

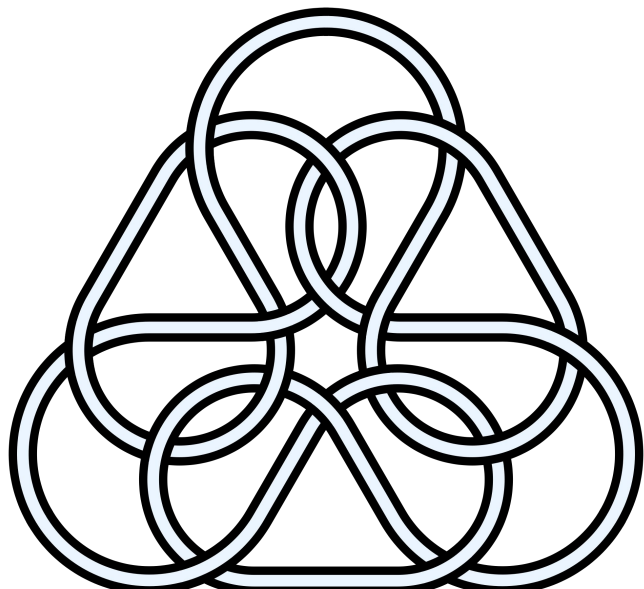


For Loops



WHILE LOOP REVIEW

- Three main parts of a while loop
 - 1) Initialization
 - 2) Terminating Condition
 - 3) Update



COUNTER

- Three main parts of a loop
 - 1) **Initial condition**

```
int j= 0;
while (j < 10) {
    printf("Skip Day Penalty\n");
    j = j + 1;
}
```



COUNTER

- Three main parts of a loop
 - 2) **Terminating Condition**

```
int j= 0;
while (j < 10) {
    printf("Skip Day Penalty\n");
    j = j + 1;
}
```



COUNTER

- Three main parts of a loop
 - 3) **Update**

```
int j= 0;
while (j < 10) {
    printf("Skip Day Penalty\n");
    j = j + 1;
}
```



FOR LOOPS

- For Loops combine all three steps into one line of code

```
int i;  
for (i = 0; i < 10; i++) {  
    printf("Skip Day Penalty\n");  
}
```



FOR LOOPS

- Mainly used for iterating over a *finite* set

```
int i;  
for (i = 0; i < 10; i++) {  
    printf("Skip Day Penalty\n");  
}
```



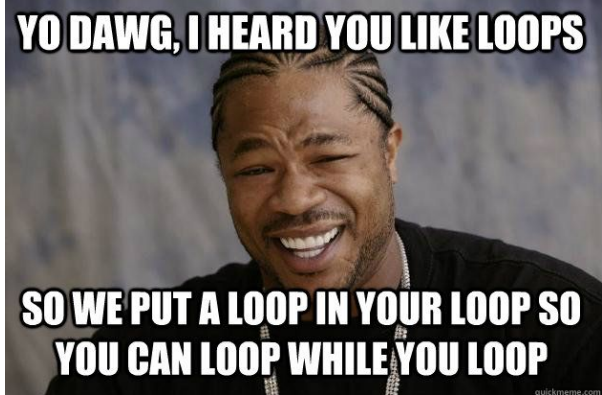
FOR LOOPS

```
int i;  
for (i = 0; i < 10; i++) {  
    /*Code here*/  
}
```

- A for loop still has all three parts of a loop

1) Initialization

a) i = 0



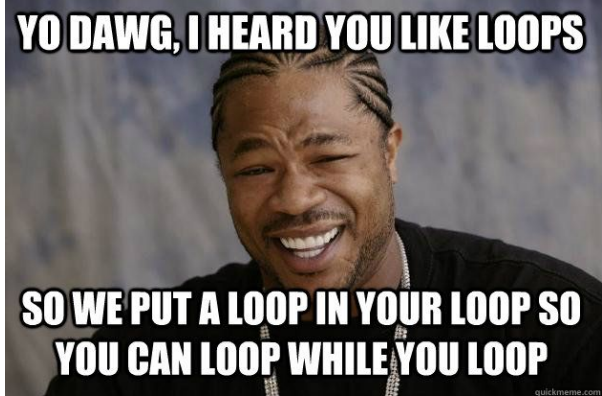
FOR LOOPS

```
int i;  
for (i = 0; i < 10; i++) {  
    /*Code here*/  
}
```

- A for loop still has all three parts of a loop

2) Update

a) **i++**



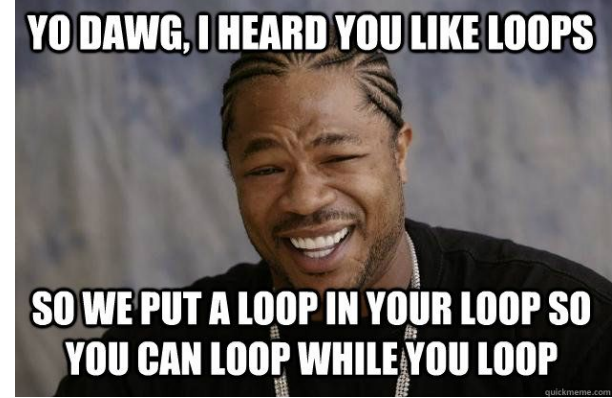
FOR LOOPS

```
int i;  
for (i = 0; i < 10; i++) {  
    /*Code here*/  
}
```

- A for loop still has all three parts of a loop

3) Termination

a) **i < 10**

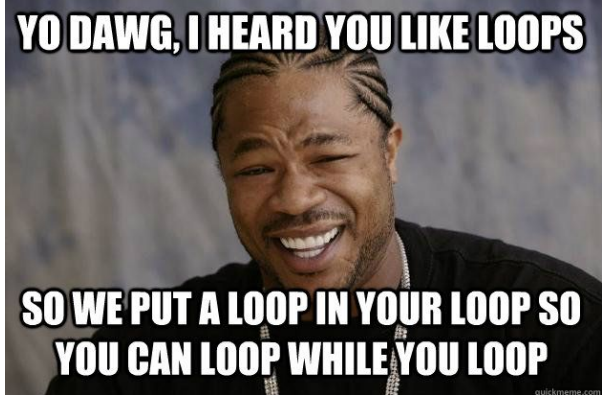


FOR LOOPS

```
int i;  
for (i = 0; i < 10; i++) {  
    /*Code here*/  
}
```

- Final format

```
For (initialization; termination; update) {  
    /*Code here*/  
}
```



FOR LOOP

- Modify the box of *'s → use *for* loops instead of *while* loops



FOR LOOP CODING CHALLENGE; PART 1-2

- 1) Create a for loop that will count from 0 - 10 by 1's
- 2) Create a for loop that will count from 0 - 10 by 2's



FOR LOOP CODING CHALLENGE; PART 3-4

- 3) Create a for loop that will count from 0 - 10 by a user-inputted number
- 4) Create a for loop that will count from a user-inputted minimum & maximum, by a user-inputted number

