

EXAMPLE 27: PASSING FUNCTIONS TO FUNCTIONS AS ARGUMENTS

EXAMPLE 27: PASSING FUNCTIONS TO FUNCTIONS AS ARGUMENTS

SAY WE HAVE A RECTANGLE OBJECT (CREATED VIA A

```
function Rectangle(length, breadth, color) {  
    this.length = length;  
    this.breadth = breadth;  
    this.color = color;  
    this.getArea = function() {  
        return this.length * this.breadth;  
    }  
}
```

EXAMPLE 27: PASSING FUNCTIONS TO FUNCTIONS AS ARGUMENTS

SAY WE HAVE A RECTANGLE OBJECT (CREATED VIA A
NOW SAY WE NEED TO 'COMPARE' RECTANGLES

```
function compareTwoRectangles(rectangle1, rectangle2)
{
    // if rectangle1 > rectangle2, return 1
    // if rectangle1 < rectangle2, return -1
    // if rectangle1 == rectangle2, return 0;
}
```

NOW SAY WE NEED TO 'COMPARE' RECTANGLES

```
function compareTwoRectangles(rectangle1, rectangle2)
{
    // if rectangle1 > rectangle2, return 1
    // if rectangle1 < rectangle2, return -1
    // if rectangle1 == rectangle2, return 0;
```

ERRM..HOW DOES ONE COMPARE 2 RECTANGLES?

BY LENGTH?

BY BREADTH?

BY AREA?

EASY! JUST PASS IN A FUNCTION THAT
DOES THE COMPARISON!

**EASY! JUST PASS IN A FUNCTION THAT
DOES THE COMPARISON!**

```
function compareTwoRectangles(rectangle1, rectangle2)
{
    // if rectangle1 > rectangle2, return 1
    // if rectangle1 < rectangle2, return -1
    // if rectangle1 == rectangle2, return 0;
}
```

**EASY! JUST PASS IN A FUNCTION THAT
DOES THE COMPARISON!**

```
function compareTwoRectangles(rectangle1, rectangle2,  
compareFunction) {  
    // if rectangle1 > rectangle2, return 1  
    // if rectangle1 < rectangle2, return -1  
    // if rectangle1 == rectangle2, return 0;  
    return compareFunction(rectangle1, rectangle2);  
}
```

BY LENGTH?

BY BREADTH?

BY AREA?

**EASY! JUST PASS IN THE CORRECT
VERSION OF THE COMPARE FUNCTION!**

EASY! JUST PASS IN THE CORRECT

A COMPARE FUNCTION TAKES IN 2
RECTANGLES

IT WILL RETURN A NUMBER > 0 IF
 $RECTANGLE1 > RECTANGLE2$

IT WILL RETURN A NUMBER $= 0$ IF
 $RECTANGLE1 == RECTANGLE2$

IT WILL RETURN A NUMBER < 0 IF
 $RECTANGLE1 < RECTANGLE2$

**EASY! JUST PASS IN A
FUNCTION THAT DOES THE**

```
function compareTwoRectangles(rectangle1, rectangle2,  
compareFunction) {  
    // if rectangle1 > rectangle2, return 1  
    // if rectangle1 < rectangle2, return -1  
    // if rectangle1 == rectangle2, return 0;  
    return compareFunction(rectangle1, rectangle2);  
}
```

BY AREA?

```
compareTwoRectangles(rectangle1, rectangle2, compareTwoRect  
lesByArea)
```


**EASY! JUST PASS IN A
FUNCTION THAT DOES THE**

```
function compareTwoRectangles(rectangle1, rectangle2,  
compareFunction) {
```

```
    // if rectangle1 > rectangle2, return 1
```

```
    // if rectangle1 < rectangle2, return -1
```

```
    // if rectangle1 == rectangle2, return 0;
```

```
    return compareFunction(rectangle1, rectangle2);
```

```
compareTwoRectangles(rectangle1, rectangle2, compareTwoRectan  
lesByArea)
```

```
function compareTwoRectanglesByArea(rectangle1, rectangle2)  
{
```

```
    var area1 = rectangle1.getArea();
```

```
    var area2 = rectangle2.getArea();
```

```
    console.log("Comparing 2 rectangles by area: " + area1 +  
and " + area2);
```

```
    return Math.sign(area1-area2);
```

```
}
```

EASY! JUST PASS IN A
FUNCTION THAT DOES THE

```
function compareTwoRectangles(rectangle1, rectangle2,  
compareFunction) {
```

```
// if rectangle1 > rectangle2, return 1
```

```
// if rectangle1 < rectangle2, return -1
```

```
// if rectangle1 == rectangle2, return 0;
```

```
return compareFunction(rectangle1, rectangle2);
```

```
}  
compareTwoRectangles(rectangle1, rectangle2, compareTwoRectan  
lesByArea)
```

```
function compareTwoRectanglesByArea(rectangle1, rectangle2)
```

```
{
```

```
var area1 = rectangle1.getArea();
```

```
var area2 = rectangle2.getArea();
```

```
console.log("Comparing 2 rectangles by area: " + area1 +  
and " + area2);
```

```
return Math.sign(area1-area2);
```

```
}
```

EASY! JUST PASS IN A
FUNCTION THAT DOES THE

```
function compareTwoRectangles(rectangle1, rectangle2,  
compareFunction) {
```

```
// if rectangle1 > rectangle2, return 1
```

```
// if rectangle1 < rectangle2, return -1
```

```
// if rectangle1 == rectangle2, return 0;
```

```
return compareFunction(rectangle1, rectangle2);
```

```
}  
compareTwoRectangles(rectangle1, rectangle2, compareTwoRectan  
lesByLength)
```

BY LENGTH?

```
function compareTwoRectanglesByLength(rectangle1,  
rectangle2) {
```

```
return Math.sign(rectangle1.length - rectangle2.length);
```

```
}
```

EASY! JUST PASS IN A
FUNCTION THAT DOES THE

```
function compareTwoRectangles(rectangle1, rectangle2,  
compareFunction) {
```

```
// if rectangle1 > rectangle2, return 1
```

```
// if rectangle1 < rectangle2, return -1
```

```
// if rectangle1 == rectangle2, return 0;
```

```
return compareFunction(rectangle1, rectangle2);
```

```
}  
compareTwoRectangles(rectangle1, rectangle2, compareTwoRectan  
lesByLength)
```

BY LENGTH?

```
function compareTwoRectanglesByLength(rectangle1,  
rectangle2) {
```

```
return Math.sign(rectangle1.length - rectangle2.length);
```

```
}
```