EXAMPLE 16: CREATE AN OBJECT USING A CONSTRUCTOR FUNCTION

EXAMPLE 16: CREATE AN OBJECT USING A CONSTRUCTOR FUNCTION

WE JUST LEARNT HOW CREATE AN OBJECT THE "JSON" WAY

"JSON" = JavaScript Object Notation

THERE IS AN ALTERNATIVE HOWEVER, WHICH INVOLVES THE USE OF A SPECIAL FUNCTION CALLED AN OBJECT CONSTRUCTOR

THERE IS AN ALTERNATIVE HOWEVER, WHICH INVOLVES THE USE OF A SPECIAL FUNCTION CALLED AN **OBJECT CONSTRUCTOR**

IF YOU HAVE ENCOUNTERED CONSTRUCTORS IN PYTHON, JAVA, OR C++, THESE ARE SIMILAR, BUT SUBTLY DIFFERENT, SO PAY CLOSE ATTENTION :-)

AN OBJECT CONSTRUCTOR

IS A FUNCTION THAT CREATES OBJECTS, ALL OF WHICH HAVE THE SAME PROPERTIES

THIS FUNCTION 'LOOKS' LIKE ANY OTHER FUNCTION, BUT IT IS SPECIAL IN A FEW WAYS

AN OBJECT CONSTRUCTOR
'LOOKS' LIKE ANY OTHER FUNCTION, BUT IT IS
SPECIAL IN A FEW WAYS

IT MAGICALLY 'KNOWS' WHAT OBJECT IT IS CREATING,
AND ACCESS IT VIA A VARIABLE CALLED `this`

IT MUST BE CALLED IN A SPECIFIC WAY, USING THE
KEYWORD `new`

THERE IS AN ALTERNATIVE HOWEVER, WHICH INVOLVES THE USE OF A SPECIAL FUNCTION CALLED

AN OBJECT CONSTRUCTOR

THIS FUNCTION 'LOOKS' LIKE ANY OTHER FUNCTION, BUT IT IS SPECIAL IN A FEW WAYS

WRITING AN OBJECT CONSTRUCTOR

IT MAGICALLY 'KNOWS' WHAT OBJECT IT IS CREATING, AND ACCESS IT VIA A VARIABLE CALLED `this`

IT MUST BE CALLED IN A SPECIFIC WAY, USING THE KEYWORD `new`

THERE IS AN ALTERNATIVE HOWEVER, WHICH INVOLVES THE USE OF A SPECIAL FUNCTION CALLED

AN OBJECT CONSTRUCTOR

THIS FUNCTION 'LOOKS' LIKE ANY OTHER FUNCTION, BUT IT IS SPECIAL IN A FEW WAYS

IT MAGICALLY 'KNOWS' WHAT OBJECT IT IS CREATING, AND ACCESS IT VIA A VARIABLE CALLED `this`

USING AN OBJECT CONSTRUCTOR

IT MUST BE CALLED IN A SPECIFIC WAY, USING THE KEYWORD `new`

THESE ARE THE BITS THAT MATTER ABOUT USING OBJECT CONSTRUCTORS, SO LET'S DIG INTO THEM.

WRITING AN OBJECT CONSTRUCTOR

USING AN OBJECT CONSTRUCTOR

WRITING AN OBJECT CONSTRUCTOR

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

AN OBJECT CONSTRUCTOR

'LOOKS' LIKE ANY OTHER FUNCTION, BUT

IT IS SPECIAL IN A FEW WAYS

IT MAGICALLY 'KNOWS' WHAT OBJECT IT IS CREATING,
AND ACCESS IT VIA A VARIABLE CALLED `this`

WRITING AN OBJECT CONSTRUCTOR

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

THERE IS AN ALTERNATIVE HOWEVER, WHICH INVOLVES THE USE OF A SPECIAL FUNCTION CALLED

'LOOKS' LIKE ANY OTHER FUNCTION

WRITING AN OBJECT CONSTRUCTOR

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

THERE IS AN ALTERNATIVE HOWEVER, WHICH INVOLVES THE USE OF A SPECIAL FUNCTION CALLED

'LOOKS' LIKE ANY OTHER FUNCTION BUT

WRITING AN OBJECT CONSTRUCTOR

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

THERE IS AN ALTERNATIVE HOWEVER WHICH INVOLVES THE USE OF A SPECIAL FUNCTION CALLED

'LOOKS' LIKE ANY OTHER FUNCTION **BUT**

IT MAGICALLY 'KNOWS' WHAT OBJECT IT IS CREATING,
AND ACCESS IT VIA A VARIABLE CALLED `this`

WRITING AN OBJECT CONSTRUCTOR

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

THE CONSTRUCTOR USES `this` TO SPECIFY, AND INITIALISE THE PROPERTIES OF THE OBJECT BEING CREATED

WRITING AN OBJECT CONSTRUCTOR

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

THE CONSTRUCTOR USES `this` TO SPECIFY, AND INITIALISE THE PROPERTIES OF THE OBJECT BEING CREATED

WRITING AN OBJECT CONSTRUCTOR

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

THE CONSTRUCTOR USES `this` TO SPECIFY, AND INITIALISE THE PROPERTIES OF THE OBJECT BEING CREATED

THE CONSTRUCTOR USES `this` TO SPECIFY, AND INITIALISE THE PROPERTIES OF THE OBJECT BEING CREATED

THAT'S HOW AN OBJECT CONSTRUCTOR ACTS LIKE A BLUEPRINT FROM WHICH OBJECTS ARE CREATED.

WRITING AN OBJECT CONSTRUCTOR

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

THE CONSTRUCTOR USES `this` TO SPECIFY, AND INITIALISE THE PROPERTIES OF THE OBJECT BEING CREATED

WRITING AN OBJECT CONSTRUCTOR

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

THE CONSTRUCTOR USES `this` TO SPECIFY, AND INITIALISE THE PROPERTIES OF THE OBJECT BEING CREATED

WRITING AN OBJECT CONSTRUCTOR

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

BTW, NOTICE THAT THE OBJECT CONSTRUCTOR NEITHER TAKES IN `this` AS AN ARGUMENT, NOR RETURNS `this`. IT JUST MAGICALLY KNOWS WHAT `this` IS.

BTW, NOTICE THAT THE OBJECT CONSTRUCTOR NEITHER TAKES IN `this` AS AN ARGUMENT, NOR RETURNS `this`. IT JUST MAGICALLY KNOWS WHAT `this` IS.

BTW, NOTICE THAT THE OBJECT CONSTRUCTOR NEITHER TAKES IN `this` AS AN ARGUMENT, **NOR RETURNS `this`.** IT JUST MAGICALLY KNOWS WHAT `this` IS.

BTW, NOTICE THAT THE OBJECT CONSTRUCTOR NEITHER TAKES IN `this` AS AN ARGUMENT, NOR RETURNS `this`. IT JUST MAGICALLY KNOWS WHAT `this` IS.

WRITING AN OBJECT CONSTRUCTOR

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

BTW, NOTICE THAT THE OBJECT CONSTRUCTOR NEITHER TAKES IN `this` AS AN ARGUMENT, NOR RETURNS `this`. IT JUST MAGICALLY KNOWS WHAT `this` IS.

THESE ARE THE BITS THAT MATTER ABOUT USING OBJECT CONSTRUCTORS, SO LET'S DIG INTO THEM.

WRITING AN OBJECT CONSTRUCTOR

USING AN OBJECT CONSTRUCTOR

USING AN OBJECT CONSTRUCTOR

```
var rectangle = new Rectangle(3.3, 2.5,"Blue");
```

AN OBJECT CONSTRUCTOR MUST BE CALLED IN A SPECIFIC WAY, USING THE KEYWORD `new`

THIS IS THE ONLY DIFFERENCE BETWEEN CALLING A CONSTRUCTOR, AND CALLING ANY OTHER FUNCTION

```
var rectangle = new Rectangle(3.3, 2.5,"Blue");
```

THAT `new` KEYWORD IS A SIGN TO THE JAVASCRIPT INTERPRETER..

TO CREATE AN EMPTY OBJECT, AND PASS IT INTO THE FUNCTION (SECRETLY) AS THE `this`..

..IN ADDITION TO THE OTHER FUNCTION ARGUMENTS

```
var rectangle = new Rectangle(3.3,
2.5,
```

THAT `new` KEYWORD IS A SIGN TO THE JAVASCRIPT INTERPRETER..

TO CREATE AN EMPTY OBJECT, AND PASS IT INTO THE FUNCTION (SECRETLY) AS THE

..IN ADDITION TO THE OTHER FUNCTION ARGUMENTS

```
var rectangle = new Rectangle(3.3, 2.5,
```

THAT

```
var this = {};
```

TO CREATE AN EMPTY OBJECT, AND PASS IT INTO THE FUNCTION (SECRETLY) AS THE `this`..

..IN ADDITION TO THE OTHER FUNCTION ARGUMENTS

INTERPRETER..

var rectangle = new Rectangle(3.3, 2.5,

THAT

```
var this = {};
```

TO CREATE AN EMPTY OBJECT, AND PASS IT INTO THE FUNCTION (SECRETLY) AS THE this..

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

INTERPRETER..

AND ADDITION TO THE OTHER FUNCTION ARGUMENTS

```
function Rectangle(this, length,breadth,color)
{
  var this = {};
  this.length = length;
  this.breadth = breadth;
  this.color = color;
  return this;
}
var rectangle = Rectangle(this,
3.3,2.5,"Blue");
```

TO CREATE AN EMPTY OBJECT, AND PASS IT INTO THE FUNCTION (SECRETLY) AS THE `this`..

..IN ADDITION TO THE OTHER FUNCTION ARGUMENTS

```
 var rectangle = new Rectangle(3.3,
2.5,"Blue");
```

IS LOGICALLY EQUIVALENT TO

```
function Rectangle(this, length,breadth,color
{
 var this = {};
   this.length = length;
   this.breadth = breadth;
   this.color = color;
   return this;
}
var rectangle = Rectangle(this,
3.3,2.5,”Blue");
```

WRITING AN OBJECT CONSTRUCTOR

REMEMBER THE MAGICAL ROLE OF `this`

USING AN OBJECT CONSTRUCTOR

REMEMBER THE MAGICAL ROLE OF `new`