*Visita →* *https://it.wikibooks.org/wiki/OpenSCAD/Primi_passi*

# DISEGNO SOLIDI PRIMITIVI

In OpenSCAD esistono tre funzioni per creare solidi elementari

| | | |
|---|---|---|
| **cube(10);** | cube([2,3,4]); | rispettivamente cubo e parallelepipedo |
| **cylinder(10,6,6);** | cylinder(10,6,2) | cilindro/prisma retto e tronco di cono/piramide |
| **sphere(5);** | | |

Apri il software e prova i seguenti script (**premi F5** per vedere il modello 3D)

```
cube(10);                          // cubo di lato 10
color ("red") cube(10, true);      // aggiunta del colore rosso al solido
cube([2,5,9]);
```

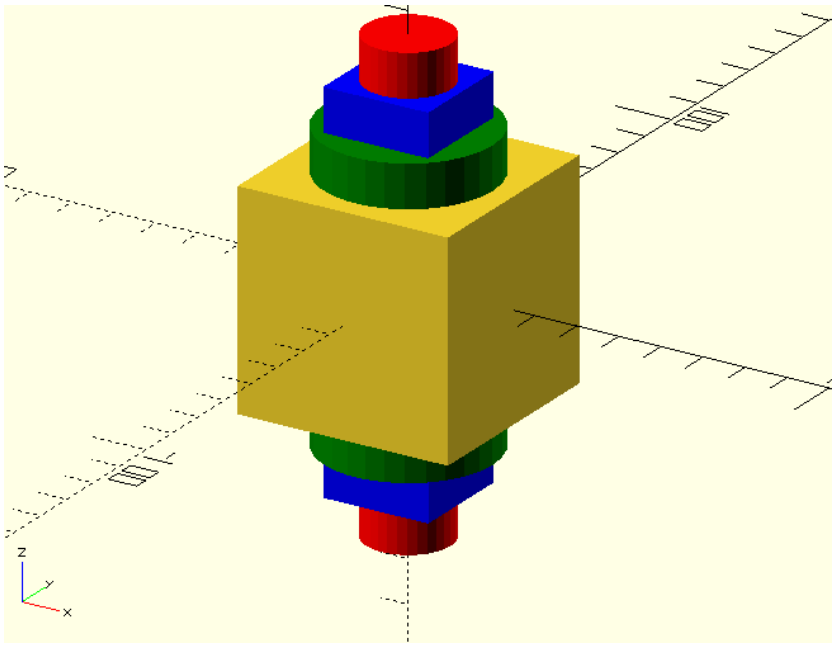
Se vuoi visualizzare un solo solido anteponi ! (il punto esclamativo) al solido scelto.
Per colorare gli oggetti qui puoi trovare i possibili colori
https://en.wikibooks.org/wiki/OpenSCAD_User_Manual/Transformations#color

Cancella lo script precedente ed inserisci e visualizza il seguente.

```
cylinder(20,5,5);
cylinder(20,10,2,$fn=5,true);
```

**DOMANDA.1**
A cosa serve il parametro true aggiunto all'interno delle funzioni cube e cylinder?
E il parametro $fn?

**ESERCIZIO.1**
Scrivi lo script per realizzare il seguente



# POSIZIONAMENTO NELLO SPAZIO

In questa sezione vedremo come posizione i solidi nello spazio tridimensionale.
translate ([x,y,z]); posiziona l'oggetto nel punto x,y,z
rotate([rotx,roty,rotz]); ruota l'oggetto rispetto agli assi

La rotazione rispetto ad un asse, per esempio Z, avviene come se Z (asse assoluto non relativo all'oggetto) funzionasse da perno fisso e il piano perpendicolare XY ruotasse in senso antiorario del valore indicato.

Apri il software e prova i seguenti script (**premi F5** per vedere il modello 3D)

```
translate ([-20,0,0]) cube(10);          // cubo posto sull'asse -x
translate ([10,0,0]) cube(10);           // cubo posto sull'asse +x
translate ([0,-10,0]) sphere (5);        // sfera posta sull'asse -y
translate ([0,10,0]) sphere (5,$fn=50);  // sfera posta sull'asse +y con risoluzione maggiore
```

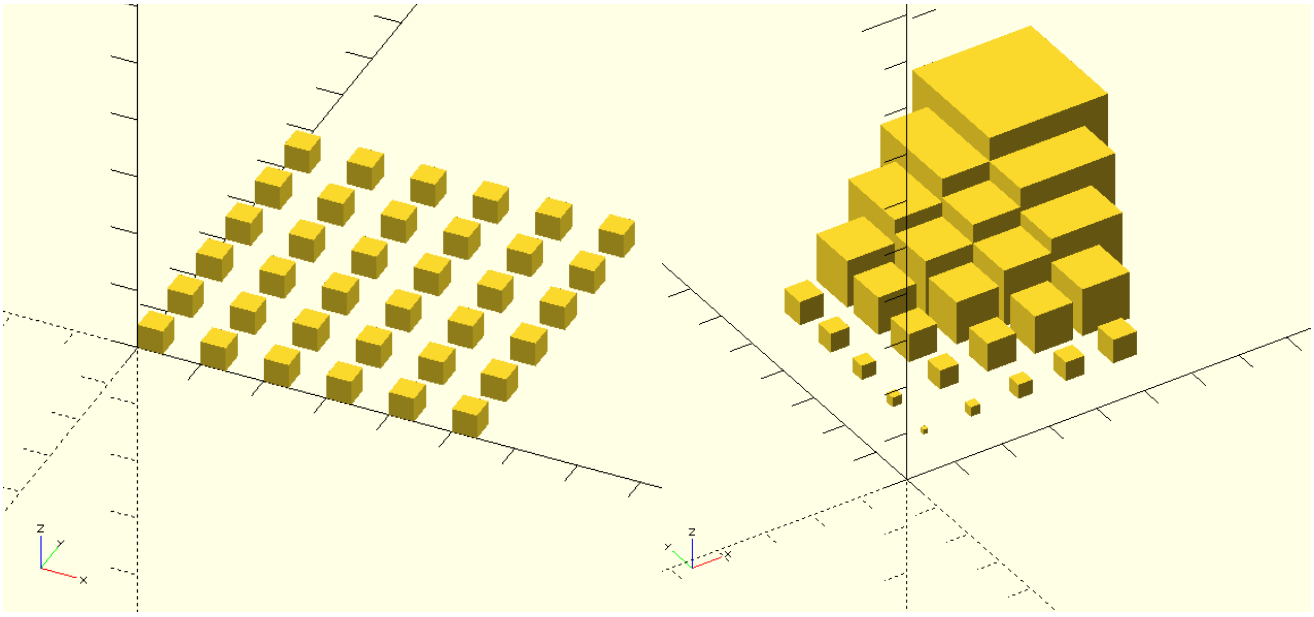
**DOMANDA.2**
*Perchè il primo cubo è traslato di -20 ed il secondo di +10 (e non di +20)?*
*Qual è il punto di riferimento per la traslazione del cubo?*
*E il punto di riferimento per la traslazione della sfera?*

*SERIE DI CUBI*

Si possono facilmente creare serie di solidi utilizzando cicli for...

```
for (i=[0:5]){          //[start : increment : end] se l'incremento non è indicato vale 1
        translate ([i*10,0,0]) cube(4);      // serie di cinque cubi posti a x=+0,10,20,30,40,50
}
```

**ESERCIZIO.2**
Scrivi lo script per realizzare i seguenti gruppi di solidi. Utilizza lo script di cui sopra.



*Visita →* *https://it.wikibooks.org/wiki/OpenSCAD/Primi_passi*

Apri il software e prova i seguenti script (**premi F5** per vedere il modello 3D)

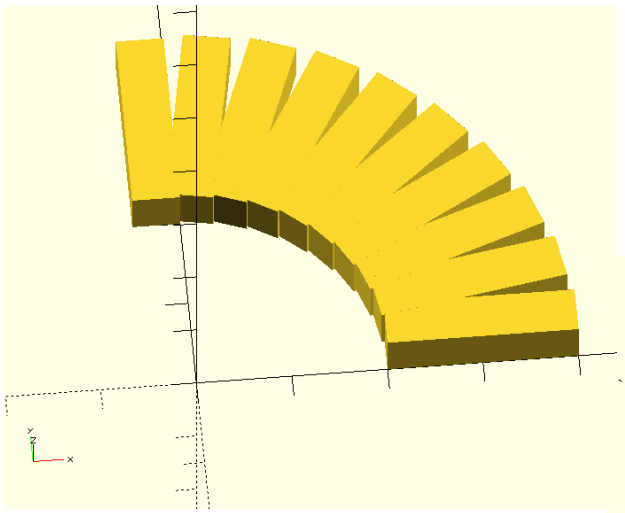
```
cylinder(30,8,8,$fn=6);                    //disegna un prisma a base esagonale
rotate([0,0,45]) cylinder(30,8,8,$fn=6);   //lo ruotare rispetto a Z di 45gradi antiorario

translate([20,0,0]) cube([10,5,5]);                    //trasla il solido
rotate([0,0,45]) translate([20,0,0]) cube([10,5,5]);   //e poi lo ruota rispetto a Z

translate([20,0,0]) rotate([0,0,45]) cube([10,5,5]);   //e qui?
```

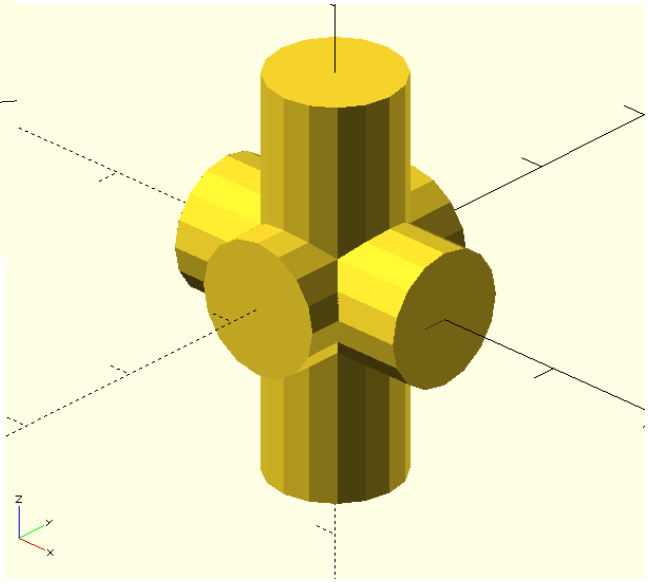
**DOMANDA.3**

Che differenza c'è tra le ultime due righe del precedente script?

**ESERCIZIO.3**

Scrivi lo script per realizzare i seguenti gruppi di solidi. Aiutati utilizzando gli script finora visti.





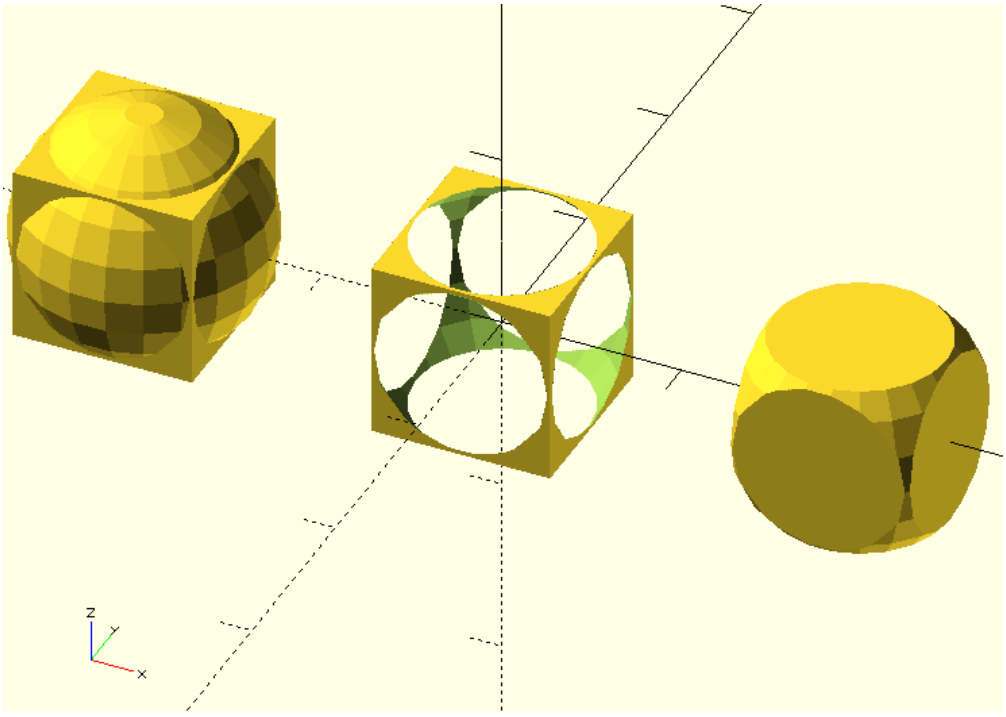
# OPERAZIONI BOOLEANE SUI SOLIDI

Nella modellazione CGS le forme solide si ottengono mediante operazioni booleane di:
UNIONE, DIFFERENZA, INTERSEZIONE.
Le funzioni di OpenSCA che eseguono questi comandi sono le seguenti:
**union(), difference(), intersection()**

Apri il software e prova i seguenti script (**premi F5** per vedere il modello 3D)

```
translate([-20,0,0])
union(){                        //unione di due solidi (somma dei volumi)
        cube (10,true);
        sphere (7);
}

difference(){                   //differenza di volumi tra un cubo e una sfera
        cube(10, center=true);
        sphere(7);              // aggiungendo l'hashtag la sfera diventerebbe visibile
}

translate([20,0,0])
intersection(){                 //intersezione (volume in comune) tra due solidi
        cube (10,true);
        sphere (7);
}
```

*Per prova aggiungi # (hashtag) al comando sphere(7) → #sphere(7)*

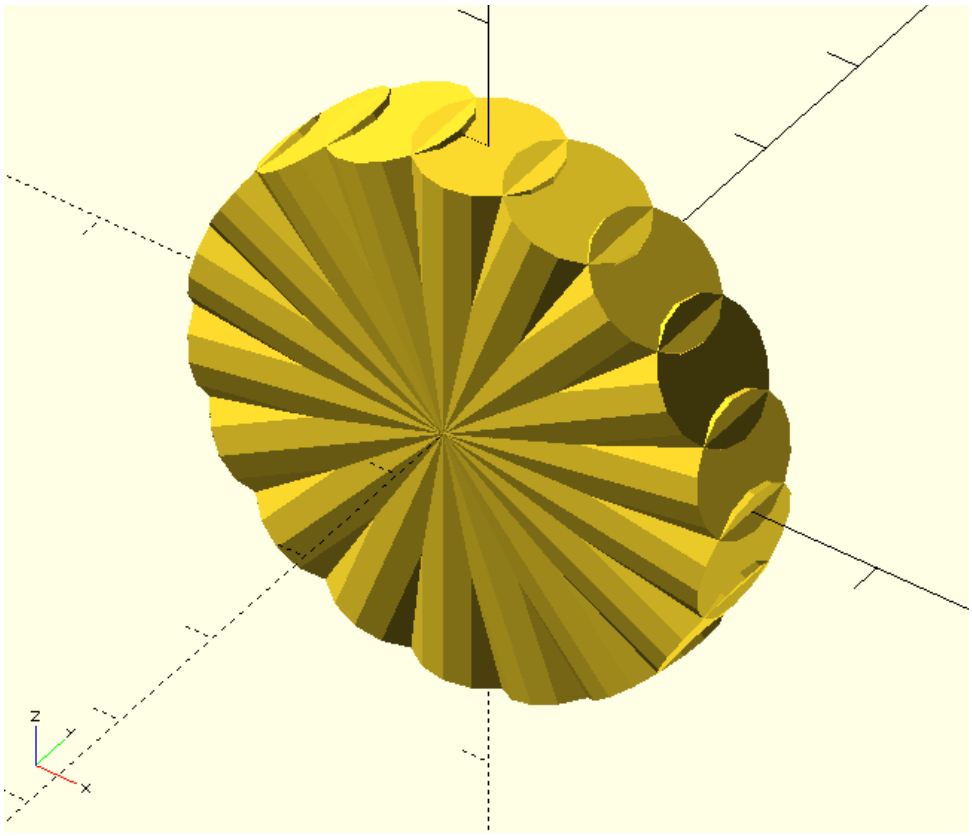


# MODULI

I moduli possono essere utilizzati per costruire procedure complesse da aggiungere alle istruzioni proprie del linguaggio. In questo modo anziché ripetere una stessa sequenza di comandi la si sostituisce con una istruzione personalizzata che entra a far parte del linguaggio nell'ambito del file .scad in esecuzione.
I moduli accettano anche parametri in ingresso. Non restituiscono valori in uscita, per questo serve utilizzare function().

```
module rotcy(rot, r, h) {                    //modulo per la creazione di cilindri ruotati
   rotate(rot)                               // non viene eseguito se non richiamato
     cylinder(h ,r ,r ,true);
}

                  //parte principale dello in cui si richiama il modulo
rotcy([0,45,0],10,30);                       //l'istruzione ruota un cilindro di raggio 10, alto 30
```

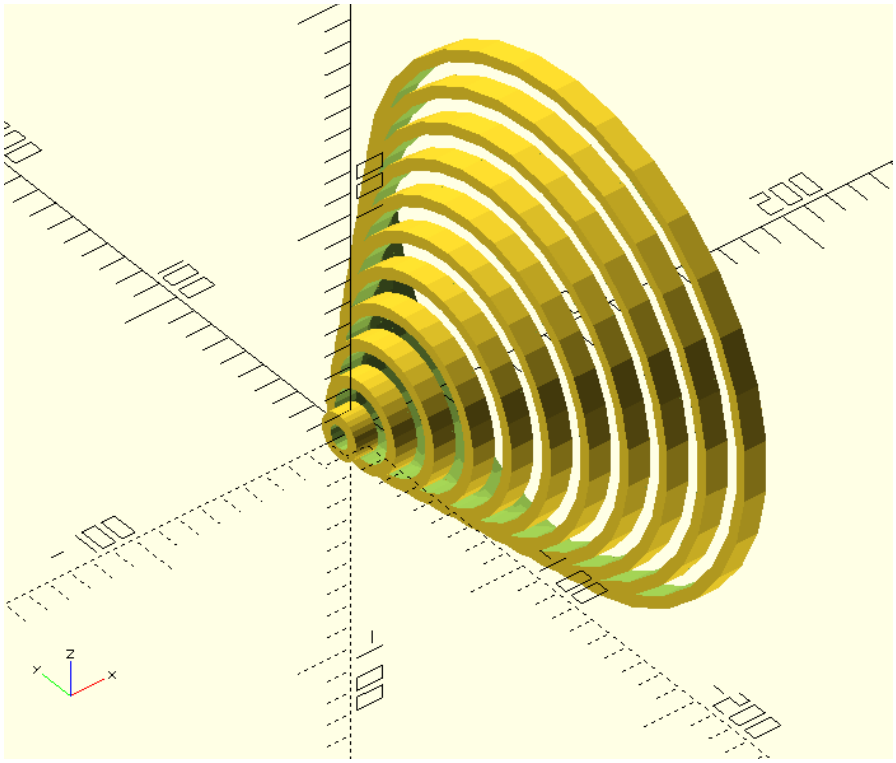
**ESERCIZIO.4**
Modifica lo script precedente in modo che realizzi il seguente gruppo di solidi. Aiutati utilizzando gli script finora visti.



## ESERCIZIO.5

Modifica lo script seguente in modo che realizzi il gruppo di solidi mostrato.

```
module anello(r) {                    //questo modulo serve a realizzare un anello di raggio r
        difference(){
                cylinder(10,r,r,true);
                cylinder(10+2,r-5,r-5,true);
        }
}
                        //l'istruzione eseguente la modellazione di un anello di raggio 10
translate([10,0,0]) rotate([0,90,0]) anello (10);
```



**DOMANDA.4**

Si vuole realizzare un porta schedine SD (30mm,25mm,2mm) di dimensione variabile in base alla richiesta.
Copia lo script e visualizza il modello. Modifica i valori di numeroSD.
Rispondi alle domande riportate più sotto.

```
numeroSD=10;                        //crea una variabile per modificare velocemente

module portaSD(num){
        for (i=[0:num-1]){
        translate([i*3,0,0])        //distanzia le schedine di 3mm l'una dall'altra
        cube([2,25,30],true);       //schedina SD
        }
}

                        //parte MAIN dello script
difference(){                       //differenza tra il box e le schedine SD
        translate ([0,0,-5])        //la dimensione del box dipende dal numero di SD
                cube([numeroSD*3+3,30,25],true);
        translate([(-numeroSD*3+3)/2,0,0])
                #portaSD(numeroSD);
}
```

*Il translate ([0,0,-5]) a cosa serve? E translate([(-numeroSD*3+3)/2,0,0])?*
*Puoi trovare una soluzione alternativa? (fai coincidere lo spigolo del box con l'asse Z)*
*A cosa serve il modificatore #?*