**Resultats de l’aprenentatge:**

RA1. Selecciona les arquitectures i tecnologies de programació sobre clients web, identificant i analitzant les capacitats i característiques de cadascuna.

RA2. Escriu sentències simples, aplicant la sintaxi del llenguatge i verificant la seva execució sobre navegadors web.

**EXERCICI 1. Pels 20 fragments següents, indica’n la seva sortida per consola i justifica la resposta (temps màxim: 30 min).**

|  |  |  |
| --- | --- | --- |
|  | **SORTIDA PER CONSOLA** | **JUSTIFICACIÓ** |
|  | **TypeError** | **Los elementos del DOM no son accesibles como propiedades del objeto document.body.** |
|  | **Lista de lenguajes** | **Los elementos del DOM estan disponibles como variables globales por su id.** |
|  | **ReferenceError** | **Los elementos del DOM no estan disponibles como variables globales por su clase.** |
|  | **JavaScript** | **El método childNodes devuelve los NODOS hijos de un elemento. Esto incluye no sólo elementos HTML, sino también elementos de texto y comentarios. En este caso, el nodo que se elimina es un nodo de texto correspondiente a un salto de línea antes del primer <li>.** |
|  | **TypeScript** | **El método children devuelve los ELEMENTOS hijos de un elemento. Esto incluye sólo elementos HTML. En este caso, el nodo que se elimina es el primer <li>.** |
|  | **TypeError** | **El método querySelector devuelve un elemento, no una lista; por tanto, no se puede indexar con [ ].** |
|  | **Python** | **El método querySelectorAll devuelve siempre una lista (HTMLCollection); por tanto, se puede indexar con [ ].** |
|  | **undefined** | **El método querySelectorAll devuelve siempre una lista (HTMLCollection); por tanto, se debe indexar con [ ].** |
|  | **Python** | **El método nextElementSibling devuelve el hermano siguiente al elemento. Aquí se encadena dos veces, con lo que accedemos al contenido del tercer <li>.** |
|  | **TypeError** | **No se puede buscar en el documento por #titulo la segunda vez, ya que se ha cambiado el id del elemento.** |
|  | **programacion** | **El setAttribute añade un atributo al elemento HTML cob valor programación, y este valor se lee a continuación.** |
|  | **undefined** | **No se puede acceder a atributos no estándar de un elemento como propiedades del objeto correspondiente.** |
|  | **deModa** | **Cuando asignamos una valor a className, se machaca el valor anterior.** |
|  | **lista deModa** | **Cuando añadimos una clase mediante classList.add, className refleja las clases anteriores y la nueva.** |
|  | **“”** | **El método classList.toggle añade la clase si el elemento no la tenía, y la quita si el elemento ya la tenía.** |
|  | **green** | **Las propiedades CSS de un elemento son accesibles como propiedades del style del elemento.** |
|  | **ReferenceError** | **Una propiedad que en CSS se escribe con guion, se debe escribir en camelCase en JS.** |
|  | **TypeError** | **No se puede acceder al quinto <li> porque no ha sido añadido aún a la lista.** |
|  | **Ruby** | **Al haber añadido el nuevo <li> al final de la lista (append), el quinto elemento resulta ser el nuevo.** |
|  | **Dart** | **Al haber añadido el nuevo <li> al principio de la lista (prepend), el quinto elemento resulta ser el que era cuarto en la lista original.** |