

ANOTACIONES JAVASCRIPT:

DECLARAR VARIABLES:

- **"Const"** no me permite modificarlo en ningún momento, pero sí puedo modificar los atributos declarados en él.
- **"Let"** si permite ser modificado en ambos casos.
- **"Var"** es global (puede dar problemas, mejor no usarla). Funciona como "Let" pero está disponible en cualquier momento de la función en la que está "var".

DIFERENCIA ENTRE "FUNCTION" Y FUNCIÓN "FLECHA":

- **"Function"** establece **"this."** al objeto cuando se utiliza.
- En las funciones "flecha" eso no ocurre.

TIPOS DE INCREMENTOS:

- **+= "variable"** significa que incremento algo en "variable" unidades.
- **++** significa que incremento algo en 1 unidad.
- **--** significa que decremento algo en 1 unidad.

DEBUGGER:

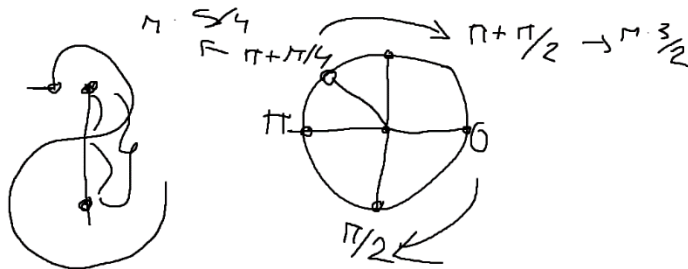
Si hay algún problema en el código, puedo usar la línea de código **"debugger;"** para que, cuando abra el HTML en Google pueda avanzar paso a paso lo que ocurre en el programa.

FOREACH:

Si queremos aplicar algo a cada elemento de una lista o "array", podemos usar el bucle **"forEach"**. En él, declaramos el elemento de array, su índice y el array.

DETALLES PARA DIBUJAR EN CANVAS:

- En la función **"arc"** se añade **"true"** al final para que dibuje en el sentido contrario al reloj.
- Usamos **".beginPath"** para reiniciar el trayecto en cada dibujo y así no se superpongan entre sí los anteriores.
- **".fill"** sigue las reglas de **".stroke"**, es decir, rellena el camino realizado.
- **".globalalpha"** hace que el fondo sea transparente.
- Para dibujar en radianes, **multiplicamos los grados por $\pi/180$** , y si queremos pasar de radianes a grados **multiplicamos por $180/\pi$** .
- Puntos de inicio para dibujar círculos y arcos:



DETALLES DE FUNCIONES PARA ARRAYS Y CÁLCULOS:

- Para calcular un número aleatorio dentro de cualquier rango, usamos **"Math.Random() * (max-min)+min"**.
- **"shift()"** quita el primer elemento del array y lo devuelve a una variable si se asocia a ésta.
- **"pop()"** lo mismo que **"shift()"** pero empieza desde el final del array.
- **"push()"** añade el elemento a un array desde el final de éste.
- **"splice(posicion inicial, nº elementos que quito, elemento que añado nº1, elemento que añado nº2.....)"**. Si no se especifica el primer argumento, empieza desde el final. Si no se especifica el segundo argumento, empieza a quitar a partir de la posición inicial del primer argumento y quita todo lo que haya detrás de ahí.

PILLAR INFORMACIÓN DE CANVAS Y DOCUMENTOS:

Para pillar información del documento usamos **"document.getElementById..."** y para conseguir el contexto del canvas y poder borrar o lo que sea, usamos **".getContext("2d")"**. Por ejemplo, usando ese contexto, podemos borrar el canvas usando **".clearRect(0, 0, canvas.width, canvas.height)"**.

LISTENER:

Utilizamos **"addEventListener('cadena del evento', función asociada { lo que hace el listener cuando ocurre })"**.

MOSTRAR PUNTAJES EN PANTALLA:

Aplicamos el método de **"document.getElementById('cadena nombre del ID').innerHTML = puntos que quieres mostrar"**.

ANOTACIONES HTML:

DETALLES DE ETIQUETAS, CLASES, NOMBRES, ETC PARA CAMBIOS VISUALES CON CSS:

- Para declarar una clase de CSS, usamos **"."**.
- Para declarar una id de CSS, usamos **"#"**.
- Para aplicar cambios a una etiqueta, simplemente ponemos el **nombre** de ésta.
- Si queremos hacer cambios visuales de algo, usamos **"<style>"**.

BOTÓN:

Para crear un botón, usamos **"<input>"**. Asociamos a **"onclick"** la función que ejecutará cuando cliqueemos, **"type"** asociamos que es **"button"** y **"value"** el texto del botón.

CONTADOR DE TIEMPO:

Para un cronómetro usamos **"setInterval(function() {lo que hace}, tiempo en milisegundos)"**.

OTRAS COSAS:

```
cond ? ifTrue : ifFalse
```

```
for (let i = 0; i < list.length; i++) {}
```

```
mousedown
click
mouseup
mousemove
keydown
keypress
keyup
load
contextmenu
```

```
<etiqueta id="mi-id" class="mi-clase"></etiqueta>
```

```
switch (miVariable) {
  case "valor1":
    // algo
    break;
  case "valor2":
    // algo
    break;
  default:
    // algo
}
```

```
// Resultado      0 0.2 0.5 0.8 1
Math.floor // Suelo    0 0   0   0 1
Math.round // Redondear 0 0   1   1 1
Math.ceil  // Techo    0 1   1   1 1
```

```
0/4 = 0   Math.floor(0/4) = 0   0%4 = 0
1/4 = 0.25 Math.floor(1/4) = 0   1%4 = 1
2/4 = 0.5   Math.floor(2/4) = 0   2%4 = 2
3/4 = 0.75  Math.floor(3/4) = 0   3%4 = 3
4/4 = 1     Math.floor(4/4) = 1   4%4 = 0
5/4 = 1.25  Math.floor(5/4) = 1   1%4 = 1
6/4 = 1.5   Math.floor(6/4) = 1   2%4 = 2
7/4 = 1.75  Math.floor(7/4) = 1   3%4 = 3
8/4 = 2     Math.floor(8/4) = 2   4%4 = 0
```

<https://github.com/PabloMarinAbenza/pablitojw>

<https://developer.mozilla.org/es/docs/Web/API/CanvasRenderingContext2D>