

# GENERADORES

# OPERADOR YIELD

- Pausa la ejecución de una función generadora
- Reanuda la ejecución de una función generadora
- Cede el valor de la expresión situada a su derecha
- Se devuelve un objeto con dos propiedades
  - {value: Object, done: Boolean}
- El último valor de un generador que se cede es “undefined”

# SINTAXIS DE UN GENERADOR

```
function* functionName(optionalParam) {  
  yield "un valor";  
}
```



Anatomía de una función generadora

# EJEMPLO DE UN GENERADOR

```
function* functionName(attempts) {  
  for (let iteration = 0; iteration < attempts; iteration++) {  
    yield Math.floor(Math.random() * 15) + 1;  
  }  
}
```

```
let sum = 0;  
const limit = 21;
```

```
for (const mystery of functionName(3)) {  
  sum += mystery;  
  console.log("Sacaste un: " + mystery + ". Acumulas: " + sum);  
}
```

```
console.log((sum > limit) ? "Perdiste" : "Ganaste");
```

Cada vez que se solicita devuelve un valor

# EJEMPLO DE GENERADOR MANUAL

```
function* functionName(attempts) {  
  for (let iteration = 0; iteration < attempts; iteration++) {  
    yield Math.floor(Math.random() * 15) + 1;  
  }  
}
```

```
let lanzaDado = functionName(3);
```

```
console.log(lanzaDado.next().value);
```



```
/* More code goes here
```

```
*
```

```
*
```

```
*
```

```
*/
```

```
console.log(lanzaDado.next().value);
```

Cada vez que se solicita devuelve un valor