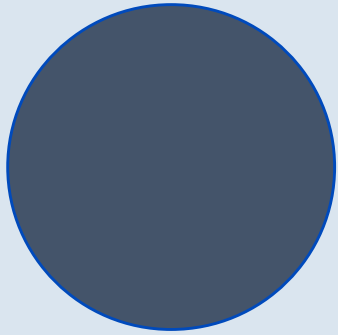


# **Introduzione ai Tipi di Dato Primitivi**



# Introduzione ai Tipi di Dato

## Primitivi in Java

- Java ha 8 tipi di dati primitivi: byte, short, int, long, float, double, boolean e char, ognuno con dimensioni e valori specifici.
- I tipi primitivi definiscono il range di valori che le variabili possono contenere e non richiedono oggetti wrapper come nelle classi.
- La tipizzazione forte di Java impone che ogni variabile sia dichiarata con un tipo di dato specifico, garantendo coerenza e robustezza nel codice.



# byte

- Dimensione: 1 byte; Intervallo di valori: da -128 a 127; Adatto per memorizzare piccoli numeri interi come età o dimensioni.
- Esempio di utilizzo: `byte age = 25`; Creazione di variabili possono contenere valori compresi tra -128 e 127.
- Nel contesto di Java, `byte` determina la dimensione e il range di valori che la variabile può rappresentare.



# short

- Dimensione: 2 byte; intervallo di valori da -32,768 a 32,767. Ogni variabile short può memorizzare numeri interi relativamente piccoli come l'esempio temperature = -10.
- I tipi di dati primitivi in Java, come short, definiscono le dimensioni e i valori che le variabili possono contenere. Utile per ottimizzare l'uso della memoria.
- La dichiarazione di variabili short consente un'efficiente gestione della memoria. Adatto per salvare dati numerici compresi nell'intervallo specificato.



# int

- Dimensione di 4 byte, con intervallo di valori compreso tra 2.147.483.648 e 2.147.483.647.
- Utilizzato per memorizzare numeri interi nell'esempio di codice: `int population = 1000000;`
- Essenziale per la programmazione in Java, definisce una variabile come un numero intero.



# long

- Dimensione: 8 byte; Intervallo di valori: da -9,223,372,036,854,775,808 a 9,223,372,036,854,775,807
- Esempio di utilizzo: long distance = 1500000000L;
- Il tipo di dato 'long' in Java viene utilizzato per memorizzare numeri interi molto grandi grazie alla sua ampia gamma di valori accettabili.



# float

- Dimensione di 4 byte, intervallo di valori con variabilità significativa. Esempio di utilizzo: `float price = 19.99f`.
- I tipi di dato primitivi in Java definiscono la dimensione tipo di valore che le variabili possono contenere.
- Java ha 8 tipi di dati primitivi, come `byte`, `short`, `int`, `long`, `double`, `boolean` e `char`, ognuno con caratteristiche specifiche



# double

- Dimensione: 8 byte; Intervallo di valori: variabilità significativa; Esempio di utilizzo: `double pi = 3.14159;`
- I tipi primitivi in Java definiscono dimensione e tipo di variabili. Il `double` rappresenta numeri in virgola mobile a doppia precisione.
- Essendo un tipo di dato più preciso rispetto al `float`, il `double` è comunemente usato per calcoli matematici che richiedono alta precisione numerica.





# boolean

- I valori booleani in Java possono essere true o false e vengono utilizzati per rappresentare lo stato di verità di una condizione nel codice.
- Nell'esempio fornito, isJavaFun è una variabile booleana che è stata dichiarata come true, indicando che Java è divertente.
- In Java, il tipo di dato boolean è essenziale per le espressioni condizionali e permette di controllare il flusso di esecuzione del programma.



# char

- Dimensione: 2 byte; Intervallo di valori: caratteri Unicode  
Esempio di utilizzo: `char grade = 'A';`
- Il tipo di dato `char` in Java rappresenta un singolo carattere Unicode, occupando 16 bit di memoria e consentendo la memorizzazione di simboli, lettere e numeri.
- Essendo un tipo di dato primitivo, il `char` può essere dichiarato e inizializzato con un singolo carattere tra apici singoli, come `'A'`.



# Riepilogo sui Tipi di Dato Primitivi

- I tipi di dato primitivi in Java come byte, short, int offrono precisione e controllo sulla dimensione dei valori memorizzati essenziali per ottimizzare lo spazio di memoria.
- La corretta scelta del tipo di dato primitivo aiuta a garantire efficienza e affidabilità nel codice, bilanciando le esigenze del programma con la natura dei dati.
- Conoscere i 8 tipi di dato primitivi di Java è fondamentale per gli sviluppatori, permettendo la creazione di codice efficiente e performante.

