**03.OBJECTS AND COMPOSISION**

1. **Objects in JS**

An object is a collection of properties in key(identifier)/value pairs  
Reference data type

Created with object literal: const name = {

firstName: “First Name”,

secondName: “Second Name”

};

A property of an object can be explained as a variable attached to the object

The properties can be any type of data and can be reassigned

The properties are accessed via `.` dot-notation or via []-bracket notation

Non existed properties are set to undefined if someone try to access them

Destructuring syntax – const {propertyOne, propertyTwo} = objectName

Properties are deleted with “delete” operator 🡪delete objectName.property

1. **Objects as Associative Arrays**
2. **Methods and Context**

for-in loop 🡪 iterates over all enumerable properties.

Object.keys(objectName) 🡪 returns an array fulfill with all the keys

Object.values(objectName) 🡪 returns an array fulfill with all the values

Object.entries(objectName) 🡪returns and array with [key,value] tuples, used when sorting objects

objectName.hasOwnProperty(property)

Objects can also have methods – functions declared inside an object. The method is property, whose value is the function.

Objects can be used as function libraries

Short-syntax to attach methods to an object is:

objectName {

methodName() {..method body..},

}

Accessing Object Context:

* Functions in JS have execution context
* Accessed with “this” keyword 🡪 instance of the method’s object
* When “this” is executed as an object method, the context is a reference to the parent object

1. **Object Composition**

Combining simple objects into more complex ones

Nested objects

Powerful for code reusage

It is considered superior to OOP inheritance

Describes objects in terms of what they do, not what they are

Object.assing({}, referenceObject) 🡪 returns new object with properties copied from the other given objects

Decorator functions 🡪 add new data and behavior to objects

1. **JSON**

Data interchange format

name/value pairs

It is used to read data from a web server and display the data on a web page

JSON.parse(data) 🡪convert the data into JS object

JSON.stringify(object) 🡪convert the given object into string