EXAMPLE 26: FAKING 'PUBLIC' AND 'PRIVATE' OBJECT PROPERTIES IN JAVASCRIPT

EXAMPLE 26: FAKING 'PUBLIC' AND 'PRIVATE' OBJECT PROPERTIES IN JAVASCRIPT

JAVA, C++ AND OTHER LANGUAGES HAVE THE NOTION OF ACCESS MODIFIERS

(WELL, IN THOSE LANGUAGES, THE TERM IS

THESE ALLOW THE PROPERTIES OF AN OBJECT TO BE MARKED AS PUBLIC, PRIVATE OR PROTECTED

PUBLIC, PRIVATE OR PROTECTED

PUBLIC PROPERTIES ARE ACCESSIBLE TO ALL CODE, INSIDE AND OUTSIDE THE OBJECT

PRIVATE PROPERTIES ARE ACCESSIBLE ONLY WITHIN THE OBJECT ITSELF.

PROTECTED PROPERTIES ARE ACCESSIBLE ONLY TO OBJECTS THAT INHERIT FROM AN OBJECT

EXAMPLE 26: FAKING ‘PUBLIC’ AND ‘PRIVATE’ OBJECT PROPERTIES IN JAVASCRIPT

JAVA, C++ AND OTHER LANGUAGES HAVE THE NOTION OF ACCESS MODIFIERS

(WELL, IN THOSE LANGUAGES, THE TERM IS

THESE ALLOW THE PROPERTIES OF AN OBJECT TO BE MARKED AS PUBLIC, PRIVATE OR PROTECTED

MANY EXPERIENCED PROGRAMMERS NATURALLY THINK IN TERMS OF PUBLIC, PRIVATE AND PROTECTED..

..SO EVEN THOUGH JAVASCRIPT DOES NOT EXPLICITLY SUPPORT THEM, WAYS HAVE BEEN FOUND TO MIMIC THEM:-)

EXAMPLE 26: FAKING 'PUBLIC' AND 'PRIVATE' OBJECT PROPERTIES IN JAVASCRIPT

THE PROPERTIES WE HAVE SEEN SO FAR ARE, BY DEFAULT, PUBLIC.

TO MAKE A PROPERTY PRIVATE, JUST MAKE IT LOCAL TO THE CONSTRUCTOR:-)

TO MAKE A PROPERTY PRIVATE, JUST MAKE IT LOCAL TO THE CONSTRUCTOR:-)

```
function Rectangle(length,breadth,color) {
  this.length = length;
  this.breadth = breadth;
  this.color = color;

  var privateVar = "I don't want anyone to know
this, but I am actually not just a rectangle, but
also a square”;

  this.sayHello = function() {
    console.log(privateVar);
  };
}
```

TO MAKE A PROPERTY PRIVATE, JUST MAKE IT LOCAL TO THE CONSTRUCTOR:-)

```
function Rectangle(length,breadth,color) {
  this.length = length;
  this.breadth = breadth;
  this.color = color;

  var privateVar = "I don't want anyone to know
this, but I am actually not just a rectangle, but
also a square";

  this.sayHello = function() {
    console.log(privateVar);
  };
}
```

TO MAKE A PROPERTY PRIVATE, JUST MAKE IT LOCAL TO THE CONSTRUCTOR:-)

```
function Rectangle(length,breadth,color) {
  this.length = length;
  this.breadth = breadth;
  this.color = color;

  var privateVar = "I don't want anyone to know
this, but I am actually not just a rectangle, but
also a square”;

  this.sayHello = function() {
    console.log(privateVar);
  };
}
```

TO MAKE A PROPERTY **PRIVATE**, JUST MAKE IT LOCAL TO THE CONSTRUCTOR:-)

WE CAN STILL ACCESS THIS VARIABLE

BUT ANY OUTSIDE CODE THAT TRIES TO ACCESS THIS WILL RESULT IN AN UNDEFINED

```
function objectStuff() {
    var rectangle2 = new Rectangle(3.3,
2.5, "Blue")
    console.log(rectangle2.privateVar);
                    undefined
```

TO MAKE A PROPERTY PRIVATE, JUST MAKE IT LOCAL TO THE CONSTRUCTOR:-)

OUTSIDE CODE CAN STILL CALL THE PROPERTY THAT ACCESSES THIS PRIVATE VARIABLE JUST FINE THOUGH!

```
function objectStuff() {
  var rectangle2 = new Rectangle(3.3,
2.5, "Blue")
  rectangle2.sayHello();
```

```
I don't want anyone to know this, but I am actually not just a rectangle, but also a square
```

TO MAKE A PROPERTY PRIVATE, JUST MAKE IT LOCAL TO THE CONSTRUCTOR:-)

```
function Rectangle(length,breadth,color) {
  this.length = length;
  this.breadth = breadth;
  this.color = color;

  var privateVar = "I don't want anyone to know
this, but I am actually not just a rectangle, but
also a square”;

  this.sayHello = function() {
    console.log(privateVar);
  };
}
```

TO MAKE A PROPERTY PRIVATE, JUST MAKE IT LOCAL TO THE CONSTRUCTOR:-)

```
function Rectangle(length,breadth,color) {
  this.length = length;
  this.breadth = breadth;
  this.color = color;

  var privateVar = "I don't want anyone to know
this, but I am actually not just a rectangle, but
also a square";

  this.sayHello = function() {
    console.log(privateVar);
  };
}
```

IN CASE YOU ARE WONDERING..

IN CASE YOU ARE WONDERING..

HOW A PROPERTY WAS ABLE TO ACCESS A VARIABLE THAT IS DEFINED OUTSIDE THAT PROPERTY -

```
function Rectangle(length,breadth,color) {
  this.length = length;
  this.breadth = breadth;
  this.color = color;

  var privateVar = "I don't want anyone to know
this, but I am actually not just a rectangle, but
also a square”;


  this.sayHello = function() {
    console.log(privateVar);
  };
}
```

IN CASE YOU ARE WONDERING..

HOW A PROPERTY WAS ABLE TO ACCESS A VARIABLE THAT IS DEFINED OUTSIDE THAT PROPERTY -

THIS IS OUR FIRST ENCOUNTER WITH

CLOSURES

THIS WAS JUST A TEASER, MORE LATER:-)