

Object Data Type

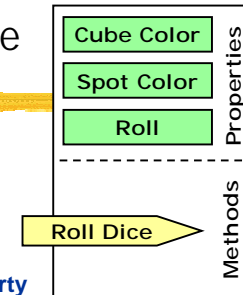
- **Primitive Data Type** holds a single value:
 - Number 1, 1983, 123.543
 - String "This is a string"
 - Boolean true false
- **Object Data Type**
 - Common description for similar "class" of objects
 - Multiple **Properties** (variables) can be contained in one aggregate Object Data Type
 - **Methods** (functions) can be associated the Object Data Type
- **Object** is an Instance of an Object Data Type

1

Copyright © 2002 R.M. Laurie

Object Data Type - Example

```
function Dice(cube, spot)
{
    this.CubeColor = cube; // Property
    this.SpotColor = spot; // Property
    this.Value;           // Property
    this.Roll = roll_dice; // Method Declaration
}
function roll_dice()
{
    this.Value = Math.ceil(6 * Math.random());
}
```



2

Copyright © 2002 R.M. Laurie

Object

- **Object** is an Instance of an Object Data Type
- Objects are created by using assignment operator **=**, the **new** operator, and the object data type **constructor** function
- Effectively you are assigning to a variable name a new object of the object data type described by the constructor

```
var dateBirthday = new Date();
```

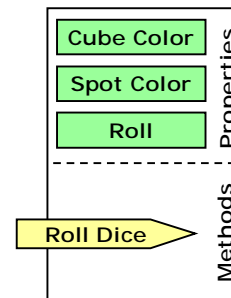
↑ Declaration ↑ Variable Name ↑ New Operator ↑ Constructor

3

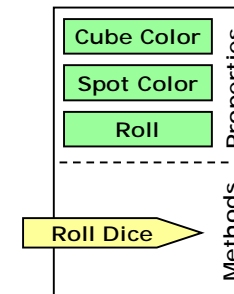
Copyright © 2002 R.M. Laurie

Object - Examples

```
var Die1 = new Dice("Red", "White");
```



```
var Die2 = new Dice("Red", "White");
```



4

Copyright © 2002 R.M. Laurie

```

var Die1 = new Dice("Red", "White");
var Die2 = new Dice("Red", "White");
var Sum;
for(var l=0; l<5; l++){
    Die1.Roll();
    Die2.Roll();
    Sum = Die1.Value + Die2.Value;
    document.writeln(Die1.Value+ " + "+Die2.Value+" = "
        +Sum+" <BR>");
}
function Dice(cube, spot){
    this.CubeColor = cube; // Property
    this.SpotColor = spot; // Property
    this.Value; // Property
    this.Roll = roll_dice; // Method Declaration
}
function roll_dice(){
    this.Value = Math.ceil(6 * Math.random());
}

```

Roll Dice 5 times - Example

```

<HTML><HEAD><TITLE>Object Demo</TITLE>
<SCRIPT LANGUAGE="JavaScript" SRC="DiceRoller.js">
</SCRIPT></HEAD>
<BODY BGCOLOR="#CCFFCC" TEXT="#000000">
Click Reload to run again
</BODY></HTML>

```

```

2 + 1 = 3
2 + 6 = 8
2 + 4 = 6
4 + 5 = 9
3 + 1 = 4
Click Reload to run again

```

6

Copyright © 2002 R.M. Laurie

Predefined Object Data Types

- JavaScript has many Predefined Object Data Types that can be used in your code
 - Document
 - Screen
 - Window
 - Array
 - Date
 - Image
 - Input
 - Layer (Netscape 4 Only)
 - HTMLElement.all (IE 4 Only)
 - Navigator (Browser ID)
 - Math
 - String
 - Number
 - Plugin
- All these object have methods (See Book)
- You can create an *Object* = new *ODT*()

7

Copyright © 2002 R.M. Laurie

Arrays

- Grouping of similarly named variables, which are grouped sequentially in memory and accessed by their element (index) number
- Element numbering begins with 0 to one less than the total number of elements
- An Array element can hold numbers, strings, Boolean, and Objects

Counter[0]	30
Counter[1]	45
Counter[2]	53
Counter[3]	2
Counter[4]	879

8

Copyright © 2002 R.M. Laurie

Creating Arrays

Declaration:

```
var J = new Array('abc', true, 32);
var Counter(5);
```

Counter[0]	30
Counter[1]	45
Counter[2]	53
Counter[3]	2
Counter[4]	879

- Reserves Counter array memory Counter[0] to Counter[4]
- No values are stored in elements

```
var Counter= new Array(30, 45, 53, 2, 879);
for(var K=0; K<50; K++)
    Counter3[K] = 0;
```

Unlike other languages declaration is optional

- Elements can also be added to an existing Array by assigning values to new array elements

9 Copyright © 2002 R.M. Laurie

Array Usage Example

```
// This Program rolls dice 10,000 times and
// prints total rolls for each count 2 to 12.
// Author: Robert M. Laurie
//
var Die1 = new Dice("Red", "White");
var Die2 = new Dice("Red", "White");
var Sum, I, J, $Output;
var Count = Array(13);
for(I=0; I<13; I++)Count[I]=0; //Clear Counter
document.writeln("<B>This Program rolls dice 10,000 times<BR>"
    + "and prints total rolls for each count 2-12<BR></B>");
for(var I=0; I<10000; I++) //Rolls Dice 10,000 times
{
    Die1.Roll();
    Die2.Roll();
    Sum = Die1.Value + Die2.Value;
    Count[Sum]++;
}
```

```
document.writeln("<TABLE BORDER=1 CELSPACING=4 CELLPADDING=2>"
    + "<TR ALIGN=CENTER BGCOLOR='#CCFFF'\><TH>Roll</TH>"
    + "<TH>Count</TH><TH>Graph scaled (Count/100)</TH></TR>");
for(I = 2; I <= 12; I++){
    $Output = "<TR ALIGN=RIGHT><TD> "+I+"</TD><TD>"
    +Count[I]+ "</TD><TD ALIGN=LEFT><TT>";
    for(J = Count[I]/100; J > 0; J--) $Output += "=";
    $Output += "</TT></TH></TR>";
    document.write($Output);
}
document.write("</TABLE>");
function Dice(cube, spot){
    this.CubeColor = cube; // Property
    this.SpotColor = spot; // Property
    this.Value; // Property
    this.Roll = roll_dice; // Method Declaration
}
function roll_dice(){ // Method Function
    this.Value = Math.ceil(6 * Math.random());
}
```

Roll Dice 10,000 Times - Example

```
<HTML><HEAD>
<TITLE>Object Demo</TITLE>
<SCRIPT LANGUAGE="JavaScript"
SRC="DiceRollCounter2.js">
</SCRIPT></HEAD>
<BODY BGCOLOR="#FFFFCC"
TEXT="#000000">
<P>Click Reload to run again</P>
</BODY></HTML>
```

This Program rolls dice 10,000 times and prints total rolls for each count 2-12

Roll	Count	Graph scaled (Count/100)
2	279	===
3	545	=====
4	830	=====
5	1108	=====
6	1397	=====
7	1684	=====
8	1373	=====
9	1132	=====
10	838	=====
11	551	=====
12	263	===

Click Reload to run again

12