

Nome e Cognome: \_\_\_\_\_ Matricola: \_\_\_\_\_

**Istruzioni:** Il tempo a disposizione per completare la prova è di **50 minuti**. Durante la prova, non è consentito consultare materiale didattico, né interagire con dispositivi elettronici o colleghi. Nel caso di domande a risposta multipla con più risposte corrette, ciascuna risposta corretta indica una corrispondente frazione del punteggio complessivo, mentre le risposte errate comportano una pari penalizzazione. In nessun caso, il punteggio complessivo relativo a una domanda sarà negativo.

Es. 1. (2 punti) Qual è il significato degli status code della famiglia 400 di HTTP?

- |                            |   |                            |                                     |
|----------------------------|---|----------------------------|-------------------------------------|
| <input type="checkbox"/> A | Errore nel processamento della richiesta. | <input type="checkbox"/> C | Errore nella trasmissione dei dati. |
| <input type="checkbox"/> B | Errore nella richiesta.                   | <input type="checkbox"/> D | Errore nella connessione.           |

Es. 2. (2 punti) Quale dei seguenti selettori CSS seleziona tutti i link contenuti in `<li>` di posto *pari* in una lista non ordinata con classe `lst`?

- |                            |  |                            |  |
|----------------------------|--|----------------------------|--|
| <input type="checkbox"/> A | <code>ol.lst li:even-child a</code>      | <input type="checkbox"/> D | <code>ul.lst::nth-child(2) a</code>      |
| <input type="checkbox"/> B | <code>ul.lst li:nth-child(even) a</code> | <input type="checkbox"/> E | <code>ol.lst li:nth-child(even) a</code> |
| <input type="checkbox"/> C | <code>a li:even-child ul .lst</code>     | <input type="checkbox"/> F | <code>ul.lst::nth-child(even) a</code>   |

Es. 3. (2 punti) Si consideri il seguente insieme di regole CSS incluse in un singolo foglio di stile importato nella pagina web il cui `<body>` è riportato di seguito. Qual è il colore finale dell'elemento `h1`?

```

1 .f { color: yellow; }
2 #g { color: blue; }
3 div h1 { color: pink !important; }
4 h1 .f { color: green !important; }
5 #cont h1.f { color: black; }
```

```

1 <body><div id="cont">
2   <h1 class="f" id="g" style="color: red;">
3     Hi!
4   </h1>
5 </div></body>
```

- |                            |       |                            |        |
|----------------------------|-------|----------------------------|--------|
| <input type="checkbox"/> A | Pink  | <input type="checkbox"/> D | Yellow |
| <input type="checkbox"/> B | Blue  | <input type="checkbox"/> E | Black  |
| <input type="checkbox"/> C | Green | <input type="checkbox"/> F | Red    |

Es. 4. (2 punti) Si consideri il seguente frammento di codice JavaScript (si assuma la presenza di `"use strict;"`).

```

1 console.log(x);
2 if (true) {
3   console.log(x);
4   let x;
5   console.log(x); }
```

Con riferimento al ciclo di vita di una variabile dichiarata con `let`, indicare quali delle seguenti affermazioni sono **vere**.

- |                            |  |                            |  |
|----------------------------|--|----------------------------|--|
| <input type="checkbox"/> A | In Linea 3, x è dichiarata ma non inizializzata.         | <input type="checkbox"/> D | Il codice esegue senza sollevare errori.         |
| <input type="checkbox"/> B | In Linea 3, x è inizializzata a <code>undefined</code> . | <input type="checkbox"/> E | In Linea 1, x è inizializzata ma non dichiarata. |
| <input type="checkbox"/> C | In Linea 5, x ha valore null.                            | <input type="checkbox"/> F | In Linea 5, x ha valore <code>undefined</code> . |

Es. 5. (2 punti) Quali dei seguenti **non** sono tipi di dato primitivi in JavaScript?

- |                            |         |                            |        |
|----------------------------|---------|----------------------------|--------|
| <input type="checkbox"/> A | array   | <input type="checkbox"/> D | number |
| <input type="checkbox"/> B | Promise | <input type="checkbox"/> E | float  |
| <input type="checkbox"/> C | boolean | <input type="checkbox"/> F | string |

Es. 6. (2 punti) Qual è il risultato dell'esecuzione del seguente frammento? Si assuma la presenza di `"use strict;"`.

```

1 try { console.log("A"); x = 1; console.log(x); }
2 catch(e) { console.log("X"); } finally { console.log("Z"); }
```

- |                            |   |                            |          |
|----------------------------|---|----------------------------|----------|
| <input type="checkbox"/> A | A, X, Z.  | <input type="checkbox"/> C | X, Z.    |
| <input type="checkbox"/> B | Il codice lancia un <code>ReferenceError</code> , senza altro output. | <input type="checkbox"/> D | A, 1, Z. |

Es. 7. (2 punti) Qual è il risultato dell'esecuzione del seguente frammento? Si assuma la presenza di "use strict;".

```

1 let prom = new Promise(function(resolve, reject) { setTimeout( () => { resolve("A"); }, 1000); });
2 prom.finally( value => console.log("B") ).then( val => console.log(val) )
3 .catch( value => console.log("X") );
4 console.log("Done");

```

- A A, B, Done.
- B Done, A, B.
- C X, B, Done.

- D Done, B, A.
- E Done, X.
- F B, A, Done.

Es. 8. (2 punti) Qual è l'output in console dopo il caricamento completo del documento HTML riportato di seguito?

```

1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <title>Web</title>
5     <meta charset="UTF-8" /></head>

```

```

6   <body>
7     <script>
8       let h = document.head?.children;
9       console.log(h[1]?.outerHTML);
10    </script></body></html>

```

- A undefined.
- B Viene lanciato un TypeError.
- C <title>Web</title>.

- D null.
- E Stringa contenente whitespace.
- F <meta charset="UTF-8" />.

Es. 9. (5 punti) Quali sono gli **svantaggi** (se ve ne sono) dell'utilizzo di un approccio *multi-threaded request handling* per un'applicazione web che effettua frequenti accessi in lettura/scrittura a un database? Motivare la risposta.

---



---



---



---



---

Es. 10. (5 punti) Si spieghino brevemente analogie e differenze tra gli attributi `async` e `defer` in un elemento `<script>`.

---



---



---



---



---

Es. 11. (5 punti) Quali sono le possibili origini di regole CSS nello step di analisi dei bucket di origine-importanza nell'algoritmo cascade di risoluzione dei conflitti?

---



---



---



---



---