Classes

Defining A Class

Python

To define a class in Python, use the `class` keyword and add your `__init__()` function (the constructor) if you're not making a static class

```
class Dog:
        # if your class is static (or it doesn't take any attributes for whatever
        # reason), then you do not need to define this.
        def __init__(self, name):
                self.name = name
```

TypeScript

To define a class in TypeScript, also use the `class` keyword, then add your attributes and finally add your `constructor` function, if you're not making a static class.

Use the `private` modifier on attributes if they should not be accessible without using a method. Ensure you add appropriate types to all attributes and parameters!

```
class Dog {
        private name: string;

        constructor(name: string) {
                this.name = name;
        }
}
```

C#

Defining a class in C# is very similar to TypeScript, with a few syntax changes. You will need to use the `public` modifier on the `class` keyword, and your attributes will be defined as `modifier type name` instead of `modifer name: type`.

In C#, the `constructor` function is defined by having a function with the same name as the class. i.e., if your class is named `Person`, your constructor function would be `public Person(...)`, and if your class is `Dog`, the constructor would be `public Dog(...)`.

In C#, `this` does exist, but is not required. Additionally, it is usually common practice to prefix any private variables or functions with `_`.

```
public class Dog
{
        private string _name;

        public Dog(string name)
        {
```

```
                _name = name;
        }
}
```