

For Loops

- Iteration code blocks must always return something, unlike void functions. If you don't return something, you will err out.
- For loops are a version of iteration statements. These loops contain particular clauses that change the initialization and iteration of a looped code block. Other than that, they are the same as while loops. The clauses include the init, condition, and iteration clauses.
- Another type of loop is the "foreach" loop. This iteration loops through each element in an enumerable object. This is useful, as almost every type in C# that are used to represent a set or list of elements are enumerable, like strings, arrays, and lists.
- Iteration statements are critical to a functioning 'loop.' Iteration statements result in the statements within the iteration are executed repeatedly. The amount of times the code is looped depends on the iteration statement parameters.

- ```
public int o;

string[] numbers = {"one", "two", "three"};

void goCounting (char c) {
 for (int o, o<3, ++o) {
 print(numbers[c]);
 print(o);
 }
}
```
- ```
public int i

string[] oddNumber = {"one", "two", "three", "four", "five"};

void oddCounting (char c) {
    for (int i, i<6, ++i) {
```

```
        print(numbers[c]);  
        print(i);  
        if (i == 2, 4) {  
            continue;  
        }  
    }  
}
```

- `public int[] numbers = {1,2,3,4,5};`

```
public void onetoFive (int i) {  
    foreach (int i in numbers) {  
        print(i);  
    }  
}
```

- `public char[] hello = {'w', 'h', 'a', 't', 's', 'u', 'p'};`

```
public void WhatsUp (char c) {  
    foreach (char c in hello) {  
        if (c == 's') continue;  
        print(c);  
    }  
}
```

- `public char[] spelling = {'p', 'e', 'a', 'w', 'n', 'u', 't'};`

```
public void SpellingBee (char c) {  
    foreach (char c in spelling) {
```

```
        if (c == 'w') goto char[4];  
        print(c);  
    }  
}  
  
• public int[] numbers = {0,2,3,4,6,8};  
  public void evenNum (int i) {  
      if (i == 3) continue;  
      foreach (int i in numbers) {  
          print(i);  
      }  
  }
```