Javascript:

1. ECMAscript: 内核，兼容性，基本完全兼容
2. DOM： document object model 操作的是HTML，兼容性，相对兼容。
3. BOM： browser object model 操作浏览器窗口的， 兼容性，没有兼容性，它几乎不兼容。

任何语言都有发展历史：

语言的语法：

1. 标识符：

定义：在计算机内为了标识一个对象，所起的名称。

命名规范：

1. 组成：字母、数字、下划线、$
2. 是有字母或者下划线开头的，（$开头的）,推荐是字母开头。
3. 字符长度推荐不超过32位。
4. 保留字（关键字）不能用作标识符。
5. 标识符，一定要代表一定的意义。
6. 符号：

0—9的数字、大小写英文字母52个、特殊符号，转义字符（转义字符是有斜杠\开始的），保留字。

3、变量：定义—在程序运行当中其值可以改变之量

4、常量：定义—在程序运行当中其值不可以改变之量

5、变量在程序中的定义

语法:

Var 变量名 ;

赋值符号 = ;

数据类型：

1）number 数值型

2）string 字符串

表示字符串：单引号或者双引号括起来的字符就是字符串。

如果是与属性值无关的字符串用双引号，否则用单引号。

3）function 函数类型

用处：事件处理程序，功能。

4）object 对象类型

用处：设置，获取，控制的对象

5）undefined 未定义类型

1. Boolean 布尔类型

两个值：true false

数据类型的转换：

转换方式：强制转换（显示转换），自动转换（隐式转换）

ParseInt（）整数转换：如果是浮点数，则转换成去掉小数部分后的整数。如果是字符串以数字开头的则可以转换成去掉第一个非数字开始的数值部分。如果不是数字开头的则提示转换错误（NaN not a number）.

isNaN()判断是不是数字

parseFloat( )浮点数转换：同理，除整数转换成浮点数的时候，还是整数。

变量的作用域：

变量的作用域就是变量的可使用的范围。

分为:全局变量和局部变量

生命周期:从定义开始，到使用结束。

运算符：

1. 算术运算符

+ - \* / % 模

运算规则：

1. 先乘除模，后加减
2. 同级别的运算符，从左向右依次运算。
3. 特性：

+ 的使用含义： 加法操作、字符串链接、正数

* 的使用含义： 减法操作 、负数
* \* 的使用含义：乘法、幂运算

/ : 整除时如果除尽，则是整数，否则是浮点数

% : 整除取余

1. 比较运算（关系运算） 最终的结果是boolean 类型。

运算符： > < >= <= == === != !==

1. 逻辑运算符

&& 与 相当于乘法

|| 或

！ 非 （取反）

运算法则：

如果是与运行，则运算的条件必须同时满足，返回的值是true，否则返回false

如果是或运算，则运算的条件有一个满足，则返回true， 否则是false。

如果是取反运算：则是遇真则假，遇假则真。

什么是真，什么是假

1. 非零的数字都是真的。
2. 未定义的也是假的。
3. Null 也是假的
4. True 是真 false 是假

4、自增自减

++ --

功能:就是原变量的基础上增加或者减少1。

法则：

如果是前缀运算符，先增加（减少）该变量的值在使用该变量。

如果是后缀运算符,先使用该变量的值，再增加或减少变量。

5、位运算(逻辑运算)：

& 与

| 或

^ 异或

~ 位非

<< 左移

>> 右移

>>> 无符号右移

运算法则：

1. 逻辑与运算：见零置零。
2. 逻辑或运算：见1置1。
3. 异或运算: 相同为0 相反为1
4. 位非 : 按位取反
5. 左移 ： 移出最高位，最低位补0
6. 右移 ： 移出的最低位，最高位补，最高位数据。
7. 无符号右移： 永远补的是0。

移位的用处：可以做加密算法，同时，左移相当于乘2，右相当于除2。

6、赋值运算

=

7、自反赋值运算符

+ = -= \*= /= %=

运算的原理： 左边的操作数与右边操作数运算后赋给左边。

a+=b; a=a+b;

8 、条件运算符

? :

表达式1?表达式2：表达式3

法则：

如果表达式1成立，取表达式2的值，否则取表达式3的值。

9、() 强制运算符

运算符按照运算的操作数的个数：分为单目、双目、三目。

10、运算符的优先级

最高的是（） 最低是赋值=