

Loops

Loops

- Loops allow your programs to execute lines of code repeatedly, saving you from needing to copy and paste or otherwise repeat lines of code.
- C provides a few different ways to implement loops in your programs, some of which likely look familiar from Scratch.

Loops

```
while (true)
{
}
}
```



- This is what we call an *infinite loop*. The lines of code between the curly braces will execute repeatedly from top to bottom, until and unless we break out of it (as with a `break; statement`) or otherwise kill our program.

Loops

```
while (boolean-expr)
{
}

```



- If the **boolean-expr** evaluates to true, all lines of code between the curly braces will execute repeatedly, in order from top-to-bottom, until **boolean-expr** evaluates to false.
- Somewhat confusingly, the behavior of the Scratch block is reversed, but it is the closest analog.

Loops

```
do  
{  
  
}  
while (boolean-expr);
```

- This loop will execute all lines of code between the curly braces once, and then, if the **boolean-expr** evaluates to true, will go back and repeat that process until **boolean-expr** evaluates to false.

Loops

```
for (int i = 0; i < 10; i++)  
{  
  
}
```



- Syntactically unattractive, but for loops are used to repeat the body of a loop a specified number of times, in this example 10.
- The process undertaken in a for loop is:
 - The counter variable(s) (here, `i`) is set
 - The Boolean expression is checked.
 - If it evaluates to `true`, the body of the loop executes.
 - If it evaluates to `false`, the body of the loop does not execute.
 - The counter variable is incremented, and then the Boolean expression is checked again, etc.

Loops

```
for (start; expr; increment)
{
}
}
```



- Syntactically unattractive, but for loops are used to repeat the body of a loop a specified number of times, in this example 10.
- The process undertaken in a for loop is:
 - The statement(s) in **start** are executed
 - The **expr** is checked.
 - If it evaluates to **true**, the body of the loop executes.
 - If it evaluates to **false**, the body of the loop does not execute.
 - The statement(s) in **increment** are executed, and then the **expr** is checked again, etc.

Loops

while

- Use when you want a loop to repeat an unknown number of times, and possibly not at all.

do-while

- Use when you want a loop to repeat an unknown number of times, but at least once.

for

- Use when you want a loop to repeat a discrete number of times, though you may not know the number at the moment the program is compiled.