



Part1 - Binary Representation of numbers:

In base 10:

The number 142 is expressed as $2 \cdot 10^0 + 4 \cdot 10^1 + 1 \cdot 10^2 = 142$;

In binary (base 2):

The number 5 can be expressed as $1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 5_{10} = 101_2$

i>clicker question solution:

Going from right to left

$$1101_2 = 1 \cdot (2^0) + 0 \cdot (2^1) + 1 \cdot (2^2) + 1 \cdot (2^3)$$

$$= 1 + 0 + 4 + 8$$

$$= 13$$

ASCII activity:

To represent the word BOW

$$\begin{aligned} B &= 66_{10} = 64 + 0 + 0 + 0 + 0 + 2 + 0 \\ &= 1 \cdot (2^6) + 0 \cdot (2^5) + 0 \cdot (2^4) + 0 \cdot (2^3) + 0 \cdot (2^2) + 1 \cdot (2^1) + 0 \cdot (2^0) \\ &= 1000010_2 \end{aligned}$$

$$\begin{aligned} O &= 79_{10} = 64 + 0 + 0 + 8 + 4 + 2 + 1 \\ &= 1 \cdot (2^6) + 0 \cdot (2^5) + 0 \cdot (2^4) + 1 \cdot (2^3) + 1 \cdot (2^2) + 1 \cdot (2^1) + 1 \cdot (2^0) \\ &= 1001111_2 \end{aligned}$$

$$\begin{aligned} W &= 87_{10} = 64 + 0 + 16 + 0 + 4 + 2 + 1 \\ &= 1 \cdot (2^6) + 0 \cdot (2^5) + 1 \cdot (2^4) + 0 \cdot (2^3) + 1 \cdot (2^2) + 1 \cdot (2^1) + 1 \cdot (2^0) \\ &= 1010111_2 \end{aligned}$$

An easy method for converting decimal to binary is explained here:

<http://www.is.wayne.edu/OLMT/BINARY/PAGE3.HTM>

Part 2 – Functions:

Functions are basically piece of code that can be used in your script. They give you ways of doing something with fewer lines of code in your program. Of course all the logic is written inside of the function definition, but in your program, it gets logically easier and clutter free

to build your code if functions are used Functions could be inbuilt or defined by you. We have learnt a couple of in built functions so far:

- `print()`
- `readline()`

Example 1:

=====

rateSchool function:

```
function rateSchool(school)
{
    if(school == "Illinois")
        return "Awesome";

    if(school == "ISU")
        return "Not so awesome";
    else
        return "unknown";
}
```

=====

Using the function rateSchool:

```
var rating = rateSchool("ISU");      - LINE 1
print(rating);                       - LINE 2
```

=====

Comments on the rateSchool function:

The word ***rateschool*** is the name of the function

The word "**school**" present inside the parenthesis is called an argument/parameter to the function `rateSchool()`. It is some information that must be given to the function for it to proceed with its code execution. There can be any number of parameters for a function.

The command: **return "unknown";** is called a return statement. This value is returned to the variable on the left hand side of the assignment with this function call. This is **rating** in the above code in LINE 1. Also, there are three return statements in the function. If a return statement is executed in the function block, the rest of the code following that is not executed.

**Global functions and local functions:**

JavaScript has a lot of functions. Hence, all functions shouldn't be at the same place to make it more organized. Some functions are global/top-level functions.

`print()` and `readline()` are examples of top-level functions that can be called from anywhere in the program.

```
var s = "Hello";
var l = s.length; // getting the length of the string, in this
                  // case it will be 5
```

(notice that there is no parenthesis after `length`, this is an exception in JavaScript)

In general, all functions are followed by a parenthesis. The function **`length`** is an exception and does not have parenthesis. ALL other functions have them.

=====

`charAt()` :

`charAt()` - This function returns the character within a string at the index inside the parenthesis

```
var str = "Hello";
var c = str.charAt(1);
```

This will return the character "e" **because the computer starts counting indices at 0**. Hence `str.charAt(0)` - "H".

Example 2:

=====

`isVowel()` : Counting the number of vowels in string.

```
// This is the function definition of isVowel() and it accepts a
// char as an argument
function isVowel(letter)
{
    if(letter == "a")
        return true;
    if(letter == "e")
        return true;
    if(letter == "i")
        return true;
    if(letter == "o")
        return true;
    if(letter=="u")
        return true;
}
```

=====

Utilizing the `isVowel()` function

```
//Initializing variables
var str = "Hello"; // The string we want to iterate through to
                  // find the number of vowels in
var i=0;          // helps iterate through the string
var count=0;      // keeps count of the vowels

while (i < str.length) // This line checks if 'i' is strictly lesser
                      // (<) than the length of the string, to
                      // iterate through the string, because
                      // computers begin counting from zero.
{
    var c= str.charAt(i);
    if (isVowel(c))
    {
        count=count+1; // incremented if isVowel returns "true"
    }

    // The next line is what advances the logic to the next character
    // of the string
    i=i+1;
}

print("Vowels: " + count);
```

=====

Comments about the above example:

Note that the program does not account for capital letters that are vowels. To make an OR condition, you can also use the following notation:

```
if ((letter=="a") || (letter=="A"))
```

– The pipe symbol `'||'` denotes an **'or'** condition. (You can type this by pressing the key above the enter key twice, whilst holding the shift key)

What are comments in JavaScript:

The comments are used for explaining something about the code. They improve the readability of any code. In JavaScript, they are the lines that start after `'//'`. They are generally shown in different color in most text editors. It is a good programming practice to add comments in your code to make it readable to others and to yourself in the future. For example:

```
// This is a code to add two numbers together.
```