



vidcode

JavaScript Vocabulary + CS Concept Overview

Computational Thinking is the transferrable part of Computer Science. It is composed of five fundamental skills that can be applied to any type of problem:

- Decomposition - breaking a problem into smaller problems
- Generalization - comparing this problem to other problems
- Abstraction - deciding which details don't matter
- Pattern Recognition - deciding which parts repeat
- Algorithm Design - solving all the problems of one type

HTML



CSS



JS



JavaScript & Creative Coding

```
playbackRate = 7;  
runVideo(playbackRate);
```



```
respond('thanks!');
```





```
1 // Part: Earth
2 function earth(){
3   movie = video();
4   movie.source = "night-stars.mp4";
5   this.shadow = tint("black",0);
6
7   // Purpose: be lit or shaded
8   this.dim = function(perc){
9     this.shadow.amount = perc;
10  }
11 }
12
13 // Part: Sun
14
```

Eclipse 2017

A simulation of the solar eclipse using parts, purposes, and complexities.

338



<https://www.vidcode.com/share/YPpdJN1KEy>

Project: Make Your Own Filter

Vocabulary: Functions + Strings

JavaScript: the language

Computers aren't like humans. They can't read English or other languages humans speak! To make them do what we want, we have to give them instructions using a programming language that they understand.

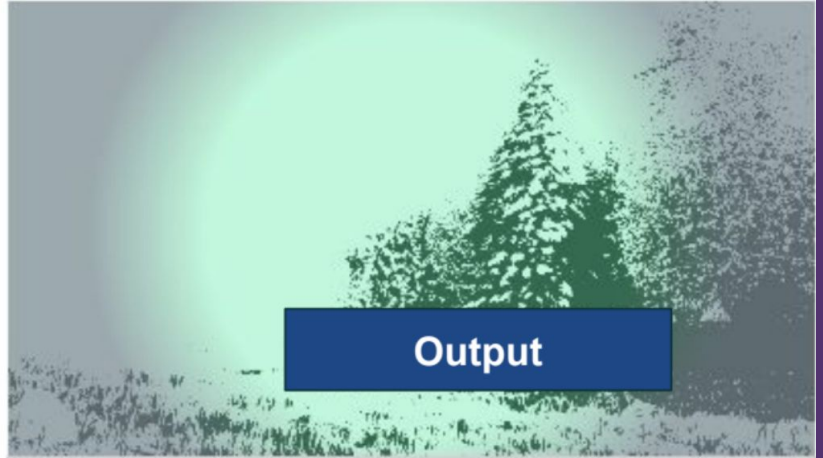
JavaScript: the language

JavaScript is one of the languages we use to speak to computers. It is just one of many programming languages out there.

JavaScript is the language used on all websites on the internet. For that reason, it is called “the language of the web”.


```
1  
2 black_and_white(1);  
3 vignette(55);  
4 tint("#71f0b2", 45);  
5
```

Input



Output



Functions

A function is an action in your code. You can think of it just like a verb in English. You can recognize a function by its parentheses.

When you add a filter to your video with the code

```
tint("red", 50);
```

or

```
invert();
```

you're using a function!

Functions

A function can either have empty parenthesis like this:

```
invert();
```

Or can have something inside the parenthesis:

```
blur(25);  
text("I am a programmer! 🤖", 100, 200);
```

When some data is inside a parenthesis, you are passing an **argument** to your function. A function can have one argument or many.

Data Types

String
Numerical
Boolean

Strings are text in your code. Unlike **numbers**, you have to put them inside quotation marks

Anytime you write a word or sentence, like a name or color, you are using a string.

```
tint("green", 10);
```

```
text("Don't forget emojis are string data ! 📡", 10, height/2);
```

Project: Make A Meme

Vocabulary: Variables + Object Properties

Variables

Are like storage containers in a program. You can recognize variables by the special JS keyword **var**

You can put any kind of data you want in the container - a **number**, a **string**, an **image** - anything you want! You should label your container with a name that makes sense based on what's inside of it.

```
var myAge = 12
```

```
var theGreeting = "Hello! I am a programmer! 🤖"
```

```
var i = false;
```

```
var theImage = graphic("lovers-gonna-love.png");
```

Variables

In programming we store data in variables so we can use or change it later. This becomes important as our programs become longer + more complex. Variables give us a way to references something in our code so we can use it over and over again.

```
var myAge = 12
```

```
var theGreeting = "Hello! I am a programmer! 🤖"
```

```
var i = false;
```

```
var theImage = graphic("lovers-gonna-love.png");
```


Objects + Properties

Just as in the real world, JavaScript objects are things we use in our code.

An **object** as **properties**. You can think of properties like adjectives used to describe an object.

In your code, **graphic** + **text** are both objects that have their own set of properties.

Objects + Properties

You can change an object's properties using **dot notation**. You can recognize dot notation by the dot that comes before the property name.

```
1  var my_text = text("Good day!");  
2  
3  my_text.color = "green";  
4  my_text.font = "Arial";
```

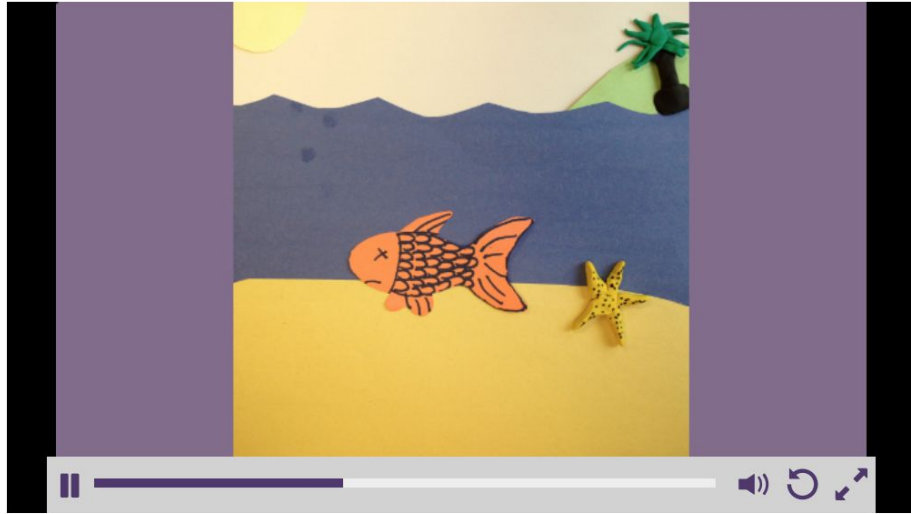
In the code above, the variable called **my_text** contains a text **object**. Using dot notation, you can change the **.color** and **.font** properties to customize our text!

Project: Slide Show

Cross-Curricular Coding Kit

Vocabulary: Arrays

Project: Slide Show



Fossil 2

```
1 movie = stopmotion();  
2 movie.frames = ["fish1.JPG","fish3.JPG", "fish2.JPG", "fish1.JPG"];  
3 movie.interval=500;  
4
```



Arrays

An array is a list of data in your code. If you make a stop motion animation, an array will hold your image data (aka photos, aka pix)! The could look like this:

```
movie.frames = ["mj_earth.jpg", "mj_working.jpg", "mj_floating.jpg", "mj_helmet.jpg"]
```

You can recognize an array by its square brackets. All of the elements in an array are separated by a comma.

Arrays

Let's say you have an array holding the string data for the names of days of the week.

```
var theWeekdays = ["Mon", "Tues", "Wednes", "Thurs", "Fri"];
```

```
var theDay = "day";
```

To retrieve data from the **array**, we have to refer to its **index** (or place, or number) in the list. Because computers start counting at **ZERO**, we will too.

Arrays

Let's say you have an array holding the string data for the names of days of the week.

```
var theWeekdays = ["Mon", "Tues", "Wednes", "Thurs", "Fri"];
```

```
var theDay = "day";
```

To retrieve data from the **array**, we have to refer to its **index** (or place, or number) in the list. Because computers start counting at **ZERO**, we will too.

0

1

2

3

4

Mon

Tues

Wednes

Thurs

Fri

Arrays

If we wanted to retrieve “Mon” from this **array**, we would write:

```
theWeekdays[0];
```

If we wanted to retrieve “Thurs” from this **array**, we would write:

```
theWeekdays[3];
```

To retrieve data from the **array**, we have to refer to its **index** (or place, or number) in the list. Because computers start counting at **ZERO**, we will too.

Project: Code the News

Cross-Curricular Coding Kit

Vocabulary: Loops

Project: Code the News

Out Women in Society Today



```
1 graphic_box = rect(420, 46, 230, 1000) ;
2 graphic_box.color = "pink";
3 var inset = graphic("img-chimamandainterviewvogue_16104120");
4 inset.x = 430
5 inset.y = 200
6 inset.scale = .6
7 var inset = graphic("18b573c178f1a6da8061ba10f00c1ac3.jpg");
8 inset.x = 430
9 inset.y = 50
10 inset.scale = .6
11 var headline = text ("Flawless Challenges Misconceptions at
12 headline.font = "Oswald" ;
13 headline.x = 0
14 headline.y = -6
15 headline.color = "pink" ;
16 headline.size = 55
```

Beyonce Flawless; Sociology

Loops

A loop is a sequence of instructions that is continually repeated in your program.

In Vidcode, loops can be recognized by the word **repeat**

Repeat is a special kind of function.

```
1 |  
2 repeat(function(){  
3   //write your repeating code here  
4  
5   }, 5);
```

Loops

In Vidcode, if you want to change something in your video so it changes over time, rather than stay the same throughout your video, you want to use a loop. Any of the instructions that you want to repeat over and over should go inside your repeat function.

In the example, we put the code that moves the text inside the repeat function. Otherwise, the text would stay in one place!

```
1  var my_text = text("hello");
2  repeat(function(){
3    //my text moves 10 steps to the right
4    //on repeat
5      my_text.x = my_text.x+10;
6    }, 1);
```

Boolean logic

Vocabulary: if statements

If statements

if statements are a way to give instructions to your computer. **If statements** follow the same logic of real world examples:

if it is raining outside, I will bring my umbrella.

In JavaScript, we write **if statements**:

```
if ( it-is-raining) {  
    umbrella  
}
```

```
if (studentGrade > 90) {  
    text("great job!");  
}
```

```
3   if (grade > 90) {  
4  
5       text("Great Job!");  
6  
7   }
```

if statements

We put the condition inside the parentheses of the if statement.

We put the instructions inside the curly brackets.

The condition will be evaluated by the computer as true or false. If it's true, the computer will follow the instructions inside the curly brackets. Otherwise, the computer won't follow the instructions.

```
3  if (grade > 90) {  
4  
5      text("Great Job!");  
6  
7  }
```

if statements

These are all examples of **if statement** conditions:

if (5 > 4) this statement would evaluate to **true**

if (10 < 2) this statement would evaluate to **false**

You can put anything in an if statement!

```
if (my_text.y > 200)
```

```
if (my_text.color === "blue")
```