Base64

First let's understand how base 64 works

```
+--first octet--+-second octet--+--third octet--+
|7 6 5 4 3 2 1 0|7 6 5 4 3 2 1 0|7 6 5 4 3 2 1 0|
+---------+----+-------+-------+----+---------+
|5 4 3 2 1 0|5 4 3 2 1 0|5 4 3 2 1 0|5 4 3 2 1 0|
+--1.index--+--2.index--+--3.index--+--4.index--+
```

Now here is where things begin to get interesting and I will show step by step

First we want a Binary Conversion Table

```
+------------+
|  0    0000 |
+------------+
|  1    0001 |
+------------+
|  2    0010 |
+------------+
|  3    0011 |
+------------+
|  4    0100 |
+------------+
|  5    0101 |
+------------+
|  6    0110 |
+------------+
|  7    0111 |
+------------+
|  8    1000 |
+------------+
|  9    1001 |
+------------+
|  A    1010 |
+------------+
|  B    1011 |
+------------+
|  C    1100 |
+------------+
|  D    1101 |
+------------+
|  E    1110 |
+------------+
|  F    1111 |
+------------+
```

```
EXAMPLE STRING: Hello World!

HEX:48656C6C6F20576F726C6421
        4   8   6   5   6   C   6   C   6   F   2   0   5   7   6   F   7   2   6   C   6   4   2
1
8-bit: 01001000 01100101 01101100 01101100 01101111 00100000 01010111 01101111 01110010 01101100 01100100 00100001
6-bit: 010010 000110 010101 101100 011011 000110 111100 100000 010101 110110 111101 110010 011011 000110 010000 100001
Dec:   18     6      21     44   | 27     6      60     32   | 21     54     61     50   | 27     6      16     33

Base64 output: SGVsbG8gV29ybGQh
```

```
            ###########################################
            #                                         #
            #          BASE64 CONVERSION TABLE        #
            #                                         #
            ###########################################
```

char	binary	dec	char	binary	dec	char	binary	dec	char	binary	dec
A	000000	0	Q	010000	16	g	100000	32	w	110000	48
B	000001	1	R	010001	17	h	100001	33	x	110001	49
C	000010	2	S	010010	18	i	100010	34	y	110010	50
D	000011	3	T	010011	19	j	100011	35	z	110011	51
E	000100	4	U	010100	20	k	100100	36	0	110100	52
F	000101	5	V	010101	21	l	100101	37	1	110101	53
G	000110	6	W	010110	22	m	100110	38	2	110110	54
H	000111	7	X	010111	23	n	100111	39	3	110111	55
I	001000	8	Y	011000	24	o	101000	40	4	111000	56
J	001001	9	Z	011001	25	p	101001	41	5	111001	57
K	001010	10	a	011010	26	q	101010	42	6	111010	58
L	001011	11	b	011011	27	r	101011	43	7	111011	59

```
+-------------------+-------------------+-------------------+-------------------+
|  M    001100  12 | c    011100  28 | s    101100  44 | 8    111100  60 |
+-------------------+-------------------+-------------------+-------------------+
|  N    001101  13 | d    011101  29 | t    101101  45 | 9    111101  61 |
+-------------------+-------------------+-------------------+-------------------+
|  O    001110  14 | e    011110  30 | u    101110  46 | +    111110  62 |
+-------------------+-------------------+-------------------+-------------------+
|  P    001111  15 | f    011111  31 | v    101111  47 | /    111111  63 |
+-------------------+-------------------+-------------------+-------------------+
```

```
JWT Header: eyJ1c2VybmFtZSI6InRlc3QxIn0
        e      y      J      1      | c      2      V      y      |
Decim: 30     50     9      53     | 28     54     21     50     |
6-bit: 011110 110010 001001 110101 | 011100 110110 010101 110010 |
8-bit: 01111011 00100010 01110101  | 01110011 01100101 01110010 |
Hex:   7  B    2  2   7  5         | 7  3   6  5   7  2         |

        b      m      F      t      | Z      S      I      6      |
Decim: 27     38     5      45     | 25     18     8      58     |
6-bit: 011011 100110 000101 101101 | 011001 010010 001000 111010 |
8-bit: 01101110 01100001 01101101  | 01100101 00100010 00111010 |
Hex:   6  E    6  1   6  D         | 6  5   2  2   3  A         |

        I      n      R      l      | c      3      Q      x      |
Decim: 8      39     17     37     | 28     55     16     49     |
6-bit: 001000 100111 010001 100101 | 011100 110111 010000 110001 |
8-bit: 00100010 01110100 01100101  | 01110011 01110100 00110001 |
Hex:   2  2    7  4   6  5         | 7  3   7  4   3  1         |

        I      n      0
Decim: 8      39     52
6-bit: 001000 100111 110100
8-bit: 00100010 01111101 00
Hex:   2  2    7  D   =

Hex str: 7B 22 75 73 65 72 6E 61 6D 65 22 3A 22 74 65 73 74 31 22 7D
ASCII :  {"username":"test1"}

JWT Payload: FWkNLg
        F      W      k      N      L      g
Decimal: 5     22     36     13     11     32
6-bit:   000101 010110 100100 001101 001011 100000
8-bit:   00010101 01101001 00001101 00101110 0000
Hex:     15 69 0D 2E ═
ASCII:   NAK i  \r .              NAK = Negative Acknowledge

JWT Key: CYgHRLkx4PxK77wLMFVyLheuEIY
        C      Y      g      H      R      L      k      x      4
Decimal: 2     24     32     7      17     11     36     49     56
6-Bit:   000010 011000 100000 000111 010001 001011 100100 110001 111000
8-bit    00001001 10001000 00000111 01000100 10111001 00110001 11100000
Hex:     0  9   8  8   0  7   8  4   B  9   3  1   D  0

        P      x      K      7      7      w      L      M      F
Decimal: 15    49     10     59     59     48     11     12     5
6-Bit:   001111 110001 001010 111011 111011 110000 001011 001100 000101
8-Bit:   11111100 01001010 11101111 10111100 00001011 00110000 01010101
Hex:     F  C   4  A   E  F   B  C   0  B   3  0   5  5

        V      y      L      h      e      u      E      I      Y
Decimal: 21    50     11     33     30     46     4      8      24
6-Bit:   010101 110010 001011 100001 011110 000100 000100 001000 011000
8-Bit:   01110010 00101110 00010111 10101110 00010000 10000110 00
Hex:     7  2   2  E   1  7   A  E   1  0   8  6   =

Hex  : 09 88 07 84 B9 31 D0 FC 4A EF BC 0B 30 55 72 2E 17 AE 10 86
ASCII: \t ê  \a ä ╫ 1 ð  ³  J  ï  ¼  \v 0  U  r  .  SI ® \b å
```