

Algorithms and Computer Programs

Lesson Objectives

- Understand algorithms and it's parallel to computer programs
 - Know basic computer operations
 - Learn the flow of an algorithm/ a program.
 - Understand concept of function as a modular algorithms
 - Understand concept of object as a wrapper that binds data and functions together.
-
- Learn to declare and assign value to a variable in JavaScript.

Algorithms

- Sequence of instructions to achieve a specific task
 - Recipes to prepare a meal
 - Driving instructions
- Algorithm to brush teeth
 - Hold your toothbrush
 - Pour small amount of toothpaste
 - Start brushing
 - Stop brushing after 2 mins or when you feel fresh enough.
 - Rinse your mouth

Example

- Algorithm for computing the average test score from a list of test scores.
 1. Set *sum* to 0
 2. Get the number of Students
 3. For each student
 - a) Get the student's test score
 - b) Add the student's test score to *sum*
 4. Divide the *sum* by the number of students
- Notice that it is nothing more than the step-by-step process for performing the calculation.

Computer Program

- Algorithm that a machine can execute
- A program is composed of data and instructions.
 - Data (like ingredients of a recipe)
 - Instructions (like steps to prepare a recipe)

Machine is Dumb

- The computer doesn't know anything
 - About anything at all
- The computer (memory) can store things
 - Data items that it is told to store
 - Algorithms/Instructions (steps) aka programs
- The computer (processor) can read and execute the instruction saved in memory to access/manipulate data in memory.

Data

- Would the algorithm for computing average test score work in a machine?
 - Does computer know about students?
 - Does computer know about their scores?

Model (Structured Data)

- Representation of real-world objects.
 - An abstraction (just containing the things you need)
 - It's always different, but always related to the task at hand.
- A person model for video game will be different from a person model for a hospital or a university.

Instructions

- The instructions in a computer program are like the steps in algorithm.
 - In programming language like JavaScript, the instructions are called statements.
- Instructions access and manipulate models
 - The program to calculate average test score would get (access) test score from every student.
 - Would update (manipulate) the value of *sum* after adding test score for each student.

Main point 1

- An algorithm is a step by **step sequence of operations** to accomplish some task. A computer program is an algorithm written using a language that the machine understands. *Science of Consciousness*, The source of thought is pure silence, and thought is the source of all languages. *We experience the field of pure silence during our practice of the Transcendental Meditation Program.*

Functions

- When a computer program gets bigger, it is broken down into smaller, modular pieces called functions.
 - `performDailyRoutine()`
 - `freshnUp()`
 - `brush()`
 - `shower()`
 - `breakfast()`
 - `doWork()`
 - `afterWork()`
 - `sleep()`
- Another approach is to write functions first (bottom up approach) and compose them into a program.

Objects

- Wrap data and related functionality as one entity

```
car{
```

```
    speed = 0;
```

```
    speedUp(){ speed = speed + 1}
```

```
    brake(){speed = 0}
```

```
}
```

```
car.speedUp() // speed is 1
```

```
car.speedUp() // speed is 2
```

```
car.brake() // speed is 0
```

Computer Operations

- Input (Receive Data)
 - Input devices: keyboard, mouse, camera, mic, sensors
 - Input from persistence storage (files and databases)
- Process (Execute instructions to operate on data)
 - Assign values to a memory location (variables holding data)
 - $x = 5;$
 - Perform different kind of operations: arithmetic, relational, logical
 - $z = x + y;$
- Output (Present computed result)
 - Output devices: monitor, speakers, printers
 - Output to persistence storage (files and databases)

The equals sign

- Many programming languages use the equals sign for assignment (storage) into variables.
- Important! The equal sign in most programming language does not indicate equality.
 - It indicates assignment

Exercise

- What is the output of the following

A = 10

B = 20

A = B

print A

print B

Control Structures

- Sequence
 - Execute statements in the order they are written (top to bottom), default
- Selection (branching)
 - if (day is Saturday or Sunday)
 - haveFun()
 - else
 - goToWork()
- Repetition (Looping)
 - Until it's 2 mins keep brushing.
 - Until I say stop, keep on counting
- Function call and return
- try, catch, and throw

Main Point 2

- A computer program, no matter how complex, has three key control structures i.e. sequence, selection & repetition. *Science of Consciousness*, All complex structures are built upon more fundamental structures. *Pure consciousness, the unified field, is the most fundamental structure of all existence.*

JavaScript in browser

- JavaScript is a high-level programming language understood by all modern browsers.
- JavaScript was originally developed to work with HTML pages in a browser application.

Statements

- A program is composed of statements.
- A single statement is equivalent to a single step in an algorithm.
- A statement is a complete command within a program.
- Statements in JavaScript ends with semicolon (;)

Hello, World!

- The traditional 'first program' for any language
 - Output "Hello, World!"
 - `alert ("Hello, World!");`
 - `console.log("Hello, World!");`
- JavaScript is case sensitive, `alert()` is different from `Alert()` or `aLert()`.

window

- When JavaScript runs on a browser, it runs inside the global environment called **window (object)**.
 - `alert()` and `prompt()` are methods (functions) of window object for alerting output and displaying prompt for user input.

Declaring Variables

- `var` keyword was the only way to declare variables in JavaScript before ES6.
 - `var x;`
 - `var speed;`
 - `var name;`
- ES6 introduced `let` keyword
 - `let x;`
 - `let speed;`
 - `let name;`
- Should follow identifier naming rules while declaring a variable (page 35).

Variables

- Memory location referenced by some identifier like x and y.
 - Machine saves data on those memory locations.
 - Machine access/ manipulates variables in a program in order to compute results.

Assigning values to a variable

- When variables are declared their default value is 'undefined'

```
let x;  
console.log(x) // undefined  
x = 5;  
console.log(x) // 5  
x = 'hello'  
console.log(x); // hello
```


Declaring & assigning constants

- ES6 also introduced const keyword when you need to declare named constants.

```
const WEEK_DAYS = 7;  
const PI = 22/7;
```

Good coding practice

- Use `const` if the value won't change after assignment
- Use `let` for variables
- **Never use `var`**

Reserve keywords

- Book page 31

Examples

- Write a program to output 5+5
- Update program to save some integer values on **constants** X and Y and print the sum.
- Update program to save some integer values on **variables** x and y and print the sum.

Defining Table

- A tool to better understand the task in hand
 - You don't want to give right answer the wrong question? Do you?

Input	Process	Output

- Example 1, Chapter 2, Page 9-10
- Exercise, Example 2

Main Point 3

- Understanding a problem is key to producing working algorithms/computer programs. *Science of Consciousness, How well a human can understand a situation depends on his/her level of perception. We perceive better when our thinking is clear, and our thinking is clear when we are free of stress.*

Assignment

- Reading Chapter 2
- Review Questions (all)
- Try following on your own for today, you can find tutorials online and videos on YouTube
 - Open account at github.com and create a repository
 - Download and install GitHub desktop and clone repository you created in step1 in your machine.
 - Download and install VSCode