



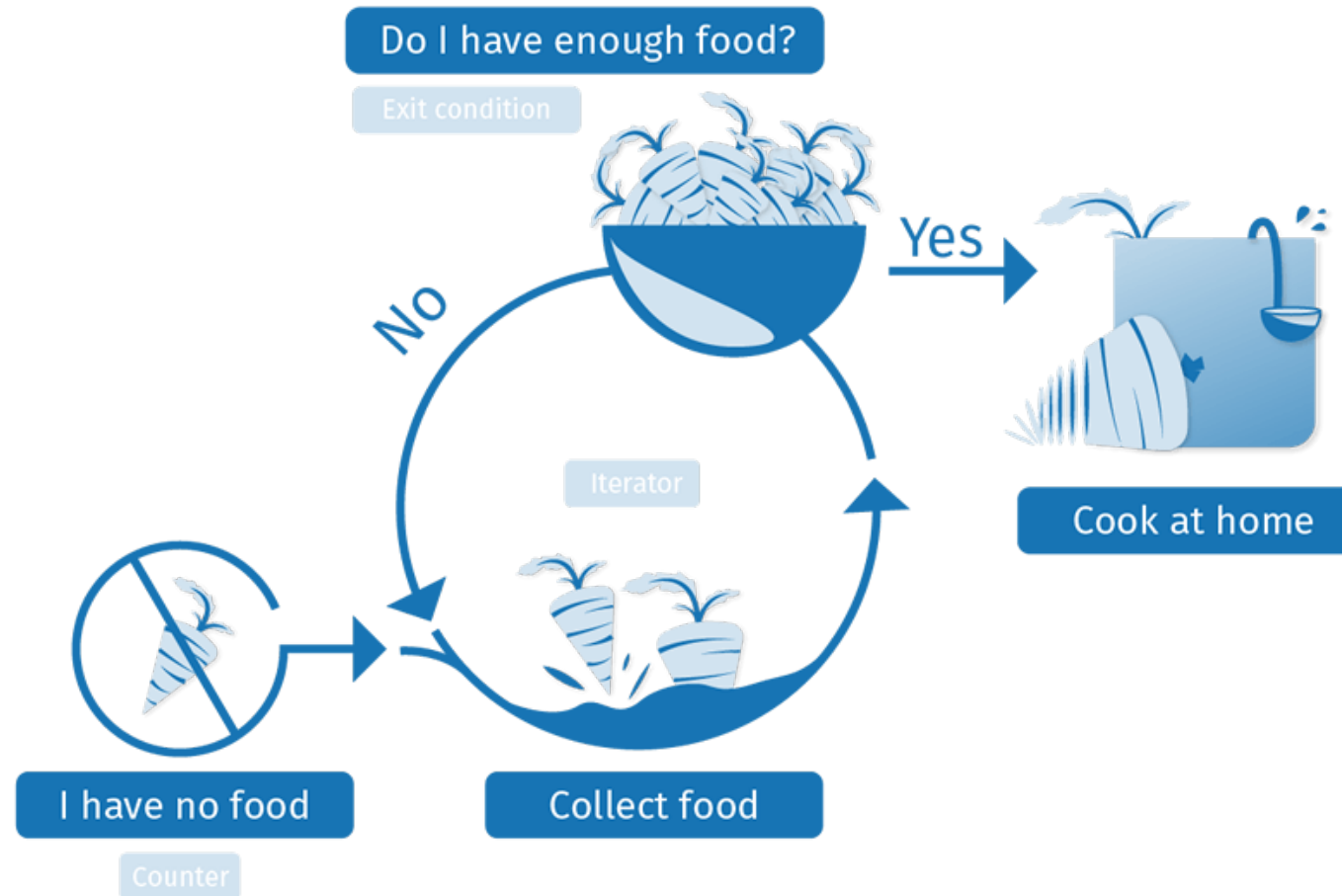
Lenguaje de Programación I



# Bucles de código



# Mantente en el Bucle



# Características del Bucle

---

- Un Contador, que se inicia con un determinado valor ( este será el valor del punto inicial del bucle).
- Una **condicion de salida**, que será el criterio bajo el cual, el bucle se romperá — normalmente un contador que alcanza un determinado valor.
- Un **iterador**, que generalmente incrementa el valor del contador en una cantidad pequeña a cada paso del bucle, hasta que alcanza la condición de salida.



# Pseudocódigo

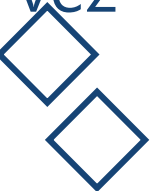
---

```
3  bucle(comida = 0; comidaNecesaria = 10) {  
4      if (comida = comidaNecesaria) {  
5          salida bucle;  
6          // Tenemos suficiente comida; vamos para casa  
7      } else {  
8          comida += 2; // Pasar una hora recogiendo 2 más de comida  
9          // Comenzar el bucle de nuevo  
10     }  
11 }
```

# El Bucle estándar **for**

---

- Un **inicializador** - Este es usualmente una variable con un número asignado, que aumenta el número de veces que el bucle ha sido ejecutado.
- Una **condición de salida** - Generalmente es una expresión que contiene un operador de comparación, una prueba para verificar que la condición de término o salida ha sido cumplida.
- Una **expresión final** - que es siempre evaluada o ejecutada cada vez que el bucle ha completado una iteración.



# Sintaxis

---

*//Sintaxys de los Bucles*

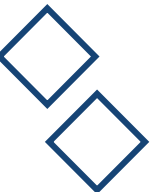
```
for (inicializador; condición de salida; expresión final) {  
    // código a ejecutar  
}
```

```
for (let i = 1; i <= 20; i++) {  
    console.log(i);  
}
```

# Ejercicios

---

- Ingresar con un *input* la cantidad de personas y mostrar en una lista.
- Listar 30 productos de un *array*.
- Hacer una cuenta regresiva del 20 al 1, dando que en ves de 1 salga un mensaje final.

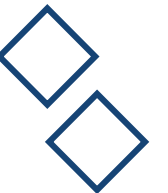


# Salir de un bucle con **break**

---

Si deseas salir de un bucle antes de que se hayan completado todas las iteraciones, puedes usar la declaración ***break***.

Ocurre lo mismo con los bucles - una declaración ***break*** saldrá inmediatamente del bucle y hará que el navegador siga con el código que sigue después



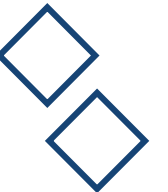


# Omitir iteraciones con **continue**

---

La instrucción ***continue*** funciona de manera similar a ***break***, pero en lugar de salirse del ciclo por completo, salta a la siguiente iteración del ciclo.

<https://github.com/christian-aj/Lenguaje-de-Programacion-I/blob/master/Unidad%2002/ejemplo-continue.html>

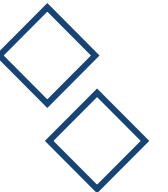


# while y do ... while

---

La instrucción **while** crea un bucle que ejecuta una instrucción especificada siempre que la condición de prueba se evalúe como verdadera. La condición se evalúa antes de ejecutar la sentencia.

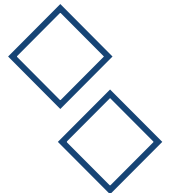
```
initializer  
while (exit-condition) {  
    // code to run  
  
    final-expression  
}
```



# Sintaxis

---

- **Condición:** Una expresión evaluada antes de cada pasada por el bucle. Si esta condición se evalúa como verdadera, **statementse** se ejecuta.
- **Declaraciones:** Una declaración opcional que se ejecuta siempre que la condición se evalúe como verdadera.



# do ... while

---

- La **do...while** crea un bucle que ejecuta una declaración especificada hasta que la condición de prueba se evalúa como falsa.
- La condición se evalúa después de ejecutar la instrucción, lo que hace que la instrucción especificada se ejecute al menos una vez

