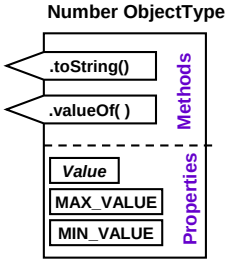
JavaScript Objects

- **Object oriented design (OOD)** breaks problem into objects in a top-down process
  - Supports *Divide and Conquer* approach
  - Supports *Code Reuse*
- **Object-Type (Class in Java or C++)**
  - Definition of a type of object
  - Describes all properties and methods associate with objects of this type
- An **Object** is a self contained **instance** of an object-type that contains
  - **Properties** (data, attributes, member variable)
  - **Methods** (functions, operations, instructions)

JavaScript Object-Type Library

- JavaScript has many Object Types from which new objects may be created
  - Global
  - Math
  - Number
  - String
  - Date
  - Array



- Global object
  - Created when scripting engine started
  - *Methods available immediately*
    - *number* parseInt(*string*)
    - *number* parseFloat(*string*)

Math Object-Type

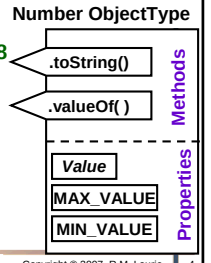
- Math Library is encapsulated in Math Object
- Properties
  - *number* Math.PI Returns 3.141592654558979
  - *number* Math.E Returns Euler's Constant = 2.718
- Methods
  - *number* Math.random() Returns value between 0 to 1
  - *number* Math.sqrt(*num*) Returns square root of num
  - *number* Math.sin(*num*) Returns sine of num
  - and many more…
  - *number* Math.pow(*x*, *y*) Returns X^y power

Output Object Method Inputs

Number Object-Type

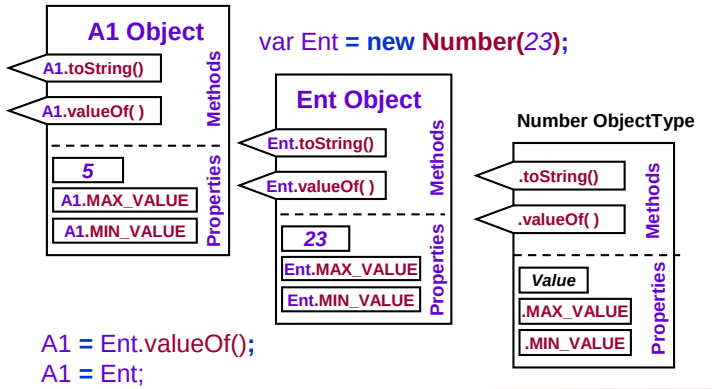
- Number Object-Type defines a container for a number and associated library of methods
- To create an Object (Instance) of the Number Object-Type use the new operator
  - var *NumberObject* = new Number(*value*);
- Properties
  - *NumberObject*.MAX_VALUE // 1.79E+308
  - *NumberObject*.MIN_VALUE // 5.00E-324
- Methods
  - *NumberObject*.valueOf( )
  - *NumberObject*.toString(*radix*)

Object Method Inputs



Creating new Number Objects

```
var A1 = new Number(5);
```




```
A1 = Ent.valueOf();
A1 = Ent;
```


String Object-Type

- String Object-Type defines a container for a string and associated library of methods
- To create an Object (Instance) of the String Object-Type use the new operator
  - var *StringObject* = new String("My Name is Bob");
- Properties
  - *StringObject*.length // length of string object
- Methods
  - *StringObject*.concat(*string, string,…*)
  - *StringObject*.toLowerCase()
  - *StringObject*.substr(*start, length*)
  - *StringObject*.charAt(*index*)
  - *StringObject*.indexOf(*substr, index*)


```
var Time24 = new String(), Hrs24 = new String();
var Hours, Minutes, AmPm, YorN;
do
{
  Time24 = window.prompt("Enter the 2400 time format:", "");
  Hrs24 = Time24.substr(0,2);
  Minutes = Time24.substr(2,2);
  if(Hrs24.charAt(0)=="0")
    Hours = parseInt(Hrs24.charAt(1));
  else
    Hours = parseInt(Hrs24);
  if(Hours < 12)
  {
    AmPm="am";
    if(Hours == 0)
      Hours = 12;
  }
  else
  {
    AmPm="pm";
    if(Hours > 12)
      Hours -= 12;
  }
  document.writeln("<h3>"+Time24+" = "+Hours+":"+Minutes+AmPm+"</h3>");
  YorN = window.prompt("Would you like to enter another 2400 time? (y or n)","y");
}while(YorN == "y" || YorN == "y");
document.writeln("<p>Program Exit</p>");
```

Date Object-Type

- Date Object-Type defines a container for a Date/Time and associated library of methods
- To create an Object (Instance) of the Date Object-Type use the new operator
  - var *DateObject* = new Date(); // Current Date&Time
  - var *DateObject* = new Date(year, month, date[, hours[, minutes[, seconds[,ms]]]]);
- Methods
  - *DateObject*.getDate()
  - *DateObject*.setDate()
  - *DateObject*.getTime()
  - *DateObject*.setTime()
  - *DateObject*.toString()
  - *DateObject*.toGMTString()

