

DADO EN HTML y CSS:

Este es el html que hace el dado y sus caras:

```
<h1>Juego de dados</h1>
<div class="container">
  <div class="cara-1">
    <div class="punto"></div>
  </div>
  <div class="cara-2">
    <div class="punto"></div>
    <div class="punto"></div>
  </div>
  <div class="cara-3">
    <div class="punto"></div>
    <div class="punto"></div>
    <div class="punto"></div>
  </div>
  <div class="cara-4">
    <div class="columna">
      <div class="punto"></div>
      <div class="punto"></div>
    </div>
    <div class="columna">
      <div class="punto"></div>
      <div class="punto"></div>
    </div>
  </div>
  <div class="cara-5">
    <div class="columna">
      <div class="punto"></div>
      <div class="punto"></div>
    </div>
    <div class="punto medio"></div>
  </div>
</div>
```

```
        <div class="columna">
          <div class="punto"></div>
          <div class="punto"></div>
        </div>
      </div>
    <div class="cara-6">
      <div class="columna">
        <div class="punto"></div>
        <div class="punto"></div>
        <div class="punto"></div>
      </div>
      <div class="columna">
        <div class="punto"></div>
        <div class="punto"></div>
        <div class="punto"></div>
      </div>
    </div>
  </div>
```

El css sería el siguiente:

```
* {
  box-sizing:inherit;
}

body {
  padding: 0;
  font-family: sans-serif;
}

h1 {
  text-align: center;
}

.container {
  display: flex;
  flex-wrap: wrap;
  justify-content: center;
```

```
}
```

```
[class^="cara-"] {  
  background-color: #555;  
  display: flex;  
  width: 120px;  
  height: 120px;  
  padding: 10px;  
  margin: 1em;  
}
```

```
.cara-1 {  
  justify-content: center;  
  align-items: center;  
}
```

```
.cara-2 {  
  justify-content: space-between;  
}
```

```
.cara-2 .punto:nth-child(2) {  
  align-self: flex-end;  
}
```

```
.cara-3 {  
  flex-direction: column;  
  justify-content: space-between;  
}
```

```
.cara-3 .punto:nth-child(2) {  
  align-self: center;  
}
```

```
.cara-3 .punto:nth-child(3) {  
  align-self: flex-end;
```

```
}
```

```
.cara-4 {  
  justify-content: space-between;  
}
```

```
.cara-4 .columna {  
  display: flex;  
  flex-direction: column;  
  justify-content: space-between;  
}
```

```
.cara-5 {  
  justify-content: space-between;  
}
```

```
.cara-5 .columna {  
  display: flex;  
  flex-direction: column;  
  justify-content: space-between;  
}
```

```
.cara-5 .medio {  
  align-self: center;  
}
```

```
.cara-6 {  
  justify-content: space-between;  
}
```

```
.cara-6 .columna {  
  display: flex;  
  flex-direction: column;  
  justify-content: space-between;
```

```
}
```

```
.cara-6 {
```

```
}
```

```
.punto {
```

```
width: 20px;
```

```
height: 20px;
```

```
background-color: #eee;
```

```
border-radius: 50%;
```

```
}
```

Cara 1:

- Representa un dado con un solo punto en el centro.
- Se define mediante `<div class="cara-1">`.
- Contiene un único `<div class="punto"></div>`.

2. Cara 2:

- Representa un dado con dos puntos.
- Se define mediante `<div class="cara-2">`.
- Contiene dos `<div class="punto"></div>`.

3. Cara 3:

- Representa un dado con tres puntos.
- Se define mediante `<div class="cara-3">`.
- Contiene tres `<div class="punto"></div>`.

4. Cara 4:

- Representa un dado con cuatro puntos dispuestos en dos columnas.
- Se define mediante `<div class="cara-4">`.
- Cada columna contiene dos puntos: `<div class="columna">`.

5. Cara 5:

- Representa un dado con cinco puntos.
- Se define mediante `<div class="cara-5">`.
- Tiene una columna con dos puntos, un punto en el centro (`<div class="punto medio"></div>`), y otra columna con dos puntos.

6. Cara 6:

- Representa un dado con seis puntos dispuestos en dos columnas.
- Se define mediante `<div class="cara-6">`.
- Cada columna contiene tres puntos: `<div class="columna">`.

1. `* { box-sizing: inherit; }`

- Esta regla establece que todos los elementos (*) heredarán el modelo de caja (box-sizing) de sus elementos padres. El valor `inherit` indica que se utilizará el modelo de caja definido en el elemento padre.

2. `body { padding: 0; font-family: sans-serif; }`

- Estas reglas aplican al elemento `body`.
- Elimina el relleno (padding) predeterminado del cuerpo del documento.
- Define la fuente (font-family) como una fuente sans-serif (sin remates).

3. `h1 { text-align: center; }`

- Aplica al elemento `h1`.
- Centra el texto horizontalmente utilizando `text-align: center`.

4. `.container { display: flex; flex-wrap: wrap; justify-content: center; }`

- Aplica a un elemento con la clase `.container`.
- Crea un contenedor flexible (`display: flex`) que envuelve los elementos internos.
- Permite que los elementos internos se envuelvan en varias filas (`flex-wrap: wrap`).
- Centra los elementos horizontalmente dentro del contenedor (`justify-content: center`).

5. `[class^="cara-"] { ... }`

- Aplica a elementos cuyas clases comienzan con "cara-" (por ejemplo, `.cara-1`, `.cara-2`, etc.).
- Establece un fondo gris oscuro (`background-color: #555`).
- Define un tamaño fijo para las caras del dado (`width: 120px; height: 120px`).
- Agrega un relleno y margen alrededor de las caras.

6. Reglas específicas para cada cara (`.cara-1`, `.cara-2`, etc.):

- Alinea los puntos en diferentes posiciones según la cantidad de puntos en cada cara.
- Utiliza flexbox para organizar los puntos en columnas o filas.

7. `.punto { ... }`

- Aplica a elementos con la clase `.punto`.
- Define un tamaño para los puntos (`width: 20px; height: 20px`).
- Establece un color de fondo gris claro (`background-color: #eee`).
- Da forma circular a los puntos con bordes redondeados (`border-radius: 50%`).