What are microtasks? What is a microtask queue? What is their role in Promises and how are they different from callbacks?

Microtasks - These are the functions executed after the functions or program which is being executed in the call stack. When the call stack is empty and just before the event loop runs, these microtasks which are in the microtask queue gets executed by making a completely empty queue.

A Microtask queue is a queue in which microtasks such as promises and any mutation API will be added to this queue. And get executed after JavaScript call stack is empty and control is not with JavaScript code then microtasks will be executed until the queue is empty and before control going to event loop to pull the tasks which are present in the task queue such as timeouts, intervals.

Explain with examples how private, protected variables can be implemented in classes and how can they be used in subclasses?

Private variables:

class privateVariableClass {

#private\_variable;

constructor(name) {

this.#private\_variable = name;

}

returnPrivateVariable() {

return this.#private\_variable;

}

#returnMessage() {

return `This object contains a private variable`;

}

}

const obj = new privateVariableClass("Hello");

console.log(obj.returnPrivateVariable()); *// Hello*

console.log(obj.#private\_variable); *// undefined*

console.log(obj["#private\_variable"]); *// Error*

console.log(obj.#returnMessage()); *// Error*

The # method for declaring private members of a class is in part of ES2019/ES10

Protected variables:

class ProtectedVariable {

\_name;

constructor(name) {

this.\_name = name;

}

get name() {

return this.\_name;

}

}

const protectObj = new ProtectedVariable("Hello");

console.log(protectObj.name()); *// Hello;*

protectObj.\_name = "Janny"; *// Cannot assign new value because it's read only variable*