Esercitazione 6 - 02/11

Ultima esercitazione prima del compitino: **esercizi di riepilogo** su tutti gli argomenti visti finora.

Argomenti:

- tipi, operatori, casting, `++`, valori truthy e falsy
- condizioni: `if`, operatori logici
- cicli: `for`, `while`, `do ... while`, cicli annidati
- funzioni: parametri, `return`, variabili locali e globali
- funzioni ricorsive

Mappa degli argomenti

Argomento	Dove l'abbiamo visto	Esercizi di oggi
Tipi, operatori, casting, `++`, operatori logici	Esercitazioni 1, 2, 3	1
Tracciare un ciclo	Esercitazioni 3, 4	2
Condizioni e operatori logici	Esercitazione 2	3
`for`, `do ... while`, accumulatori	Esercitazioni 2, 3	4
`while` con sentinella	Esercitazione 2	5
Funzioni: errori tipici	Esercitazione 4	6
Variabili locali e globali, scope	Esercitazioni 4, 5	7
Scrivere funzioni	Esercitazioni 3, 4	8
Ricorsione	Esercitazione 5	9
Cicli annidati	Esercitazioni 3, 5	8

Indice degli esercizi

Esercizio	Argomento
Esercizio 1	Quiz a risposta multipla
Esercizio 2	Ciclo while: cosa stampa
Esercizio 3	Anno bisestile: scrivere il programma
Esercizio 4	Multipli di 3: completare il programma
Esercizio 5	Numeri fino a zero: positivi e multipli di 2
Esercizio 6	Funzioni: trovare gli errori
Esercizio 7	Variabili locali e globali: cosa stampa
Esercizio 8	Scrivere funzioni: massimo, tavola pitagorica, multipli
Esercizio 9	Ricorsione: scrivere una funzione

Esercizio 1

Dieci domande di ripasso. Rispondete **senza eseguire** il codice.

1. Quale di queste righe provoca un errore?

- a) `let x = 1; x = 2;`
- b) `const y = 1; y = 2;`
- c) `let z; z = 3;`
- d) `const w = 4; console.log(w);`

2. Quale di questi confronti vale false?

- a) `2 == "2"`
- b) `3 >= 3`
- c) `"a" < "b"`
- d) `2 === "2"`

3. "Chiedi un dato finché non è valido, con almeno una richiesta". Quale ciclo è il più adatto?

- a) for

- b) while
- c) do ... while
- d) Nessuno dei tre

4. Perché `if (x = 5)` è sempre un errore?

- a) Perché `=` assegna, non confronta
- b) Perché 5 non è un booleano
- c) Perché x dovrebbe essere const
- d) Perché manca l'else

5. Qual è la differenza tra parametro e argomento di una funzione?

- a) Il parametro è il valore restituito
- b) L'argomento sta nella definizione
- c) Parametro nella definizione, argomento nella chiamata
- d) Il parametro è sempre un numero

6. Dove è visibile una variabile dichiarata con let nel corpo di un for?

- a) In tutto il programma
- b) Solo dentro quel blocco
- c) Solo dopo la fine del ciclo
- d) Solo dentro le altre funzioni

7. Una funzione ricorsiva non termina mai. Cosa manca quasi certamente?

- a) Un parametro
- b) Il console.log
- c) Il return
- d) Il caso base

8. Quale condizione verifica che n sia pari e positivo?

- a) `n % 2 == 0 && n > 0`
- b) `n % 2 && n > 0`
- c) `n % 2 == 0 || n > 0`
- d) `n / 2 == 0 && n > 0`

9. while (i < 5) { console.log(i); } con i = 0, mai modificata nel corpo. Cosa succede?

- a) Stampa 0 1 2 3 4
- b) Stampa 0 una volta

- c) Errore di sintassi
- d) Stampa 0 all'infinito

10. Quale espressione dà il numero 42 a partire dalla stringa "42"?

- a) `Number("42")`
- b) `String(42)`
- c) `typeof "42"`
- d) `"42" + 0`

Esercizio 2

Cosa stampa questo programma? Compilate una tabella con i valori di `a` e `b` dopo ogni giro per scoprirlo.

```
let a = 20;
let b = 1;
while (a > b) {
  a = a - b;
  b = b * 2;
  console.log(a, b);
}
console.log("fine", a, b);
```

A. Cosa stampa il programma?

B. Quante volte viene eseguito il corpo del ciclo?

Esercizio 3

Un anno è **bisestile** se è divisibile per 4, **tranne** quelli divisibili per 100, a meno che non siano divisibili anche per 400. Ad esempio il 2024 e il 2000 sono bisestili, il 1900 e il 2023 no.

Scrivete un programma che legge un anno dall'input (controllando che sia un numero intero positivo) e stampa se l'anno inserito è bisestile o meno.

Esercizio 4

Il programma legge un intero positivo `n` e stampa la **somma dei multipli di 3** tra 1 e `n` (ad esempio per `n = 10`: `3 + 6 + 9 = 18`).

A. Completate gli spazi.

```
let n;
do {
  n = Number(prompt("Inserire un intero positivo: "));
} while (_____________________________);

let somma = ____;
for (let i = ____; i <= n; i = i + ____) {
  somma = somma + i;
}
console.log("Somma dei multipli di 3 fino a", n, ":", somma);
```

B. Cosa stampa il programma con `n = 10`? E con `n = 2`?

C. Cosa succede se l'utente inserisce `0`, poi `-4`, poi `2.5`, poi `9`?

Esercizio 5

Scrivete un programma che legge numeri dall'utente **finché non viene inserito `0`**. Alla fine stampa:

- quanti numeri **positivi** e quanti **multipli di 2** sono stati inseriti.

Se il primo numero è `0`, il programma stampa `Nessun numero inserito`. Provatelo con `3, -2, 8, 0` e con `-5, -3, 0`.

Esercizio 6

Ciascuna di queste parti contiene un errore. Dite cosa stampa il codice e correggetelo.

A.

```
function area(base, altezza) {
  let a = base * altezza / 2;
}
let risultato = area(4, 5);
console.log("Area:", risultato);
```

B.

```
function doppio(x) {
  return x * 2;
  console.log("calcolo finito");
}
console.log(doppio(4));
```

C.

```
function media(a, b) {
  return (a + b) / 2;
}
console.log(media(4));
```

D. Il programma vuole incrementare `k` di 1.

```
function incrementa(n) {
  n = n + 1;
}
let k = 5;
incrementa(k);
console.log(k);
```

Esercizio 7

A. Cosa stampa questo programma? Tenete traccia del valore della variabile globale `x`.

```
let x = 10;

function f(x) {
  x = x + 1;
  return x * 2;
}

function g() {
  let y = x + 1;
  x = y;
  return y;
}

console.log(f(x));
console.log(x);
console.log(g());
console.log(x);
console.log(f(x));
```

B. Cosa succede eseguendo questo programma?

```
function conta() {
  let c = 0;
  c++;
}
conta();
console.log(c);
```

C. Cosa succede eseguendo questo programma?

```
for (let i = 0; i < 3; i++) {
  console.log(i);
}
console.log("fine:", i);
```

Esercizio 8

A. Scrivete la funzione `massimo3(a, b, c)` che **restituisce** il più grande dei tre numeri. Provatela con `(3, 9, 4)` e `(-1, -5, -3)`.

B. Scrivete la funzione `tavola(n)` che **stampa** la tavola pitagorica `n × n`: una riga per ogni numero da 1 a `n`, con i prodotti separati da uno spazio. Usate **due cicli annidati**. Con `tavola(4)` stampa:

```
1 2 3 4
2 4 6 8
3 6 9 12
4 8 12 16
```

C. Scrivete un programma che chiede un intero positivo `N` (ripetendo la domanda se non è valido) e stampa **su un'unica riga** tutti i multipli di 2 o di 3 fino a `N`. Ad esempio con `N = 20`:

```
2 3 4 6 8 9 10 12 14 15 16 18 20
```

Esercizio 9

Scrivete la seguente funzione **ricorsiva**. Prima individuate il **caso base** e il **passo ricorsivo**.

`potenza(base, esp)` che restituisce `base` elevato a `esp`, con `esp` intero maggiore o uguale a 0. Non usate `**` né `Math.pow`. Provatela con `potenza(2, 10)`, `potenza(3, 3)`, `potenza(5, 0)`.


Ludovica Pannitto - Università di Pisa - CC BY-NC-SA 4.0