

### Lienzo

Establecer las dimensiones del lienzo con la instrucción `size()` y el parámetro **P3D** para trabajar en 3D: `size(x, y, P3D)`

### Propiedades del lienzo

Establecer color de fondo, un color para la línea y un color de relleno con:

`background(color)`  
`stroke(color)`  
`fill(color)`

### Definir lugar de dibujo

Establecer el punto para dibujar, inicialmente el punto de dibujo se encuentra en posición **(0,0,0)**, esquina superior izquierda del lienzo. Con la instrucción:

`translate(x, y, z)`  
`translate(x, y)`

Trasladamos el origen de coordenadas al punto indicado.

### Vista del dibujo

Para tener una vista en perspectiva de la figura, imponemos una rotación en sus ejes como:

`rotateX(angulo)`  
`rotateY(angulo)`  
`rotateZ(angulo)`

### Dibujar 3D funciones box and sphere

Hay instrucciones que permiten dibujar ya en 3D en processing, como:

`box(x)` -> Dibuja un cubo de longitud de sus lados **x**  
`sphere(x)` -> Dibuja una esfera con radio **x**

### Dibujar figuras 3D personalizadas

Las figuras personalizadas se dibujan mediante:

`beginShape()`  
`endShape()`  
`vertex(x,y,z)`

Colocando varios poligonos uno al lado del otro, esto se realiza haciendo polígono por polígono o con un ciclo que nos permita realizar esto.

### Paso 1

Definir el numero de lados de la base en una variable, las el ángulo en que se recorrerá, la altura y el radio:

```
int sides = 10;
float angle = 360 / sides;
float halfHeight = height - t / 2;
float r = 100;
```

### Paso 2

Dibujar bases de la figura usando ciclos y las funciones **sin** y **cos**. Estos pasos se repiten para dibujar la base inferior y superior con la diferencia de la variable `halfHeight`, donde una es positiva y la otra negativa, o se puede omitir una en caso de no ocuparla:

```
beginShape();
for (int i = 0; i < sides; i++) {
    float x = cos( radians( i * angle ) ) * r;
    float y = sin( radians( i * angle ) ) * r;
    vertex ( x , y , (-)halfHeight);
}
```

### Paso 2 (cont)

```
> }
endShape(CLOSE);
```

### Paso 3

Dibujar contorno de la figura, para realizar esto es lo mismo de la anterior pero a diferencia se debe

especificar **QUAD\_STRIP** en el `beginShape` para

dibujar rectángulos, y para `vertex` se debe definir tanto negativo como positivo.

```
beginShape(QUAD_STRIP);
for (int i = 0; i < sides + 1; i++) {
    float x = cos( radians( i * angle ) ) * r;
    float y = sin( radians( i * angle ) ) * r;
    vertex ( x , y , halfHeight);
    vertex ( x , y , -halfHeight);
}
endShape(CLOSE);
```

**Nota:** Todos los parámetros de cada ciclo pueden cambiar dependiendo de la figura deseada, debido a que no todas presentan las mismas características.