

Variables

var

let

const

Calling A Class

```
const varName = new classname(inputVars);
```

Template Literals

string text

```
string text ${expr ession} string text
```

Template literals are string literals allowing embedded expressions. You can use multi-line strings and string interpolation features with them.

Creating Function

```
if ( what you're testing ) {
  what to do when true;
} else {
  what to do when false;
}
};
```

Create Iteration

```
for ( let singleObject in dataset) {
  code;
};
```

single object = would be the specific item in the data set
dataset = would be the entire data set

Create If, Then, Else

```
if ( what you're testing ) {
  what to do when true;
} else {
  what to do when false;
}
};
```

Creating Classes

```
class className {
  con str inputVars) {
    //Input Variable Mapping
    inputVar = var;
  }
  //Methods, Setters and Getters
  methodName(inputVars) {
    code // e.g. return
  }
  methodName (inputVars) {
    code // e.g. return
  }
};
```

* No comma between items inside classes

Extending a Class

```
class newClass extends parentClass {
  con str inputVars) {
    inputVar() // calls the
parent class
    inputVar = var //creates
a subclass specific variable
  }
};
```

You must always call the super method before you can use the this keyword — if you do not, JavaScript will throw a reference error. To avoid reference errors, it is best practice to call super on the first line of subclass constructors.

