



Laboratorio I

a.a. 2026/2027

TypeScript - controllo di flusso iterativo

Contenuti

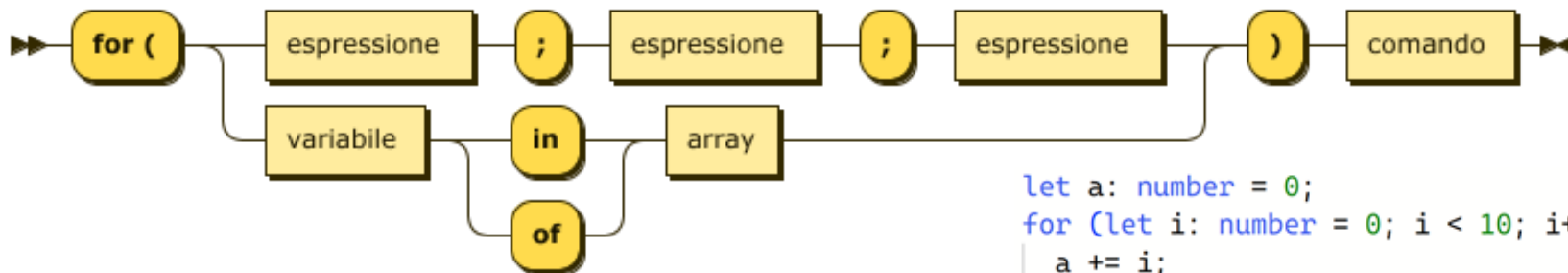
- Controllo del flusso
 - Comandi condizionali: `if`, `switch`
 - **Comandi iterativi:** `for`, `while` (... this lecture!)

Controllo del flusso

- Programma - serie di comandi da eseguire in ordine
- A volte serve cambiare il codice in base ai risultati dei calcoli, o in base all'input dell'utente
 - Comandi condizionali - una parte del codice viene eseguita solo se una condizione è soddisfatta
- **A volte serve ripetere la stessa operazione un numero di volte**
 - **Comandi iterativi**
 - Un numero fissate di volte - `for`
 - Un numero sconosciuto di volte - `while`



Comando iterativo - for



```
comando_for ::= 'for' '('  
    ( espressione ';' espressione ';' espressione  
    | variabile ( 'in' | 'of' ) array )  
    ')' comando
```

```
let a: number = 0;  
for (let i: number = 0; i < 10; i++) {  
    a += i;  
}
```

```
let studenti: string[] = ['Matt', 'Alice', 'Bob'];  
for (let i in studenti) {  
    console.log(studenti[i]);  
}
```

```
let studenti: string[] = ['Matt', 'Alice', 'Bob'];  
for (let s of studenti) {  
    console.log(s);  
}
```

- Permette di ripetere un comando o set di comandi un numero di volte
- Spesso usato quando si conosce il numero di iterazioni
- Esempio: somma dei primi n numeri naturali

Comando iterativo - while, do while



- Altro comando per ripetere istruzioni.
- Esegue *comando* fino a quando *espressione* diventa *false*.
- *espressione* di solito è Booleano
 - Ma può essere qualsiasi altro tipo (cast automatico come per **IF**). TS segnala errore se si confrontano tipi incompatibili!
- Di solito usato quando non si conosce il numero di iterazioni
- Esempio: sommare numeri in input finché l'utente inserisce -1

```
comando_while ::= while ( espressione ) comando
```



```
comando_do ::= do comando while ( espressione )
```



```
while(condizione){  
    //comandi  
}  
  
do{  
    //comandi  
} while (condizione);
```

Comandi iterativi

- A volte bisogna fermare una o tutte le iterazioni
- Comandi :
 - `break` - esce dal comando iterativo (`for`, `while`) fermandolo completamente e continua il programma
 - `continue` - salta l'iterazione corrente e passa alla prossima, continuando l'iterazione

```
for (let i: number = 0; i < 10; i++) {  
  if (i === 5) break;           // esce dal ciclo quando i vale 5  
  if (i % 2 === 0) continue;    // salta i numeri pari, passa al successivo  
  console.log(i);              // stampa solo 1, 3  
}
```

Misurare il tempo di esecuzione

- Comando `Date.now()` - tempo - *timestamp* in millisecondi
- Chiedere il tempo all'inizio e alla fine del programma

```
let start:number = Date.now();
```

```
// programma
```

```
let end:number = Date.now();
```

```
console.log(`Tempo di esecuzione ${end-start} ms`);
```

Q & A

Esercizi

(**Sommatoria**) Si scriva un programma che dato un numero naturale n calcoli e stampi la somma dei numeri fino a n .

(**Accumulatore**) Si scriva un programma che legge numeri finché la somma non supera 101, quindi stampi la somma.

(**Primo**) Si scriva un programma che legge un numero naturale n e valuti se è primo (un numero è primo se non esiste un numero d in $[2, n-1]$ tale che n sia divisibile per d). Si stampi il tempo di esecuzione del programma.

(**MultiPrimo**) Si modifichi il programma di sopra per leggere 10 numeri n e calcoli per ciascuno se si tratta di un numero primo.

Esercizi Extra

(**Fattoriale**) Si scriva un programma che dato un intero n calcoli e stampi il suo fattoriale. Si ricorda che il fattoriale di n è $n! = 1*2*...*(n-1)*n$

(**Asterischi**) Si scriva un programma che dato un intero n stampi n asterischi sulla prima linea, $n - 2$ asterischi sulla seconda linea, $n - 4$ sulla terza e così via, fino ad arrivare a stampare uno o due asterischi sull'ultima linea

- (**Stampa selettiva**) Si scriva un programma che legga da tastiera degli interi e stampi l'elemento letto se rispetta una delle seguenti proprietà: (i) è positivo e pari, oppure (ii) è negativo e preceduto (nell'ordine di inserimento) da un intero con valore maggiore o uguale. Terminare la l'acquisizione alla lettura di uno zero.

(**Media**) Si scriva un programma che legga da tastiera 10 interi e stampi la media aritmetica di tutti i valori diversi da zero e di segno uguale all'ultimo valore della sequenza.