SNS COLLEGE OF TECHNOLOGY

Coimbatore-35
An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A+' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF INFORMATION TECHNOLOGY

WEB TECHNOLOGY

III YEAR - V SEM

UNIT 2 – CSS AND CLIENT-SIDE SCRIPTING

Topic 6 - Syntax-Variables and Data Types

Basic JavaScript Syntax

```
// HighLow.js

var thinkingOf;  // Number the computer has chosen (1 through 1000)
var guess;       // User's latest guess

// Initialize the computer's number
thinkingOf = Math.ceil(Math.random()*1000);

// Play until user guesses the number
guess = window.prompt("I'm thinking of a number between 1 and 1000." +
                      "  What is it?", "");
```

Basic JavaScript Syntax

```
// HighLow.js

var thinkingOf;  // Number the computer has chosen (1 through 1000)
var guess;       // User's latest guess

// Initialize the computer's number
thinkingOf = Math.ceil(Math.random()*1000);

// Play until user guesses the number
guess = window.prompt("I'm thinking of a number between 1 and 1000." +
                      "  What is it?", "");
```

Notice that there is no main() function/method

Basic JavaScript Syntax

// HighLow.js

Comments like Java/C++ (/* */ also allowed)

```
var thinkingOf;  // Number the computer has chosen (1 through 1000)
var guess;       // User's latest guess

// Initialize the computer's number
thinkingOf = Math.ceil(Math.random()*1000);

// Play until user guesses the number
guess = window.prompt("I'm thinking of a number between 1 and 1000." +
                      "  What is it?", "");
```

Basic JavaScript Syntax

Variable declarations:
- Not required
- Data type not specified

```
// HighLow.js

var thinkingOf;  // Number the computer has chosen (1 through 1000)
var guess;       // User's latest guess

// Initialize the computer's number
thinkingOf = Math.ceil(Math.random()*1000);

// Play until user guesses the number
guess = window.prompt("I'm thinking of a number between 1 and 1000." +
                      "  What is it?", "");
```

Basic JavaScript Syntax

```
// HighLow.js

var thinkingOf;   // Number the computer has chosen (1 through 1000)
var guess;        // User's latest guess

// Initialize the computer's number
thinkingOf = Math.ceil(Math.random()*1000);

// Play until user guesses the number
guess = window.prompt("I'm thinking of a number between 1 and 1000." +
                      "  What is it?", "");
```

Semi-colons are usually not required, but always allowed at statement end

Basic JavaScript Syntax

```
// HighLow.js

var thinkingOf;  // Number the computer has chosen (1 through 1000)
var guess;       // User's latest guess

// Initialize the computer's number
thinkingOf = Math.ceil(Math.random()*1000);

// Play until user guesses the number
guess = window.prompt("I'm thinking of a number between 1 and 1000." +
                      "  What is it?", "");
```

Arithmetic operators same as Java/C++

Basic JavaScript Syntax

```
// HighLow.js

var thinkingOf;  // Number the computer has chosen (1 through 1000)
var guess;       // User's latest guess

// Initialize the computer's number
thinkingOf = Math.ceil(Math.random()*1000);

// Play until user guesses the number
guess = window.prompt("I'm thinking of a number between 1 and 1000." +
                      "  What is it?", "");
```

String concatenation operator as well as addition

Basic JavaScript Syntax

```
// HighLow.js

var thinkingOf;  // Number the computer has chosen (1 through 1000)
var guess;       // User's latest guess

// Initialize the computer's number
thinkingOf = Math.ceil(Math.random()*1000);

// Play until user guesses the number
guess = window.prompt("I'm thinking of a number between 1 and 1000." +
                      "  What is it?", "");
```

Arguments can be any expressions

Argument lists are comma-separated

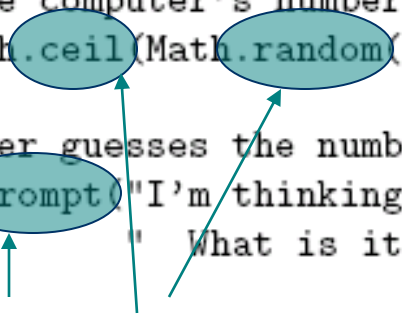
Basic JavaScript Syntax

```
// HighLow.js

var thinkingOf;  // Number the computer has chosen (1 through 1000)
var guess;       // User's latest guess

// Initialize the computer's number
thinkingOf = Math.ceil(Math.random()*1000);

// Play until user guesses the number
guess = window.prompt("I'm thinking of a number between 1 and 1000." +
                      "  What is it?", "");
```



Object dot notation for method calls as in Java/C++

Basic JavaScript Syntax

```
while (guess != thinkingOf)
{

  // Evaluate the user's guess
  if (guess < thinkingOf) {
     guess = window.prompt("Your guess of " + guess +
                           " was too low.  Guess again.", "");
  }
  else {
     guess = window.prompt("Your guess of " + guess +
                           " was too high.  Guess again.", "");
  }
}

// Game over; congratulate the user
window.alert(guess + " is correct!");
```

Basic JavaScript Syntax

```
while (guess != thinkingOf)
{

  // Evaluate the user's guess
  if (guess < thinkingOf) {
     guess = window.prompt("Your guess of " + guess +
                           " was too low.  Guess again.", "");
  }
  else {
     guess = window.prompt("Your guess of " + guess +
                           " was too high.  Guess again.", "");
  }
}

// Game over; congratulate the user
window.alert(guess + " is correct!");
```

Many control constructs and use of { } identical to Java/C++

Basic JavaScript Syntax

```
while (guess != thinkingOf)
{

  // Evaluate the user's guess
  if (guess < thinkingOf) {
     guess = window.prompt("Your guess of " + guess +
                           " was too low.  Guess again.", "");
  }
  else {
     guess = window.prompt("Your guess of " + guess +
                           " was too high.  Guess again.", "");
  }
}

// Game over; congratulate the user
window.alert(guess + " is correct!");
```

Most relational operators syntactically same as Java/C++

Basic JavaScript Syntax

```
while (guess != thinkingOf)
{

  // Evaluate the user's guess
  if (guess < thinkingOf) {
     guess = window.prompt("Your guess of " + guess +
                           " was too low.  Guess again.", "");
  }
  else {
     guess = window.prompt("Your guess of " + guess +
                           " was too high.  Guess again.", "");
  }
}

// Game over; congratulate the user
window.alert(guess + " is correct!");
```

Automatic type conversion:
`guess` is String,
`thinkingOf` is Number

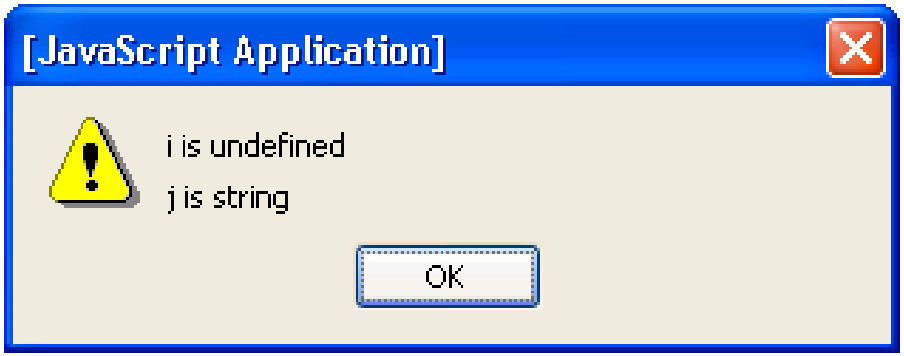
Variables and Data Types

- Type of a variable is dynamic: depends on the type of data it contains
- JavaScript has six data types:
  - Number
  - String
  - Boolean (values `true` and `false`)
  - Object
  - Null (only value of this type is `null`)
  - Undefined (value of newly created variable)
- Primitive data types: all but Object

Variables and Data Types

- `typeof` operator returns string related to data type
  - Syntax: `typeof` *expression*
- Example:

```
// TypeOf.js
var i;
var j;
j = "Not a number";
alert("i is " + (typeof i) + "\n" +
      "j is " + (typeof j));
```



Variables and Data Types

TABLE 4.1: Values returned by `typeof` for various operands.

Operand Value	String typeof Returns
null	"object"
Boolean	"boolean"
Number	"number"
String	"string"
native Object representing function	"function"
native Object not representing function	"object"
declared variable with no value	"undefined"
undeclared variable	"undefined"
nonexistent property of an Object	"undefined"

Variables and Data Types

- Common automatic type conversions:
  - Compare String and Number: String value converted to Number
  - Condition of `if` or `while` converted to Boolean
  - Array accessor (*e.g.*, `3` in `records[3]`) converted to String

Variables and Data Types

TABLE 4.2: Data type conversions to Boolean.

Original Value	Value as Boolean
undefined	`false`
`null`	`false`
0	`false`
NaN	`false`
"" (empty string)	`false`
any other value	`true`

Variables and Data Types

TABLE 4.3: Data type conversions to String.

Original Value	Value as String
undefined	`"undefined"`
`null`	`"null"`
`true`, `false`	`"true"`, `"false"`
NaN	`"NaN"`
Infinity, -Infinity	`"Infinity"`, `"-Infinity"`
other Number up to ≈20 digits	integer or decimal representation
Number over ≈20 digits	scientific notation
Object	call to `toString()` method on the object

Variables and Data Types

TABLE 4.3: Data type conversions to String.

Original Value	Value as String
undefined	`"undefined"`
`null`	`"null"`
`true`, `false`	`"true"`, `"false"`
NaN	`"NaN"`
Infinity, -Infinity	`"Infinity"`, `"-Infinity"`
other Number up to ≈20 digits	integer or decimal representation
Number over ≈20 digits	scientific notation
Object	call to `toString()` method on the object

Special Number values (“Not a Number” and number too large to represent)

Variables and Data Types

TABLE 4.4: Data type conversions to Number.

Original Value	Value as Number
undefined	`NaN`
`null`, `false`, `""` (empty string)	0
`true`	1
String representing number	represented number
other String	`NaN`
Object	call to `valueOf()` method on the object

Variables and Data Types

- Syntax rules for names (identifiers):
  - Must begin with letter or underscore (_)
  - Must contain only letters, underscores, and digits (or certain other characters)
  - Must not be a reserved word

Variables and Data Types

abstract	boolean	break	byte	case	catch
char	class	const	continue	debugger	default
delete	do	double	else	enum	export
extends	false	final	finally	float	for
function	goto	if	implements	import	in
instanceof	int	interface	long	native	new
null	package	private	protected	public	return
short	static	super	switch	synchronized	
this	throw	throws	transient	true	try
typeof	var	void	volatile	while	with

FIGURE 4.6: JavaScript reserved words.

Variables and Data Types

- A variable will automatically be created if a value is assigned to an undeclared identifier:

`var` is not required

```
testing = "Does this work?";
window.alert(testing);
```

- Recommendation: declare all variables
  - Facilitates maintenance
  - Avoids certain exceptions