring** , the properties (or keys) and their corresponding values can be pulled out from an object.

```
var num = {x: 100, y: 200};
var {x, y} = num;  // Destructuring object

console.log(x); // 100
console.log(y); // 200
```

Destructuring arrays

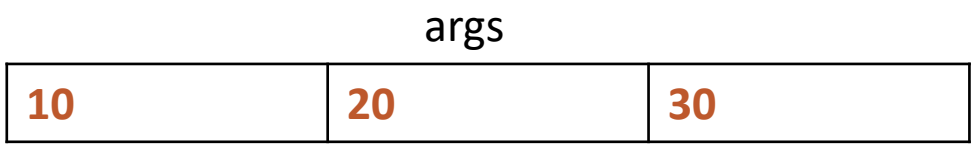
We destructure an array on basis of index. We extract values from an array and put them in new variables.

```
var arr = ["Hello", "World"]

// Destructuring array
var [first, second] = arr;

console.log(first); // Hello
console.log(second); // World
```

6. Rest Parameter

- By using the **rest parameter**, a function can be called with any number of arguments.

```
function show(...args) {
  let sum = 0;
  for (let i of args) {
     sum += i;
  }
  console.log("Sum = "+sum);
}

show(10, 20, 30);
```



7. Spread operator (...)

- By using the **spread operator**, an iterable can be expanded in places where more than zero arguments are expected.

Example

```
let colors = ['Red', 'Yellow'];
let newColors = [...colors, 'Violet', 'Orange', 'Green'];
console.log(newColors);
```

Output

```
[ 'Red', 'Yellow', 'Violet', 'Orange', 'Green' ]
```

8. JS Classes

Real life analogy:

Object :

Anything that has one or more **properties**.

Examples: any person (John , David) , any car (Honda Civic) etc.

Class :

A class defines the **structural blueprint** of its objects. It is a category for objects.

It lists down the properties and associated functions(methods) of the objects belonging to that class.

Examples: human, car.

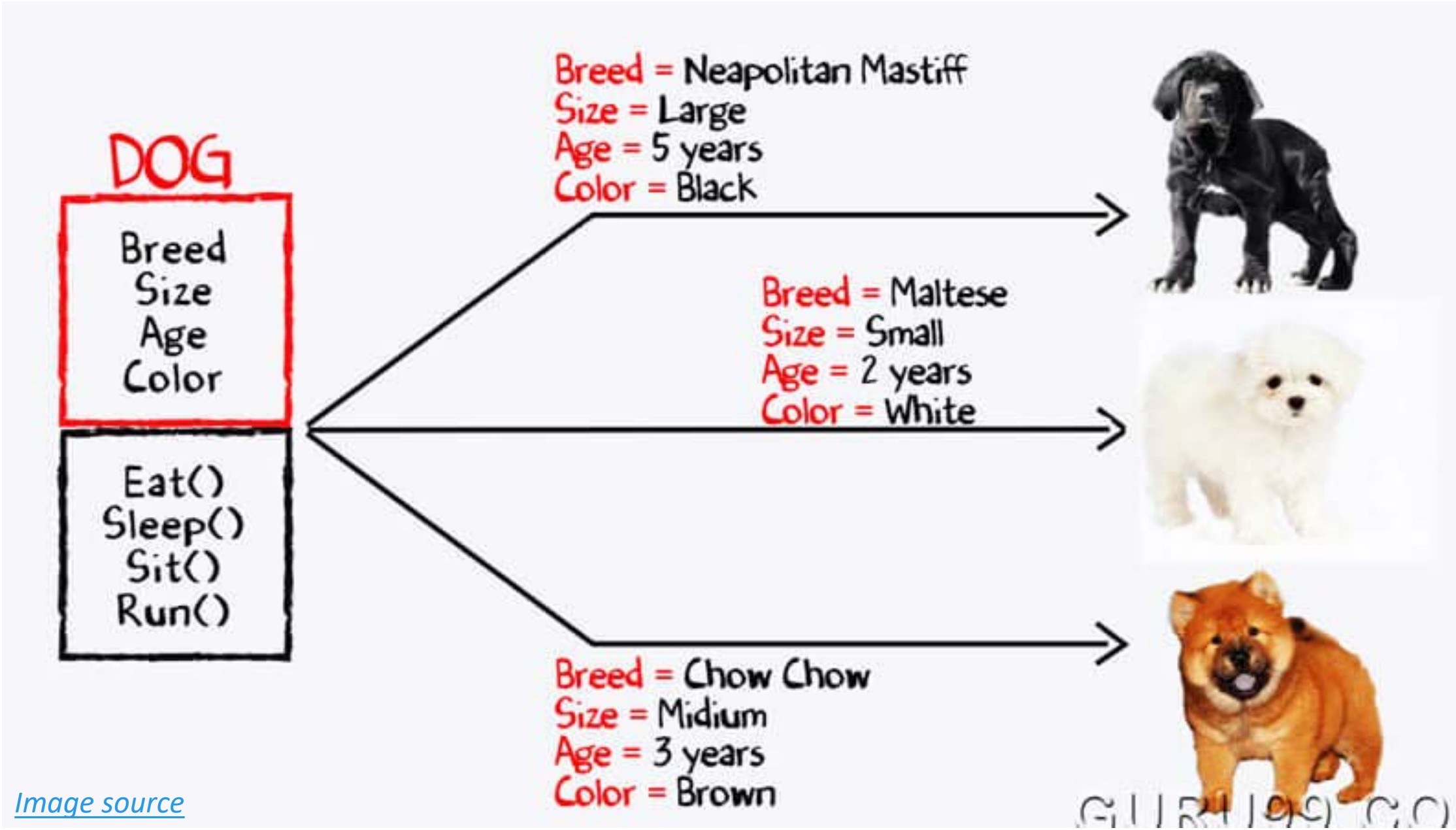


Image source

```
class Car {

  //constructor to initialise carname
  constructor(name) {
    this.carname = name;
  }


  printBrand() {
    return "I have a " + this.carname;
  }
}

car1 = new Car("Ford 16");
car1.printBrand(); //Output: “ I have a Ford 16 “
```

9. Use strict

"use strict";

Defines that JavaScript code should be executed in "strict mode". Strict mode changes previously accepted "bad syntax" into real errors.

- Using a variable/object, without declaring it, is not allowed.
- Deleting variable/object/function is not allowed.
- Duplicating a parameter name is not allowed:
- Some words like ***eval*** , ***with*** , ***arguments*** , ***let*** , ***public*** , ***static*** etc can not be used as names for variables/functions.
- Writing to a read-only property is not allowed

```
"use strict";
x = 3.14;       // This will cause an error
because x is not declared


function myFunction() {
  "use strict";
  y = 3.14;   // This will also cause an error because y is not declared
}

"use strict";
function x(p1, p1) {};   // This will cause an
error


"use strict";
var obj = {};
Object.defineProperty(obj, "x", {value:0, writable:false});

obj.x = 3.14;  //This will cause error
```

10. What is Babel ?

- Babel is a JavaScript transcompiler that is mainly used to **convert ECMAScript 2015+ code into a backwards compatible version of JavaScript**, that can be run by older JavaScript engines.
- Babel is a popular tool for using the newest features of the JavaScript programming language.