EXAMPLE 43: CLOSURES CAN BE BUILT AROUND BOTH DECLARED FUNCTIONS AND FUNCTION LITERALS

EXAMPLE 43: CLOSURES CAN BE BUILT AROUND BOTH DECLARED FUNCTIONS AND FUNCTION LITERALS

CLOSURE = SAY WE HAVE A NESTED FUNCTION + VARIABLES LOCAL TO THE OUTER SCOPE

"REFERENCING ENVIRONMENT"

THE NESTED FUNCTION COULD BE EITHER A DECLARED FUNCTION OR A FUNCTION LITERAL

```
var PI = 3.14;

window.onload = function(){
  var circle1 = new Circle(3);
  var circle2 = new Circle(4);
  var circle3 = new Circle(5);

  var circleArray = [circle1,circle2,circle3];
  var PI = 3.1415;
```

THE NESTED FUNCTION COULD BE EITHER A DECLARED FUNCTION OR A FUNCTION LITERAL

```
  var printStuffAboutCircleArray = function(circleArray) {
    var PI = 3.14159;

    var getArea = function(circle){
      console.log("Inside the nested function getArea, PI = " + PI);
      return PI * circle.radius * circle.radius;
    };
    for (var i = 0;i<circleArray.length;i++) {
       var c = circleArray[i];
       console.log(c.radius + " " + getArea(c));
    }
   return getArea;
  }

  var areaFunction = printStuffAboutCircleArray(circleArray);
  areaFunction(circle1);


};
```

CLOSURE = SAY WE HAVE A NESTED +

VARIABLES LOCAL TO THE OUTER SCOPE

“REFERENCING

THE NESTED FUNCTION COULD BE EITHER A DECLARED FUNCTION OR A FUNCTION LITERAL

```
 var printStuffAboutCircleArray = 
function(circleArray) {
  var PI = 3.14159;

   var getArea = function(circle){
     console.log("Inside the nested function getArea, PI = " + PI);
     return PI * circle.radius * circle.radius;
  };
  for (var i = 0;i<circleArray.length;i++) {
     var c = circleArray[i];
     console.log(c.radius + "," + getArea(c));
  }
 return getArea;
}
```

FUNCTION LITERALS

CLOSURE = SAY WE HAVE A NESTED +

VARIABLES LOCAL TO THE OUTER SCOPE

“REFERENCING

THE NESTED FUNCTION COULD BE EITHER A DECLARED FUNCTION OR A FUNCTION LITERAL

```
printStuffAboutCircleArray(circleArray) {
  var PI = 3.14159;

   getArea(circle){
    console.log("Inside the nested function getArea, PI = " + PI);
    return PI * circle.radius * circle.radius;
  };
  for (var i = 0;i<circleArray.length;i++) {
     var c = circleArray[i];
     console.log(c.radius + "," + getArea(c));
  }
 return getArea;
}
```

DECLARED FUNCTIONS

CLOSURE = SAY WE HAVE A NESTED + VARIABLES LOCAL TO THE OUTER SCOPE

"REFERENCING