# ADVANCED JS TOPICS

# LEARNING OBJECTIVES

- Define arrays
- Practice using indexes to access array elements
- Create objects and access and update values in objects.

# REVIEW

# WHAT ARE VARIABLES?

- ➤ We can tell our program to remember (store) values for us to use later on.
- The "container" we use to store the value is called a variable



# **DECLARING A VARIABLE**

var age = 29;

# JAVASCRIPT — UPDATING THE VALUE OF A VARIABLE

Declaring a variable:

Update the value of the variable:

# **ASSIGNMENT OPERATORS**

	Initial Value	Operator	Example	Result
Assign value to variable	var num = 8	=	num = 6	6
Add value to variable	var num = 8	+=	num += 6	14
Subtract value from variable	var num = 8	-=	num -= 6	2

# **COMPARISON OPERATORS**

Comparison Operators		
<	Less than	
>	Greater than	
<=	Less than or equal to	
>=	Greater than or equal to	

# **COMPARISON OPERATORS**

Equality Operators		
===	Strict equal to	
==	Equal to	
!==	Strict not equal to	
!=	Not equal to	

# ASSIGNMENT VS. COMPARISON — DON'T GET THEM CONFUSED!

Assignment	Comparison	
var number = 7;	if (number === 8) { // Do something }	

# JAVASCRIPT — IF STATEMENT

# **COMPARISON OR EQUALITY OPERATOR**

```
if (age > 65) {
   console.log("Senior Discount Applied");
}
```

Comparison Operators		
<	Less than	
>	Greater than	
<=	Less than or equal to	
>=	Greater than or equal to	

<b>Equality Operators</b>		
===	Strict equal to	
==	Equal to	
!==	Strict not equal to	
!=	Not equal to	

# IF STATEMENTS

```
if (age > 65) {
    console.log("Senior Discount Applied");
} else if (age < 18) {
    console.log("Student Discount Applied");
} else {
    console.log("Sorry, you don't qualify for a discount");
}</pre>
```

# WHAT CAN BE STORED IN VARIABLES?

# **DATA TYPES:**

1. Numeric	2. String	3. Boolean
Handles numbers	Consists of letters and/or other characters	Handles true or false values
Ex: 200.54 Ex: 893	Ex: 'GA@ga.co' Ex: "How are you user?"	Ex: true Ex: false
Used for tasks that involve counting or calculating	Used when working with any kind of text Written with single or double quotes	Used when there are two options for a value (i.e. yes/no, on/off, true/false)

"Numbers, Booleans, and strings are the bricks that data structures are built from. But you can't make much of a house out of a single brick. Objects allow us to group values—including other objects—together and thus build more complex structures."

Marijn Haverbeke "Eloquent JavaScript"

# ARRAYS

# **ACTIVITY — ARRAYS PART 1**



#### KEY OBJECTIVE

► Follow the steps under part 1 in arrays\_practice

#### TYPE OF EXERCISE

Individual

#### **LOCATION**

starter code> arrays\_practice

#### **TIMING**

1 min

- Follow instructions in main.js under part1.
- Check the console as you go to make sure you're on track!



# **ARRAYS — STORING LISTS OF VALUES**

- An array can be used to store a list of values in a single variable
- ▶ They are an ordered collection of values

0. "milk"

- 1. "eggs"
- 2. "cheese"

# **DECLARING ARRAYS**

var descriptiveNameHere = [item1, item2, item3];

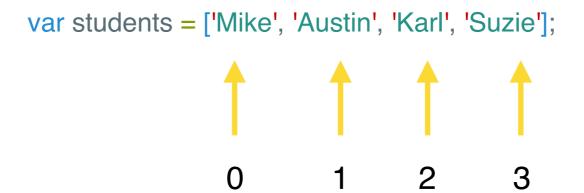
# **DECLARING ARRAYS**

```
var beverages = ["coffee", "tea", "hot chocolate", "milk"];
```

var leapYears = [2016, 2020, 2024, 2028];

#### **ARRAYS - ACCESSING ITEMS BY INDEX**

- Each item in an array has an index, by which you can access that item.
- ▶ The first item has an index of 0, the second item 1, the third item 2, etc.



#### ARRAYS — ACCESSING ITEMS IN AN ARRAY

var students = ['Mike', 'Austin', 'Karl', 'Suzie'];

# ACCESSING ITEMS (RETRIEVING VALUES):

students[1] --- 'Austin' students[2] --- 'Karl' students[0] --- 'Mike' students[3] --- 'Suzie'

# ARRAYS — ACCESSING ITEMS IN AN ARRAY

We can save what we find in a variable like so:

#### ARRAYS - ADDING A VALUE/REPLACING A VALUE

# **INSERTING A NEW VALUE**

To add a new value to the array, specify the index where the new value should be added.

students[4] = 'Matt';

```
['Mike', 'Austin', 'Karl', 'Suzie', 'Matt']
```

# **UPDATING VALUES**

If there's already an item at that position, it will be replaced with the new value.

```
students[4] = 'Sophie';
```

```
['Mike', 'Austin', 'Karl', 'Sophie', 'Matt']
```

### **ARRAYS - LENGTH**

➤ We can use the .length property to find out how many items are in an array

The length property is also useful for accessing the last item in an array

# **ACTIVITY — ARRAYS PART 1**



#### KEY OBJECTIVE

▶ Define arrays and practice using indexes to access array elements

#### TYPE OF EXERCISE

Individual

#### **LOCATION**

starter code> arrays\_practice

#### TIMING

5 min

- Follow the instructions under Part 2
- Be sure to check your console as you work!

# ARRAYS LAB

# **ACTIVITY — QUOTE CAROUSEL**



#### KEY OBJECTIVE

Apply JS and jQuery knowledge to program a quote carousel.

#### **TIMING**

20 min

- 1. Open the page in your browser.
- 2. Follow the instructions in main.js
- 3. Keep your console open, use the debugger, log values to the console to check things
- 4. Bonus: Create an array of colors (these should be strings in hex format, e.g. '#17A9F8'). Each time the quote changes, use the .css() method to change the background color as well. Hex codes: '#17A9F8','#59B776', '#E8519C'
- 5. **Super bonus:** Switch between more than 3 background colors.

# OBJECTS

### THE NEED FOR OBJECTS

- In our programs, we'll sometimes want the ability to model real life objects.
- We could store information about an object in an array like so:

```
var artist = ["Michael", "Jackson", 1958, 13];
```

What does the number 13 refer to? How old Michael Jackson was when he sang his very first hit song? The number of best selling albums he made?

# THE NEED FOR OBJECTS

Objects allow us to associate keys with values:

```
var artist = {
  firstName: "Michael",
  lastName: "Jackson",
  birthYear: 1958,
  numberOneHits: 13
};
```

# **OBJECT SYNTAX**

```
var user = {
    firstName: "Bill",
    lastName: "Smith",
    email: "billsmith@email.com",
    age: 29
    };

Values
```

### **ACCESSING VALUES IN AN OBJECT**

```
var user = {
    firstName: "Bill",
    lastName: "Smith",
    email: "billsmith@email.com",
    age: 29
};
Values
```

• To access values in an object: use the object name followed by a dot followed by the property we want to access:

#### **UPDATING VALUES IN AN OBJECT**

To update values in an object: use the object name followed by a dot followed by the property we want to update. Then assign a new value:

```
user.email = "billy@email.com"
user.age = 30
```

#### **BEFORE UPDATING:**

```
var user = {
  firstName: "Bill",
  lastName: "Smith",
  email: "billsmith@email.com",
  age: 29
};
```

#### **AFTER UPDATING:**

```
var user = {
  firstName: "Bill",
  lastName: "Smith",
  email: "billy@email.com",
  age: 30
};
```

# **ACTIVITY — OBJECTS PART 1**



#### KEY OBJECTIVE

 Define objects and practice using dot notation to update and access properties in an object

#### **LOCATION**

starter code> objects

#### **TIMING**

5 min

Follow instructions in main.js for part 1.

# **ARRAYS CONTAINING OBJECTS**

```
var fruits = [
  color: "red",
  type: "apple",
  price: 79
  color: "green",
  type: "avacado",
  price: 1.50
```

# **ARRAYS CONTAINING OBJECTS**

```
var fruits = [
  color: "red",
  type: "apple",
  price: 79
  color: "green",
  type: "avacado",
  price: 1.50
```

Accessing a value:

fruits[0].color

fruits[1].price

Updating a value:

fruits[0].color = "green";

fruits[1].price = 1.00;

# **ACTIVITY — OBJECTS PART 2**



#### KEY OBJECTIVE

 Define arrays and practice using indexes to access array elements

#### TYPE OF EXERCISE

Individual/Paired

#### **LOCATION**

starter code> objects

#### TIMING

5 min

Follow instructions in main.js for part 2.

# **ACTIVITY — OBJECTS PART 2**



#### KEY OBJECTIVE

 Define objects and practice using dot notation to update and access properties in an object

#### **LOCATION**

starter code> objects

#### TIMING

5 min

Follow instructions in main.js for part 2.

# LAB

# **ACTIVITY — IMAGE CAROUSEL**



#### KEY OBJECTIVE

▶ Apply JS and jQuery knowledge to program an image carousel.

#### **TIMING**

- 1. Follow the instructions in main.js
- 2. Bonus: Complete the bonus version.

# LEARNING OBJECTIVES

- Define arrays
- Practice using indexes to access array elements
- Create objects and access and update values in objects.

# **ADVANCED CSS**

# HOMEWORK

# **HTML BASICS**

# EXIT TICKETS