KICKSTARTER:

JAVASCRIPT
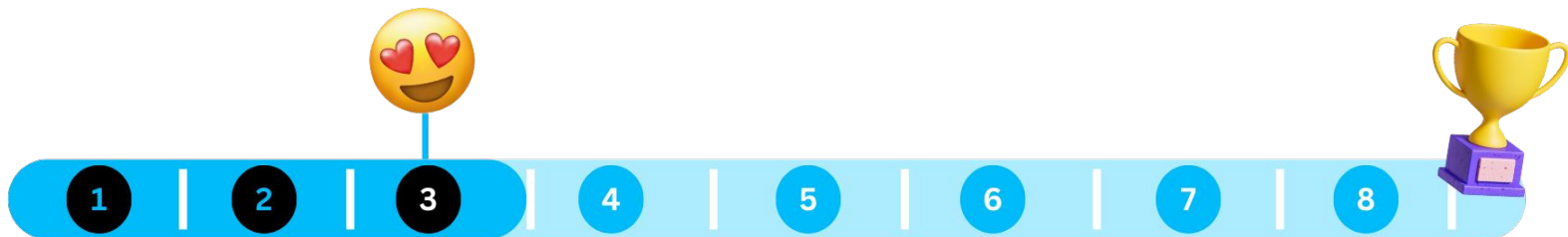
Used for for creating dynamic and interactive web applications

JS

YOUR JAVASCRIPT JOURNEY

INTRODUCTION TO JAVASCRIPT

1 | 2 | 3 | 4 | 5 | 6 | 7 | 8

1 Hello world!		4 DOM part one		7 Projects & presenting skills	
2 Making decisions		5 DOM part two		8 Project presentations	
3 Working with data		6 Project intro			

LESSON THREE: WORKING WITH DATA
SESSION OVERVIEW

+ Arrays

+ Arrays: Push and Pop

+ Objects

+ Loops: For

+ Loops: While

TOPIC OVERVIEW // ARRAYS
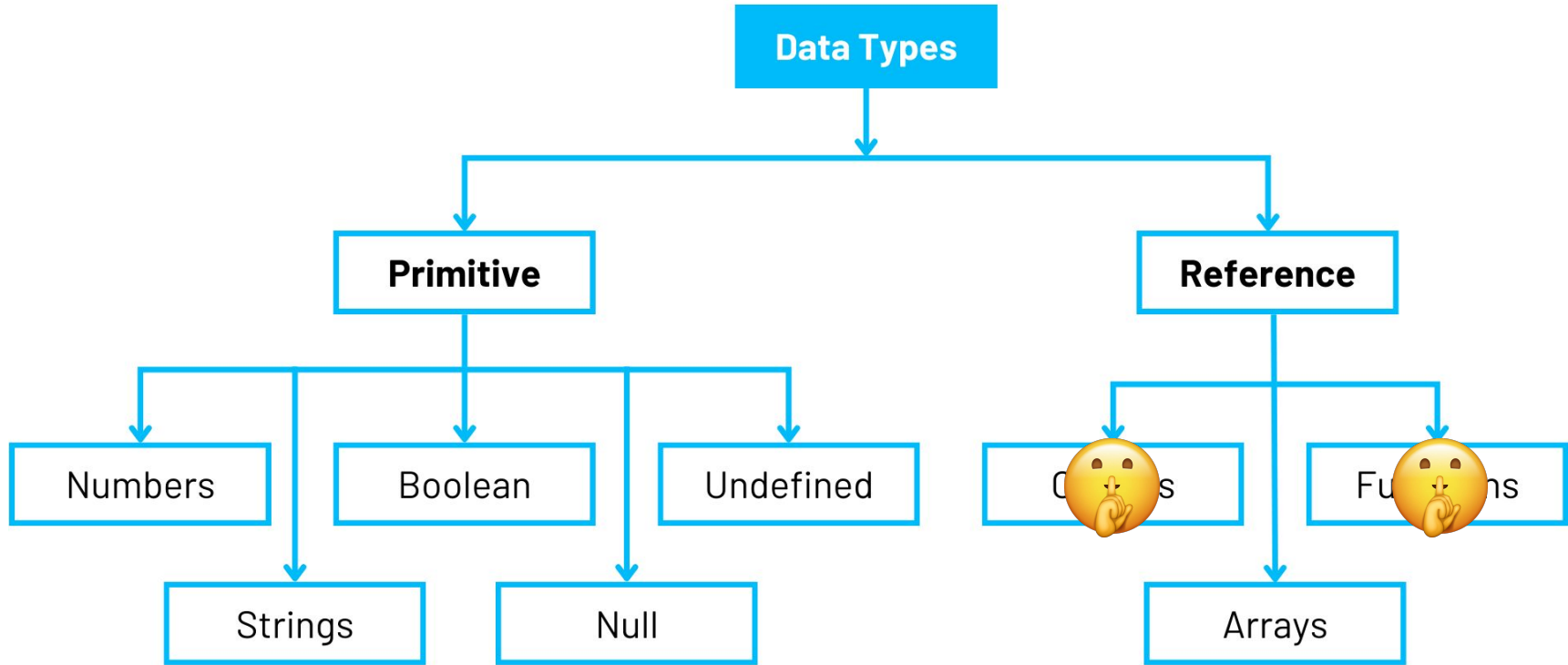
1. What is an array

2. Indexing through data

3. Syntax & Structure

DATA TYPES
LEARNING SESSION

DATA TYPE: ARRAY
LEARNING SESSION

A reference data type which allows you to store data in collections
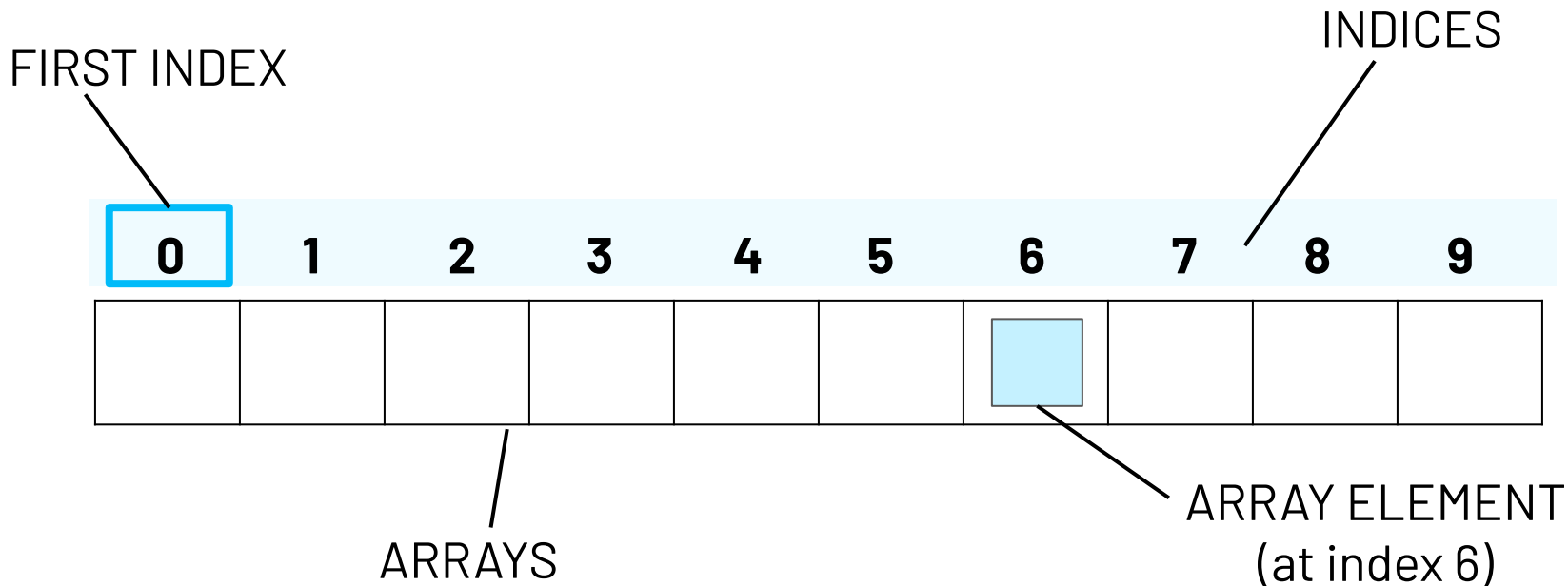
Imagine you wanted to store the names of lots of different kinds of fruit:

"apple", "plum", "cherry"

It wouldn't be very convenient to have to create a new variable for each one!

INDEXED ARRAYS

Indexed arrays have a numeric index.

Array elements are accessed using an index.

INDICES

FIRST INDEX

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

ARRAYS

ARRAY ELEMENT
(at index 6)

INDEXING THROUGH DATA
LEARNING SESSION

- Array indexes start from 0 (zero).

- Syntax: [element1, element2, ...]

- Array elements are accessed using an index.

- The length of the property returns the number of elements.

```
let fruit = ["apple, "plum", "cherry", "avocado"]

        console.log(fruit.length)
        console.log(fruit[0])
        console.log(fruit[1])
        console.log(fruit[2])
```

ARRAY

ELEMENT

INDEX
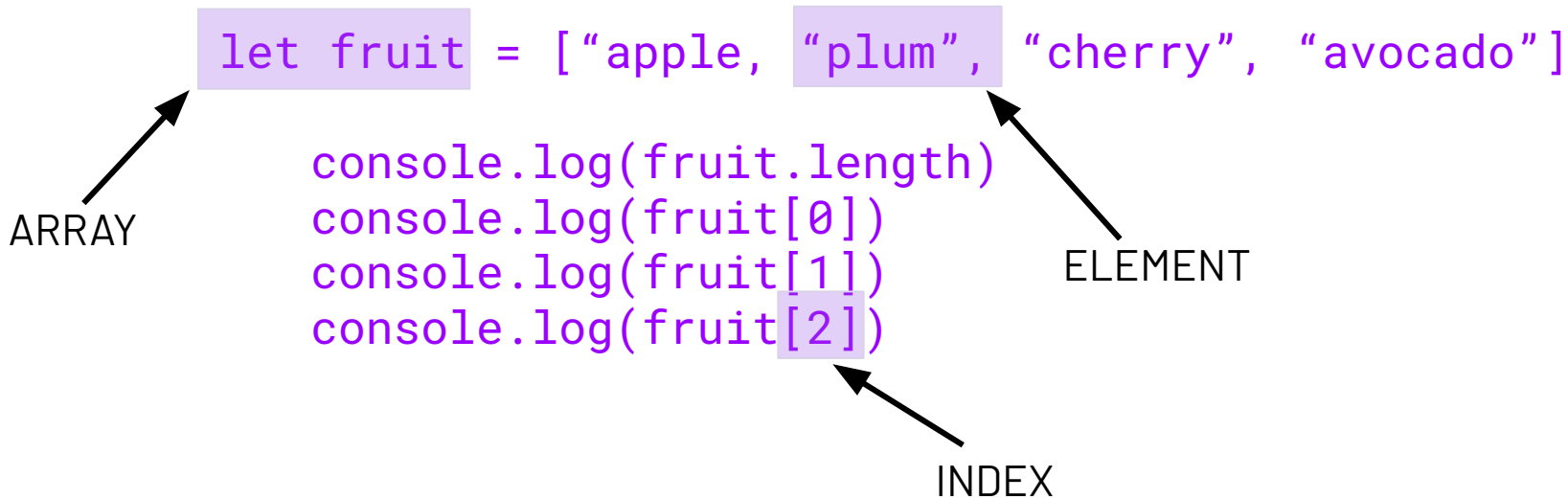
TOPIC 2: ARRAYS:
PUSH AND POP

TOPIC OVERVIEW // ARRAYS: PUSH AND POP

1. Adding and removing elements

2. Push and Pop

PUSH AND POP

push adds an element at the end of an array
pop removes the last element of an array and returns that element.

+ What if we decide that avocado isn't a fruit and shouldn't be in the array:
["apple", "plum", "cherry", "avocado"]

TOPIC 3: OBJECTS

TOPIC OVERVIEW // OBJECTS

1. What are objects?

2. Defining and using objects

3. Syntax

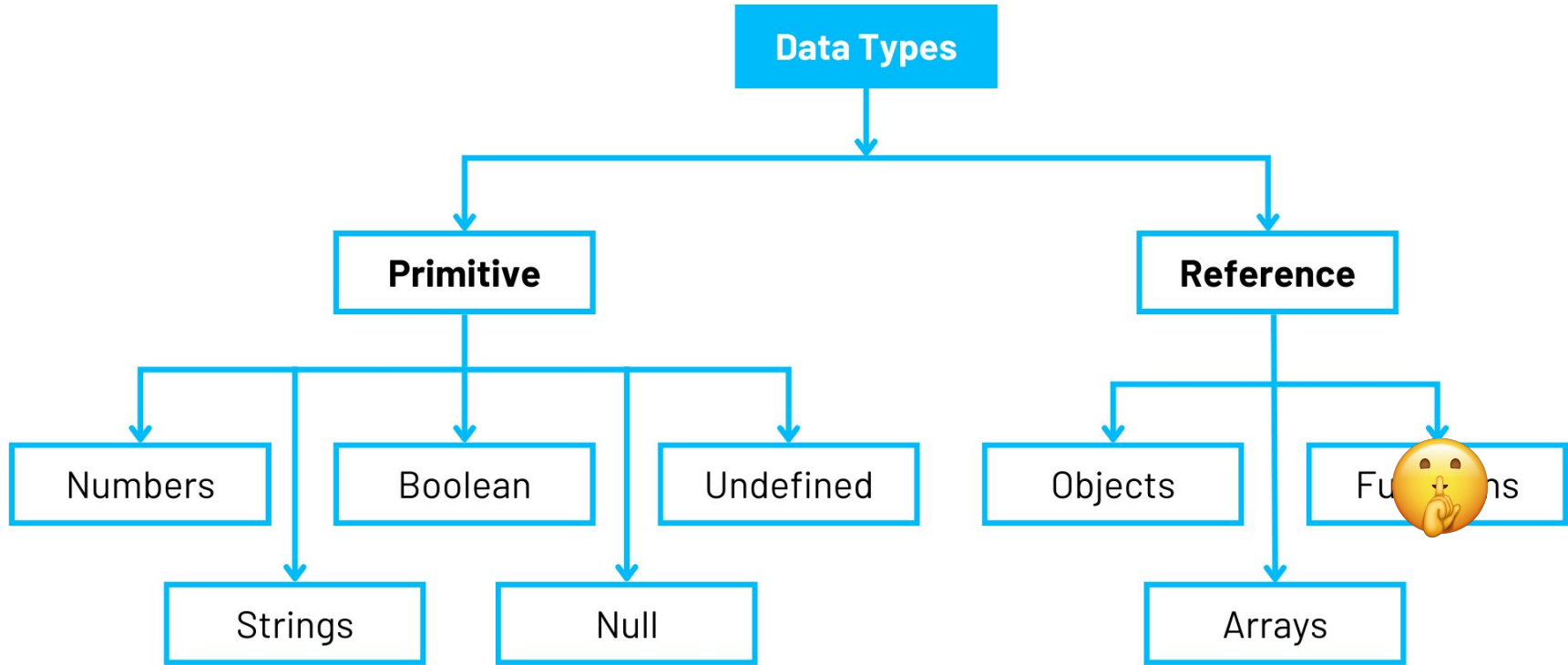4. Nested Objects

DATA TYPE: OBJECTS
LEARNING SESSION

The **Object** data type allows us to store multiple data values together

+ Sometimes it's useful to group multiple data values together with labels (keys)

+ For example, a book has a title, an author, and a publication year

+ It gets a bit messy if we have to define three variables for every book

DATA TYPES
LEARNING SESSION

```
                        ┌─────────────┐
                        │ Data Types  │
                        └──────┬──────┘
              ┌────────────────┴────────────────┐
        ┌─────┴─────┐                      ┌─────┴─────┐
        │ Primitive │                      │ Reference │
        └─────┬─────┘                      └─────┬─────┘
    ┌─────┬───┴───┬─────┐              ┌─────────┼─────────┐
┌───┴───┐ │ ┌─────┴───┐ │          ┌───┴───┐     │     ┌───┴────┐
│Numbers│ │ │ Boolean │ │          │Objects│     │     │ Fu🤫ns │
└───────┘ │ └─────────┘ │          └───────┘     │     └────────┘
      ┌───┴───┐     ┌───┴──┐                  ┌───┴───┐
      │Strings│     │ Null │                  │Arrays │
      └───────┘     └──────┘                  └───────┘
```

Numbers Boolean Undefined Objects Fu🤫ns

Strings Null Arrays

OBJECTS

```
const invisibleWomen = {
    title: "Invisible Women",
    subtitle: "Exposing the gender bias women face
    every day",
    author: "Caroline Criado-Perez",
    year: 2019
}
```

PROPERTY

```
console.log(invisibleWomen.title + ": " +
invisibleWomen.subtitle)
```

USE THE PROPERTY

```
let girl = {
    name: "Ana", // String
    age: 23, // Number
    friends: ["Mei", "Sabina", "Nuala"], // Array
    address: {
        // Object
        number: 123,
        street: "Cambridge Road",
        city: "London"
    }
};
console.log(girl.name + " lives in " +
girl.address.city)
```

NESTED
OBJECTS

USE THE NESTED PROPERTY

TOPIC 4: FOR LOOPS

TOPIC OVERVIEW // FOR LOOPS

1. Repeating code

2. Structure of a for loop

LOOPS: FOR
LEARNING SESSION: REPEATING CODE

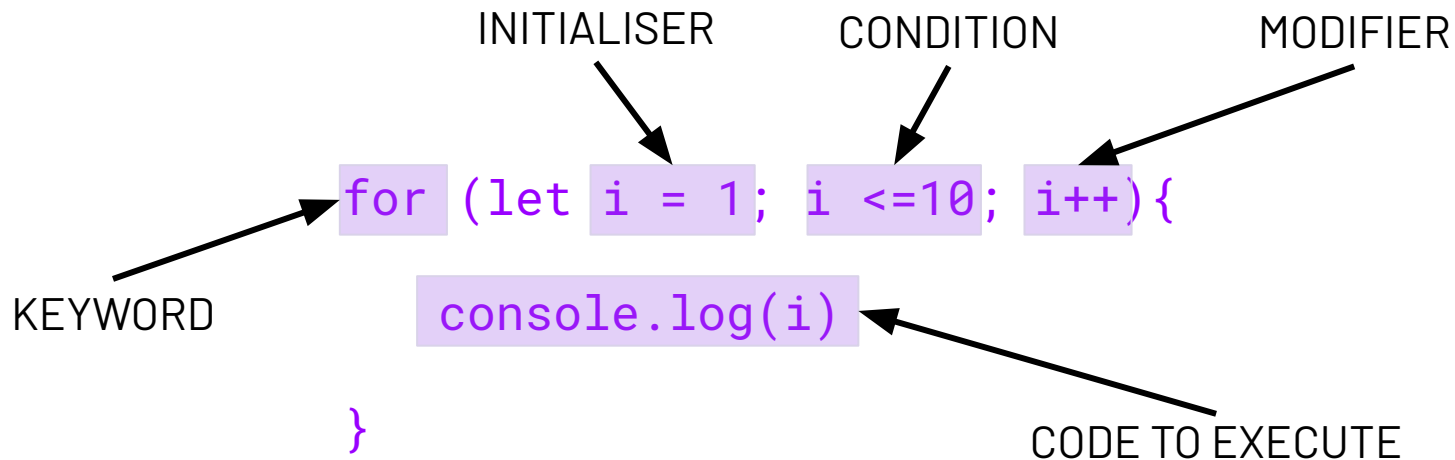A for loop repeats a block of code for a number of iterations.

+   Sometimes we want to repeat a block of code

+   We might want to do something 10 times or 42 times

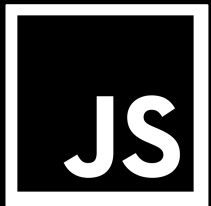+   It's often useful to have a loop counter variable that tells us which repetition of the loop we are on

LOOPS: FOR
LEARNING SESSION: SYNTAX

INITIALISER   CONDITION   MODIFIER

KEYWORD

```
for (let i = 1; i <=10; i++){

        console.log(i)

}

// Output:
// 1
// 2
// 3 …
```

CODE TO EXECUTE

TOPIC 5: WHILE LOOPS

TOPIC OVERVIEW // WHILE LOOPS

1. Repeating code

2. Structure of a while loop

3. Infinite Loops
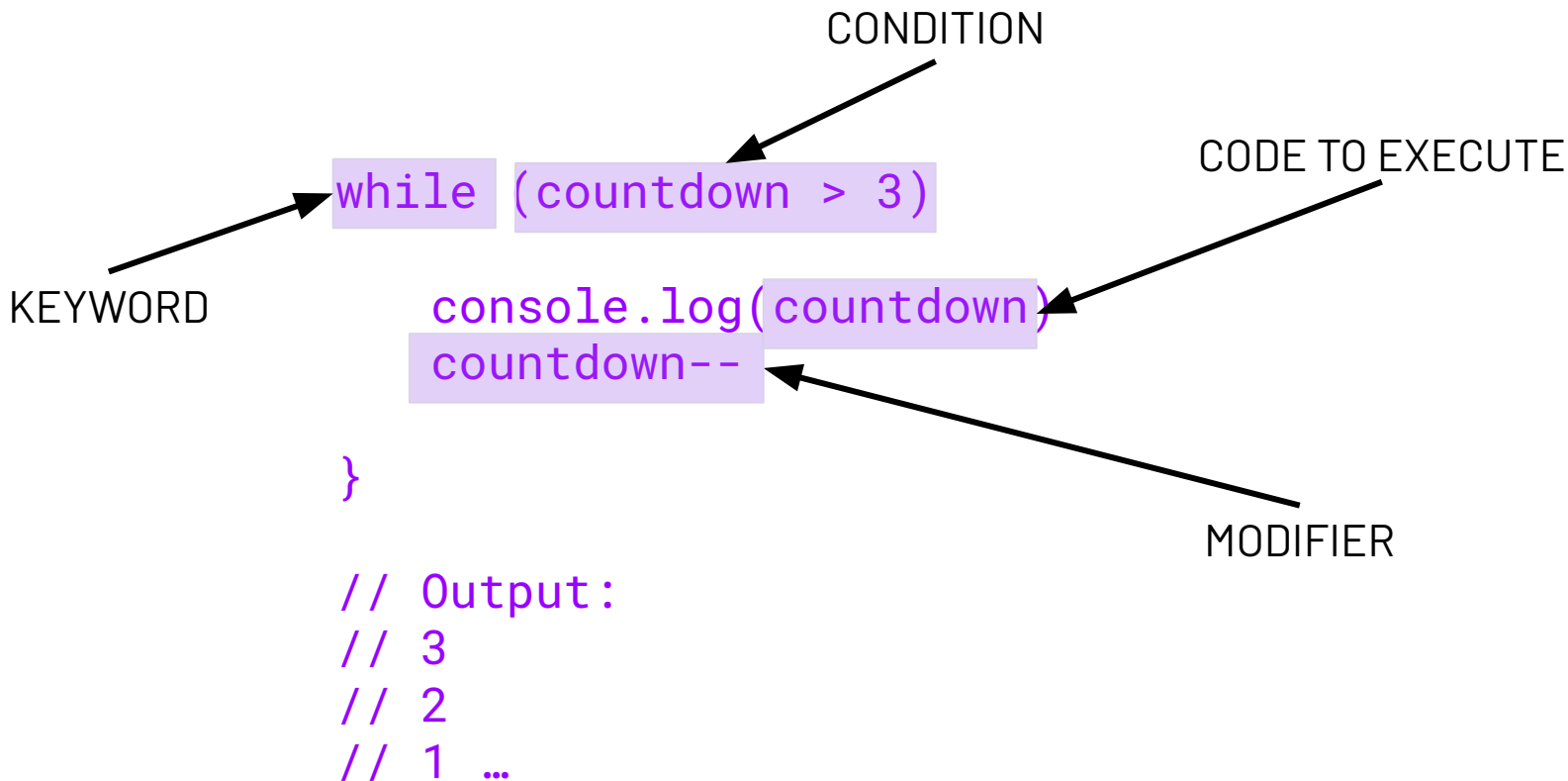
LOOPS: WHILE
LEARNING SESSION: REPEATING CODE

A **while** loop repeats a block of code while a condition is true.

+ Also used when we want to repeat an action

+ Let's say we don't know how many times we need a loop to run and can't use a for loop

+ While loops will repeat so long as a condition is true

LOOPS: WHILE
LEARNING SESSION: SYNTAX

CONDITION

CODE TO EXECUTE

```
while (countdown > 3)
      console.log(countdown)
      countdown--
}

// Output:
// 3
// 2
// 1 …
```

KEYWORD

MODIFIER

INFINITE LOOP
LEARNING SESSION

Infinite loops are coding structures that continuously execute a block of code due to a missing or incorrect condition, they can cause a program to crash.

+ Make sure to always change the loop variable inside of the loop

+ Otherwise it may run forever!

+ It could freeze your computer!

+ This is a bug called infinite loop

INFINITE LOOP EXAMPLE

⚠️

Do not do this as it would crash your codepen.

```javascript
let countdown = 5

while (countdown > 3)

    console.log(countdown)
    countdown++

}

// Output:
// 5 ...
// 22374 ...
// 12365487 ...
```

LIVE LAB
CHECKPOINT FOR LEARNING OUTCOMES

SKILL APPLICATION:

+   To create an array to store a collection of values.

+   Access specific elements within an array using their indices.

+   Use push to add an element at the end of an array.

+   Use pop to remove the last element of an array.

+   Create objects to store multiple data elements together.

+   To create well structured HTML documents by nesting elements within each other.

+   Create for loops to repeat blocks of code for a number of iterations

+   Create while loops to run for as long as a certain condition is true.

JOIN US FOR OUR LIVE LAB SESSIONS

★ **Remember to complete the quizzes for each topic this week to help build knowledge and understanding!**

★ **Live Lab sessions:**

● Elevate your learning experience by joining us in the live lab session.

● You'll get to put into practice what you have learned by completing exercises and working with your fellow students and instructors!

● You'll also have the chance to be put in a group of students to support each other through the class and work together on your final project!

● You can communicate through LearnDash and look out for the details of the live lab session taking place this week