| **EVIDENCIAS DE COMPETENCIA**  **Demostración de conocimientos y estrategias cognitivas** | |
| --- | --- |
| **Resultados a comprobar** | |
| CE1.1 Distinguir y explicar pautas de accesibilidad al contenido en los documentos elaborados para permitir una mejor navegación y comprensión de los usuarios.  CE1.2 Distinguir y explicar pautas de usabilidad al contenido en los documentos elaborados para permitir una mejor calidad, efectividad y satisfacción de los usuarios. | |
| **INDICADORES DE LOGRO** | **SISTEMA DE VALORACIÓN** |
| Identificación de la respuesta correcta entre varias alternativas | En los ítems tipo test, con cuatro respuestas posibles, se aplicará la siguiente fórmula:  **P.D= A-(E/n-1)** |
| **METODOS E INSTRUMENTOS DE EVALUACION** | |
| Formulación de preguntas mediante una prueba objetiva de 20 ítems distribuidos como sigue:  20 tipo test, con cuatro alternativas, en donde sólo una es la correcta.  Antes de responder, hay que leer atentamente la pregunta y las posibles respuestas, con el fin de evitar errores.  Marca con un círculo la letra de la respuesta correcta. En caso de equivocarse, tacha la marcada con una cruz y marcar, con un nuevo círculo, la correcta.  **Su duración total será de 1 hora aproximadamente.** | |

**1. Al declarar una función, entre paréntesis escribimos**

1. Nada
2. Los argumentos
3. Los parámetros
4. Las Strings

**2. Es obligado que una función devuelva (*return*) un valor**

a) Siempre

b) Nunca

c) A veces

d) Las funciones no devuelven valores.

**3. En esta llamada a una función: *calcula(a, b).* Tanto *a* como *b* son**

a) Objetos

b) Parámetros

c) Argumentos

d) Undefined

**4. Una función anónima se define como aquella que**

a) Se declara dentro de un objeto

b) Se declara sin asignarle un nombre

c) Se declara siempre sin argumentos

d) Se declara siempre con argumentos

**5. Los objetos admiten que se creen funciones en su ámbito de ejecución**

a) Siempre y cuando tengan argumentos

b) No. Nunca.

c) Si. Sin problemas

d) Siempre y cuando devuelvan algún valor

**6. ¿Cuál es la principal ventaja de las funciones?**

a) Son más rápidas que el resto del código

b) Son reusables en nuestro código

c) Son muy buenas para trabajar con números flotantes

d) Ninguna de las anteriores

**7. La declaración *nuevosColores = [...masColores]***

a) Añade el array *masColores* al array *nuevosColores*

b) Crea un objeto *masColores* igual al array *nuevosColores*

c) Crea un array *nuevosColores* exactamente igual a *masColores*.

d) Elimina del array *nuevosColores* el contenido de *masColores*.

**8. La declaración *const [, , , pececito] = peces***

a) Crea una constante llamada *pececito* que contiene el valor de la variable *peces*

b) Crea una variable *pececito* que copia el valor de la cuarta posición en el array *peces*

c) Crea una variable *pececito* que copia el valor de la cuarta posición en el objeto *peces*

d) Es inválida

**9. Método y función se distinguen…**

a) Solo en que la función siempre devuelve un valor

b) Solo en que el método siempre devuelve un valor

c) En nada. Son dos nombres para lo mismo.

d) Solo en que un método es cualquier función asociada a un objeto.

**10. La denominación key:value se utiliza en la definición de**

a) Arrays

d) Variables

c) Elementos del DOM

d) Objetos

**11. Una función anónima se define como aquella que**

a) Se declara dentro de un objeto

b) Se declara sin asignarle un nombre

c) Se declara siempre sin argumentos

d) Se declara siempre con argumentos

**12. Para verificar que un array contiene determinado elemento usamos el método**

a) contains

b) indexOf

c) includes

d) parent

**13**. **¿En un objeto denominado** *gatito* **podemos usar la propiedad** *raza* **(ósea** *gatito.raza***) sin que esta se haya creado previamente?**

a) Si. Javascript lo permite.

b) Si. Pero solo si el objeto está vacío

c) No. No se puede

d) Los objetos no admiten propiedades

**14. La siguiente expresión:** *const {raza} = gatito***, realiza una operación denominada**

a) Estructuring de objetos

b) Estructuring de Arrays

c) Destructuring de Arrays

d) Destructuring de Objetos

**15. El iterador** *for..of* **se utiliza para iterar sobre:**

a) Los elementos de un array

b) Los elementos de un objeto

c) Equivale a un bucle *for* normal

d) No existe

**16. Un resultado de -1 devuelto por el método** *indexOf* **significa que:**

a) El elemento se encuentra en la posición -1 del array

b) El elemento no existe en el array

c) El elemento está duplicado en el array

d) *indexOf* nunca devuelve -1

**17. El método** *includes* **siempre:**

a) Devuelve *true* o devuelve *false*

b) Devuelve un valor numérico con la posición del elemento

c) Devuelve *null*

d) Devuelve *undefined*

**18. Un String realmente se comporta como:**

a) Una variable de tipo booleana

b) Un array

c) Un objeto

d) Un prototipo

**19. El método** *split* **aplicado a un String devuelve**

a) Otro string de elementos separados por espacios

b) Otro string de elementos separados por comas

c) Un array de elementos

d) No existe

**20. Para una asignación** *a = ‘-2023.456’* **la operación** *+a* **devuelve**

a) undefined

b) -2023.456

c) -2023

d) ‘-2023’