1. **Alerts**

* Pop up that displays a message string
* alert();
* not capitalized
* keyword
* Values:
  + Strings
  + Variables
  + Expressions
  + Arrays

1. **Variables for Strings**

* var = “string”;
* strings are written in quotes “”
* word referring to a particular value
* variables can be declared w/o any value
* you can assign new values

**Variables for Numbers**

* var = 1;
* variables can only have one value
* variables can have concatenated values?
* Can’t be used as initial number for name
* Variables can contain variables/numbers only/combination

**Variable Names Legal and Illegal**

* Variables can have any name except js keywords/reserved words
* Can’t start with a number
* Can have $ \_ symbols
* Names can’t be keywords or assigned words by js
* No quotes
* No spaces
* camelCase
* Make var name descriptive

**Math expressions: Familiar operators**

* + addition, - subtraction, \* multiply, / divide

**Math expressions: Unfamiliar operators**

* % modulus
* Remainder of division

**8. Math expressions: Eliminating ambiguity**

* Use parenthesis () to specify which expressions to execute in order
* PEMDAS
* JS uses math precedents PEMDAS
* Overrides all built-in precedence rules
* Helps make intentions clear
* Change order of expression

**Concatenating text strings**

* use + to combine strings
* var helloWorld = (“hello “ + “world!”);
* strings must include spaces manually
* connect any combo of strings and variables

**9. Prompts**

Var response = prompt(“What’s your name?”, defaultAnswer);

* a prompt is a popup that requires user raw input
* the first part of the prompt is the question, second part is the defaultAnswer after user inputs
* you can make a variable = response and execute a statement
* similar to alert
* input & response field
* responses come back as strings
* if user clicks OK entering nothing – var is assigned an empty “” string

**If statements**

* if statements are test conditions
* they test comparison operators
* after test, statements under the test condition execute

**Comparison operators**

* === equals
* !== not equal
* < less than
* > greater than
* <= less than or equal to
* >= greater than or equal to

**If..else and else if statements**

* if statement is the first test
  + can have multiple if statements
* else if is the next test if, the if fails
  + makes code more readible
* else executes if all the above fail

**Testing sets of conditions**

* && and
* || or
* and means this and this have to pass
* or means or this has to pass
* eliminate ambiguity with ()

**If statements nested**

* multiple if statements can be done
* alternative to Testing sets of conditions

**Arrays**

* arrays are variables
* contain multiple elements
* 0-based

**Arrays: Adding and removing elements**

* .pop() removes the element at the end of an array
* .push() adds elements to the end of an arra

**Arrays: Removing and inserting elements**

* Shift – the beginning element of an array
* Unshift – adds elements the end of an array
* Splice (1-based) – add/remove elements of an array
* Slice – copy and put into an array --

**For loops**

* for () {

{

* for loops are cycles that have a starting position that (++) increase/decrease (--)
* for (var i = 0; i <= 10; i++) {

}

* the first declaration assigns the position; amount of loops; increment/decrement
* variables can have any name
* i is commonly used as “interger”

**For loops: Flags, Booleans, array length, ad loopus interruptus**

Flags

* beginning variable of an if statement
* var matchFound = false;

if (x === y) {

matchFound = true;

alert(“Match found”);

}

If (matchFound = false;) {

Alert(“Match not found”);

}

**For loops nested**

* for loops nested are indented twice per loop
* per full cycle of an inner array/outer array goes through one element
* inner array “seconds” hand/ outer array “minute” hand

**Changing Case**

.toLowerCase()

.toUpperCase()

* Useful if matching a user’s input for a prompt
* Js is literal
* Changing case makes the users input all lower/upper case which makes it easy to equal the array element