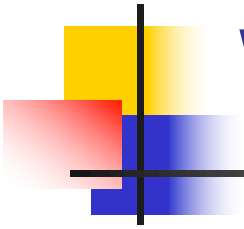
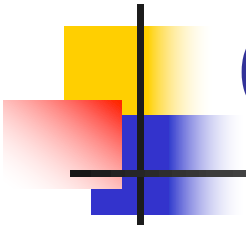
Introduction to JavaScript



Why JavaScript?

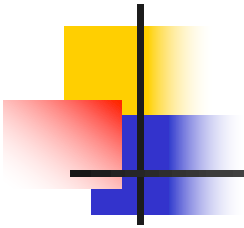
- Adds interactivity and visual enhancement
- Mainly used for client-side execution
  - Processing client requests that do not require server processing
  - No Internet traffic
- Cons
  - The client may not have enough processing resources
  - JavaScript can be turned off for security reasons



Origins…

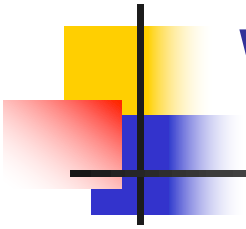
- Scripting language developed by Netscape – *Mocha*, *LiveScript*
- Netscape and Sun collaborated – *JavaScript* (1995)
- Microsoft creates *Jscript* (1996)
- *ECMAScript* is now the "standard"

European Computer Manufacturing Association



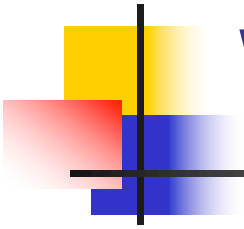
(JavaScript is not JAVA)

1. No graphical user interface
2. No read/write file access on clients
3. Not a class-based object model
4. No multithreading
5. Does not need the Java Virtual Machine (needs a browser)



What JavaScript can do

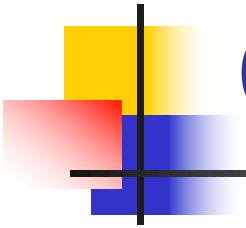
- Client-side:
  - Control browser features
  - Modify document appearance
  - Modify document content
  - Store & use information about user
  - Manipulate images
  - React to state of browser and client system
  - More



When to Use JavaScript

- CSS can handle:
  - Hover and focus events
  - Image swaps
  - Basic animation
  - Drop-down menus
- JavaScript is used to achieve:
  - Visual enhancements that CSS can't do
  - Advanced animation

Where Does One's JavaScript Code Go?



- Two general scenarios ...

1. Embedding JavaScript in HTML

```
<head>
   <title>JavaScript template</title>
   <script  type="text/javascript">
            // JavaScript code goes here
   </script>
</head>

<body>

   ...
```

Actually,

- JavaScript code can be inserted anywhere in the HTML, including in the body.
- There are no limits on how many <script> tags can be embedded in the HTML code of a web page.

2. Putting the JavaScript in its Own File

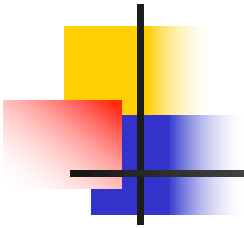
```
<head>
    <title>Client-side input validation</title>
    <script src="inputValidator.js"
            type="text/javascript"></script>
</head>
```

Attributes:

src - **Specify URL of external code source.**

type - **specifies scripting language ("text/javascript")**

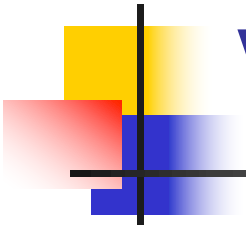
This is the best way for code reuse.



Development Environment

- *A text editor such as VS Code*
- *A browser* to run and debug code
  - Right-click on a page to open the browser's inspect tool (Need to enable the developer's menu in some browsers)
  - Use the JavaScript console to test your code

The Language Basics



Variables

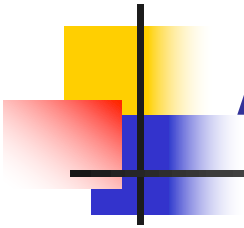
- Valid variable names
  - Consist of letters, digits, underscores, and dollar signs
  - Do not begin with a digit
  - Not a reserved JavaScript keyword
- JavaScript is case sensitive
- A *var* or *let* statement to declare a variable:

```
var varName=1;
```

Dynamic Types

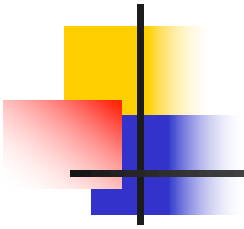
- JavaScript variables have dynamic data types

```
var length = 16;                                    // Number
var lastName = "Johnson";                           // String
var cars = ["Saab", "Volvo", "BMW"];                // Array
var x = {firstName:"John", lastName:"Doe"};         // Object
```



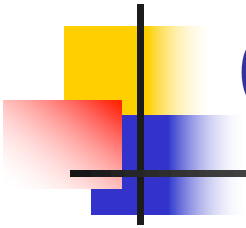
Arithmetic Operators…

+	addition
-	subtraction
*	multiplication
/	division
%	modulus (remainder)
++	increment
--	decrement



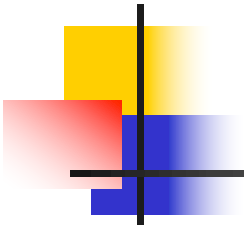
Example:

```
var a=10;
var b;
b = a++;          (a=11, b=10)
b = ++a;          (a=11, b=11)
b = a--;          (a=9, b=10)
b = --a;          (a=9, b=9)
```



Comparison Operators…

Meaning	JavaScript	Example…
equal	==	(x==y)
strict equal	===	(x===y)
not equal	!=	(x!=y)
greater than	>	(x>y)
(or equal to)	>=	(x>=y)
less than	<	(x<y)
(or equal to)	<=	(x<=y)

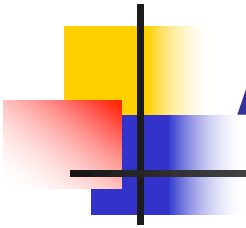


Example:

```
var a = 3;
var b = "3";
var c = (a==b);          (c=true)
var d = (a===b);         (d=false)
```

Logical Operators…

Meaning	JavaScript	Example…
And	&&	(A<B) && (C>D)
Or	||	(A<B) || (C>D)
Not	!	!(E>F)



Assignment Operators…

JavaScript	Example	Same As
=	x=y	x=y
+=	x+=y	x=x+y
-=	x-=y	x=x-y
*=	x*=y	x=x*y
/=	x/=y	x=x/y

Comment Tags

```
//  JavaScript comment
```

```
/*
   Multiple line JavaScript comment
*/
```

```
<!--  HTML comment -->
```

```
<!--
   Multiple line HTML comment
-->
```

String Operators

- String concatenation (+)

Ex.: x = "hello " + "world" (x="hello world")

x = "hello" + 5 (x="hello5")

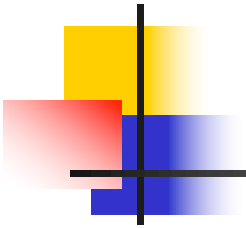
x = "5" + 5 (x="55")

x = Number("5") + 5 (x=10)

Number is a Javascript function that converts a value to a number data type

Work with Strings

var x=”University of Delaware"
var y=x.length (y=22)
y=x.indexOf("i") (y=2) //counts from 0
y=x.lastIndexOf(”e") (y=21)
var z=x.substring(14,8) (z=”Delaware")
z=x.substr(14,8) (z="Delaware")
x=x.toUpperCase() (z=”DELAWARE")
x=x.toLowerCase() (z=”delaware")
Link to other string methods



Date and Time

- var a = new Date(); //current date and time
- Date methods:
  - getHours //0-23
  - getDay //0-6
  - getDate //1-31
  - getMinutes //0-59
  - getMonth //0-11
  - getFullYear //yyyy

Control structures: Conditions

```
if (condition) {
        // area when condition is true
}
else {
        // area when condition is false
};
```

Control structures: Conditions

```
switch (expression) {
    case n;
        // code when case is true
        break;
    …
    default:
        // default code
        break;
};
```

Control structures: Loop

```
while (condition)
{
        // loop code goes here
};
```

Control structures: Loop

```
do
{
        // loop code goes here
} while (condition);
```

Control structures: Loop

```
for(statement 1; condition; statement 2){
        // loop code goes here
};
```

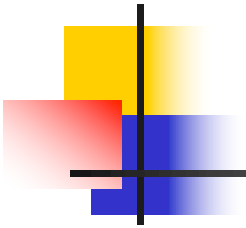
Example:

```
for (var i=0; i<5; i++) {
   …
};
```

Input/Output in JavaScript

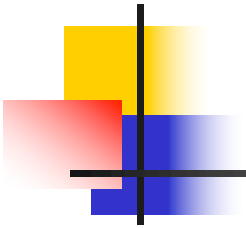
(Limited because of security)

- Output
  - window.alert() - popup message box
  - document.writeln() or document.write()
- Input
  - window.prompt() - input box
  - confirm() - ok/cancel box
- Debug
  - console.log()
  - Console.error()



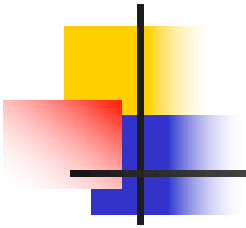
Demo 1

a. Embedded (in <*head*> or in <*body*>):
b. Separate (external) JavaScript file:



Demo 2

- Write JavaScript code to greet the user based on the current time



Demo 3

- Write JavaScript code to get student grades from the user, calculate the GPA for the class, and display the GPA