

Titolo unità didattica: Concetto di Variabile

[02]

Titolo modulo : Operazione di assegnazione

[02-T]

Operazioni di base sulle variabili in programmazione

Argomenti trattati:

- ✓ operazione di assegnazione di un valore a una variabile
- ✓ valutazione delle espressioni

Prerequisiti richiesti: AP-02-01-T

l'operazione che consente di **associare un valore a una variabile** è l'operazione fondamentale in un algoritmo

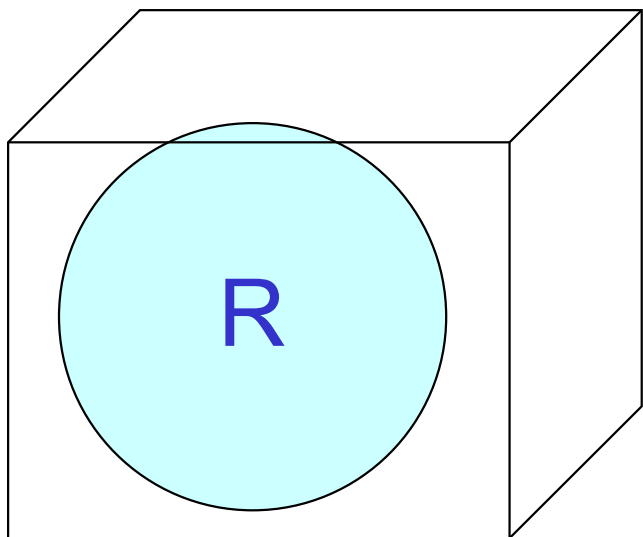
due diverse istruzioni:



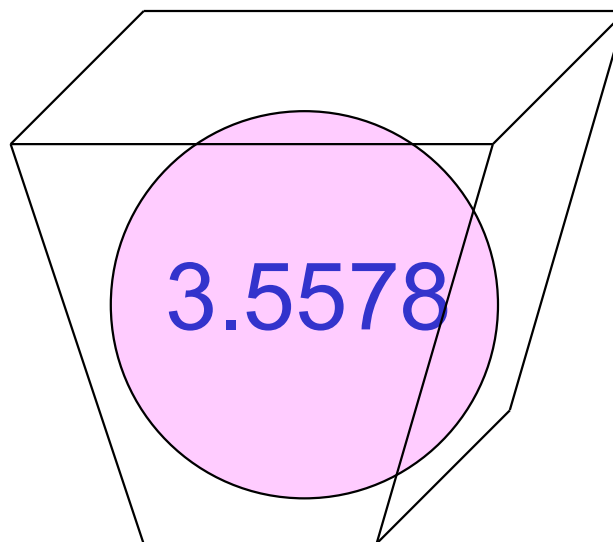
istruzione di **assegnazione**



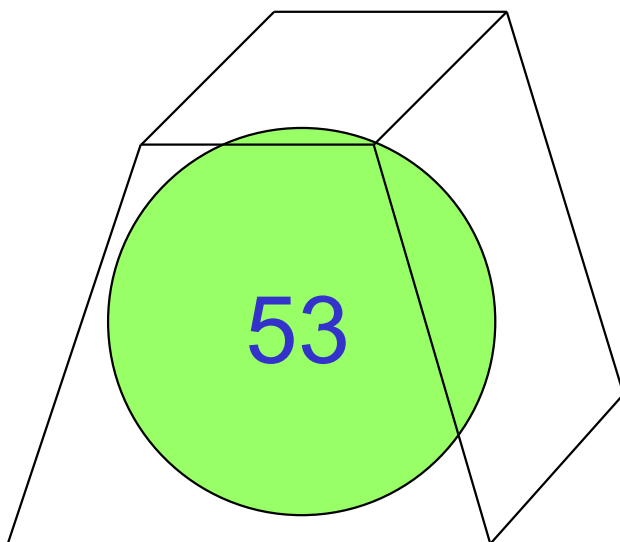
istruzione di **lettura da dispositivo esterno di input**



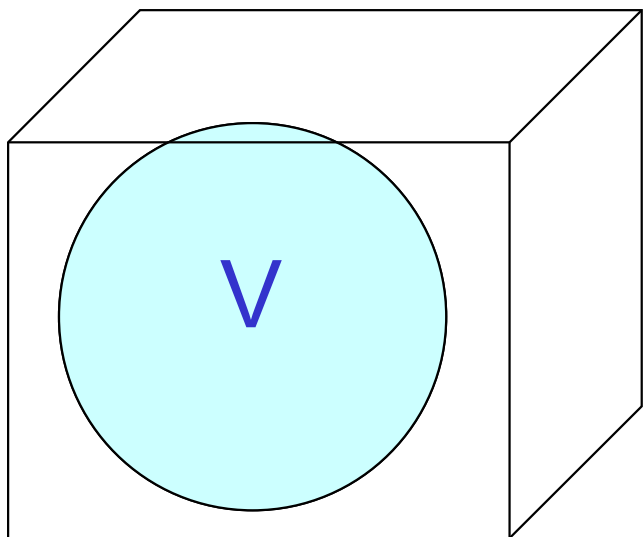
iniz_cognome



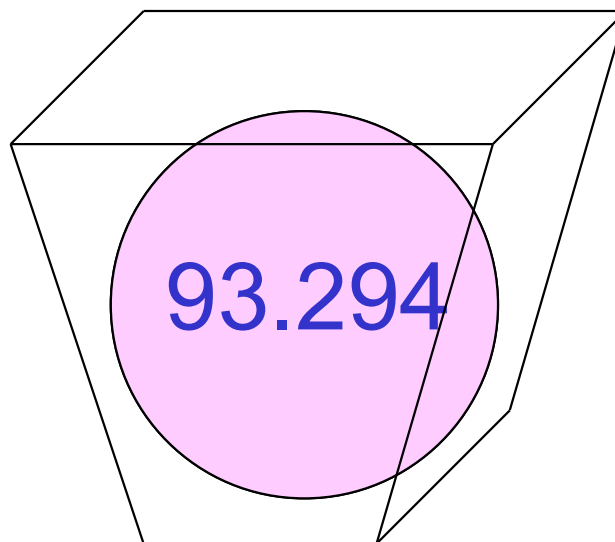
lato



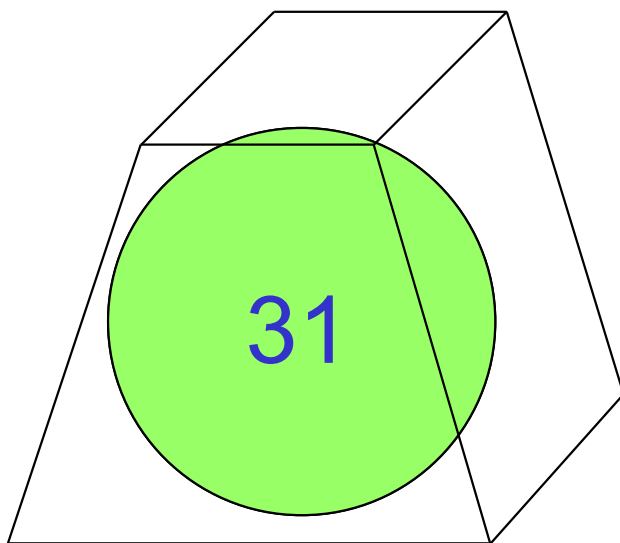
eta_anni



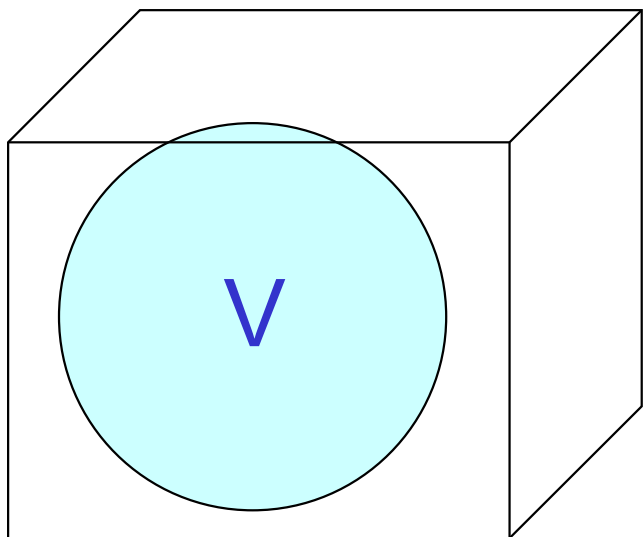
iniz_cognome



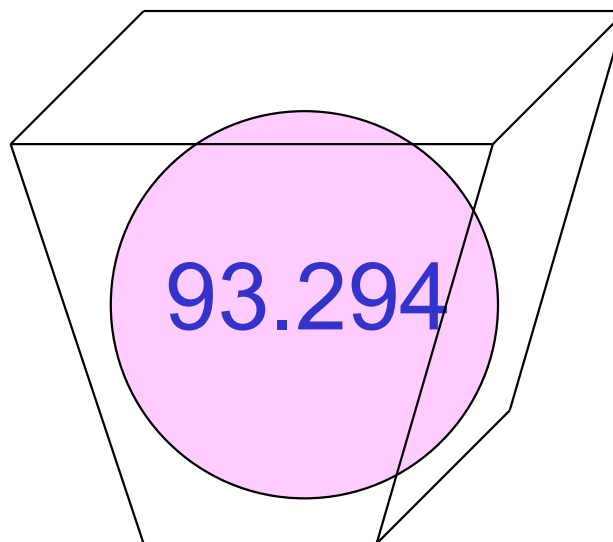
lato



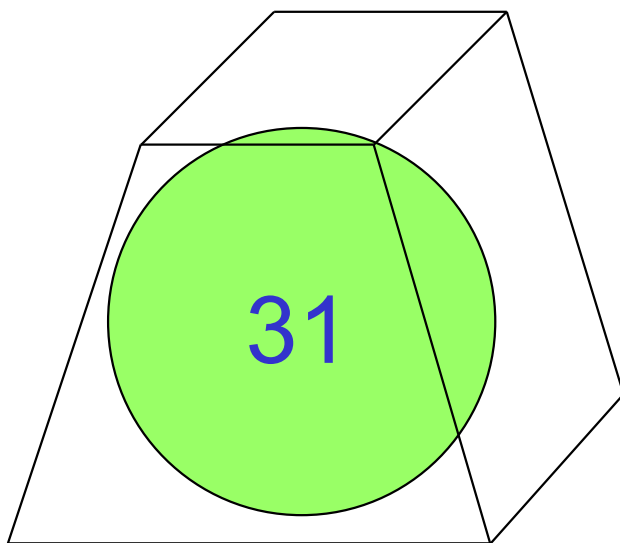
eta_anni



iniz_cognome



lato



eta_anni

istruzione di **assegnazione**:

l'**assegnazione** deve specificare

- ✓ la **variabile** (destinazione) a cui deve essere assegnato il valore
 - ✓ il **valore** da assegnare
- oppure**
- ✓ una espressione, la cui valutazione fornisce il valore da assegnare

<variabile> = <valore> ;

<variabile> = <espressione> ;

istruzione di **assegnazione**:

Esempio:

```
lato = 3.12 ;  
eta_anni = 27 ;
```

- ✓ il valore **3.12** è associato alla variabile **lato**
- ✓ il valore **27** è associato alla variabile **eta_anni**

<valore> deve essere un dato del tipo di cui è stata dichiarata la **<variabile>**

istruzione di **assegnazione**:

Esempio:

lato = 4.23 + 2.1 ;

eta_anni = (27 - 7)/2 ;

- ✓ viene valutato il valore dell'espressione **4.23 + 2.1**, che è **6.33**
- ✓ il valore **6.33** è associato alla variabile **lato**
- ✓ viene valutato il valore dell'espressione **(27 - 7)/2**, che è **10**
- ✓ il valore **10** è associato alla variabile **eta_anni**

istruzione di **assegnazione**:

Esempio:

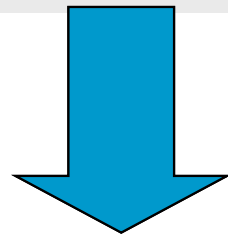
```
float area_rett, base, altezza;  
    base = 4.0 ;  
    altezza = 3.1 ;  
    area_rett = base * altezza ;
```

- ✓ il valore **4.0** è associato alla variabile **base**
- ✓ il valore **3.1** è associato alla variabile **altezza**
- ✓ viene valutato il valore dell'espressione
 (base * altezza), che è **12.4**
- ✓ il valore **12.4** è associato alla variabile
 area_rett

una **espressione** è una combinazione di variabili e dati connessi da operatori

nella **valutazione** di una espressione si utilizzano i valori delle variabili e i dati

una espressione deve essere **corretta**



- deve coinvolgere variabili, dati e operatori rispettando la congruenza di tipo
- non deve contenere variabili **indeterminate** o **indefinite**
- deve avere un **unico** valore

Esempio:

```
int alfa,beta,gamma;  
    alfa = 4 ;  
    beta  = 9 ;  
    gamma = 3*(alfa+5)-beta ;
```

- dopo la prima assegnazione il valore associato a **alfa** è **4**
- dopo la seconda quello associato a **beta** è **9**
- viene valutato il valore dell'espressione **$3*(alfa+5)-beta$** , che è **18**
- dopo la terza assegnazione il valore associato a **gamma** è **18**

Esempio:

```
int i ;  
    i = 1 ;  
    i = i+1 ;
```

- **dopo** la prima assegnazione il valore associato a *i* è **1**
- viene valutato il valore dell'espressione *i* + **1**, che è **2**
- **dopo** la seconda assegnazione il valore associato a *i* è **2**

Esempio:

```
float alfa,beta;
```

```
logical delta;
```

```
    alfa = 4. ;
```

```
    beta = 9.1 ;
```

```
    delta = alfa < beta ;
```

- dopo la prima assegnazione a **alfa** è associato **4**
- dopo la seconda a **beta** è associato **9.1**
- viene valutato il valore dell'espressione **alfa < beta**, che è **true**
- dopo la terza a **delta** è associato **true**

Esempio:

```
int alfa,beta,gamma;
```

```
    alfa = 4 ;
```

```
    beta = 9 ;
```

```
    beta = 3*(alfa+5) - gamma ;
```

- dopo la prima assegnazione a **alfa** è associato **4**
- dopo la seconda a **beta** è associato **9**
- viene valutato il valore dell'espressione
 $3*(alfa+5) - gamma$, che è **indefinito**
poiché **gamma** è **indefinita**
- dopo la terza **beta** è **indefinita**,

Esempio:

```
int alfa,beta;
```

```
    alfa = 4 ;
```

```
    beta = 9 ;
```

```
    alfa+beta = beta ;
```

- dopo la prima assegnazione a **alfa** è associato **4**
- dopo la seconda a **beta** è associato **9**
- viene valutato il valore dell'espressione **beta** che è **9**
- ma - **ERRORE** - un dato non può essere assegnato a una espressione (**alfa+beta**)

dichiarazione di costanti simboliche

```
const <tipo> <costante> = <valore> ;
```

```
const float pi_greco = 3.1415926;
```

```
const float e_neper = 2.7182818;
```

sono indicate **8** cifre significative, cioè il massimo consentito dalla **singola precisione** del **sistema aritmetico floating point** di un pc

valori di tipo **reale**

una sequenza di cifre decimali precedute eventualmente dal segno, e contenente il punto decimale (.) oppure in notazione esponenziale contenente la **E**.

Esempio:

3.

-0.417

5**E**9

+298976543212**E**-26

-3.4**E**2

ATTENZIONE: **E**9 significa 10^9 , **E-26** significa 10^{-26} ,
E2 significa 10^2