Basic Types

number
bool
string
any

Declarations

var x: any	Basic declaration
var x: {a:any; b:any;};	x is an object
var x: Foo;	x is instance of class Foo

Parameters

x:any	Basic parameter
x:() => string	Function that returns string.
x:{a:any;-b:any;}	Object that contains a and b
x:Foo	Object that is a class of Foo
x?:any	Optional parameter

Function

```
function a(x:bool);
function a(x:number);
function a(x:any):bool{
    return x%2 == 0;
}
a(2);
```

Basic Class

```
class Goose{
    a:number;
    private b:bool;
    constructor(x: number, y:bool = true){
        this.a = x;
        this.b = y;
    }
}
var x: Goose = new Goose(50);
```

Class Inheritance

```
class Suzy extends Goose{
    constructor(public c: string){
        super(0, true);
    }
}
var y: Suzy = new Suzy("foo");
console.log(y.c + " | " + y.a);
var z: Goose = new Suzy("bar")
```

Interface Example

```
interface Foo{
    a(b:number):bool;
}
class Bar implements Foo{
    a(b:number){
        return false;
    }
}
var x:Foo = new Bar();
```

Function Explination

Overload functions with bool and number.
Create new function that takes a number x and returns a bool.
Execute function.

Class Explination

Create a new class.
Public attribute.
Private attribute.
constructor with attributes x and y
y is optional and defaulted to true.
Instantiate with x as 50 and use default for y

Inheritance Explination

New class that extends Goose
constructor. creates a public var c
calls inherited constructor.
New instance of the class.
Accessing public attribute c and a.
Making a new Goose using class Suzy.

Interface Explination

Create a new Interface
    Which much have function a
Create a new class that implement Foo
    Implement everything in Foo
Create a new instance of type Foo